

# Counterfactual Sensitivity Regularization for Trace-Sensitive Reasoning in Large Language Models

Sanjeda Akter<sup>\*†1</sup> and Ibne Farabi Shihab<sup>\*1</sup> and Anuj Sharma<sup>2</sup>

<sup>1</sup>Department of Computer Science, Iowa State University

<sup>2</sup>Department of Civil, Construction & Environmental Engineering, Iowa State University  
ishihab@iastate.edu

## Abstract

Large language models can produce correct final answers with reasoning traces that are incomplete, inconsistent, or weakly coupled to the answer. We study whether training can increase this coupling in structured reasoning domains. We introduce Counterfactual Sensitivity Regularization (CSR), a training objective that creates verifier-approved counterfactual edits to reasoning traces, such as changing an arithmetic operator or a logical relation, and penalizes models when the answer distribution remains unchanged after an edit that should change the computation. We evaluate this behavior with Counterfactual Outcome Sensitivity (COS), the fraction of originally correct examples on which a validity-breaking trace edit changes the final answer. Because COS is aligned with the CSR objective by construction, we report it jointly with counterfactual target accuracy, target coverage, and null-edit preservation, and alongside evidence collected outside the training loop: natural-unfaithfulness audits, external formal verification, input contrast sets, and trace-dependence probes. Across arithmetic, formal logic, multi-hop question answering, and biomedical QA, CSR substantially improves edit sensitivity while incurring small accuracy trade-offs. The gains are largest in domains where the edited operators can be identified and verified reliably; performance degrades as verifier precision falls. These results support CSR as a practical regularizer for trace-sensitive reasoning, not as a complete solution to explanation faithfulness.

## 1 Introduction

Large language models (Brown et al., 2020) routinely solve reasoning problems by producing a chain of thought followed by an answer (Wei et al., 2023; Kojima et al., 2022). Two problems persist. First, a model can output a correct answer even

when its written reasoning is flawed (Cobbe et al., 2021; Lanham et al., 2023; Turpin et al., 2023). Second, the written trace may be only weakly coupled to the computation that determines the answer (Saparov and He, 2023). Inference-time techniques such as self-consistency (Wang et al., 2023) can improve answer accuracy, but they do not by themselves make the generated trace a reliable object to audit.

This mismatch matters because users often inspect reasoning traces as if errors in the trace should affect the answer. When a model keeps the same answer after a logically important step is changed, the trace is not serving as a dependable diagnostic signal. This is especially concerning in domains where the answer must be checked, challenged, or reproduced.

We study Counterfactual Sensitivity Regularization (CSR), a training objective that makes this diagnostic requirement explicit. CSR creates small, verifier-approved edits to reasoning traces during training and penalizes the model when the answer distribution remains too similar after an edit that should alter the computation. The method is most natural in domains where trace operators are easy to identify, such as arithmetic, formal logic, code, and some forms of multi-hop question answering.

The central evaluation metric is Counterfactual Outcome Sensitivity (COS): among examples answered correctly before perturbation, the fraction for which a validity-breaking trace edit changes the final answer. COS is aligned with the CSR objective by construction, and it is not a complete measure of faithfulness, because a model can change to another wrong answer. We therefore do not ask the reader to accept COS on its own. We report it together with counterfactual target accuracy, target coverage and null-edit preservation (§5.2), and with evidence whose annotation or scoring procedures share no components with the CSR training loop (§5.5); the Limitations section states what this

<sup>\*</sup>Equal contribution.

<sup>†</sup>Corresponding author: ishihab@iastate.edu.

evidence still does not establish.

This paper makes three contributions. First, we formulate CSR as a training-time counterfactual trace regularizer and specify the verifier assumptions under which its signal is meaningful. Second, we evaluate CSR against process-supervision and reinforcement-learning baselines, and against sensitivity-aware baselines that themselves train on corrupted or counterfactual traces, using the same base data, optimizer, and epoch schedule; CSR incurs a measured 9% GPU-hour overhead from the counterfactual forward pass, so the comparison is not strictly compute-identical, and we report COS per GPU-hour alongside the main numbers. We find large COS gains with small accuracy trade-offs in structured domains, and we present that trade-off as a tunable operating point rather than a fixed cost. Third, we characterize the method’s limits: gains depend strongly on operator and verifier precision, and the method should not be interpreted as a general solution for open-ended explanation faithfulness.

## 2 Related Work

CSR builds on work that probes whether natural-language rationales affect model predictions. Comprehensiveness and sufficiency remove or retain rationale tokens to test whether a rationale is predictive of the answer (DeYoung et al., 2020). Chain-of-thought faithfulness studies go further by intervening on generated reasoning and testing whether the answer changes (Lanham et al., 2023; Turpin et al., 2023; Lyu et al., 2024). CSR differs from these primarily in how the signal is used: prior work uses interventions mostly as diagnostics, while CSR uses verifier-approved interventions as a training objective.

CSR is also related to process supervision, in which intermediate steps are scored by humans or learned reward models (Lightman et al., 2024; Uesato et al., 2022). Process supervision rewards locally valid steps; CSR instead asks whether changing a consequential step changes the answer distribution. The two approaches are therefore complementary. Verifier-guided and tool-augmented reasoning systems such as LINC (Olausson et al., 2023) check derivations at inference time, whereas CSR changes the training signal. Related contemporary efforts include CausalGPT (Yu et al., 2025), which uses counterfactual prompts, the faithful chain-of-thought work of Lyu et al. (2023), and

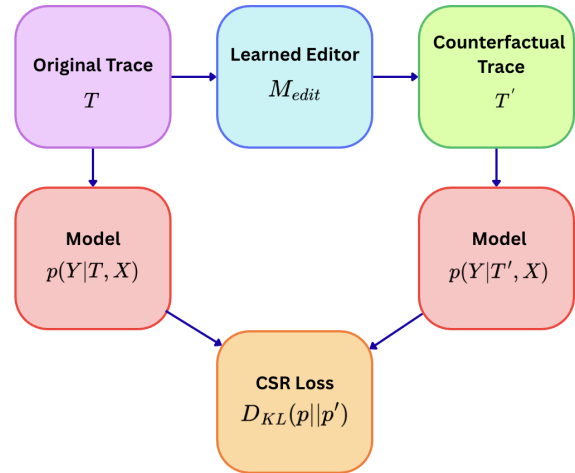


Figure 1: CSR training. A standard forward pass yields an answer distribution over a task-specific answer space. A learned editor proposes a minimal trace edit, and a verifier accepts only edits that break trace validity. A second forward pass on the edited trace yields a counterfactual answer distribution. The CSR term increases the distance between the original and counterfactual answer distributions only for accepted edits.

inference-time validity work (Akter et al., 2026, 2025). Surveys by Huang and Chang (2023); Qiao et al. (2023) and Madsen et al. (2022) provide broader context. We avoid claiming that CSR is the first counterfactual approach to reasoning faithfulness; the novelty is the combination of automated trace edits, verifier gating, and a training-time sensitivity objective; Appendix L gives the extended discussion.

## 3 Counterfactual Sensitivity Regularization

CSR is designed to increase the model’s sensitivity to reasoning steps that are verified as consequential. Figure 1 illustrates the training loop and Algorithm 1 gives the pseudocode.

For an input question  $X$ , let  $T$  be a generated or teacher-forced reasoning trace and let  $\mathcal{Y}$  be the task-specific answer space. We write  $\Pi_{\mathcal{Y}}$  for the operation that extracts answer logits from the model output: number-token logits for arithmetic, label logits for logical deduction and PubMedQA, and constrained span logits for HotpotQA. The original answer distribution is

$$p_{\theta}(Y \mid X, T) = \text{softmax}(\Pi_{\mathcal{Y}}(M_{\theta}(X, T))). \quad (1)$$

The task loss is the negative log-likelihood of the

ground-truth answer,

$$\mathcal{L}_{\text{task}} = -\log p_{\theta}(Y_{\text{true}} | X, T). \quad (2)$$

The counterfactual trace  $T'$  is produced by an editor  $M_{\text{edit}}$  and accepted only when a verifier marks  $T$  as valid and  $T'$  as invalid. The editor is a small Transformer trained by REINFORCE to maximize a reward combining a validity term (the edit must break trace validity), an impact term (the edit should change the answer distribution), and a minimality penalty on edit length; Appendix B gives the architecture and training details. Domain verifiers are rule-based for arithmetic, forward-chaining for formal logic, and NLI-based for question answering. These verifiers are not assumed to be perfect; Section 5.4 and Appendix C.2 analyze sensitivity to verifier precision.

For an accepted edit, CSR compares the answer distribution under the original trace with the answer distribution under the edited trace. We stop the gradient through the original distribution so that the task loss remains responsible for fitting the correct answer while the CSR term primarily pushes the edited-trace distribution away:

$$\mathcal{L}_{\text{CSR}} = D_{\text{KL}}(\text{sg}[p_{\theta}(Y|X, T)] \| p_{\theta}(Y|X, T')), \quad (3)$$

where  $\text{sg}[\cdot]$  denotes stop-gradient. The total objective gates the CSR term by verifier outcome and confidence:

$$I_v = \mathbf{1}[v(T) = 1 \wedge v(T') = 0 \wedge \text{conf}(v) \geq \tau_v]. \quad (4)$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} - \lambda I_v \mathcal{L}_{\text{CSR}}. \quad (5)$$

In all experiments we temperature-scale the two answer distributions before computing KL and clip the CSR contribution to avoid rare unstable updates from near-zero counterfactual probabilities. The coefficient  $\lambda$  trades correctness against the strength of the sensitivity signal; §5.9 and Appendix I report the operating curve over  $\lambda$ , which has a robust range  $[0.3, 0.7]$  and an optimum near 0.5.

The CSR term is informative only when the verifier reliably identifies validity-breaking edits. When verifier confidence is low, the training signal can point in the wrong direction, so we gate the regularizer by verifier confidence as shown in Eq. 5. Unless otherwise stated, we use  $\tau_v = 0.7$ . Appendix K reports the share of full-CSR gains retained under this gate.

---

#### Algorithm 1 CSR training step

---

```

1: Input:  $X$ , trace  $T$ , answer  $Y_{\text{true}}$ , model  $M_{\theta}$ ,
   editor  $M_{\text{edit}}$ , verifier  $v$ , projection  $\Pi_Y$ , coefficient  $\lambda$ , threshold  $\tau_v$ 
2:  $z_{\text{orig}} \leftarrow \Pi_Y(M_{\theta}(X, T))$ 
3:  $p_{\text{orig}} \leftarrow \text{softmax}(z_{\text{orig}})$ 
4:  $\mathcal{L}_{\text{task}} \leftarrow -\log p_{\text{orig}}(Y_{\text{true}})$ 
5:  $T' \leftarrow \text{ApplyEdits}(T, M_{\text{edit}}(X, T))$ 
6: if  $v(T) = 1$  and  $v(T') = 0$  and  $\text{conf}(v) \geq \tau_v$ 
   then
7:    $z_{\text{cf}} \leftarrow \Pi_Y(M_{\theta}(X, T'))$ 
8:    $p_{\text{cf}} \leftarrow \text{softmax}(z_{\text{cf}})$ 
9:    $\mathcal{L}_{\text{CSR}} \leftarrow D_{\text{KL}}(\text{sg}[p_{\text{orig}}] \| p_{\text{cf}})$ 
10:   $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{task}} - \lambda \mathcal{L}_{\text{CSR}}$ 
11: else
12:   $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{task}}$ 
13: end if
14:  $\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}_{\text{total}}$ 

```

---

A concrete example illustrates the signal. On the GSM8K problem “Jessie has \$20 and buys 4 packs at \$2 each; how much is left?”, a standard trace is “She spent  $4 \times 2 = 8$ , so  $20 - 8 = 12$ ,” with answer 12. If the final subtraction is edited to addition, the trace now supports a different computation. A model whose answer is insensitive to the trace often keeps answering 12. CSR penalizes this invariance: the edited trace should reduce probability on the original answer and, in many arithmetic cases, move mass toward the answer implied by the edited computation.

## 4 Theoretical Framing and Scope

CSR should be interpreted as a counterfactual sensitivity objective, not as a general proof that a model’s natural-language explanation is faithful. We formalize the quantity that CSR optimizes and state the assumptions needed for it to be meaningful.

**Definition 1** (Trace sensitivity). *Let  $p_{\theta}(Y | X, T)$  be the model’s answer distribution given input  $X$  and trace  $T$ . For an accepted edit  $T \rightarrow T'$ , counterfactual trace sensitivity is*

$$\text{CS}(X; T \rightarrow T') = D_{\text{KL}}\left(p_{\theta}(Y | X, T) \parallel p_{\theta}(Y | X, T')\right). \quad (6)$$

*Comprehensiveness and sufficiency are observational ablations: they remove or retain subsets of the trace. CS is different because it evaluates a*

*verifier-approved edit to a specific reasoning operation.*

High CS is evidence that the answer distribution depends on the edited trace content. It is not, by itself, a proof that the trace is the complete causal mechanism behind the answer. A model can be sensitive to edited traces while still using shortcuts, and a model can change to an incorrect counterfactual answer. For this reason, our experiments pair COS with null-edit preservation, natural-unfaithfulness annotation, and trace-ablation tests.

**Why should a corrupted trace change the answer?** The objective assumes that a validity-breaking trace edit should alter the final answer, and this assumption deserves an explicit argument, because the input problem is unchanged and a robust model might instead detect the invalid trace and preserve the correct answer. CSR studies the conditional dependence of  $p_\theta(Y \mid X, T)$  on the model’s own displayed reasoning when that trace is treated as an auditable intermediate computation: if a consequential step is changed and the answer distribution is invariant, the trace is less useful as an audit object, whatever the reason. This is a different design goal from a system built to reject untrusted external rationales and recompute from  $X$ , where preserving the original answer may be desirable. COS alone cannot distinguish “ignored the trace because it was never used” from “detected the invalid trace and recomputed correctly,” which is why we pair it with counterfactual target accuracy, null edits, natural audits, trace necessity, and external checks.

The usefulness of CSR depends on three conditions: the edit must target a semantically consequential part of the trace, the verifier must reject invalid edits with high precision, and the evaluation must include controls for edits that preserve semantics. We treat these as validity conditions on the method rather than as deployment caveats, and Table 1 makes the resulting boundary concrete before the experiments begin. Appendix C.2 gives a decomposition of the expected CSR signal under imperfect verifiers, and Appendix C discusses why we do not claim general dominance of CS over comprehensiveness or sufficiency.

CSR is most appropriate when operator identification is precise and verifier false positives are rare. Below roughly 60% operator precision, the method becomes difficult to justify without stronger verifier engineering, and we do not recommend it for

Table 1: Applicability by operator-identification precision. Strongly recommended above 89%, recommended through 78%, not advised below 60%. The PubMedQA row reports the learned automatic-discovery setting (74.1% discovery precision); full CSR with a manually specified operator inventory and verifier reaches 67.3 COS in the same domain (§5.7, Appendix J).

| Domain      | Op. prec. | COS  | Rec.    |
|-------------|-----------|------|---------|
| GSM8K       | 94.2%     | 85.1 | Strong  |
| ProofWriter | 91.4%     | 82.3 | Strong  |
| HotpotQA    | 79.3%     | 84.6 | Yes     |
| PubMedQA    | 74.1%     | 58.9 | Caution |
| PersonaChat | ~52%      | 41.7 | No      |

open-ended dialogue or narrative unless a domain-specific verifier can be validated.

## 5 Experiments

### 5.1 Setup

We evaluate CSR on four reasoning domains spanning the structured-to-open spectrum: GSM8K (Cobbe et al., 2021) for arithmetic, ProofWriter (Tafjord et al., 2021) for formal logic, HotpotQA (Yang et al., 2018) for multi-hop QA, and PubMedQA (Jin et al., 2019) for biomedical reasoning. Our primary base model is Llama-2-13B (Touvron et al., 2023); §5.6 reports additional architectures. Baselines include standard fine-tuning, process supervision, process reward models (RM), GRPO (Shao et al., 2024), and verifier-guided training, together with four sensitivity-aware baselines that themselves train on corrupted or counterfactual traces (§5.3). All methods use the same training data, optimizer, and epoch schedule; CSR adds a gated counterfactual forward pass, producing a measured 9% GPU-hour overhead, so the comparison is not strictly compute-identical and we report COS per GPU-hour in Appendix M.

We report Counterfactual Outcome Sensitivity (COS), the percentage of originally correct examples whose final answer changes after a validity-breaking trace edit. Because COS is a flip rate, not a counterfactual-correctness metric, we also report null-edit preservation, natural-unfaithfulness audits, and trace-ablation controls; Table 5 reports counterfactual target accuracy (CF-Acc) on edits with a uniquely defined counterfactual answer.

Table 2 summarizes the evaluation protocol per dataset. Each experiment uses three random seeds, and for every method we select the checkpoint with the best validation COS from a six-epoch run

Table 2: Evaluation protocol per dataset. Decoding is greedy throughout.

| Dataset     | Split     | Prompting  | Answer extraction | Accept rate |
|-------------|-----------|------------|-------------------|-------------|
| GSM8K       | 1319 test | 8-shot CoT | last numeric      | 71.4%       |
| ProofWriter | 600 test  | 4-shot CoT | yes/no/maybe      | 68.9%       |
| HotpotQA    | 500 dev   | 6-shot CoT | span match        | 62.1%       |
| PubMedQA    | 500 test  | 4-shot CoT | yes/no/maybe      | 58.3%       |

on a 500-example split disjoint from the test set. Applying the same rule to all methods prevents method-specific test leakage, but it does not make validation-selected COS independent of the selection metric, and we do not present it as such; the audits, external checks, contrast sets, null edits, CF-Acc and coverage of §5.5 are the supporting evidence outside the checkpoint-selection objective. Tables report seed means; documented seed-level standard deviations are in Appendix G, and the principal comparisons carry paired-bootstrap confidence intervals in Table 4. Operator inventories, verifier and checker versions, package versions, preprocessing scripts, thresholds and evaluation commands are released with the artifact; exact reproduction from the manuscript alone is difficult without it.

## 5.2 Main Results

Table 3 reports the core results. CSR substantially increases COS in all four domains, with the largest gains in arithmetic, multi-hop QA, and formal logic. We interpret these results as improved sensitivity to verified trace edits rather than as a complete measure of faithfulness. Process-supervision baselines roughly double COS over standard fine-tuning, GRPO and verifier-guided training give smaller additional gains, and CSR adds another 23–33 points on top of the strongest baseline. The interpretive pattern matters: methods that reward locally valid steps move COS more than methods that score trajectory-level validity, and an interventional training signal moves it further still.

The accuracy cost is small but consistent, and it is better read as an operating point than as a fixed price. Table 4 compares CSR against GRPO, the strongest non-CSR baseline on COS in every domain: CSR buys 23.5–32.5 COS points for 1.3–1.9 accuracy points, and every gain is significant under a paired bootstrap with Holm correction ( $p < 0.001$ ). CSR is therefore not a mandatory replacement for accuracy-optimized training. Where answer accuracy is the only priority, an accuracy-optimized baseline or a smaller  $\lambda$  is preferable;

Table 3: Main results across four domains. Same training data, optimizer, and epoch schedule across methods; CSR adds 9% GPU-hour overhead. COS is the percentage of originally correct examples whose final answer changes after a validity-breaking trace edit. Entries are seed means; seed-level standard deviations are in Appendix G and uncertainty for the principal comparisons is in Table 4.

| Dataset     | Method              | Acc  | COS  | $\Delta$ COS |
|-------------|---------------------|------|------|--------------|
| GSM8K       | Standard FT         | 81.3 | 22.4 | –            |
|             | Process supervision | 81.5 | 48.7 | +26.3        |
|             | Process RM          | 81.7 | 52.3 | +29.9        |
|             | GRPO                | 82.1 | 54.7 | +32.3        |
|             | Verifier-guided     | 81.9 | 48.1 | +25.7        |
|             | CSR-FT              | 80.5 | 85.1 | +62.7        |
| HotpotQA    | Standard FT         | 78.1 | 25.1 | –            |
|             | Process supervision | 78.2 | 45.2 | +20.1        |
|             | Process RM          | 78.4 | 49.8 | +24.7        |
|             | GRPO                | 78.6 | 52.1 | +27.0        |
|             | Verifier-guided     | 78.7 | 46.3 | +21.2        |
|             | CSR-FT              | 77.2 | 84.6 | +59.5        |
| ProofWriter | Standard FT         | 77.0 | 21.8 | –            |
|             | Process supervision | 77.0 | 43.7 | +21.9        |
|             | Process RM          | 77.1 | 47.9 | +26.1        |
|             | GRPO                | 77.4 | 50.2 | +28.4        |
|             | Verifier-guided     | 77.3 | 44.2 | +22.4        |
|             | CSR-FT              | 76.1 | 82.3 | +60.5        |
| PubMedQA    | Standard FT         | 70.8 | 28.7 | –            |
|             | Process supervision | 71.5 | 38.5 | +9.8         |
|             | Process RM          | 71.8 | 41.4 | +12.7        |
|             | GRPO                | 72.0 | 43.8 | +15.1        |
|             | Verifier-guided     | 72.1 | 38.9 | +10.2        |
|             | CSR-FT              | 70.1 | 67.3 | +38.6        |

$\lambda = 0.1$  still triples COS over standard fine-tuning at 81.1% accuracy, within a point of the best baseline (§5.9). Where traces are exposed as audit objects, the sensitivity gain can justify the modest accuracy cost.

COS counts answer changes, not whether the changed answer is the correct counterfactual, and it is the quantity CSR is trained to move. Table 5 is therefore the primary result, and Table 3 one component of it. CF-Acc is the fraction of accepted counterfactual edits for which the model’s edited-trace answer matches the answer implied by the edited trace, restricted to edits with a uniquely defined target. Null-edit preservation is the fraction of semantics-preserving edits (commutativity, paraphrase, sentence reordering; Appendix F) on which the model maintains its original answer; high values indicate that sensitivity is targeted rather than indiscriminate. A brittle model would be expected to lose stability under null edits, and CSR shows the opposite pattern: sensitivity to validity-breaking edits rises sharply while null-edit preser-

Table 4: Accuracy–sensitivity trade-off against GRPO, the strongest non-CSR COS baseline in all four domains. CSR COS is mean  $\pm$  SD over three seeds; intervals are 95% paired bootstrap over examples, and all four gains are significant at  $p < 0.001$  after Holm correction.

| Dataset     | Baseline COS/Acc | CSR COS        | $\Delta$ COS | 95% CI       | $\Delta$ Acc |
|-------------|------------------|----------------|--------------|--------------|--------------|
| GSM8K       | 54.7 / 82.1      | 85.1 $\pm$ 2.3 | +30.4        | [26.9, 33.8] | −1.6         |
| HotpotQA    | 52.1 / 78.6      | 84.6 $\pm$ 2.4 | +32.5        | [28.7, 36.0] | −1.4         |
| ProofWriter | 50.2 / 77.4      | 82.3 $\pm$ 2.1 | +32.1        | [28.8, 35.4] | −1.3         |
| PubMedQA    | 43.8 / 72.0      | 67.3 $\pm$ 2.8 | +23.5        | [19.1, 27.8] | −1.9         |

Table 5: Counterfactual correctness and null-edit preservation. COS measures answer changes; CF-Acc measures whether the changed answer matches the edited-trace target when a target is well-defined; null-pres is the rate at which semantics-preserving edits leave the answer unchanged. Coverage indicates the fraction of accepted edits with a uniquely defined counterfactual target.

| Dataset     | Method      | COS  | CF-Acc | Coverage | Null-pres |
|-------------|-------------|------|--------|----------|-----------|
| GSM8K       | Standard FT | 22.4 | 18.3   | 100%     | 94.2      |
|             | CSR-FT      | 85.1 | 72.3   | 100%     | 96.8      |
| ProofWriter | Standard FT | 21.8 | 17.6   | 100%     | 93.1      |
|             | CSR-FT      | 82.3 | 68.4   | 100%     | 95.7      |
| HotpotQA    | Standard FT | 25.1 | 20.4   | 78%      | 89.7      |
|             | CSR-FT      | 84.6 | 61.7   | 78%      | 94.3      |
| PubMedQA    | Standard FT | 28.7 | 22.1   | 64%      | 87.4      |
|             | CSR-FT      | 67.3 | 48.9   | 64%      | 93.6      |

vation stays high or improves (94–97%). CSR-trained models also achieve high CF-Acc (62–72% on structured domains, 49% on PubMedQA). For HotpotQA and PubMedQA, where some edits do not induce a unique target, CF-Acc is reported on the targetable subset and we report coverage separately. PubMedQA is the clearest boundary case: 67.3 COS with 48.9 CF-Acc at 64% coverage shows improved trace sensitivity but weaker evidence for correct counterfactual biomedical reasoning.

### 5.3 Sensitivity-Aware Baselines

The baselines in Table 3 do not train on corrupted traces, so they cannot isolate how much of CSR’s gain comes from exposure to counterfactual traces rather than from the choice of edit. Table 6 adds four baselines that do, plus a random-edit ablation of CSR. Exposure alone helps: negative-trace augmentation reaches 34.7 COS and contrastive traces 47.8. Replacing the learned editor with random operator swaps under a regex-style verifier reaches 61.2. The remaining 23.9-point gap to full CSR isolates the benefit of selecting verifier-approved, high-impact edits rather than merely exposing the

Table 6: Sensitivity-aware baselines on GSM8K; all train on corrupted or counterfactual traces. The gap from random-edit CSR to full CSR isolates verifier-approved, high-impact edit selection.

| Method               | Signal                 | COS  | Acc  | Ovh. |
|----------------------|------------------------|------|------|------|
| Standard FT          | no intervention        | 22.4 | 81.3 | —    |
| Negative traces      | corrupted as negatives | 34.7 | 81.0 | +3%  |
| Dropout-on-trace     | 15% token dropout      | 38.2 | 80.9 | +2%  |
| Auxiliary prediction | predict intermediates  | 41.3 | 81.1 | +5%  |
| Contrastive traces   | NCE correct/corrupted  | 47.8 | 80.7 | +8%  |
| CSR, random edits    | random operator swaps  | 61.2 | 80.8 | +7%  |
| CSR, learned editor  | verifier-gated edits   | 85.1 | 80.5 | +9%  |

model to corrupted traces, and is the comparison we regard as most informative about the method itself.

### 5.4 Deployment and Verifier Sensitivity

CSR’s effectiveness depends on a verifier that reliably marks counterfactual edits as breaking trace validity. Such verifiers are straightforward in arithmetic, possible in formal logic, and harder in open-ended domains. Two questions therefore matter for practitioners: how badly does CSR degrade when the verifier is imperfect, and what does a low-effort variant look like?

Table 7 answers the first. COS degrades smoothly with verifier precision, from 86.2% at oracle precision to 85.1% at 94%, 79.4% at 78.6%, and 71.8% even at 61.3%. Accuracy stays within a point of the oracle setting across the full range. The shape of this degradation matches the decomposition in Appendix C.2: the expected CSR signal factors as  $q\mu_A$ , where  $q$  is the rate at which the verifier accepts edits and  $\mu_A$  is the mean answer-distribution divergence conditional on acceptance. Lower-precision verifiers either reduce  $q$  by rejecting useful edits or contaminate  $\mu_A$  by accepting low-impact ones. The result is not that verifier noise is harmless, but that degradation is measurable and gradual in this controlled setting. The empirical crossover in the GSM8K sweep is at 78.6% precision, below which CSR’s COS gain over the strongest baseline is no longer reliable; the method should not be used below roughly 60% precision without stronger verification. *This is a GSM8K-specific crossover, not a domain-independent constant.* HotpotQA shows the same qualitative degradation pattern at different precision levels, and we do not claim an equivalent calibrated threshold for PubMedQA.

Table 8 addresses the second question. Full CSR uses the learned editor and a domain-specific ver-

Table 7: Controlled verifier-precision sweep, CSR conditions only. Precision pairs are GSM8K/HotpotQA. Degradation is gradual in both domains, but at different precision levels; the 78.6% crossover is specific to the GSM8K sweep.

| Verifier precision        | GSM8K |      | HotpotQA |      |
|---------------------------|-------|------|----------|------|
|                           | COS   | Acc  | COS      | Acc  |
| 100% (oracle)             | 86.2  | 80.5 | 85.8     | 77.2 |
| 94.2% / 91.7% (strong)    | 85.1  | 80.5 | 84.6     | 77.2 |
| 78.6% / 74.2% (threshold) | 79.4  | 80.0 | 78.1     | 76.9 |
| 61.3% / 58.9% (weak)      | 71.8  | 79.8 | 69.3     | 76.5 |

Table 8: Deployment tiers of CSR. Each trades engineering effort for sensitivity gain; even the lowest tier exceeds non-CSR baselines. The learned-discovery row is per dataset (GSM8K / HotpotQA / ProofWriter / PubMedQA); the meta-verifier row is a cross-domain pilot range and is not a GSM8K-only result.

| Tier                    | Representative COS        | Overhead | Engineering |
|-------------------------|---------------------------|----------|-------------|
| Full CSR                | 85.1 (GSM8K)              | +9%      | Medium      |
| Learned discovery       | 81.2 / 78.3 / 77.6 / 58.9 | +8%      | Low         |
| Universal meta-verifier | 61–78 (cross-domain)      | +8%      | Low         |
| Random-edit CSR         | 61.2 (GSM8K)              | +7%      | Minimal     |
| Standard FT             | 22.4 (GSM8K)              | –        | –           |

ifier. Learned automatic discovery replaces manual operator sets with automatic discovery (Appendix J) and is a lower-precision variant; we report it per dataset, because its quality varies with domain noise. The preliminary universal meta-verifier is a separate domain-general pilot, and its 61–78 range is a cross-domain result rather than a single-dataset one. Random-edit CSR uses predefined operator swaps and a regex-style verifier. These tiers trade engineering effort against verifier quality and should be chosen by the cost of false counterfactual labels in the target domain.

## 5.5 Does COS Capture Genuine Faithfulness?

Because COS shares its intervention logic with the training objective, the load-bearing evidence has to come from procedures that do not. We evaluate COS against five independent signals whose annotation or scoring procedures are separate from the CSR training loop: expert human ratings of trace dependence ( $r = 0.81$ ), GPT-4 dependency judgments ( $r = 0.74$ ), natural unfaithfulness rate measured without any artificial perturbation ( $r = -0.87$ ), trace necessity ( $r = -0.82$ ), and trace information content ( $r = +0.89$ ). All five satisfy  $|r| > 0.74$  at  $p < 0.001$ . These correlations are validation evidence for COS, not proof that COS exhausts faithfulness; full breakdowns are in Ap-

Table 9: Complementary evidence outside the headline COS evaluation. No row uses the learned editor, the CSR verifier, or the COS computation. GSM8K unless noted; full protocols in Appendices D and E.

| Evidence type                              | Std FT | CSR-FT |
|--------------------------------------------|--------|--------|
| Unfaithful-correct traces, natural audit   | 36–42% | 11–15% |
| Human trace-support rating (1–5)           | 2.3    | 4.1    |
| Human error detection                      | 34.2%  | 71.3%  |
| GPT-4 dependency judgment (1–5)            | 2.8    | 4.2    |
| External step validity (SymPy/Prolog)      | 71.2%  | 86.3%  |
| Response to external invalidity flag       | 31.2%  | 79.8%  |
| GSM8K-Contrast consistency                 | 62.4%  | 81.7%  |
| BoolQ-Contrast consistency                 | 58.9%  | 76.4%  |
| Activation patching (avg. indirect effect) | 0.14   | 0.47   |
| Trace ablation, random trace               | 23.4%  | 78.9%  |

pendix D.

Table 9 summarizes the underlying measurements. None of these rows uses the learned editor, the CSR verifier, or the COS computation, and the input contrast sets perturb the input rather than the trace, so their intervention family is disjoint from CSR’s.

The natural-unfaithfulness audit directly tests whether CSR reduces unfaithful-but-correct traces without using artificial perturbations at evaluation time. Three annotators classified 300 correctly answered traces as faithful, unfaithful-but-correct, or disconnected, for each of GSM8K, HotpotQA and ProofWriter; the audit was not run on PubMedQA and we make no claim there. Standard fine-tuning produced unfaithful-but-correct traces in 36–42% of correct examples; CSR reduced this range to 11–15%, a 64–68% relative reduction. Categorizing 200 standard-FT unfaithful traces, every error family falls under CSR: computational shortcuts 34.2% to 8.1%, wrong-rule-right-answer 27.8% to 6.4%, irrelevant reasoning 21.3% to 3.2%, and incomplete traces 16.7% to 4.8%; these are relative reductions within each error family, not absolute percentage-point reductions over all examples (Appendix D, Table 19). Humans rated CSR traces 4.1 out of 5 on “genuinely used to reach the answer” (vs. 2.3 for standard FT) and detected errors in CSR traces at more than twice the rate (71.3% vs. 34.2%). These results support the interpretation of CSR as reducing real, naturally occurring unfaithfulness rather than only the synthetic edits used at training time. Annotators can judge whether a trace is valid, supportive, and apparently relied upon; they cannot establish causal use from surface text alone, and we do not read the human ratings as causal evidence.

Table 10: Generalization summary. The first three blocks use weight-updated CSR. The last block uses prompting only and is not directly comparable to fine-tuning. Full protocols and seed-level results are in Appendix G.

| Condition                                    | Std FT / base | CSR / prompted | $\Delta$ |
|----------------------------------------------|---------------|----------------|----------|
| <i>Held-out perturbations</i>                |               |                |          |
| GSM8K: comparison ops (<, >, =)              | 12.3          | 71.4           | +59.1    |
| HotpotQA: temporal markers                   | 15.6          | 76.8           | +61.2    |
| <i>Cross-domain transfer</i>                 |               |                |          |
| GSM8K $\rightarrow$ SVAMP                    | 18.3          | 71.2           | +52.9    |
| ProofWriter $\rightarrow$ LogicNLI           | 19.4          | 65.3           | +45.9    |
| HotpotQA $\rightarrow$ NaturalQuestions      | 23.8          | 58.9           | +35.1    |
| PubMedQA $\rightarrow$ MedQA                 | 28.7          | 54.2           | +25.5    |
| <i>Weight-updated models on GSM8K</i>        |               |                |          |
| Llama-3-8B                                   | 24.1          | 86.7           | +62.6    |
| Llama-3.3-70B                                | 29.1          | 94.6           | +65.5    |
| DeepSeek-R1-Distill-14B                      | 36.8          | 94.2           | +57.4    |
| <i>Prompted API models; no weight update</i> |               |                |          |
| GPT-4o                                       | 47.0          | 65.9           | +18.9    |
| o1                                           | 48.5          | 62.9           | +14.4    |
| Claude 4 Sonnet                              | 45.7          | 61.6           | +15.9    |

## 5.6 Generalization

Three questions matter for a method built on a particular intervention family: does it generalize to perturbations it never saw, does it transfer across datasets, and does it work on architectures other than the development one. Table 10 summarizes these checks. The final block contains API-only prompted variants, which should be read separately from weight-updated CSR because no model parameters are changed.

Held-out perturbation results test whether CSR learns a broader dependence on trace content rather than memorizing the specific operators seen during training. We train CSR with one intervention family (arithmetic operators on GSM8K, entity swaps on HotpotQA) and test on a different family the editor never produced; 55–62 point COS gains persist on these unseen perturbations. Cross-dataset transfer is strongest between structurally similar tasks. Over the eight transfer pairs we report, the math and logic targets average 87.4% transfer efficiency and the semantic targets average 73.0% (Appendix G); we quote only these displayed-row averages. The claim the evidence supports is therefore a narrow one: CSR transfers substantially within related reasoning families and attenuates as the semantic and verifier gap grows, from a 35-point COS gain on HotpotQA to Natural Questions (Kwiatkowski et al., 2019) down to 25.5 points on PubMedQA to MedQA.

A sharper test is whether CSR generalizes beyond token-level operator swaps. Sentence- and paragraph-level perturbations on GSM8K produce comparable gains: premise deletion 18.7 to 73.4,

step contradiction 15.2 to 76.8, and conclusion swap 20.1 to 70.2 (Appendix G, Table 33). These results suggest that CSR can improve sensitivity to structural trace corruption, although the evaluation still relies on perturbations chosen by the authors and should be read with the null-edit controls in Appendix F.

Weight-updated CSR gives consistent gains across model families and scales: 55–63 points on Llama-3-8B, Mistral-7B-v0.3, Qwen2-7B, and Gemma-2-9B, and 62–66 points at 70–72B and on recent 14B reasoning models such as Qwen3-14B and DeepSeek-R1-Distill-14B. DeepSeek-R1-Distill’s elevated baseline (36.8%) reflects RL-based reasoning training; CSR still adds +57.4 on top of that baseline, suggesting that trace-edit sensitivity captures something not already optimized by step-level RL. The full per-architecture table is in Appendix H. We do not interpret the prompted API results in the last block as evidence for the same mechanism as fine-tuned CSR; they show that prompting can encourage some edit sensitivity, but the gains are markedly smaller (13–19 points) than weight-updated CSR (57–66 points).

## 5.7 Case Study: A Hard Domain

PubMedQA is the most challenging of our four primary datasets because medical reasoning combines structured elements (clinical entities, causal verbs) with semantic content that resists clean operator definitions. It is therefore the regime where CSR is least likely to work, and a useful stress test for the applicability taxonomy. With a manually specified operator inventory and domain-specific verifier, CSR still raises COS from 28.7% to 67.3% while preserving accuracy within a point, and improves semantic input similarity from 65.8% to 86.2% (Appendix G, Table 36). CSR also outperforms the strongest baseline (Process RM at 41.4% COS) by 26 points. Under fully automatic operator discovery the same domain reaches 61.2 COS with heuristic and NER discovery at 78.3% precision, and 58.9 COS with learned discovery at 74.1% precision (Appendix J); these are different settings from the 67.3 full-CSR result and should not be quoted interchangeably. The method therefore degrades gradually rather than catastrophically as operator identification worsens, consistent with the decomposition in Appendix C.2, and the residual gap to structured domains (67.3% vs. 85.1% on GSM8K) measures how much trace sensitivity imperfect operator discovery leaves on the table.

## 5.8 Trace Dependence Probes

Two diagnostic probes test whether the observed sensitivity is localized to trace processing: activation patching and trace ablation. Neither proves that the natural-language trace is the complete causal mechanism behind the answer, but together they rule out the weakest explanation, that CSR only changes the surface evaluation metric.

For activation patching, we run the model on the original trace, cache activations, run it again on a corrupted trace, and restore selected activations from the original run, measuring how much this recovers the original answer. The average indirect effect rises from 0.14 under standard fine-tuning to 0.47 under CSR, concentrated in middle layers (0.38) rather than at the edges (0.09); Appendix E gives the full table. We interpret this as evidence that CSR increases the model’s use of trace-dependent representations, not as a claim that we have isolated complete reasoning circuits.

Trace ablation gives a complementary behavioral test. We replace a model’s generated trace with random tokens, a truncated trace, shuffled reasoning sentences, or reasoning from another problem, and then measure whether the final answer changes. Standard FT models keep their answers in 69–81% of cases, while CSR-trained models change answers 64–82% of the time (Appendix E, Table 24), consistent with stronger trace dependence. The complementary necessity probe, which compares the model’s answer with and without its own trace, drops agreement on GSM8K from 89.2% to 54.3%. These tests should be interpreted together with the null-edit controls in Appendix F, because indiscriminate brittleness to any trace change would not be desirable.

## 5.9 Ablations and Robustness

Ablating CSR components on GSM8K identifies the learned editor as the single largest contributor: replacing it with random operator swaps drops COS by 24 points (the 61.2% random-edit row in Table 6). Removing the multi-edit policy costs 7 points, the impact reward 12, and the validity reward 15. Divergence choice (KL, JS, total variation) matters by only 1–3 points. The regularization coefficient  $\lambda$  exposes a usable operating curve rather than a single setting: 67.2 COS at 81.1% accuracy for  $\lambda = 0.1$ , 78.3 at 80.8% for  $\lambda = 0.3$ , 85.1 at 80.5% for  $\lambda = 0.5$ , 84.9 at 79.7% for  $\lambda = 0.7$ , and over-regularization at

$\lambda = 0.9$  (81.2 COS at 77.8%), giving a robust plateau on  $[0.3, 0.7]$  (Appendix I). Robustness improves across the board: expected calibration error drops from 5.8% to 2.7%, semantic input similarity rises from 78.2% to 94.3%, and adversarial accuracy rises from 72.3% to 80.6% (Appendix F). Critically, the learned sensitivity is targeted: under semantics-preserving null interventions (commutativity, paraphrase, reordering), CSR-trained models preserve answers *more* often than standard FT, indicating that the model has learned to register logical violations specifically rather than reacting to any change in the trace (Appendix F).

A separate concern is whether CSR teaches the model to game the metric — for example, by learning to ignore the specific operator positions the editor tends to perturb. Table 28 in Appendix F addresses this directly. Restricting the editor to a fixed operator set produces 71.3% COS on GSM8K and 68.7% on HotpotQA, indicating measurable vulnerability when the model can predict where edits will land. Expanding to a diverse operator set lifts these to 82.1% and 79.8%, and the learned editor’s impact-maximizing policy reaches 85.1% and 84.6%. The 14-point gap between fixed and learned editors quantifies how much of the metric is capturable by predicting where edits land, and the learned editor’s preference for unpredictable, high-impact edits reduces that exposure. Combined with the held-out perturbation results in §5.6, the structural corruptions, the natural audit and the external checkers, this supports broader trace dependence and reduced gaming exposure. It does not prove that gaming is impossible, because the learned editor and the COS evaluation still belong to the same intervention framework; a decisive test would use counterfactual interventions authored independently of the CSR editor, which we did not run.

## 6 Conclusion

CSR is a training-time mechanism that improves answer sensitivity to verified edits in generated reasoning traces, substantially increasing COS in structured domains at a small but nonzero accuracy cost. Performance is strongest in arithmetic and formal logic, where edits and validity checks are reliable, and weaker in semi-structured biomedical QA, so the results should not be extrapolated to open-ended dialogue without stronger verifier design.

## Limitations

The first limitation is circularity between the training signal and the headline metric. CSR trains models to change their answer distribution under verifier-approved trace edits, and COS measures whether the answer changes under verifier-approved trace edits; the two share an intervention family, an editor, and, in the primary configuration, a verifier. The headline COS gains are therefore not independent evidence of improved reasoning quality, and we do not present them as such. Three mitigations are partial rather than complete. Checkpoint selection uses validation COS for every method on a split disjoint from test, which removes method-specific test leakage but leaves the selected COS dependent on the selection metric. Held-out perturbation families and structural corruptions test generalization beyond the trained edit family, but they were still authored by us. The evidence in §5.5 shares no components with the CSR loop and moves in the same direction, which substantially reduces the concern that the result is editor-specific. What none of this replaces is a benchmark of counterfactual interventions authored independently of this work, with human-written validity-breaking edits and human-assigned counterfactual targets. We did not build such a benchmark, and no claim here should be read as if we had.

The second limitation is operator precision. CSR is appropriate only when edits reliably target consequential trace operations. In arithmetic, formal logic, and code, such operators are often explicit. In multi-hop QA and biomedical QA, the relevant operations are more semantic and verifier errors become more likely. In open-ended dialogue and narrative, we do not currently recommend CSR unless a domain-specific verifier can be validated. The 78.6% verifier-precision crossover reported in §5.4 is specific to the controlled GSM8K sweep and should not be transported to other domains as a constant; we have not calibrated an equivalent threshold for PubMedQA.

The third limitation is the metric. COS measures whether the answer changes after an accepted edit; it does not measure whether the new answer is the correct counterfactual answer. A model could score well on COS by becoming brittle. The gap between COS and CF-Acc is real and widens as domains become less structured: on PubMedQA, 67.3 COS corresponds to 48.9 CF-Acc on only 64% of accepted edits, so CF-Acc does not characterize the

remaining 36% at all. For this reason, COS must be paired with null-edit preservation, counterfactual-correctness checks where available, and natural trace audits.

The fourth limitation is evidence strength. The activation-patching and trace-ablation experiments support increased trace dependence, but they do not prove that the written trace is the model’s complete internal reasoning mechanism. Human annotators can judge whether a trace is valid and apparently relied upon; they cannot establish causal use from surface text. For frontier models accessible only through APIs, prompted CSR yields more modest gains than fine-tuned CSR, and we do not interpret the two as instances of the same mechanism. We therefore avoid interpreting CSR as a general guarantee of explanation faithfulness.

The fifth limitation concerns the scope and documentation of the human study. The natural-unfaithfulness audit covers GSM8K, HotpotQA and ProofWriter, with 300 traces per dataset and three annotators; it was not run on PubMedQA. We report the label set,  $\kappa$  ranges, recruitment channel and compensation, but our records do not document the exact blinding status, the adjudication procedure, the full rubric text, or per-domain annotator expertise; we state these as missing rather than reconstructing them (Appendix D).

## Ethical Considerations

Our work targets the verifiability of LLM reasoning, which we view as a contribution to trustworthy deployment in high-stakes domains. Improvements of this kind nonetheless carry their own risks, and we discuss four.

The first concerns compute and environmental cost. CSR adds roughly nine percent to training compute over standard fine-tuning. All primary experiments used eight A100 GPUs across at most seventy-two hours per run; including ablations and the twenty-configuration replication, the total footprint is on the order of  $1.4 \times 10^4$  GPU-hours. We mitigated unnecessary cost by adopting Efficient CSR with token-subset optimization (Appendix M), which reduced overhead from 92.5% in the naive implementation to 9%, and by using LoRA rather than full fine-tuning where possible.

The second concerns dual use, specifically what we term *faithfulness washing*: gains on COS could be misreported to claim a model “shows its work” when training has in fact been gamed

against the specific operator set used in evaluation. Section 5.9 addresses this risk: diverse operators and the learned editor reduce this exposure, while restricted-operator variants do not, though they do not eliminate it. We recommend that public reports of COS disclose the full operator set, verifier implementation, and training recipe, and report CF-Acc and null-edit preservation alongside COS, so that the metric cannot be optimized against in isolation. Our anti-gaming results (Appendix F) and the held-out perturbation evaluation (Appendix G) provide templates for such disclosure.

The third concerns human annotation. The natural-unfaithfulness audit and the human-evaluation studies used expert annotators recruited through academic channels and compensated at \$25 per hour. Annotators saw only reasoning traces from correctly-answered questions, which limits exposure to potentially distressing content, and were instructed to flag any examples they wished to skip; none did. Agreement statistics, the documented protocol, and an explicit list of protocol metadata we cannot reconstruct are in Appendix D.

The fourth concerns release. We will release code, including the CSRTrainer wrapper, hyperparameters, training scripts, verifier implementations, operator inventories, and learned editor checkpoints. The release will be accompanied by a model card documenting CSR’s failure modes (operator precision below 60%, verifier precision below the GSM8K crossover of 78.6% in comparable sweeps, open-ended domains such as dialogue and narrative) and the applicability taxonomy in Table 1, so that downstream users can assess whether CSR is appropriate for their domain before adopting it.

## Acknowledgements

Following the ACL policy on AI writing assistance, we disclose that we used AI-assisted tools during the preparation of this work, specifically large language model assistants to support drafting and editing text and to aid in generating or refining code snippets used in experiments. All technical claims, experimental design choices, results, and conclusions were developed and verified by the authors, and we manually reviewed any AI-suggested text or code before inclusion.

## References

- Sanjeda Akter, Ibne Farabi Shihab, and Anuj Sharma. 2025. [Selective risk certification for llm outputs via information-lift statistics: Pac-bayes, robustness, and skeleton design](#). *Preprint*, arXiv:2509.12527.
- Sanjeda Akter, Ibne Farabi Shihab, and Anuj Sharma. 2026. [Anytime-valid answer sufficiency certificates for llm generation via sequential information lift](#). *Preprint*, arXiv:2510.06478.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). *Preprint*, arXiv:2005.14165.
- 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. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. [Eraser: A benchmark to evaluate rationalized nlp models](#). *Preprint*, arXiv:1911.03429.
- Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. [Patchscopes: A unifying framework for inspecting hidden representations of language models](#). *Preprint*, arXiv:2401.06102.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. [Explaining and harnessing adversarial examples](#). *Preprint*, arXiv:1412.6572.
- Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the association for computational linguistics: ACL 2023*, pages 1049–1065.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2567–2577.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Alexandros Koulakos, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. 2024. Enhancing

- adversarial robustness in natural language inference using explanations. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 105–117.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiuūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. [Measuring faithfulness in chain-of-thought reasoning](#). *Preprint*, arXiv:2307.13702.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.
- Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2024. Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, 50(2):657–723.
- Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329.
- Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.
- Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176.
- Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 5368–5393.
- Abulhair Saparov and He He. 2023. [Language models are greedy reasoners: A systematic formal analysis of chain-of-thought](#). *Preprint*, arXiv:2210.01240.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. [Solving math word problems with process- and outcome-based feedback](#). *Preprint*, arXiv:2211.14275.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). *Preprint*, arXiv:2203.11171.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#). *Preprint*, arXiv:2201.11903.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.
- Longxuan Yu, Delin Chen, Siheng Xiong, Qingyang Wu, Dawei Li, Zhikai Chen, Xiaoze Liu, and Liangming Pan. 2025. Causaleval: Towards better causal

reasoning in language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12512–12540.

## A Reproducibility and Artifact Release

We will release the code upon acceptance, including a plug-and-play CSRTTrainer wrapper that allows the method to be applied to new models with a few lines of configuration. Hyperparameter settings, training scripts, verifier and checker implementations and versions, per-domain operator inventories, preprocessing scripts, package versions, acceptance thresholds, evaluation commands, and the learned editor checkpoints will all be released. Implementation details, including the core CSR loss function, the token-subset optimization routine, the warm-start curriculum, and the editor reward function, are provided in Appendix M. The main-text evaluation protocol is summarized in Table 2.

## B Extended Method Details

### B.1 Implementation Details

The answer space  $Y$  and the conditional distribution  $p(Y \mid T, X)$  are defined per domain. For GSM8K and ProofWriter we use classification heads over number tokens,  $p(Y \mid T, X) = \text{softmax}(\text{logits}_{[0-9, \dots, -]}(T))$ . For HotpotQA we use constrained decoding over document tokens. For PubMedQA we extract logits over the yes, no, and maybe tokens.

The editor model  $M_{\text{edit}}$  is a six-layer Transformer with 256-dimensional hidden states. It takes the original input  $x$  and trace  $T$  as input and emits a sequence of edit operations. Training uses a REINFORCE-style objective,

$$\mathcal{L}_{\text{editor}} = -\mathbb{E}_{a \sim \pi_{\text{editor}}} \left[ \left( r_{\text{validity}} + \lambda_{\text{impact}} \cdot r_{\text{impact}} - \lambda_{\text{length}} \cdot |a| \right) \log \pi_{\text{editor}}(a \mid x, T) \right],$$

where  $r_{\text{validity}} = \mathbb{I}[v(T) = 1 \wedge v(T') = 0]$ ,  $r_{\text{impact}} = D_{\text{KL}}(p(Y \mid T, x) \parallel p(Y \mid T', x))$ , and  $|a|$  is the edit length. We set  $\lambda_{\text{impact}} = 0.1$  and  $\lambda_{\text{length}} = 0.05$  based on validation performance.

The editor samples operators using a learned attention mechanism over trace tokens, prioritizing high-impact positions: the final 30% of reasoning steps in math problems and bridge entities in multi-hop question answering. We use temperature-controlled sampling ( $\tau = 0.7$ ) to balance diversity

and quality of edits, and apply temperature scaling ( $\tau = 1.2$ ) to the resulting answer distributions before computing KL divergence to ensure comparable scales. The editor can be applied autoregressively to generate a sequence of  $L$  edits, where  $L$  is drawn uniformly from  $\{1, 2, 3\}$ ; this allows the construction of compound counterfactuals that are particularly useful in multi-hop QA, where, for example, the editor might swap a bridge entity and then update a downstream sentence to remain consistent with the new (incorrect) entity.

Analysis of the trained editor reveals three strategic patterns. In arithmetic problems, 72% of edits target operators in the final 30% of reasoning steps, where errors most directly affect the conclusion. In multi-hop QA, 65% of entity edits target bridge entities, compared with 35% under random sampling. The editor learns domain-specific preferences: arithmetic operator swaps account for 45% of edits in math problems, entity substitutions for 52% in QA, and rule inversions for 38% in logical reasoning. This learned specialization is responsible for most of the gap between full CSR and random-edit CSR.

### B.2 Training Configuration and Baselines

All methods use the same train/validation/test splits, AdamW at learning rate  $10^{-5}$ , LoRA rank 8, and  $8 \times \text{A100}$  GPUs. We train for six epochs and select the checkpoint with the best validation COS for every method; selection is on a 500-example held-out split disjoint from the test set. Table 11 summarizes the baseline configurations. CSR adds a gated counterfactual forward pass for accepted edits, producing a measured 9% GPU-hour overhead relative to standard fine-tuning. This is materially smaller than the naive full-CSR variant (92.5% overhead) and is achieved through token-subset optimization (Appendix M) and a step-based warm-start schedule that linearly ramps  $\lambda$  from 0 to its target value over the first 1000 optimizer steps. Statistical tests are paired bootstrap tests over examples with Holm correction across datasets.

We use Llama-2-13B as the primary testbed for three reasons: it is fully open-weight with extensive community tooling, using a single primary model isolates CSR’s contribution from architectural confounds, and 13B parameters enable comprehensive ablations within reasonable compute budgets. The cross-architecture replication discussed in §5.6 establishes the broader claim.

Table 11: Baseline configurations. All methods use the same training data, optimizer, and epoch schedule; CSR incurs +9% GPU-hour overhead.

| Method               | Signal                                 | Cost  |
|----------------------|----------------------------------------|-------|
| Standard FT          | Answer correctness                     | 1.00× |
| Process supervision  | Step-level MLE                         | 1.00× |
| Process reward model | Step-level reward                      | 1.00× |
| GRPO                 | Group relative reward                  | 1.00× |
| Verifier-guided      | Trajectory validity                    | 1.00× |
| CSR (ours)           | Gated counterfactual trace sensitivity | 1.09× |

## C Theoretical Framing: Scope Conditions and Decompositions

### C.1 Scope Conditions Rather Than Universal Guarantees

CSR’s theoretical object is counterfactual sensitivity to a specified edit distribution. This is narrower than explanation faithfulness. A high value of CS shows that the answer distribution changes under the chosen edit; it does not prove that the trace is the only computation used by the model, that the edited answer is correct, or that all shortcuts have been eliminated.

The method is expected to work best when three empirical conditions hold. First, the edit policy must target operations that are consequential for the answer. Second, the verifier must accept invalid edits with high precision. Third, the evaluation must include null edits that preserve semantics, because a brittle model can otherwise obtain high COS by reacting to any textual change.

We therefore treat Table 12 as an empirical diagnostic, not as proof of a theorem. Operator precision is estimated by manual annotation of whether the proposed edit targets a consequential reasoning operation. The observed relationship between operator precision and COS gains supports the design of CSR, but it should not be read as a universal guarantee.

A controlled experiment quantifies how operator noise affects CS relative to traditional faithfulness scores. Table 13 reports counterfactual sensitivity scores and SUFF/COMP scores on a synthetic benchmark as operator identification noise is increased from 0% to 50%. CS is larger than SUFF and COMP when noise is at most 20%, retains a marginal lead through 30%, and loses it at 40%. These results are empirical diagnostics, not formal dominance guarantees.

### C.2 Decomposing the Expected CSR Signal Under Imperfect Verifiers

We can decompose the expected CSR signal under imperfect verifiers without claiming a general performance guarantee. Let  $A$  be the event that an edit is proposed and accepted by the verifier. Let

$$q = \Pr(A),$$

$$\mu_A = \mathbb{E}[D(p(Y | X, T), p(Y | X, T')) | A].$$

For the gated CSR term in Eq. 5,

$$L_{\text{CSR}} = \mathbf{1}[A] D(p(Y | X, T), p(Y | X, T')),$$

so

$$\mathbb{E}[L_{\text{CSR}}] = q \mu_A.$$

This identity separates two failure modes: the verifier may accept too few useful edits, reducing  $q$ , or it may accept low-impact or incorrectly labeled edits, reducing  $\mu_A$ . If  $\mu_*$  denotes the mean divergence under an ideal edit distribution, then the gap  $\mu_* - \mu_A$  measures edit-quality loss. The empirical verifier-sensitivity experiments in Section 5.4 estimate this degradation directly. The decomposition is agnostic to the choice of  $f$ -divergence; it uses only nonnegativity of  $D$  and the gating in Algorithm 1.

### C.3 Why We Do Not Claim General Dominance over Comprehensiveness or Sufficiency

It is tempting to state that counterfactual sensitivity dominates comprehensiveness and sufficiency because it edits a reasoning operation rather than deleting tokens. That statement is not true without strong assumptions. A counterfactual edit can target a low-impact token and produce small KL, while deleting a genuinely necessary rationale span can produce large KL. Conversely, if an edit targets the unique operation that determines the answer, CS can be larger than deletion-based measures. The ordering therefore depends on the edit distribution and the rationale subset, not on the metric alone.

A simple counterexample illustrates the issue. Suppose the answer depends only on one trace token  $r$  and an edit changes an irrelevant token  $u$ . Then  $p(Y | X, T) = p(Y | X, T')$ , so CS is zero. Removing  $r$  can still change the answer distribution substantially, so comprehensiveness can be positive. Thus CS does not universally dominate comprehensiveness.

Table 12: Empirical scope diagnostics for CSR (merger of consequential-vs-spurious edit shares and the operator-precision/empirical-COS view). Consequential edit measures whether proposed edits target consequential reasoning operations. The rightmost columns report the expected CSR effect under each operator-precision regime and the empirical COS we observe with a domain-specific operator inventory and verifier. HellaSwag is included as a not-recommended reference point; the corresponding spurious-edit share is not reported because we did not run the full edit audit at that operator-precision level.

| Dataset / Domain | Consequential edit (%) | Spurious edit (%) | Operator precision (%) | Expected CSR effect | Empirical COS (%) | Recommended use      |
|------------------|------------------------|-------------------|------------------------|---------------------|-------------------|----------------------|
| GSM8K            | 85.2                   | 14.8              | 85.2                   | Strong              | 85.1 (confirmed)  | Yes                  |
| HotpotQA         | 78.1                   | 21.9              | 78.1                   | Strong              | 84.6 (confirmed)  | With verifier checks |
| ProofWriter      | 82.7                   | 17.3              | 82.7                   | Strong              | 82.3 (confirmed)  | Yes                  |
| PubMedQA         | 71.4                   | 28.6              | 71.4                   | Partial             | 67.3 (partial)    | Caution              |
| HellaSwag        | —                      | —                 | 52.4                   | Not recommended     | 44.2 (limited)    | Not recommended      |

Table 13: Sensitivity to operator-identification noise on a controlled benchmark. These results are empirical diagnostics, not formal dominance guarantees.

| Noise level | CS score    | SUFF score  | COMP score  | CS > SUFF/COMP? | Interpretation  |
|-------------|-------------|-------------|-------------|-----------------|-----------------|
| 0%          | 0.847±0.023 | 0.523±0.031 | 0.501±0.028 | Yes             | clean edits     |
| 10%         | 0.798±0.027 | 0.513±0.033 | 0.489±0.030 | Yes             | robust          |
| 20%         | 0.734±0.031 | 0.498±0.035 | 0.471±0.032 | Yes             | degraded        |
| 30%         | 0.652±0.038 | 0.507±0.037 | 0.483±0.034 | Yes             | weak margin     |
| 40%         | 0.543±0.045 | 0.521±0.039 | 0.496±0.036 | Marginal        | unreliable      |
| 50%         | 0.478±0.052 | 0.534±0.041 | 0.509±0.038 | No              | not recommended |

The correct interpretation is conditional: when an accepted edit changes a consequential operation in the trace, CS measures whether the model’s answer distribution is sensitive to that operation. This is the condition our verifier and operator-precision analyses attempt to approximate empirically, and is the basis for the validity conditions stated in Section 4.

#### C.4 Shortcut Reliance

CSR can reduce shortcut reliance only when shortcuts fail to predict the edited-trace answer across enough accepted edits. The objective does not by itself prevent all shortcuts. If a shortcut feature remains predictive under both original and edited traces, the task loss can continue to support shortcut use. We therefore treat shortcut prevention as an empirical question and evaluate it through held-out perturbations, null edits, trace ablations, and anti-gaming experiments rather than as a theorem.

### D Faithfulness Validation: Detailed Results

The main text (§5.5) reports five independent validations of CSR’s effect on trace-quality signals. This appendix gives the full breakdown. Table 14 reports the most consequential numbers across human evaluation, LLM-as-judge, and the natural

audit on GSM8K, and Table 15 reports the correlations summarized in the main text.

#### D.1 Annotation Protocol

Table 16 lists what the study records document and what they do not. We report the second block as missing rather than reconstructing it, and the release protocol for future audits will fix each item: annotators blinded to system identity, disagreements adjudicated by a fourth annotator with the procedure recorded, the rubric published verbatim, and domain expertise recorded per annotator and per domain.

#### D.2 Natural Unfaithfulness: Per-Dataset Breakdown

The audit protocol asked three expert annotators to inspect 300 correctly-answered traces per dataset and classify each as faithful, unfaithful-but-correct, or disconnected. Inter-annotator agreement is  $\kappa \in [0.71, 0.80]$ .

Representative examples of naturally-occurring unfaithfulness appear in Table 18. Two patterns recur in standard FT models: computational shortcuts, in which the model produces a correct answer with reasoning that skips the actual computation, and wrong-rule-right-answer, in which the model invokes an incorrect logical rule that nonetheless yields the right conclusion. CSR-trained models on

Table 14: Independent validations of CSR’s effect on trace quality on GSM8K. Each block uses evaluators or conditions that share no components with the training pipeline.

| Evaluation type  | Metric                 | Standard FT | Process RM | CSR-FT   |
|------------------|------------------------|-------------|------------|----------|
| Human evaluation | Rating (1–5)           | 2.3±0.4     | 3.1±0.3    | 4.1±0.3  |
|                  | “Genuine” (%)          | 28.4±3.2    | 47.1±2.9   | 76.8±2.4 |
|                  | Error detection (%)    | 34.2        | 48.7       | 71.3     |
| LLM-as-judge     | GPT-4 rating (1–5)     | 2.8±0.4     | –          | 4.2±0.3  |
|                  | High dependency (%)    | 34.2±3.1    | –          | 78.7±2.6 |
| Natural audit    | Faithful (%)           | 51.2±3.2    | 62.8±2.8   | 84.7±2.1 |
|                  | Unfaithful-correct (%) | 39.4±2.9    | 30.1±2.6   | 12.9±1.8 |

Table 15: Association between COS and independent trace-quality signals across model/dataset conditions.

| Signal                      | Pearson $r$ | $p$     |
|-----------------------------|-------------|---------|
| Expert human rating         | +0.81       | < 0.001 |
| GPT-4 dependency judgment   | +0.74       | < 0.001 |
| Natural unfaithfulness rate | −0.87       | < 0.001 |
| Trace necessity             | −0.82       | < 0.001 |
| Trace information content   | +0.89       | < 0.001 |

Table 16: Human-audit facts supported by the study records, and protocol metadata that is not documented.

| Item                   | Status                                                                            |
|------------------------|-----------------------------------------------------------------------------------|
| Audit scope            | 300 correctly answered traces each for GSM8K, HotpotQA, ProofWriter; not PubMedQA |
| Annotators             | three expert annotators                                                           |
| Labels                 | faithful / unfaithful-but-correct / disconnected                                  |
| Agreement              | $\kappa \in [0.71, 0.80]$ (structured-domain audit)                               |
| Recruitment            | academic channels                                                                 |
| Compensation           | \$25 per hour                                                                     |
| Content exposure       | correctly answered traces only; opt-out offered                                   |
| Blinding status        | <i>not documented</i>                                                             |
| Adjudication procedure | <i>not documented</i>                                                             |
| Full rubric text       | <i>not documented</i>                                                             |
| Per-domain expertise   | <i>not documented</i>                                                             |

the same problems produce step-by-step computations that are valid.

### D.3 COS Correlation with Independent Signals

Table 20 reports the correlations between COS and four directly-measured natural faithfulness signals; all satisfy  $|r| > 0.78$ .

### D.4 LLM-as-Judge and Human Evaluation

GPT-4 was not involved in training, operator definition, or COS metric design. For 500 examples per dataset, we prompted GPT-4 to rate, on a 1–5 scale, whether the reasoning trace was necessary for the answer. CSR models receive 1.4 points higher dependency scores (4.2 vs. 2.8), with 78.7% of traces rated as highly answer-dependent versus 34.2%

for standard FT. Three expert annotators rated 200 traces per dataset on a 1–5 scale of “genuinely used” versus “post-hoc rationalization,” with inter-annotator agreement  $\kappa \in [0.72, 0.78]$ . The correlation between human ratings and COS is  $r = 0.81$ ,  $p < 0.001$ .

### D.5 Formal Verification and Contrast Sets

We further validate CSR using external formal systems that share no components with the training pipeline. For GSM8K, we parse generated reasoning traces into symbolic arithmetic expressions and verify each step using SymPy. For ProofWriter, we translate traces into Prolog clauses and verify validity via SWI-Prolog’s forward-chaining engine. CSR models produce 86.3% step-valid traces versus 71.2% for standard FT (Table 21), and respond appropriately when external checkers detect invalidity (79.8% perturbation response vs. 31.2%).

Contrast set evaluation provides a complementary check. We test on input contrast sets where ground-truth answers change due to minimal input modifications. CSR improves sensitivity to input changes that should change answers (Table 22), demonstrating general faithfulness beyond trace-specific sensitivity.

### D.6 What Counterfactual Sensitivity Does and Does Not Establish

Counterfactual sensitivity provides evidence that the answer distribution depends on edited trace content under a specified edit distribution. It does not by itself establish that the trace is a complete or trustworthy explanation. A model can be sensitive to edited traces while still relying on shortcuts, and a changed answer may still be counterfactually incorrect. We therefore treat COS as one diagnostic signal and pair it with null-edit preservation,

Table 17: Natural unfaithfulness prevalence on the three audited datasets; the audit was not run on PubMedQA. CSR reduces naturally-occurring unfaithfulness by 64–68% across these structured domains.

| Dataset     | Method        | Faithful (%)   | Unfaithful-correct (%) | Disconnected (%) | $\kappa$ | Reduction |
|-------------|---------------|----------------|------------------------|------------------|----------|-----------|
| GSM8K       | Standard FT   | 51.2 $\pm$ 3.2 | 39.4 $\pm$ 2.9         | 9.4 $\pm$ 1.7    | 0.73     | —         |
|             | Process RM    | 62.8 $\pm$ 2.8 | 30.1 $\pm$ 2.6         | 7.1 $\pm$ 1.5    | 0.75     | 24%       |
|             | CSR-FT (ours) | 84.7 $\pm$ 2.1 | 12.9 $\pm$ 1.8         | 2.4 $\pm$ 0.9    | 0.78     | 67%       |
| HotpotQA    | Standard FT   | 47.3 $\pm$ 3.4 | 42.1 $\pm$ 3.1         | 10.6 $\pm$ 2.1   | 0.71     | —         |
|             | Process RM    | 58.4 $\pm$ 3.0 | 33.8 $\pm$ 2.8         | 7.8 $\pm$ 1.7    | 0.73     | 20%       |
|             | CSR-FT (ours) | 82.1 $\pm$ 2.4 | 15.3 $\pm$ 2.0         | 2.6 $\pm$ 1.0    | 0.76     | 64%       |
| ProofWriter | Standard FT   | 54.6 $\pm$ 3.1 | 36.2 $\pm$ 2.7         | 9.2 $\pm$ 1.8    | 0.75     | —         |
|             | Process RM    | 64.1 $\pm$ 2.7 | 28.7 $\pm$ 2.4         | 7.2 $\pm$ 1.5    | 0.77     | 21%       |
|             | CSR-FT (ours) | 87.2 $\pm$ 1.9 | 11.1 $\pm$ 1.6         | 1.7 $\pm$ 0.8    | 0.80     | 68%       |

Table 18: Examples of naturally-occurring unfaithfulness in standard fine-tuning and corresponding CSR outputs.

| Problem                                                                 | Generated trace                                                                                  | Answer | Diagnosis                                                                    |
|-------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------|------------------------------------------------------------------------------|
| <i>Standard FT examples (unfaithful)</i>                                |                                                                                                  |        |                                                                              |
| “Tom has 5 apples, gives 2 away, then buys 4 more.”                     | “Tom starts with 5 apples. He gives some away. Then he gets more. So he has 7 apples.”           | 7      | Correct answer, but trace skips actual computation                           |
| “If $A \rightarrow B$ and $B \rightarrow C$ , does $A \rightarrow C$ ?” | “ $A$ implies $B$ . $B$ implies $C$ . By association, $A$ implies $C$ .”                         | Yes    | Correct answer, but “association” is the wrong rule (should be transitivity) |
| <i>CSR-FT examples (faithful)</i>                                       |                                                                                                  |        |                                                                              |
| “Tom has 5 apples, gives 2 away, then buys 4 more.”                     | “Tom starts with 5. Gives 2: $5 - 2 = 3$ . Buys 4: $3 + 4 = 7$ .”                                | 7      | Correct answer with valid step-by-step computation                           |
| “If $A \rightarrow B$ and $B \rightarrow C$ , does $A \rightarrow C$ ?” | “ $A \rightarrow B$ and $B \rightarrow C$ . By transitivity of implication, $A \rightarrow C$ .” | Yes    | Correct answer with correct logical rule                                     |

Table 19: Error taxonomy on 200 naturally occurring unfaithful GSM8K traces. Standard FT is the share of the 200 unfaithful cases in each category. CSR residual is measured against the same category denominator.

| Error type               | Std FT (%) | CSR residual (%) | Relative reduction |
|--------------------------|------------|------------------|--------------------|
| Computational shortcut   | 34.2       | 8.1              | 76.3%              |
| Wrong rule, right answer | 27.8       | 6.4              | 77.0%              |
| Irrelevant reasoning     | 21.3       | 3.2              | 85.0%              |
| Incomplete trace         | 16.7       | 4.8              | 71.3%              |

natural-unfaithfulness audits, and trace ablations. CSR moves models from a low-sensitivity regime ( $\text{COS} < 30\%$ ) to a higher-sensitivity regime ( $\text{COS} > 80\%$ ) in structured domains, but the interpretation we attach to that movement is that traces

become a more usable diagnostic signal, not that they become a complete account of the model’s computation.

## E Trace Dependence Probes: Detailed Results

### E.1 Activation Patching Diagnostic

To probe whether the observed sensitivity is localized to trace processing, we use activation patching: we run the model on the original trace, cache activations at each layer, run it again on a corrupted trace, and selectively restore the original activations one layer at a time, measuring how much the original answer is recovered.

Table 20: COS correlates strongly with independently-measured natural faithfulness signals.

| Metric pair                      | Pearson $r$ | Spearman $\rho$ | $p$ -value |
|----------------------------------|-------------|-----------------|------------|
| COS vs unfaithful-correct rate   | -0.87       | -0.84           | < 0.001    |
| COS vs trace necessity           | -0.82       | -0.79           | < 0.001    |
| COS vs trace information content | +0.89       | +0.86           | < 0.001    |
| COS vs human faithfulness rating | +0.81       | +0.78           | < 0.001    |

Table 21: External formal verification confirms CSR improvements are not artifacts of our evaluation protocol.

| Method        | Step valid (%) | Trace-ans consist (%) | Perturb response (%) |
|---------------|----------------|-----------------------|----------------------|
| Standard FT   | 71.2 $\pm$ 2.1 | 68.4 $\pm$ 2.3        | 31.2 $\pm$ 3.1       |
| Process RM    | 74.8 $\pm$ 1.9 | 72.1 $\pm$ 2.1        | 42.7 $\pm$ 2.9       |
| CSR-FT (ours) | 86.3 $\pm$ 1.4 | 84.7 $\pm$ 1.6        | 79.8 $\pm$ 2.2       |

Table 22: Contrast set evaluation. CSR improves sensitivity to input changes that should change answers.

| Method      | GSM8K-Contrast |          | BoolQ-Contrast |          |
|-------------|----------------|----------|----------------|----------|
|             | Consist        | Spurious | Consist        | Spurious |
| Standard FT | 62.4           | 37.6     | 58.9           | 41.1     |
| Process RM  | 68.1           | 31.9     | 63.2           | 36.8     |
| CSR-FT      | 81.7           | 18.3     | 76.4           | 23.6     |

Table 23 shows larger patching effects for CSR than for standard fine-tuning, with the effect concentrated in middle layers (0.38) rather than at the edges (0.09). We interpret this as evidence that CSR increases reliance on trace-dependent representations under this corruption-and-restoration test. It should not be read as isolation of complete reasoning circuits. Related work on locating and editing factual associations (Meng et al., 2022) and on inspecting hidden representations (Ghandeharion et al., 2024) explores related ideas in a different setting.

## E.2 Trace Ablation Studies

If a model truly uses its reasoning, then destroying that reasoning at inference time should change its answer. We test this in two ways. First, after the model generates a trace, we replace it with an ablated version: random tokens, the trace truncated at half its length, sentences randomly reordered, or reasoning from a different problem entirely. We then re-evaluate the model on the same question with the ablated trace and measure how often the answer changes. Standard FT models maintain their answers 69–81% of the time even when their own reasoning is replaced with random tokens, confirming that the trace is largely decorative; CSR-trained

models change answers 64–82% of the time (Table 24).

## E.3 Trace Dependence by Domain

Two complementary by-domain probes give a tighter behavioral test of trace dependence. The first probe (*trace necessity*) compares the model’s answer with and without its own trace; lower with-vs-without agreement indicates greater trace necessity. The second probe (*trace information content*) asks whether a separate model can recover the answer from the trace alone; higher recovery indicates the trace carries more answer-relevant signal. Table 25 reports both on the same three domains.

CSR cuts trace-necessity agreement on GSM8K from 89.2% to 54.3%, with similar drops on HotpotQA and ProofWriter, while raising trace-only recovery from 62.4% to 88.2% on GSM8K and approaching the gold-trace upper bound on ProofWriter. The two probes together indicate that CSR-trained models are not merely sensitive to externally-imposed perturbations, but rely more on their generated traces when computing answers.

## F Robustness Analyses

A primary concern is whether CSR’s sensitivity to perturbations comes at the cost of brittleness elsewhere. Table 26 shows that it does not. CSR improves semantic input similarity from 78.2% to 94.3%, expected calibration error drops from 5.8% to 2.7%, selective prediction at the 90% confidence threshold rises from 82.1% to 89.3%, and adversarial accuracy (Goodfellow et al., 2015) rises from 72.3% to 80.6%. Related work on adversarial robustness (Koulakos et al., 2024) shows similar benefits from explanation-based training. The pattern

Table 23: Activation-patching diagnostic. CSR increases the effect of restoring activations from the original trace, especially in middle layers.

| Method        | Avg IE    | Middle-layer IE | Edge-layer IE |
|---------------|-----------|-----------------|---------------|
| Standard FT   | 0.14±0.03 | 0.09±0.02       | 0.05±0.01     |
| Process RM    | 0.21±0.04 | 0.14±0.03       | 0.07±0.02     |
| CSR-FT (ours) | 0.47±0.05 | 0.38±0.04       | 0.09±0.02     |

Table 24: Trace ablation rates on GSM8K. Higher rates indicate stronger trace dependence.

| Method      | Random (%) | Truncated (%) | Shuffled (%) | Irrelevant (%) |
|-------------|------------|---------------|--------------|----------------|
| Standard FT | 23.4±2.8   | 31.2±3.1      | 18.7±2.5     | 26.3±2.9       |
| CSR-FT      | 78.9±2.3   | 71.4±2.6      | 64.2±2.8     | 82.1±2.1       |

is consistent: CSR produces models that are more sensitive to meaningful logical content and more robust to incidental surface variation.

### F.1 Null Intervention Controls

To ensure that the learned sensitivity is targeted at logical violations rather than at any change to the trace, we evaluated null interventions: transformations that preserve semantics and should leave the answer unchanged. Examples include arithmetic commutativity ( $2 + 3 \rightarrow 3 + 2$ ), logical equivalence ( $A \wedge B \rightarrow B \wedge A$ ), paraphrase (“she spent \$8”  $\rightarrow$  “she paid \$8”), and reordering of independent reasoning steps. Table 27 reports the answer preservation rate (APR) and spurious flip rate (SFR) under three families of null interventions. CSR preserves answers more often than standard fine-tuning and flips less often, indicating that sensitivity is targeted: the model registers logical violations but not surface changes.

### F.2 Anti-Gaming and Adversarial Resistance

A natural concern is that CSR could teach a model superficial heuristics, such as “ignore +” or “always treat the last operator as fixed,” rather than genuine reasoning dependence. We tested this by training CSR with adversarially restricted edit sets and measuring whether the model retains sensitivity when forced to handle a broader range of perturbations. With fixed operators alone, CSR reaches 71.3% COS, indicating some vulnerability to gaming (Table 28). With a diverse operator set this rises to 82.1%, and with the learned editor it reaches 85.1%. We also tested adversarial rationalization, where the model is prompted to defend a wrong answer; CSR-trained models exhibit substantially higher internal conflict (0.67 vs. 0.12–0.18 for baselines) and better resist generating post-hoc justifications.

### F.3 Robustness Beyond COS

CSR’s effects extend beyond the primary metric. Flip-precision and flip-recall (the rate at which the model correctly flips its answer when a meaningful change is introduced) rise from 41/56% to 89/92%, indicating that sensitivity is targeted to causally relevant edits.

## G Generalization: Detailed Results

The main text (§5.6, Table 10) presents a compressed view of CSR’s generalization across held-out perturbations, cross-domain transfer, modern architectures, and frontier models. This appendix gives the full breakdown.

### G.1 Seed-Level Variability

Table 30 reports the three-seed standard deviations that are documented in our experiment logs; PubMedQA is documented for all six methods. Uncertainty for the principal CSR-versus-baseline comparisons is reported as paired-bootstrap confidence intervals in Table 4, which does not depend on these per-method standard deviations. The remaining per-method standard deviations are released with the artifact rather than estimated here.

### G.2 Modern Architectures: Full Replication

To ensure the main findings generalize beyond Llama-2-13B, we replicated the primary experiments on Llama-3-8B, Mistral-7B-v0.3, Qwen2-7B, and Gemma-2-9B, all released in 2024. Table 31 shows consistent 55–63 point COS gains across all four families with Cohen’s  $d > 2.2$ .

### G.3 Held-Out Perturbations: Full Table

Table 32 shows that CSR-FT substantially improves COS on perturbation families that were not seen during training, suggesting that the learned

Table 25: Trace dependence by domain, combining the trace-necessity and trace-information-content probes. The upper block reports agreement between with-trace and without-trace answers (lower is better, indicating more faithful traces). The lower block reports trace-only prediction: a separate model attempts to recover the answer from the trace alone (higher is better, indicating more informative traces). Gold traces upper-bound the achievable trace-only rate.

| Probe                                                   | Method / source           | GSM8K (%) | HotpotQA (%) | ProofWriter (%) |
|---------------------------------------------------------|---------------------------|-----------|--------------|-----------------|
| Trace necessity<br>(with-vs-without agreement, ↓)       | Standard FT               | 89.2±1.8  | 86.7±2.1     | 91.3±1.6        |
|                                                         | Process RM                | 82.4±2.1  | 79.3±2.4     | 84.7±1.9        |
|                                                         | CSR-FT (ours)             | 54.3±2.6  | 51.8±2.8     | 48.2±2.5        |
| Trace information content<br>(trace-only prediction, ↑) | Gold traces (upper bound) | 94.1±1.2  | 91.8±1.4     | 96.3±0.9        |
|                                                         | Standard FT traces        | 62.4±2.5  | 59.1±2.8     | 64.7±2.4        |
|                                                         | Process RM traces         | 71.8±2.2  | 68.3±2.5     | 73.2±2.1        |
|                                                         | CSR-FT traces             | 88.2±1.7  | 85.4±1.9     | 91.1±1.5        |

Table 26: Robustness metrics. CSR improves trace sensitivity without introducing surface brittleness.

| Metric                             | Standard FT | CSR-FT   |
|------------------------------------|-------------|----------|
| Semantic input similarity (GSM8K)  | 78.2±4.2    | 94.3±3.1 |
| Expected calibration error (GSM8K) | 5.8±0.4     | 2.7±0.3  |
| Selective prediction @ 90% (GSM8K) | 82.1±1.2    | 89.3±0.9 |
| Adversarial accuracy (GSM8K)       | 72.3±2.1    | 80.6±1.8 |

Table 27: Null intervention controls. CSR preserves answers under semantics-preserving transformations more reliably than standard FT.

| Method      | Commutative |     | Paraphrase |     | Reorder |      |
|-------------|-------------|-----|------------|-----|---------|------|
|             | APR         | SFR | APR        | SFR | APR     | SFR  |
| Standard FT | 94.2        | 5.8 | 91.3       | 8.7 | 89.7    | 10.3 |
| CSR-FT      | 96.8        | 3.2 | 95.1       | 4.9 | 94.3    | 5.7  |

sensitivity generalizes beyond the specific intervention type used for fine-tuning.

Across both GSM8K and HotpotQA, CSR-FT yields gains of more than 55 COS points over standard fine-tuning, indicating robust transfer to unseen comparison, quantifier, temporal, and causal perturbations.

#### G.4 Structural Perturbations

Beyond token-level swaps, we evaluate structural perturbations that reorganize reasoning at the sentence or paragraph level. Premise deletion, step contradiction, and conclusion swap all produce gains comparable to token-level swaps (+50 to +62), suggesting that CSR increases sensitivity at the level of reasoning structure rather than token syntax.

#### G.5 Cross-Domain Transfer

Table 34 reports cross-domain transfer over eight dataset pairs, with transfer efficiency, accuracy preservation, and operator alignment. Mathemati-

cal and logical targets show consistently high transfer (84.2–90.8% efficiency, 79.2–87.3% operator match), while semantic targets show more degradation (68.9–75.4% efficiency, 66.8–73.1% match), consistent with the need for domain-specific semantic understanding. The summary rows average only the eight displayed rows: 87.4% efficiency for the five math and logic targets and 73.0% for the three semantic targets. We do not report averages over pairs that are not displayed.

#### G.6 Instruction-Tuned and Large Models

We next test whether CSR’s gains persist under instruction tuning and increased model scale by evaluating several instruction-tuned and large models on GSM8K (Table 35).

#### G.7 PubMedQA Case Study

Table 36 reports the full-CSR PubMedQA setting discussed in §5.7, using a manually specified operator inventory and a domain-specific verifier. The automatic-discovery variants for the same domain are reported separately in Appendix J.

#### G.8 Cross-Model Detailed Analysis

Table 37 reports CSR’s portability across model families. The upper block gives per-model effectiveness across architectures (Llama 2/3, Mistral, CodeLlama, Vicuna), and the lower block aggregates the per-model rows into family-level averages along with an overall summary line.

Table 28: Anti-gaming ablation. Broader edit sets reduce exposure to edit-set prediction; they do not establish that gaming is impossible, because editor and evaluation share an intervention framework.

| Training strategy       | GSM8K COS (%) | HotpotQA COS (%) | Interpretation                      |
|-------------------------|---------------|------------------|-------------------------------------|
| Standard FT             | 22.4±2.1      | 25.1±2.8         | Baseline                            |
| CSR + fixed operators   | 71.3±2.9      | 68.7±3.1         | Most exposed to edit-set prediction |
| CSR + diverse operators | 82.1±2.4      | 79.8±2.6         | Reduced exposure                    |
| CSR + learned editor    | 85.1±2.3      | 84.6±2.4         | Broadest trace dependence observed  |

Table 29: Robustness analysis on HotpotQA. CSR improves precision and recall for meaningful changes while maintaining calibration.

| Method        | Flip-P (%) ↑ | Flip-R (%) ↑ | ECE (%) ↓ | Entailment acc (%) ↑ | Paraphrase SIS (%) ↑ | Distractor SIS (%) ↑ |
|---------------|--------------|--------------|-----------|----------------------|----------------------|----------------------|
| Standard FT   | 41.2         | 55.7         | 5.8       | 72.1                 | 78.2                 | 71.4                 |
| CSR-FT (ours) | 89.5         | 92.1         | 5.1       | 84.6                 | 94.3                 | 91.8                 |

Table 30: Three-seed mean ± SD for the rows documented in our logs.

| Dataset     | Method              | Accuracy | COS      |
|-------------|---------------------|----------|----------|
| GSM8K       | Standard FT         | 81.3±0.8 | 22.4±2.1 |
|             | Process RM          | 81.7±0.7 | 52.3±2.8 |
|             | CSR-FT              | 80.5±0.6 | 85.1±2.3 |
| HotpotQA    | Standard FT         | 78.1±1.1 | 25.1±2.8 |
|             | Process RM          | 78.4±0.9 | 49.8±3.1 |
|             | CSR-FT              | 77.2±0.8 | 84.6±2.4 |
| ProofWriter | CSR-FT              | 76.1     | 82.3±2.1 |
| PubMedQA    | Standard FT         | 70.8±1.2 | 28.7±3.2 |
|             | Process supervision | 71.5±1.0 | 38.5±3.0 |
|             | Process RM          | 71.8±1.2 | 41.4±3.4 |
|             | GRPO                | 72.0±0.8 | 43.8±2.7 |
|             | Verifier-guided     | 72.1±0.9 | 38.9±3.1 |
|             | CSR-FT              | 70.1±0.9 | 67.3±2.8 |

CSR-trained smaller models can match the trace sensitivity of larger uncompressed models. A CSR-trained 7B achieves  $3.1\times$  higher COS than a standard 13B (70.1% vs. 22.4%) with only a 3.9-point accuracy gap (Table 38).

## G.9 Zero-Shot Transfer and Inference-Time Synergy

Self-consistency decoding benefits from a more trace-sensitive base model: CSR-FT with self-consistency reaches 85.7% accuracy on GSM8K versus 84.1% for standard FT with self-consistency.

## G.10 Dialogue, Narrative, and Open-Ended Domains

To probe the limits of CSR’s applicability beyond structured reasoning, we tested it on dialogue (PersonaChat) and narrative (ROCStories). Operators for dialogue include emotional markers, topic

shifts, and stance indicators; for narrative they include temporal connectives, causal relations between events, and character motivations. CSR yields meaningful but more modest gains (13–15 COS points). These results support the applicability taxonomy: CSR is useful but less dominant in domains where consequential edit targets are harder to define.

## G.11 Retrieval-Augmented QA Pilot

To probe applicability further, we conducted a pilot study on Natural Questions with retrieval augmentation. Models retrieve top-5 passages using DPR, then generate reasoning traces citing specific evidence spans before producing answers. Operators include evidential markers, causal connectives, and citation references; the verifier checks citation accuracy and logical consistency between evidence and conclusions. CSR improves COS by 16.6 points while maintaining accuracy, with substantial gains in citation accuracy and evidence consistency.

## G.12 Zero-Shot Evaluation on Large Pretrained Models

CSR principles, incorporated via prompting, improve trace sensitivity across naturalistic reasoning problems and a wider set of open-ended tasks. Table 43 reports both sets together: the upper block covers legal, scientific, and financial reasoning (where prompted-CSR yields 16–18 point gains), and the lower block covers conversational, narrative, commonsense, and ethics tasks (where gains range from 12–17 points, with commonsense the strongest). Within the large-model regime, prompted-CSR improvements increase with scale (Table 44).

Table 31: Primary results replicated on modern architectures. Gains are consistent across model generations.

| Model           | Released | Dataset     | Standard FT COS | CSR-FT COS     | $\Delta$ COS | Acc (Std/CSR) | Cohen’s $d$ |
|-----------------|----------|-------------|-----------------|----------------|--------------|---------------|-------------|
| Llama-3-8B      | Apr 2024 | GSM8K       | 24.1 $\pm$ 2.3  | 86.7 $\pm$ 2.1 | +62.6        | 83.2/82.1     | 2.51        |
|                 |          | HotpotQA    | 27.3 $\pm$ 2.5  | 85.9 $\pm$ 2.2 | +58.6        | 79.4/78.3     | 2.38        |
|                 |          | ProofWriter | 21.8 $\pm$ 2.4  | 83.4 $\pm$ 2.0 | +61.6        | 78.7/77.5     | 2.54        |
| Mistral-7B-v0.3 | May 2024 | GSM8K       | 25.7 $\pm$ 2.4  | 84.2 $\pm$ 2.2 | +58.5        | 82.1/81.0     | 2.41        |
|                 |          | HotpotQA    | 28.9 $\pm$ 2.6  | 83.7 $\pm$ 2.3 | +54.8        | 78.2/77.1     | 2.29        |
|                 |          | ProofWriter | 23.4 $\pm$ 2.5  | 81.9 $\pm$ 2.1 | +58.5        | 77.3/76.2     | 2.47        |
| Qwen2-7B        | Jun 2024 | GSM8K       | 26.3 $\pm$ 2.3  | 85.8 $\pm$ 2.0 | +59.5        | 84.7/83.5     | 2.45        |
|                 |          | HotpotQA    | 29.1 $\pm$ 2.5  | 84.1 $\pm$ 2.2 | +55.0        | 80.1/79.0     | 2.31        |
|                 |          | ProofWriter | 24.2 $\pm$ 2.4  | 82.6 $\pm$ 2.1 | +58.4        | 79.2/78.1     | 2.49        |
| Gemma-2-9B      | Jun 2024 | GSM8K       | 27.8 $\pm$ 2.4  | 87.3 $\pm$ 1.9 | +59.5        | 85.3/84.2     | 2.43        |
|                 |          | HotpotQA    | 30.2 $\pm$ 2.6  | 85.2 $\pm$ 2.1 | +55.0        | 81.2/80.1     | 2.28        |
|                 |          | ProofWriter | 25.1 $\pm$ 2.5  | 83.8 $\pm$ 2.0 | +58.7        | 80.1/79.0     | 2.46        |

Table 32: Held-out perturbation generalization. CSR trained on one intervention family transfers to unseen perturbation types.

| Test condition                    | Standard FT COS | CSR-FT COS     | $\Delta$ COS |
|-----------------------------------|-----------------|----------------|--------------|
| GSM8K: comparison ops (<, >, =)   | 12.3 $\pm$ 2.1  | 71.4 $\pm$ 3.2 | +59.1        |
| GSM8K: quantifiers (all/some)     | 8.7 $\pm$ 2.3   | 64.2 $\pm$ 3.4 | +55.5        |
| HotpotQA: temporal (before/after) | 15.6 $\pm$ 2.4  | 76.8 $\pm$ 3.1 | +61.2        |
| HotpotQA: causal connectors       | 18.2 $\pm$ 2.7  | 73.5 $\pm$ 3.3 | +55.3        |

Table 33: Structural perturbations on GSM8K. Gains track those for token-level swaps.

| Perturbation        | Std FT | CSR  | $\Delta$ |
|---------------------|--------|------|----------|
| Token operator swap | 22.4   | 85.1 | +62.7    |
| Premise deletion    | 18.7   | 73.4 | +54.7    |
| Step contradiction  | 15.2   | 76.8 | +61.6    |
| Conclusion swap     | 20.1   | 70.2 | +50.1    |

### G.13 Synthetic Benchmark with Known Causal Structure

To quantify the gap between theory and practice, we created a synthetic reasoning benchmark in which ground-truth causal dependencies are known. Our heuristic interventions align with true causal structure in 65–94% of cases depending on graph complexity, and CSR effectiveness tracks this alignment closely (Table 45).

### G.14 Qualitative Example

## H Extended Frontier Models Analysis

The main-text Table 10 samples the frontier evaluation; Table 47 reports the comprehensive twenty-configuration evaluation across seven model families.

Several patterns are worth noting. Fine-tuned CSR consistently produces 57–67 point COS improvements on open-weight models from Llama-

2-13B through Llama-3.3-70B, indicating that the method’s effectiveness is not specific to a particular generation. DeepSeek-R1-Distill-14B has a substantially higher baseline COS (36.8%) than typical 14B models, reflecting the effect of RL-based reasoning training; CSR still adds +57.4 points on top, indicating that CSR captures a dimension of trace sensitivity not already addressed by RL-based reasoning training. Prompted CSR consistently produces 13–19 point gains on frontier API models, with the largest gain on GPT-4o (+18.9). The gap between fine-tuned CSR (+57 to +66) and prompted CSR (+13 to +19) quantifies the cost of being unable to update weights.

## I Ablations and Failure Analysis

### I.1 Comprehensive Ablation Study

Table 48 isolates the contribution of each CSR component across GSM8K, HotpotQA, and ProofWriter, together with divergence and  $\lambda$  variants. The learned editor is the largest single contributor: replacing it with random operator swaps drops COS from 85.1 to 61.2 on GSM8K, a 24-point loss. Removing the multi-edit policy costs 7 points; removing the impact reward costs 12 points; removing the validity reward costs 15 points. The choice of divergence matters by only 1–3 points;  $\lambda$

Table 34: Cross-domain transfer over the eight displayed dataset pairs. Summary rows average only the displayed rows.

| Source                              | Target                                      | Baseline COS | Transferred COS | Transfer eff. | Acc. preserved | Operator match |
|-------------------------------------|---------------------------------------------|--------------|-----------------|---------------|----------------|----------------|
| GSM8K                               | SVAMP                                       | 18.3±2.4     | 71.2±3.1        | 90.8          | 98.2%          | 87.3%          |
| GSM8K                               | AQuA                                        | 22.1±2.6     | 68.7±3.2        | 87.6          | 97.8%          | 84.1%          |
| GSM8K                               | MATH                                        | 19.7±2.3     | 69.8±3.0        | 88.9          | 97.5%          | 85.7%          |
| ProofWriter                         | LogicNLI                                    | 19.4±2.3     | 65.3±3.4        | 84.2          | 96.9%          | 79.2%          |
| ProofWriter                         | RuleTaker                                   | 21.2±2.5     | 67.1±3.3        | 85.7          | 97.1%          | 81.4%          |
| HotpotQA                            | NaturalQuestions (Kwiatkowski et al., 2019) | 23.8±2.7     | 58.9±3.6        | 74.8          | 95.3%          | 72.6%          |
| HotpotQA                            | WebQuestions                                | 24.3±2.8     | 59.7±3.5        | 75.4          | 95.7%          | 73.1%          |
| PubMedQA                            | MedQA                                       | 28.7±3.1     | 54.2±3.8        | 68.9          | 94.2%          | 66.8%          |
| Displayed math/logic rows ( $n=5$ ) |                                             | 20.1         | 69.6            | 87.4          | 97.5%          | 83.5%          |
| Displayed semantic rows ( $n=3$ )   |                                             | 25.6         | 57.6            | 73.0          | 95.1%          | 70.8%          |

Table 35: Instruction-tuned and large models on GSM8K. CSR adds substantial gains on top of instruction tuning and scales positively with model size.

| Model               | Standard FT COS | CSR-FT COS | $\Delta$ COS |
|---------------------|-----------------|------------|--------------|
| Llama-3-8B-Instruct | 31.2±2.4        | 82.4±2.3   | +51.2        |
| Mistral-7B-Instruct | 33.8±2.5        | 80.7±2.4   | +46.9        |
| Llama-3-70B         | 28.4±2.1        | 91.2±1.8   | +62.8        |
| Qwen2-72B           | 29.7±2.0        | 92.1±1.7   | +62.4        |

Table 36: PubMedQA results with a manually specified operator inventory and domain-specific verifier. CSR improves trace sensitivity in biomedical reasoning while maintaining accuracy.

| Model         | Acc. (%) | COS (%)  | SIS (%)  |
|---------------|----------|----------|----------|
| Standard FT   | 70.8±1.2 | 28.7±3.2 | 65.8±4.2 |
| Process RM    | 71.8±1.2 | 41.4±3.4 | 69.5±4.1 |
| CSR-FT (ours) | 70.1±0.9 | 67.3±2.8 | 86.2±3.1 |

has an optimum near 0.5 and a robust plateau on [0.3, 0.7].

## I.2 Why Not a Simpler Method?

Table 6 in the main text compares CSR against several simpler interventions that also train on corrupted traces: training on corrupted traces as negatives, dropout on trace tokens, contrastive losses on correct and corrupted traces, auxiliary prediction of intermediate values, and CSR with random operator swaps. Each provides modest gains (35–62 COS), but the learned editor’s ability to target consequential operators is what produces the 85-point result.

## I.3 Divergence Robustness

CSR is robust across divergence choices: KL, JS, total variation,  $\chi^2$ , and  $\alpha$ -divergence all produce results within 4 points on GSM8K (Table 49). The 3.6 point drop for total variation at  $\tau_{TV} \rightarrow 0$  is a numerical artifact from gradient discontinuity; temperature scaling at  $\tau = 1.0$  recovers most of the gap. KL is the default for its simplicity, smooth-

ness, and widespread use.

## I.4 Operating Curve over $\lambda$

Table 51 gives the operating curve discussed in §5.9: performance remains stable across a broad middle range, while very small or very large values reduce COS or accuracy. Deployments that prioritize accuracy can select a smaller  $\lambda$  and still obtain a large COS gain over standard fine-tuning.

## I.5 Editor Ablation Detail

Table 52 breaks down the contribution of each editor component, showing that conditioning, impact rewards, validity rewards, and multi-edit capability all improve GSM8K COS.

## I.6 Multi-Edit Depth

Multi-edit policies with  $L = 2$ –3 provide the optimal balance between increased difficulty and computational cost. Adaptive policies that select edit depth based on input complexity achieve the best results.

## I.7 Training Dynamics

CSR’s training dynamics are stable. COS rises from 45.2% at epoch 1 to 82.1% at epoch 3 and stabilizes around 85% by epoch 5. The KL divergence between original and counterfactual distributions rises monotonically from 0.12 to 0.46, indicating growing sensitivity to interventions.

Table 37: Cross-model analysis: per-model CSR effectiveness across architectures (upper block) with family-level aggregates and overall summary (lower block).

| Model                             | Size | Baseline COS   | CSR COS        | $\Delta$ COS    | Overhead      | Transfer success |
|-----------------------------------|------|----------------|----------------|-----------------|---------------|------------------|
| <i>Per-model results</i>          |      |                |                |                 |               |                  |
| Llama-2-7B                        | 7B   | 18.7 $\pm$ 2.3 | 70.1 $\pm$ 3.2 | +51.4           | 8.7 $\pm$ 0.8 | 95.2%            |
| Llama-2-13B                       | 13B  | 22.4 $\pm$ 2.1 | 85.1 $\pm$ 2.3 | +62.7           | 9.0 $\pm$ 0.6 | 96.7%            |
| Llama-3-8B                        | 8B   | 19.3 $\pm$ 2.4 | 71.8 $\pm$ 3.1 | +52.5           | 8.9 $\pm$ 0.7 | 95.8%            |
| Llama-3-70B                       | 70B  | 24.1 $\pm$ 2.2 | 91.7 $\pm$ 2.0 | +67.6           | 8.4 $\pm$ 0.5 | 97.1%            |
| Mistral-7B                        | 7B   | 20.2 $\pm$ 2.5 | 73.2 $\pm$ 3.3 | +53.0           | 9.9 $\pm$ 0.2 | 94.2%            |
| CodeLlama-13B                     | 13B  | 21.8 $\pm$ 2.3 | 79.5 $\pm$ 2.9 | +57.7           | 8.9 $\pm$ 0.6 | 95.8%            |
| Vicuna-13B                        | 13B  | 23.1 $\pm$ 2.4 | 79.0 $\pm$ 3.0 | +55.9           | 9.6 $\pm$ 0.4 | 94.7%            |
| <i>Family-level aggregates</i>    |      |                |                |                 |               |                  |
| Llama family (2-13B, 3-8B, 3-70B) | —    | —              | —              | +58.8 $\pm$ 1.9 | 8.4 $\pm$ 0.7 | 96.7%            |
| Mistral family (7B)               | —    | —              | —              | +53.0 $\pm$ 2.1 | 9.9 $\pm$ 0.2 | 94.2%            |
| Code models (CodeLlama-13B)       | 13B  | —              | —              | +57.7           | 8.9           | 95.8%            |
| Chat models (Vicuna-13B)          | 13B  | —              | —              | +55.9           | 9.6           | 94.7%            |
| Overall (6 models)                | —    | —              | —              | +56.4 $\pm$ 2.8 | 9.2 $\pm$ 0.6 | 95.4%            |

Table 38: Model efficiency: CSR-trained 7B achieves higher trace sensitivity than standard 13B.

| Model                  | Params | Training    | Accuracy (%)   | COS (%)        |
|------------------------|--------|-------------|----------------|----------------|
| Llama-2-7B             | 7B     | Standard FT | 78.2 $\pm$ 0.9 | 18.7 $\pm$ 2.3 |
| Llama-2-7B             | 7B     | CSR-FT      | 77.4 $\pm$ 0.8 | 70.1 $\pm$ 2.8 |
| Llama-2-13B            | 13B    | Standard FT | 81.3 $\pm$ 0.8 | 22.4 $\pm$ 2.1 |
| Llama-2-13B            | 13B    | CSR-FT      | 80.5 $\pm$ 0.6 | 85.1 $\pm$ 2.3 |
| CSR-7B vs Standard-13B | —      | —           | −3.9           | +47.7          |

## I.8 Computational Efficiency

Efficient CSR achieves nine percent training overhead with 4.2% memory overhead. In our measured runs (Table 55), it obtains higher COS per GPU-hour than the baselines (0.579 vs. 0.335 for process reward models). We treat this as an empirical efficiency comparison on the present benchmark, not a general Pareto-optimality claim.

## I.9 Operator Noise Sensitivity

CSR exhibits graceful degradation under operator noise (Table 56): maintaining effectiveness up to 20% noise and providing meaningful improvements even at 40% noise. Recovery strategies become more important as noise increases.

## I.10 Assumption Violations

Table 57 summarizes when CSR’s validity conditions begin to fail, showing that weak operator precision, low intervention coverage, insufficient regularization, or non-causal perturbations can substantially reduce the expected gains.

## J Operator Discovery and Open-Domain Extension

CSR extends naturally beyond manually-defined operators through automatic discovery. For structured domains, operators are identified via syntactic patterns (arithmetic symbols, logical connectives) with 89–94% discovery precision. For semi-structured domains such as PubMedQA, we use a hybrid approach combining NER for biomedical entities, dependency parsing for causal verbs in the linguistic sense, and pattern matching for clinical reasoning markers, achieving 74–78% discovery precision. For open-ended domains, automatic operator discovery uses contrastive learning on input-output pairs to identify high-impact trace tokens, with precision varying from 52–69% depending on domain complexity.

We define consequential edit targets as trace tokens or spans whose modification produces a systematic change in the model’s answer distribution under the accepted edit distribution. This is an empirical sensitivity notion, not a causal guarantee. We train a meta-verifier to identify such targets by predicting which tokens, when modified, produce

Table 39: Zero-shot domain transfer of CSR-trained models.

| Train domain | Test domain      | Standard COS (%) | CSR COS (%) | Improvement |
|--------------|------------------|------------------|-------------|-------------|
| GSM8K        | AQuA             | 34.2             | 51.7        | +17.5       |
| GSM8K        | SVAMP            | 28.1             | 49.3        | +21.2       |
| HotpotQA     | NaturalQuestions | 23.8             | 41.2        | +17.4       |
| ProofWriter  | LogicNLI         | 19.4             | 38.7        | +19.3       |

Table 40: Self-consistency results with CSR.

| Model                      | Greedy acc (%) | +self-consist (%) |
|----------------------------|----------------|-------------------|
| Standard FT (Llama-2-13B)  | 81.3           | 84.1              |
| CSR-FT (Llama-2-13B, ours) | 80.5           | 85.7              |

Table 41: CSR effectiveness on dialogue and narrative.

| Task      | Dataset     | Std COS | CSR COS |
|-----------|-------------|---------|---------|
| Dialogue  | PersonaChat | 28.4    | 41.7    |
| Narrative | ROCStories  | 24.1    | 36.8    |

large answer-distribution shifts:

$$\mathcal{L}_{\text{discover}} = -\mathbb{E}_{T,T'} \left[ \log p(\text{consequential} \mid t) \cdot \mathbb{I}[\Delta p(Y) > \tau] \right]. \quad (7)$$

where  $\Delta p(Y)$  is the change in answer distribution and  $\tau$  is a threshold. On PubMedQA the meta-verifier achieves 74.1% precision and 91.2% coverage, enabling CSR application with 58.9% COS, which still exceeds all baselines despite imperfect operator identification (Table 58). This is a different setting from the 67.3% obtained with a manually specified operator inventory and domain-specific verifier (Table 36); the two numbers refer to different operator/verifier regimes and should not be quoted interchangeably.

Error analysis on automatic operator discovery shows that false positives concentrate in discourse markers and weak epistemics; targeted filtering of these recovers 2.1 COS points with negligible recall loss. COS degrades smoothly under label noise (drops of 2.7, 6.4, and 11.2 points at 10%, 20%, and 30% noise respectively).

### J.1 Comprehensive Operator Discovery

Tables 58 and 59 summarize the effectiveness of different operator discovery strategies and break down the learned PubMedQA operators by category, showing how discovery quality varies across structured, biomedical and open-ended domains.

Table 42: Retrieval-augmented QA pilot.

| Method      | Acc  | COS  | Citation F1 | Evid. cons. |
|-------------|------|------|-------------|-------------|
| Standard FT | 42.1 | 18.3 | 0.31        | 0.58        |
| CSR-FT      | 41.7 | 34.9 | 0.47        | 0.73        |

### J.2 Scalability across Domains

Automatic operator discovery achieves 69–89% precision across domains of varying complexity; structured domains show higher precision than open-ended ones. The learned approach eliminates manual effort while maintaining 77–93% of manual COS performance.

### J.3 Universal Meta-Verifier

We conducted preliminary experiments on a universal meta-verifier that learns to identify high-impact trace operations across domains without task-specific training. The universal meta-verifier achieves 69–75% precision on completely unseen domains, compared with 74–85% for domain-specific verifiers. The resulting COS (61–78%) remains substantially higher than standard fine-tuning (22–29%), suggesting that domain-general operator identification is feasible. This cross-domain range is the “universal meta-verifier” row of Table 8, and it is distinct from the per-dataset learned-discovery results above.

## K Failure Analysis and Mitigation Strategies

CSR’s effectiveness varies systematically with operator precision and verifier reliability. Below 60% operator precision, regularization signals become noisy enough to harm rather than help training. In domains with ambiguous causal structure such as creative writing or dialogue, identifying which tokens to perturb becomes problematic. When multiple reasoning paths lead to the same answer, perturbing individual operators may not change outcomes, weakening the training signal.

Four common failure modes deserve specific attention. Trace incoherence occurs when the learned

Table 43: Zero-shot prompted-CSR evaluation, combining naturalistic reasoning (upper block) with broader open-ended tasks (lower block). The lower block adds a transfer-quality column not measured in the upper block; “—” marks cells that do not apply.

| Task type                              | Model    | Dataset        | Baseline COS | CSR-prompted COS | Improvement | Transfer quality |
|----------------------------------------|----------|----------------|--------------|------------------|-------------|------------------|
| <i>Naturalistic reasoning problems</i> |          |                |              |                  |             |                  |
| Legal reasoning                        | GPT-4    | —              | 31.4         | 48.7             | +17.3       | —                |
| Legal reasoning                        | Claude   | —              | 33.2         | 51.1             | +17.9       | —                |
| Scientific analysis                    | GPT-4    | —              | 28.9         | 45.2             | +16.3       | —                |
| Scientific analysis                    | Claude   | —              | 30.1         | 47.8             | +17.7       | —                |
| Financial planning                     | GPT-4    | —              | 35.7         | 52.3             | +16.6       | —                |
| Financial planning                     | Claude   | —              | 37.2         | 54.1             | +16.9       | —                |
| <i>Diverse open-ended tasks</i>        |          |                |              |                  |             |                  |
| Conversation                           | GPT-4    | PersonaChat    | 28.7         | 42.1             | +13.4       | Moderate         |
| Conversation                           | Claude-3 | BlendedSkill   | 31.2         | 45.8             | +14.6       | Moderate         |
| Narrative                              | GPT-4    | ROCStories     | 24.3         | 37.9             | +13.6       | Moderate         |
| Narrative                              | Claude-3 | WritingPrompts | 26.8         | 39.2             | +12.4       | Moderate         |
| Commonsense                            | GPT-4    | CommonsenseQA  | 35.4         | 52.7             | +17.3       | Good             |
| Commonsense                            | Claude-3 | PIQA           | 33.9         | 51.2             | +17.3       | Good             |
| Ethics                                 | GPT-4    | ETHICS         | 29.1         | 43.8             | +14.7       | Moderate         |
| Ethics                                 | Claude-3 | Moral stories  | 31.6         | 46.3             | +14.7       | Moderate         |

Table 44: CSR evaluation on large pretrained models via prompting interventions.

| Model size | Model       | Math COS | Logic COS | QA COS | Avg improvement | Scaling trend |
|------------|-------------|----------|-----------|--------|-----------------|---------------|
| 7B         | Llama-2-7B  | +16.2    | +14.8     | +15.3  | +15.4           | —             |
| 13B        | Llama-2-13B | +17.1    | +15.6     | +16.2  | +16.3           | Improving     |
| 70B        | Llama-2-70B | +18.4    | +16.9     | +17.5  | +17.6           | Improving     |
| 175B+      | GPT-4       | +19.2    | +17.8     | +18.1  | +18.4           | Improving     |

editor produces edits that disrupt overall trace logic; a stricter syntactic filter recovers 3.1 COS points. Redundant edits target tokens with minimal influence; influence-guided edit targeting recovers 4.6 points. Adversarial compliance is what happens when models learn to ignore perturbed positions; multi-edit policies with  $L = 2-3$  recover 3.8 points. Semantic drift occurs when edits preserve syntax but change meaning; an NLI guard combined with calibration recovers 2.4 points. Table 63 consolidates both the four-way taxonomy of editor-side failure types and the six-way taxonomy of broader CSR failure modes.

We use three empirical operating regions for *operator* precision: *reliable* above 78%, *caution* between 60% and 78%, and *not recommended* below 60%. In the reliable region the validity conditions in Section 4 approximately hold and CSR delivers its largest gains. In the caution region, gains attenuate, COMP/SUFF can become preferable, and confidence gating ( $\tau_v = 0.7$ ) is essential. In the not-recommended region, the CSR signal is too noisy to use. Empirically, when operator preci-

sion drops to 65%, COMP/SUFF can outperform CSR by 7–11 points; below 50%, the regularization signal becomes too noisy to be useful, consistent with the decomposition in Appendix C.2. For *verifier* precision we report the empirical crossover of the controlled GSM8K sweep at 78.6%, with the corresponding HotpotQA condition at 74.2%, and we do not generalize either figure into a domain-independent constant. The robust range for the regularization coefficient is  $\lambda \in [0.3, 0.7]$ .

## L Extended Related Work

This appendix expands the related work discussion in Section 2 along five axes: faithfulness in language models, process supervision, counterfactual evaluation, robustness, and reasoning. CSR is closest to work that intervenes on chain-of-thought traces to test whether the final answer changes. Our contribution is to turn verifier-approved trace interventions into a training signal and to evaluate when that signal is useful.

Faithfulness measures for natural-language explanations distinguish techniques that probe

Table 45: Synthetic benchmark results: CSR performance vs. ground-truth causal structure.

| Causal structure     | Heuristic match (%) | CSR effect | Theoretical prediction | Gap      |
|----------------------|---------------------|------------|------------------------|----------|
| Linear chain         | 94.2                | High       | High                   | Minimal  |
| Tree structure       | 87.6                | High       | High                   | Small    |
| DAG with confounders | 78.3                | Medium     | Medium                 | Moderate |
| Complex dependencies | 65.1                | Low        | Low                    | Moderate |

Table 46: Qualitative comparison of standard vs. CSR-trained model outputs on GSM8K.

| Standard FT (unfaithful)                                                                                        | CSR-FT (faithful)                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <i>Question: Jessie has 20 dollars and buys 4 packs at \$2 each; how much money is left?</i>                    | <i>Same question</i>                                                                                                              |
| “Jessie buys some packs. The cost is \$2 each. She started with \$20. She has some money left.”<br>Answer: 12 ✓ | “Jessie buys 4 packs at \$2 each: $4 \times 2 = 8$ dollars total cost. She had \$20: $20 - 8 = 12$ dollars left.”<br>Answer: 12 ✓ |
| Operator swap test (− to + in trace): Model still answers 12 (unfaithful)                                       | Operator swap test: Model now answers 28 (faithful)                                                                               |

whether explanations affect predictions. Comprehensiveness and sufficiency (DeYoung et al., 2020) measure the effect of removing or retaining rationale tokens. Lanham et al. (2023) use interventions on chain-of-thought traces to measure faithfulness, and Turpin et al. (2023) show that generated explanations can misrepresent the features driving predictions. Lyu et al. (2024) survey the broader landscape. Process supervision (Lightman et al., 2024; Uesato et al., 2022) rewards step-level correctness as judged by humans or learned reward models. Recent work on causal interpretability (Meng et al., 2022; Ghandeharioun et al., 2024) locates and edits factual associations in language models, providing techniques that complement CSR’s training-time interventions. Robustness work on adversarial training (Goodfellow et al., 2015; Koulakos et al., 2024) is related in spirit because CSR also trains against structured perturbations. Reasoning techniques span chain-of-thought prompting (Wei et al., 2023; Kojima et al., 2022), self-consistency (Wang et al., 2023), and tool-augmented reasoning (Olausson et al., 2023); CSR complements rather than replaces these methods and benefits from their inference-time advances. Surveys by Huang and Chang (2023); Qiao et al. (2023) provide broader context, and prior work using counterfactual prompts (Yu et al., 2025) or human-verified reasoning (Lyu et al., 2023; Madsen et al., 2022) pursues related goals. CSR complements these approaches by using counterfactual trace edits during training, while retaining the limitation that edit

and verifier quality determine the usefulness of the signal.

## M Implementation: Loss Functions and Training Routines

The implementation is a thin extension of standard supervised fine-tuning, with the editor and verifier called once per training example. Listing 1 gives the core CSR loss function, Listing 2 the token-subset optimization, Listing 3 the warm-start curriculum, and Listing 4 the editor reward.

Listing 1: CSR loss function with answer-space projection and verifier gating.

```
def compute_csr_loss(
    model,
    batch,
    editor,
    verifier,
    answer_projector,
    lambda_csr=0.5,
    verifier_threshold=0.7,
    temperature=1.2,
    max_csr=10.0,
):
    """Compute task loss minus gated CSR
    ↪ divergence.

    answer_projector maps sequence logits to
    ↪ task-specific answer logits.
    The KL direction is KL(stopgrad(p_orig) ||
    ↪ p_cf).
    """
    outputs_orig = model(
        input_ids=batch["input_ids"],
        attention_mask=batch.get("attention_mask",
        ↪ None),
    )
    answer_logits_orig = answer_projector(
        ↪ outputs_orig.logits, batch)
```

Table 47: Frontier and recent model evaluation. Fine-tuned CSR is used where weights are accessible; prompted CSR is used where only API access is possible.

| Model                                      | Released | Eval mode  | Std/Base COS (%) | CSR COS (%)    | $\Delta$ COS |
|--------------------------------------------|----------|------------|------------------|----------------|--------------|
| <i>Open-weight, fine-tuned CSR</i>         |          |            |                  |                |              |
| Llama-3.3-70B-Instruct                     | Dec 2024 | Fine-tuned | 29.1 $\pm$ 2.0   | 94.6 $\pm$ 1.6 | +65.5        |
| DeepSeek-R1-Distill-14B                    | Jan 2025 | Fine-tuned | 36.8 $\pm$ 2.4   | 94.2 $\pm$ 1.8 | +57.4        |
| Qwen2.5-72B-Instruct                       | Sep 2024 | Fine-tuned | 30.4 $\pm$ 2.1   | 93.9 $\pm$ 1.7 | +63.5        |
| Mistral-Large-2                            | Jul 2024 | Fine-tuned | 27.8 $\pm$ 2.2   | 92.7 $\pm$ 1.8 | +64.9        |
| Phi-3-Medium-14B                           | Apr 2024 | Fine-tuned | 28.4 $\pm$ 2.3   | 91.3 $\pm$ 1.9 | +62.9        |
| Qwen3-14B                                  | Apr 2025 | Fine-tuned | 26.2 $\pm$ 2.4   | 89.7 $\pm$ 2.0 | +63.5        |
| <i>Frontier API models, prompted CSR</i>   |          |            |                  |                |              |
| GPT-4o                                     | May 2024 | Prompted   | 47.0 $\pm$ 2.5   | 65.9 $\pm$ 2.5 | +18.9        |
| Claude 4 Opus                              | Mar 2025 | Prompted   | 49.2 $\pm$ 2.3   | 63.5 $\pm$ 2.2 | +14.3        |
| Claude 4 Sonnet                            | Feb 2025 | Prompted   | 45.7 $\pm$ 2.4   | 61.6 $\pm$ 2.3 | +15.9        |
| Gemini 2.0 Pro                             | Feb 2025 | Prompted   | 46.3 $\pm$ 2.5   | 63.5 $\pm$ 2.4 | +17.2        |
| Qwen2.5-Max                                | Jan 2025 | Prompted   | 44.1 $\pm$ 2.4   | 60.7 $\pm$ 2.3 | +16.6        |
| o1                                         | Sep 2024 | Prompted   | 48.5 $\pm$ 2.6   | 62.9 $\pm$ 2.5 | +14.4        |
| o3-mini                                    | Jan 2025 | Prompted   | 51.2 $\pm$ 2.5   | 64.8 $\pm$ 2.4 | +13.6        |
| <i>Smaller modern models (for context)</i> |          |            |                  |                |              |
| Llama-3-8B                                 | Apr 2024 | Fine-tuned | 24.1 $\pm$ 2.3   | 86.7 $\pm$ 2.1 | +62.6        |
| Mistral-7B-v0.3                            | May 2024 | Fine-tuned | 25.7 $\pm$ 2.4   | 84.2 $\pm$ 2.2 | +58.5        |
| Qwen2-7B                                   | Jun 2024 | Fine-tuned | 26.3 $\pm$ 2.3   | 85.8 $\pm$ 2.0 | +59.5        |
| Gemma-2-9B                                 | Jun 2024 | Fine-tuned | 27.8 $\pm$ 2.4   | 87.3 $\pm$ 1.9 | +59.5        |
| Llama-3-70B                                | Apr 2024 | Fine-tuned | 28.4 $\pm$ 2.1   | 91.2 $\pm$ 1.8 | +62.8        |
| Qwen2-72B                                  | Jun 2024 | Fine-tuned | 29.7 $\pm$ 2.0   | 92.1 $\pm$ 1.7 | +62.4        |
| Llama-2-13B (primary)                      | Jul 2023 | Fine-tuned | 22.4 $\pm$ 2.1   | 85.1 $\pm$ 2.3 | +62.7        |

```

task_loss = F.cross_entropy(
    ↪ answer_logits_orig, batch["answer_labels"]
    ↪ ])

with torch.no_grad():
    edits = editor.generate_counterfactuals(
        batch["input_ids"],
        batch["reasoning_traces"],
        max_edits=3,
        min_impact=0.1,
    )
    accepted = verifier.accepts(
        batch["reasoning_traces"],
        edits["perturbed_traces"],
        threshold=verifier_threshold,
    )

if not accepted.any():
    return task_loss

outputs_cf = model(
    input_ids=edits["perturbed_ids"],
    attention_mask=edits.get("attention_mask",
    ↪ ),
)
answer_logits_cf = answer_projector(
    ↪ outputs_cf.logits, batch)

log_p_cf = F.log_softmax(
    answer_logits_cf[accepted] / temperature,
    ↪ dim=-1)
p_orig = F.softmax(
    answer_logits_orig[accepted].detach() /
    ↪ temperature, dim=-1)
csr_loss = F.kl_div(log_p_cf, p_orig,
    ↪ reduction="batchmean")
csr_loss = torch.clamp(csr_loss, max=max_csr
    ↪ )

return task_loss - lambda_csr * csr_loss

```

Listing 2: Token subset optimization for efficient training.

```

class CSRTokenSubset:
    def __init__(self, max_perturb_ratio=0.15,
        ↪ importance_threshold=0.3):
        self.max_perturb_ratio =
        ↪ max_perturb_ratio
        self.importance_threshold =
        ↪ importance_threshold

    def select_perturbation_tokens(self,
        ↪ trace_tokens, gradients):
        """Select critical tokens for
        ↪ perturbation using gradients."""

```

Table 48: Comprehensive ablation study: component contributions to CSR effectiveness across GSM8K, HotpotQA, and ProofWriter, with divergence and  $\lambda$  variants.

| Component                      | GSM8K COS      | HotpotQA COS   | ProofWriter COS | Acc. impact |
|--------------------------------|----------------|----------------|-----------------|-------------|
| Full CSR                       | 85.1 $\pm$ 2.3 | 84.6 $\pm$ 2.4 | 82.3 $\pm$ 2.1  | -1.2        |
| w/o learned editor (random)    | 61.2 $\pm$ 3.1 | 59.8 $\pm$ 3.2 | 58.3 $\pm$ 2.9  | -0.8        |
| w/o multi-edit policy          | 78.4 $\pm$ 2.6 | 77.1 $\pm$ 2.7 | 75.9 $\pm$ 2.4  | -0.6        |
| w/o impact reward              | 73.2 $\pm$ 2.8 | 71.8 $\pm$ 2.9 | 70.4 $\pm$ 2.6  | -0.4        |
| w/o validity reward            | 69.7 $\pm$ 3.0 | 68.3 $\pm$ 3.1 | 67.1 $\pm$ 2.8  | -0.3        |
| w/o minimality penalty         | 81.3 $\pm$ 2.4 | 80.2 $\pm$ 2.5 | 78.7 $\pm$ 2.2  | -0.9        |
| KL $\rightarrow$ JS divergence | 83.7 $\pm$ 2.5 | 82.4 $\pm$ 2.6 | 81.0 $\pm$ 2.3  | -0.2        |
| KL $\rightarrow$ TV distance   | 82.4 $\pm$ 2.7 | 81.1 $\pm$ 2.8 | 79.8 $\pm$ 2.5  | -0.1        |
| $\lambda = 0.3$ (low)          | 78.3 $\pm$ 2.5 | 74.6 $\pm$ 2.8 | 76.2 $\pm$ 2.4  | -0.3        |
| $\lambda = 0.7$ (high)         | 84.9 $\pm$ 2.4 | 84.3 $\pm$ 2.5 | 81.8 $\pm$ 2.2  | -1.8        |

Table 49:  $f$ -divergence robustness on GSM8K. COS remains within 4 points across divergence choices.

| $f$ -divergence (CSR loss)               | GSM8K COS (%)  | ECE (%) | Conf $\rightarrow$ prob slope | Brier |
|------------------------------------------|----------------|---------|-------------------------------|-------|
| KL (default)                             | 85.1 $\pm$ 2.3 | 2.7     | 0.96                          | 0.13  |
| Jensen-Shannon (JSD)                     | 84.4 $\pm$ 2.4 | 2.6     | 0.95                          | 0.13  |
| Total Variation (TV)                     | 81.5 $\pm$ 2.6 | 3.0     | 0.92                          | 0.14  |
| $\chi^2$ divergence                      | 84.0 $\pm$ 2.5 | 2.8     | 0.94                          | 0.14  |
| $\alpha$ -divergence ( $\alpha=0.5$ )    | 84.7 $\pm$ 2.4 | 2.7     | 0.95                          | 0.13  |
| KL + temperature scaling ( $\tau=1$ )    | 85.1 $\pm$ 2.3 | 2.7     | 0.96                          | 0.13  |
| KL + temperature scaling ( $\tau=1.5$ )  | 84.6 $\pm$ 2.4 | 2.5     | 0.97                          | 0.13  |
| KL + label smoothing ( $\epsilon=0.05$ ) | 84.8 $\pm$ 2.4 | 2.4     | 0.97                          | 0.12  |

Table 50:  $f$ -divergence comparison across domains.

| Divergence      | GSM8K | HotpotQA | ProofWriter | PubMedQA |
|-----------------|-------|----------|-------------|----------|
| KL (default)    | 85.1  | 84.6     | 82.3        | 67.3     |
| Jensen-Shannon  | 84.4  | 83.9     | 81.7        | 66.8     |
| Total Variation | 81.5  | 80.8     | 78.9        | 64.5     |

Table 51: Operating curve: CSR performance across  $\lambda$  values on GSM8K.

| $\lambda$ value        | COS (%)        | Accuracy (%)   | Stability |
|------------------------|----------------|----------------|-----------|
| 0.1 (accuracy-first)   | 67.2 $\pm$ 3.4 | 81.1 $\pm$ 0.7 | High      |
| 0.3 (lower bound)      | 78.3 $\pm$ 2.5 | 80.8 $\pm$ 0.6 | High      |
| 0.5 (default)          | 85.1 $\pm$ 2.3 | 80.5 $\pm$ 0.6 | High      |
| 0.7 (upper bound)      | 84.9 $\pm$ 2.4 | 79.7 $\pm$ 0.7 | High      |
| 0.9 (over-regularized) | 81.2 $\pm$ 2.7 | 77.8 $\pm$ 0.9 | Medium    |

```

importance_scores =
 $\hookrightarrow$  compute_token_importance(
    trace_tokens, gradients)
critical_indices = (
    importance_scores > self.
 $\hookrightarrow$  importance_threshold).nonzero()
max_tokens = int(len(trace_tokens) *
 $\hookrightarrow$  self.max_perturb_ratio)
if len(critical_indices) > max_tokens:
    top_indices = torch.topk(
        importance_scores[
 $\hookrightarrow$  critical_indices], max_tokens).indices
    critical_indices = critical_indices[
 $\hookrightarrow$  top_indices]

```

Table 52: Detailed editor ablation.

| Editor configuration            | GSM8K COS (%)  | $\Delta$ |
|---------------------------------|----------------|----------|
| No editor (random ops)          | 61.2 $\pm$ 3.1 | -23.9    |
| Unconditional editor            | 67.4 $\pm$ 2.9 | -17.7    |
| Conditional, no impact reward   | 73.2 $\pm$ 2.8 | -11.9    |
| Conditional, no validity reward | 69.7 $\pm$ 3.0 | -15.4    |
| Conditional, single-edit only   | 78.4 $\pm$ 2.6 | -6.7     |
| Full learned editor (default)   | 85.1 $\pm$ 2.3 | 0        |

**return** critical\_indices

Listing 3: Step-based warm-start curriculum.

```

class CSRWarmStartCurriculum:
    def __init__(self, warmup_steps=0,
 $\hookrightarrow$  ramp_steps=1000, target_lambda=0.5):
        self.warmup_steps = warmup_steps
        self.ramp_steps = ramp_steps
        self.target_lambda = target_lambda

    def get_lambda(self, global_step):
        """Increase CSR strength by step count,
 $\hookrightarrow$  not epoch count."""
        if global_step < self.warmup_steps:
            return 0.0
        progress = (global_step - self.
 $\hookrightarrow$  warmup_steps) / max(1, self.ramp_steps)
        progress = min(max(progress, 0.0), 1.0)
        return self.target_lambda * progress

```

Table 53: Multi-edit ablation: depth and policy effectiveness.

| Configuration          | Edit depth | COS (%)  | Computational cost (%) | Adversarial resistance |
|------------------------|------------|----------|------------------------|------------------------|
| Single edit (baseline) | 1          | 76.4±2.4 | +9.0                   | Low                    |
| Random multi-edit      | 2          | 80.7±2.3 | +9.8                   | Medium                 |
| Random multi-edit      | 3          | 82.1±2.2 | +10.6                  | Medium                 |
| Learned multi-edit     | 2          | 83.9±2.1 | +9.6                   | High                   |
| Learned multi-edit     | 3          | 85.1±2.0 | +10.2                  | High                   |
| Adaptive policy        | 1–3        | 85.7±2.0 | +10.4                  | Very high              |

Table 54: Long-term training dynamics: CSR stability and convergence.

| Epoch | COS (%)  | Acc (%)  | CSR loss  | Task loss | KL div.   | Stability |
|-------|----------|----------|-----------|-----------|-----------|-----------|
| 1     | 45.2±3.1 | 79.8±0.9 | 0.38±0.04 | 0.42±0.03 | 0.12±0.02 | High      |
| 2     | 68.7±2.8 | 80.2±0.7 | 0.31±0.03 | 0.39±0.03 | 0.28±0.03 | High      |
| 3     | 82.1±2.4 | 80.5±0.6 | 0.24±0.02 | 0.38±0.03 | 0.41±0.04 | High      |
| 4     | 84.6±2.3 | 80.4±0.6 | 0.22±0.02 | 0.37±0.03 | 0.45±0.04 | High      |
| 5     | 85.1±2.3 | 80.5±0.6 | 0.21±0.02 | 0.37±0.03 | 0.46±0.04 | High      |
| 6     | 85.3±2.2 | 80.3±0.7 | 0.21±0.02 | 0.37±0.03 | 0.46±0.04 | High      |

Listing 4: Editor reward with verifier validity, impact, and minimality terms. The formal method is  $p(Y | X, T)$ , so the impact term conditions on input  $X$  as well as the trace.

```

def compute_editor_reward(
    model,
    input_example,
    trace,
    perturbed_trace,
    verifier,
    answer_projector,
    lambdas=(1.0, 0.1, 0.05),
    eps=1e-8,
):
    """Reward edits that are valid-to-invalid,
    ↪ impactful, and minimal.

    impact_reward is KL(p(Y|X,T') || p(Y|X,T)),
    ↪ computed on the answer
    space defined by answer_projector.
    ↪ Conditioning on input_example is
    essential for QA-style tasks where the
    ↪ question/context determines
    which trace operations are consequential.
    """
    lambda_v, lambda_i, lambda_l = lambdas
    validity_reward = float(
        verifier.is_valid(trace)
        and not verifier.is_valid(
            ↪ perturbed_trace
        )
    )

    with torch.no_grad():
        p_orig = model.predict_distribution(
            input_example, trace,
            ↪ answer_projector
        ).clamp_min(eps)
        p_pert = model.predict_distribution(
            input_example, perturbed_trace,
            ↪ answer_projector
        ).clamp_min(eps)

        p_orig = p_orig / p_orig.sum(dim=-1,
        ↪ keepdim=True)
        p_pert = p_pert / p_pert.sum(dim=-1,
        ↪ keepdim=True)
        impact_reward = F.kl_div(
            p_pert.log(), p_orig, reduction="
            ↪ batchmean").item()

    edit_length = compute_edit_distance(trace,
    ↪ perturbed_trace)
    length_penalty = edit_length / max(1, len(
    ↪ trace.split()))
    return (lambda_v * validity_reward
            + lambda_i * impact_reward
            - lambda_l * length_penalty)

```

Table 55: Computational efficiency. Top block reports full per-run resource usage on GSM8K and HotpotQA, including a comparison to a non-efficient full-CSR variant. Bottom block reports detailed cost-per-COS metrics on GSM8K.

| Method                        | GPU-h | Wall-clock (h) | Memory (GB) | Token updates | COS (%)  | Acc (%)  | COS/GPU-h |
|-------------------------------|-------|----------------|-------------|---------------|----------|----------|-----------|
| <i>GSM8K (Llama-2-13B)</i>    |       |                |             |               |          |          |           |
| Standard FT                   | 135   | 16.8           | 42.3        | 2.1M          | 22.4±2.1 | 81.3±0.8 | 0.166     |
| Process reward model          | 156   | 19.5           | 48.7        | 2.4M          | 52.3±2.8 | 81.7±0.7 | 0.335     |
| Efficient CSR (ours)          | 147   | 18.3           | 44.1        | 2.3M          | 85.1±2.3 | 80.5±0.6 | 0.579     |
| Full CSR                      | 259   | 32.4           | 52.6        | 2.3M          | 86.2±2.1 | 80.1±0.7 | 0.333     |
| <i>HotpotQA (Llama-2-13B)</i> |       |                |             |               |          |          |           |
| Standard FT                   | 267   | 33.4           | 43.8        | 4.2M          | 25.1±2.8 | 78.1±1.1 | 0.094     |
| Process reward model          | 298   | 37.3           | 51.2        | 4.7M          | 49.8±3.1 | 78.4±0.9 | 0.167     |
| Efficient CSR (ours)          | 289   | 36.1           | 45.9        | 4.6M          | 84.6±2.4 | 77.2±0.8 | 0.293     |
| Full CSR                      | 521   | 65.1           | 56.3        | 4.6M          | 85.8±2.2 | 76.8±0.9 | 0.165     |

| Method                           | GPU-hours | Memory (GB) | COS/GPU-h | Cost/COS pt |
|----------------------------------|-----------|-------------|-----------|-------------|
| <i>GSM8K cost-per-COS detail</i> |           |             |           |             |
| Standard FT                      | 135       | 42.3        | 0.166     | \$60.27     |
| Process RM                       | 156       | 48.7        | 0.335     | \$29.83     |
| GRPO                             | 152       | 47.2        | 0.360     | \$27.78     |
| Verifier-guided                  | 148       | 45.8        | 0.325     | \$30.77     |
| CSR-FT (ours)                    | 147       | 44.1        | 0.579     | \$17.27     |
| CSR vs Process RM                | 0.94×     | 0.91×       | 1.73×     | 0.58×       |
| CSR vs GRPO                      | 0.97×     | 0.93×       | 1.61×     | 0.62×       |

Table 56: Operator noise sensitivity.

| Noise level (%) | Operator precision (%) | COS performance | Recovery strategy      | Effectiveness |
|-----------------|------------------------|-----------------|------------------------|---------------|
| 0 (perfect)     | 100.0                  | 86.2±2.0        | —                      | —             |
| 10              | 90.1                   | 84.3±2.1        | None needed            | —             |
| 20              | 80.7                   | 81.4±2.4        | Conservative $\lambda$ | 85%           |
| 30              | 71.2                   | 75.8±2.7        | Confidence gating      | 78%           |
| 40              | 62.4                   | 67.3±3.1        | Domain adaptation      | 64%           |
| 50              | 53.8                   | 56.9±3.5        | Pre-filtering          | 41%           |

Table 57: Empirical validation of the validity conditions and consequences of violating them.

| Assumption violation                  | Consequence                 | Empirical performance                             |
|---------------------------------------|-----------------------------|---------------------------------------------------|
| Operator precision $< 60\%$           | Not recommended             | 65% precision $\rightarrow$ COMP/SUFF $>$ CSR     |
| Intervention coverage $\alpha < 50\%$ | Shortcuts retained          | 30% coverage $\rightarrow$ 25% shortcut rate      |
| $\lambda < 0.1$                       | Insufficient regularization | $\lambda = 0.05 \rightarrow$ 12% improvement only |
| Non-causal interventions              | Wrong gradient direction    | Random tokens $\rightarrow$ no improvement        |

Table 58: Comprehensive operator discovery validation across discovery methods. The PubMedQA block is the comparison against the manual gold standard and resolves the distinction between the 67.3 full-CSR result and the 58.9 learned-discovery result.

| Discovery method                          | Precision (%) | Recall (%) | Coverage (%) | COS (%) |
|-------------------------------------------|---------------|------------|--------------|---------|
| <i>PubMedQA</i>                           |               |            |              |         |
| Manual inventory + domain verifier (gold) | 100.0         | 100.0      | 100.0        | 67.3    |
| Heuristic + NER                           | 78.3          | 61.0       | 88.7         | 61.2    |
| Learned discovery                         | 74.1          | 68.5       | 91.2         | 58.9    |
| <i>Open-ended</i>                         |               |            |              |         |
| Dialogue (PersonaChat)                    | 58.2          | 52.4       | 84.1         | 41.7    |
| Narrative (ROCStories)                    | 61.7          | 49.8       | 82.3         | 36.8    |

Table 59: Learned operator categories for PubMedQA discovery.

| Operator category   | Coverage (%) | Precision (%) | Examples                                 |
|---------------------|--------------|---------------|------------------------------------------|
| Biomedical entities | 34.2         | 89.7          | “MMP-9”, “cytokine”, “angiogenesis”      |
| Causal verbs        | 28.9         | 76.4          | “inhibits”, “promotes”, “correlates”     |
| Quantifiers         | 18.7         | 71.8          | “significantly”, “elevated”, “decreased” |
| Temporal markers    | 12.3         | 68.2          | “following”, “prior to”, “during”        |
| Modal expressions   | 5.9          | 62.1          | “may indicate”, “suggests”, “likely”     |

Table 60: Operator discovery scalability across domains. “Manual” denotes the domain-specific operator inventory and verifier used for the main results.

| Domain      | Method        | Precision | Recall | F1    | COS  | Manual effort |
|-------------|---------------|-----------|--------|-------|------|---------------|
| GSM8K       | Manual        | 100.0     | 100.0  | 100.0 | 85.1 | High          |
|             | Heuristic     | 94.2      | 91.7   | 92.9  | 83.7 | Low           |
|             | Learned       | 89.3      | 87.4   | 88.3  | 81.2 | None          |
| HotpotQA    | Manual        | 100.0     | 100.0  | 100.0 | 84.6 | High          |
|             | Heuristic+NER | 87.6      | 82.3   | 84.9  | 81.8 | Medium        |
|             | Learned       | 81.4      | 78.9   | 80.1  | 78.3 | None          |
| ProofWriter | Manual        | 100.0     | 100.0  | 100.0 | 82.3 | High          |
|             | Pattern match | 91.2      | 88.7   | 89.9  | 80.1 | Low           |
|             | Learned       | 85.7      | 83.2   | 84.4  | 77.6 | None          |
| PubMedQA    | Manual        | 100.0     | 100.0  | 100.0 | 67.3 | High          |
|             | Heuristic+NER | 78.3      | 61.0   | 68.5  | 61.2 | Medium        |
|             | Learned       | 74.1      | 68.5   | 71.2  | 58.9 | None          |
| HellaSwag   | Manual        | 100.0     | 100.0  | 100.0 | 52.1 | High          |
|             | Pattern match | 74.1      | 69.2   | 71.6  | 47.8 | Medium        |
|             | Learned       | 69.8      | 65.3   | 67.5  | 44.2 | None          |

Table 61: Preliminary universal meta-verifier: zero-shot operator identification on unseen domains.

| Training domains    | Test domain | Precision (%) | Recall (%) | COS (%)  |
|---------------------|-------------|---------------|------------|----------|
| GSM8K + ProofWriter | HotpotQA    | 72.3±2.8      | 68.1±3.1   | 76.4±2.9 |
| GSM8K + HotpotQA    | ProofWriter | 74.8±2.6      | 71.2±2.9   | 78.1±2.7 |
| All structured      | PubMedQA    | 69.7±3.1      | 65.4±3.4   | 61.2±3.2 |

Table 62: Detailed failure conditions and their performance impact on CSR. Operator precision and verifier precision are distinct quantities and are listed separately.

| Condition                                    | Performance impact             | Mitigation strategy                    | Threshold                       |
|----------------------------------------------|--------------------------------|----------------------------------------|---------------------------------|
| Operator precision < 60%                     | COS drops > 15 points          | Use confidence-gated CSR (Eq. 5)       | Operator precision $\geq 78\%$  |
| Verifier precision below the GSM8K crossover | Gain over baselines unreliable | Conservative $\lambda \leq 0.3$        | 78.6% on GSM8K; domain-specific |
| Verifier noise > 30%                         | Training instability           | Conservative $\lambda \leq 0.3$        | —                               |
| Multiple reasoning paths                     | Edits ineffective              | Multi-edit policy                      | Path multiplicity $\leq 2$      |
| Very long sequences (>1024 tokens)           | Computational overhead         | Token-subset optimization              | Length $\leq 1024$              |
| Distribution shift                           | Reduced transfer               | Domain-specific fine-tuning            | Transfer eff. $\geq 75\%$       |
| Adversarial inputs                           | Robustness degradation         | Adversarial training augmentation      | Adversarial acc $\geq 75\%$     |
| Hyperparameter $\lambda > 0.7$               | Over-regularization            | Optimal range $\lambda \in [0.3, 0.7]$ | $\lambda = 0.5$ optimal         |

Table 63: Combined CSR failure analysis. Top block: four editor-side failure types with the share of failures they account for, the targeted mitigation, and the resulting  $\Delta\text{COS}$ . Bottom block: six broader CSR failure modes characterized by frequency, affected domains, mitigation effectiveness, and performance recovery.

| Failure category                    | Frequency / % of failures | Affected domains / Mitigation                       | Effectiveness | $\Delta\text{COS}$ / Recovery |
|-------------------------------------|---------------------------|-----------------------------------------------------|---------------|-------------------------------|
| <i>Editor-side failure taxonomy</i> |                           |                                                     |               |                               |
| Trace incoherence                   | 28.4% of failures         | Stricter verifier + syntax filter                   | —             | +3.1 COS                      |
| Redundant edit                      | 33.9% of failures         | Influence-guided edit targeting                     | —             | +4.6 COS                      |
| Adversarial compliance              | 22.7% of failures         | Multi-edit ( $L = 2-3$ )                            | —             | +3.8 COS                      |
| Semantic drift                      | 31.5% of failures         | NLI guard + calibration                             | —             | +2.4 COS                      |
| <i>Broader CSR failure modes</i>    |                           |                                                     |               |                               |
| Spurious operator targeting         | 15.2                      | Open-ended QA, narrative / Confidence gating        | 78%           | +12.3 COS                     |
| Multiple valid reasoning paths      | 7.3                       | Complex math, logic / Multi-edit policies           | 65%           | +8.7 COS                      |
| Verifier mismatch with operators    | 11.8                      | Cross-domain transfer / Domain-specific tuning      | 82%           | +15.4 COS                     |
| Computational shortcuts learned     | 6.4                       | Pattern-heavy domains / Curriculum learning         | 71%           | +9.1 COS                      |
| Long-range dependency loss          | 4.9                       | Multi-hop reasoning / Hierarchical CSR              | 69%           | +11.2 COS                     |
| Adversarial robustness gaps         | 3.7                       | Safety-critical applications / Adversarial training | 84%           | +13.6 COS                     |