WHICH REASONING STEPS DRIVE THE ANSWER? A CAUSAL ANALYSIS OF LLM REASONING STEPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models often generate multi-step reasoning traces to solve complex problems, yet many evaluations assess only the final answer. The accuracy of the final answer alone, however, does not reveal which intermediate steps causally influence the answer or how their effects propagate through reasoning. This paper introduces a causal framework for identifying reasoning steps with statistically significant effects on the final answer and for characterizing the pathways through which these effects arise. We use e-values to assess the statistical significance of each reasoning step's total effect, apply multiple testing procedures to select significant steps, and decompose their total effects into the interventional direct and indirect effects for visualization. We demonstrate the utility of our approach for evaluating LLMs' reasoning capabilities across multiple reasoning models and datasets.

1 INTRODUCTION

Reasoning can substantially enhance the problem-solving capabilities of modern LLMs by enabling them to decompose complex problems into a sequence of intermediate subproblems and reason through them step by step, resembling how humans approach complex problems (Wei et al., 2022). Nonetheless, it remains an open question whether the reasoning trajectories produced by LLMs faithfully reflect the processes underlying their final solutions, or whether models are merely "pretending" to think, with reasoning traces that play little causal role in arriving at the solution (Turpin et al., 2023; Lanham et al., 2023).

There is a growing literature on evaluating the quality and faithfulness of LLM reasoning, often through causal interventions (see Section 2 for a review). However, establishing that a reasoning trace affects the final answer is only part of the story. A trace may contain many reasoning steps, only some of which influence the solution. Moreover, the effect of an individual step may arise directly from that step or indirectly through subsequent steps that are altered along the way. *The heart of the problem is not simply whether reasoning matters, but which reasoning steps matter and how their effects propagate to the final answer*.

This paper addresses these questions by bridging causal mediation analysis and multiple testing with the evaluation of LLM reasoning. Causal mediation analysis traces how the effect of an intervention reaches an outcome, separating the effect that acts directly from the effect transmitted through intermediate variables, known as mediators (Robins & Greenland, 1992; Pearl, 2001; VanderWeele et al., 2014; Vansteelandt & Daniel, 2017). Multiple testing addresses a different challenge: it finds needles (signals) in a haystack of high-dimensional noise while controlling the false discovery rate (Benjamini & Hochberg, 1995). Both procedures are well-suited to the evaluation of LLM reasoning, where a reasoning trace contains multiple steps whose effects must be assessed simultaneously, and the influence of any individual step may propagate through subsequent reasoning.

We summarize **our contributions** as follows:

1. We propose to apply multiple testing for identifying important reasoning steps with statistically significant causal effects on the final answer, using e-values (Vovk & Wang, 2021) to control the false discovery rate across steps.
2. For each selected reasoning step, we decompose its total effect into a direct effect on the final answer and an indirect effect transmitted through subsequent reasoning. While causal mediation

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

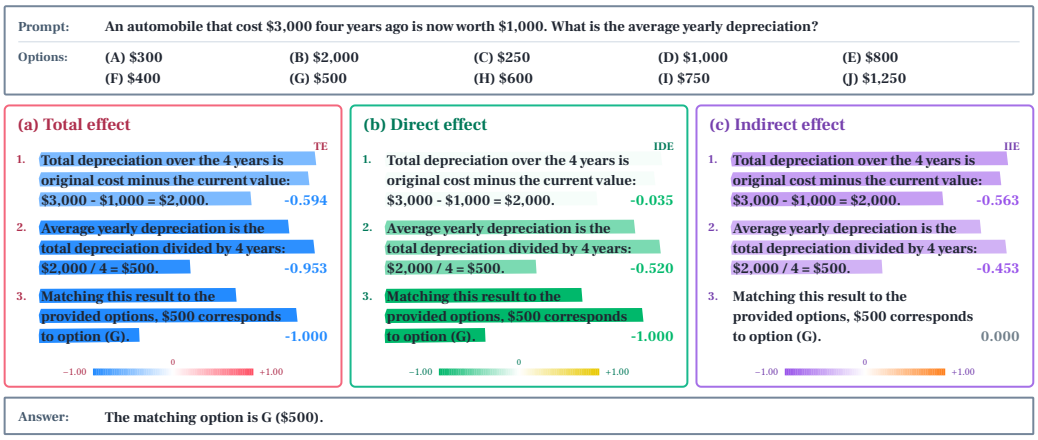

Figure 1: An example from our empirical study ([MODEL], MMLU-Pro), showing the prompt (top), the reasoning trace (middle), and the final answer (bottom). The three middle panels show, from left to right, the total, direct, and indirect effect of each reasoning step; color indicates the sign of the effect and its intensity the magnitude. All three steps have significant total effects yet their mechanism under our decomposition differs.

analysis has previously been applied to LLM reasoning, to our knowledge, such a decomposition has not been developed at the level of individual reasoning steps.

3. We visualize the total, direct, and indirect effects across a reasoning trace, offering guidance on how the causal effect emerges and propagates through the reasoning process.

To illustrate the last point, Figure 1 shows an example from our empirical study (Section 4). The top and bottom panels show the prompt and the model's answer. The three middle panels visualize, from left to right, the total, direct, and indirect effects of each individual reasoning step. Color indicates the direction of the effect (positive or negative), while color intensity represents its effect size. All three reasoning steps have significant total effects, but they reach the answer in different ways. The effect of Step 1, which computes the total depreciation ($3000 - 1000 = 2000$), is transmitted almost entirely through the subsequent reasoning; its direct effect is close to zero. Step 3, which maps the computed value to option (G), acts directly on the answer with no indirect component. Step 2, which computes the yearly depreciation ($2000/4 = 500$), has direct and indirect components of comparable size: even when the subsequent reasoning is held at the distribution induced by the original step, altering Step 2 lowers the probability of a correct answer. Because the indirect effect is defined with respect to the entire subsequent reasoning, it indicates that part of a step's effect is carried by Step 3.

2 Related Work

This paper falls within the literature on the evaluation of LLM reasoning, which can be broadly divided into *black-box* and *white-box* methods, depending on whether the analysis requires access to internal model states, such as hidden-layer representations or activations (white-box), or relies only on prompts and model outputs (black-box). Black-box methods can be further distinguished by their level of analysis, including *prompt-level* analyses, which examine how an LLM's answer or reasoning trace changes in response to modifications to the prompt; *trace-level* analyses, which examine the effect of an entire reasoning trace; and *step-level* analyses, which focus on the effects of individual reasoning steps. Our proposal is most closely related to black-box step-level analyses. Table 1 summarizes representative approaches that are most closely related to our work and their major differences from ours. Below, we review these lines of work in detail.

Black-box prompt-level analyses. Prompt-level analyses study how changes to the input prompt affect model responses and whether the reasoning trace acknowledges the factors driving these changes. Turpin et al. (2023) introduce biasing cues into prompts and find that chain-of-thought

Table 1: Methods for analyzing reasoning traces, compared along five dimensions. Definitions of treatments and outcomes may differ across methods. *Causal formulation*: whether the quantity of interest is defined as a causal effect through an explicit causal model, rather than merely as a score or rate. *Direct*, *indirect effects* and *Total*: whether the analyzed effect captures the direct influence on the outcome, the indirect influence mediated through subsequent reasoning steps, or both, respectively. *Uncertainty quantification*: whether uncertainty in the estimated effects is quantified. ✓: addressed, fully or in part; ✗: not explicitly addressed, or not applicable to the method's target. Our method differs from existing approaches by (i) analyzing direct, indirect, and total effects at the step level and (ii) quantifying uncertainty with multiple testing control.

Method	Causal formulation	Total effect	Direct effect	Indirect effect	Uncertainty quantification
Black-box, prompt level					
Turpin et al. (2023)	✗	✓	✗	✗	✓
Black-box, trace level					
Lanham et al. (2023)	✗	✓	✗	✗	✗
FRODO (Paul et al., 2024)	✓	✗	✓	✓	✗
Black-box, step level					
Thought Anchors (Bogdan et al., 2025)	✓	✓	✓	✗	✓
Thought Branches (Macar et al., 2026)	✓	✓	✗	✓	✓
AttriCoT (Wei et al., 2026)	✓	✗	✓	✗	✗
FACT-E (Sun et al., 2026)	✓	✓	✗	✗	✗
White-box					
CRV (Zhao et al., 2026)	✗	✗	✓	✗	✗
CIF (Asiaee, 2026)	✓	✗	✗	✗	✓
Our work	✓	✓	✓	✓	✓

explanations often fail to acknowledge their effect, even when these cues change the model's final answer. Later studies extend such evaluations to newer reasoning models (Chen et al., 2025; Chua & Evans, 2025). However, failing to acknowledge an influential prompt cue does not, by itself, establish that the reasoning trace has no causal effect on the answer (Zaman & Srivastava, 2025; Von Arx & Deng, 2025). The latter question is the central focus of our work.

Black-box trace-level analyses. Trace-level analyses modify the reasoning trace to examine how the model's final answer depends on that trace. Lanham et al. (2023) assess this dependence by considering several types of modifications, including early answering, paraphrasing, replacement with filler tokens, and the insertion of mistakes. Paul et al. (2024) propose a causal mediation framework that treats the input prompt as the treatment, the entire reasoning chain as the mediator, and the final answer as the outcome. Xiong et al. (2025) modify the text at the end of the reasoning trace and examine whether the subsequently generated answers remain consistent with the modified drafts. In comparison, our work focuses on step-specific causal effects, distinguishing an individual step's direct effect on the answer from its indirect effect through subsequent reasoning.

Black-box step-level analyses. Step-level analyses consider the effect of an individual reasoning step, making them closely aligned with our objective. Intervention-based step-level analyses can be further divided into two types, depending on whether the reasoning trace following the target step is regenerated or held fixed. In the former approach, Bogdan et al. (2025) and Macar et al. (2026) resample individual sentences and measure the resulting changes in the answer distribution. However, unlike our method, they do not study the decomposition of a step's effect. Yu et al. (2025) propose a necessity-and-sufficiency framework for pruning reasoning traces. Sun et al. (2026) generate perturbed continuations from fixed prefixes and compare them with the original continuations to assess the faithfulness of the reasoning trace. In the latter approach, which does not regenerate the continuation, Wei et al. (2026) remove individual units and evaluate changes in the log probabilities of downstream units within the existing trace. Gao (2023) uses Shapley values to quantify the effects of individual tokens in a reasoning trace. Fu et al. (2025) model causal relationships among reasoning steps and use LLM-based scoring to identify and refine problematic steps. As summarized in

Table 1, our method differs from these approaches in jointly estimating the direct, indirect, and total effects of each individual reasoning step while quantifying uncertainty with multiple testing control.

White-box analyses. White-box methods study model behavior with model internal states (Geiger et al., 2021; 2025; Olah et al., 2020). One approach is activation patching, which applies causal mediation analysis (Vig et al., 2020a;b) to investigate the roles of internal model components (Meng et al., 2022; Wang et al., 2023; Hanna et al., 2023). For reasoning tasks, studies have investigated how internal model components implement multi-step reasoning (Cabannes et al., 2024; Dutta et al., 2024; Brinkmann et al., 2024), how interventions on internal activations can steer reasoning (Venhoff et al., 2025), and how hidden states can be probed to predict answer correctness (Zhang et al., 2025). Relatedly, Lindsey et al. (2025) use attribution graphs to trace internal computational pathways, Zhao et al. (2026) use graph features to verify individual reasoning steps, and Minegishi et al. (2025) analyze reasoning graphs constructed from hidden state representations. Hu et al. (2026) provide a survey of this literature. On the statistical side, Asiaee (2026) introduces certified interventional fidelity (CIF), which develops confidence intervals and anytime-valid confidence sequences for uncertainty quantification. Different from these white-box methods, our approach instead intervenes on generated reasoning text and measures its effects on the final answer, without accessing internal model states or activations.

3 A Causal Framework for Evaluating LLM Reasoning Steps

We introduce a causal framework for identifying individual reasoning steps that causally affect a model's final answer and for characterizing how these effects arise. Our complete procedure is summarized in Algorithm 1 and consists of three steps: (1) estimate the total effect of each reasoning step and compute the corresponding e-value; (2) apply the e-Benjamini-Hochberg procedure (Wang & Ramdas, 2022) to identify statistically significant reasoning steps; and (3) decompose the total effects of the selected steps into a direct effect and an indirect effect. These three steps address the *whether*, *which*, and *how* of reasoning-step effects on the final answer, respectively. Before detailing these three steps, we first introduce our causal model.

Given a prompt X, a language model produces a sequence of reasoning steps $(R_1, \ldots, R_T)$ followed by a final response Y, where T denotes the number of reasoning steps. Consider a target step R_i whose causal effect on the response we aim to evaluate, and treat R_i as the treatment. Let $C_i = (X, R_1, \ldots, R_{i-1})$ denote the context preceding R_i, and let $M_i = (R_{i+1}, \ldots, R_T)$ denote the subsequent reasoning steps.

Conditional on C_i, there are two causal pathways through which R_i can affect Y: (i) a direct pathway, $R_i \to Y$, and (ii) an indirect pathway through subsequent reasoning, $R_i \to M_i \to Y$. Thus, M_i naturally serves as a mediator between the target reasoning step and the final answer. These pathways lead to a *direct effect*, which captures the change in the answer attributable to the ith step while holding downstream reasoning fixed, and an *indirect effect*, which captures the change transmitted through subsequent reasoning. Taking both pathways into consideration, we obtain the *total effect*, which captures the overall change in the answer, allowing subsequent reasoning to change accordingly.

Let $\mathcal{K}_i$ denote a finite set of interventions applicable to reasoning step R_i. This intervention set may vary across steps. We first classify each reasoning step into one of three categories: (i) content-bearing steps, which contain important reasoning content; (ii) structural steps, which primarily serve organizational or transitional purposes, such as "let us begin"; and (iii) invalid steps, which do not follow the prescribed step format.

For content-bearing steps, we further classify each step into one or more of eight functional categories, with category-specific interventions defined accordingly. For structural steps, we use a deletion intervention that removes the step. Invalid steps are excluded from the analysis.

For each $k \in \mathcal{K}_i$, let $a_{i,k}^{(0)}$ and $a_{i,k}^{(1)}$ denote the original and intervened versions of the target step under the kth intervention, respectively. The do operation $\mathrm{do}(R_i = a)$ sets the text of the target step to a and then runs the same model from the resulting prefix to generate the subsequent reasoning trace and final response. Since the response Y is textual, we first extract an answer $A = A(Y)$ from the response and assign a special symbol NULL to invalid or unparseable answers. We then

Algorithm 1 Step-level causal analysis of LLM reasoning

Require: A prompt, its associated reasoning trace and response, the LLM that generated them, an outcome function h, a sampling budget N and a significance level α.

1: **for** each step-intervention pair (i, k) **do**
2: Generate N independent continuations from each of the two prefixes formed by concatenating C_i with the two arms $a_{i,k}^{(0)}$ and $a_{i,k}^{(1)}$, respectively.
3: Estimate its total effect according to equation 2
4: Compute its e-value $e_{i,k}$ according to equation 3.
5: **end for**
6: Aggregate $\{e_{i,k}\}_{k\in\mathcal{K}_i}$ to compute step-level e-value e_i by equation 5.
7: Apply e-Benjamini–Hochberg procedure (see equations 7 and 8) to obtain the set of important steps $\mathcal{D}$.
8: **for** each $i \in \mathcal{D}$ **do**
9: Estimate its interventional direct and indirect effects according to equation 9.
10: **end for**
11: **return** The set of important steps $\mathcal{D}$, together with the total, direct, and indirect effects for each step in $\mathcal{D}$

study how the distribution of A changes in response to an intervention on the target reasoning step. Specifically, our quantities of interest are expressed as expectations of prespecified functions $h(A)$. For example, taking $h(A) = \mathbb{I}\{A = A^\star\}$ measures whether the generated answer agrees with the gold answer $A^\star$. Below, we detail our three-step procedure. Each step addresses a specific question.

Does a reasoning step matter? Using the above notation, we define the total effect for a given step-intervention pair (i, k) as

$$\mathrm{TE}_{i,k} = \mathbb{E}\Big[h(A) \,\Big|\, \mathrm{do}(R_i = a_{i,k}^{(1)}),\, C_i\Big] - \mathbb{E}\Big[h(A) \,\Big|\, \mathrm{do}(R_i = a_{i,k}^{(0)}),\, C_i\Big]. \tag{1}$$

In words, the total effect measures how the intervention changes $\mathbb{E}[h(A)]$, while allowing the subsequent reasoning to evolve naturally under the same model.

Estimation of the total effect is straightforward. Conditional on C_i, the randomness arises only from stochastic model generation. We therefore use Monte Carlo sampling, drawing N independent continuations from each arm. Specifically, for arm $a \in \{0, 1\}$, we obtain $\{(M_{i,a,s}, Y_{a,s})\}_{s=1}^N$, where $a = 0$ corresponds to the original step and $a = 1$ to the intervened step. We estimate the total effect by

$$\widehat{\mathrm{TE}}_{i,k} = \frac{1}{N}\sum_{s=1}^{N} h(A_{1,s}) - \frac{1}{N}\sum_{s=1}^{N} h(A_{0,s}), \tag{2}$$

where $A_{a,s} = A(Y_{a,s})$ denotes the extracted answer from the sth draw in arm a.

To determine whether the estimated total effect is significant, we compute its e-value. Consider the hypotheses

$$\mathcal{H}_{0,i,k} : \mathrm{TE}_{i,k} = 0 \quad \text{versus} \quad \mathcal{H}_{1,i,k} : \mathrm{TE}_{i,k} \neq 0.$$

Under the null hypothesis $\mathcal{H}_{0,i,k}$, intervening on the ith reasoning step has no average effect on $h(A)$ for the kth intervention, whereas under the alternative hypothesis $\mathcal{H}_{1,i,k}$, it has a nonzero average effect.

An e-value provides a measure of evidence against $\mathcal{H}_{0,i,k}$. Specifically, an e-value $e_{i,k}$ is a nonnegative statistic satisfying $\mathbb{E}(e_{i,k}) \leq 1$ under the null hypothesis (Wang & Ramdas, 2022). Larger e-values provide stronger evidence against the null. Given a significance level α, Markov's inequality implies that, under $\mathcal{H}_{0,i,k}$, $\mathbb{P}\left(e_{i,k} \geq 1/\alpha\right) \leq \alpha\,\mathbb{E}(e_{i,k}) \leq \alpha$. Thus, the rejection rule $e_{i,k} \geq 1/\alpha$ controls the type-I error at level α, i.e., the probability of incorrectly rejecting a true null hypothesis. We therefore reject $\mathcal{H}_{0,i,k}$ when $e_{i,k} \geq 1/\alpha$. Otherwise, we fail to reject the null.

For any outcome function h taking values in $[0, 1]$, an e-value can be constructed using betting-based methods (Waudby-Smith & Ramdas, 2024). Recall that $\{(M_{i,a,s}, Y_{a,s})\}_{s=1}^N$ denotes the N

independent mediator–response draws from arm $a \in \{0, 1\}$. We define

$$e_{i,k} = \prod_{s=1}^{N} \Big\{ 1 + \frac{\lambda_s}{2} [h(A(Y_{1,s})) - h(A(Y_{0,s}))] \Big\}, \tag{3}$$

where

$$\lambda_s = \arg \max_{\lambda \in [-1,1]} \sum_{t<s} \log \Big\{ 1 + \frac{\lambda}{2} [h(A(Y_{1,t})) - h(A(Y_{0,t}))] \Big\}.$$

By definition, the betting parameter λ_s is chosen adaptively using only the observations from the preceding $s-1$ samples. Its independence of the current observation guarantees the validity of the resulting e-value. In particular, under the null, the current paired difference $h(A(Y_{1,s})) - h(A(Y_{0,s}))$ has conditional mean zero given the past, so each multiplicative factor in equation 3 has conditional expectation one and remains nonnegative. Iterating this argument over s yields $\mathbb{E}(e_{i,k}) \leq 1$. Unlike asymptotic tests, this construction is valid for any finite N, and hence the sampling budget may also be allocated parsimoniously (Howard et al., 2021; Waudby-Smith & Ramdas, 2024).

Meanwhile, the choice of λ above maximizes the cumulative e-value based on the previous samples. This adaptive maximization is intended to improve the power (i.e., the probability of correctly rejecting a false null hypothesis) under the alternative hypothesis. It chooses bets that promote faster growth of the e-value, in order to accumulate as much evidence as possible against the null.

Which reasoning steps matter? Each individual e-value provides evidence about the effect of a given intervention on a reasoning step. To identify significant reasoning steps, we must aggregate results across interventions within each step and account for multiple testing across steps.

First, consider the following step-level hypotheses

$$\mathcal{H}_{0,i} = \bigcap_{k \in \mathcal{K}_i} \mathcal{H}_{0,i,k} \quad \text{versus} \quad \mathcal{H}_{1,i} = \bigcup_{k \in \mathcal{K}_i} \mathcal{H}_{1,i,k}. \tag{4}$$

In words, the null hypothesis $\mathcal{H}_{0,i}$ holds if every intervention on the ith reasoning step has zero TE, whereas the alternative $\mathcal{H}_{1,i}$ holds if at least one intervention attains a nonzero TE. We say that the ith reasoning step is important if $\mathcal{H}_{1,i}$ holds. Otherwise, we refer to it as a null step.

By linearity of expectation, an average of the intervention-specific e-values,

$$e_i = \frac{1}{|\mathcal{K}_i|} \sum_{k \in \mathcal{K}_i} e_{i,k}, \tag{5}$$

is itself a valid e-value under the intersection null $\mathcal{H}_{0,i}$ in equation 4.

Second, let $\mathcal{N}$ denote the set of null steps. Our goal is to identify important steps in its complement, $\mathcal{N}^c$, while controlling the false discovery rate (FDR), defined as the expected proportion of falsely selected null steps among all selected steps:

$$\mathrm{FDR}(\mathcal{D}) = \mathbb{E}\left[\frac{|\mathcal{D} \cap \mathcal{N}|}{\max(1, |\mathcal{D}|)} \right], \tag{6}$$

at a prespecified target level α, where $\mathcal{D}$ denotes the set of selected steps.

This is a classical multiple testing problem (Benjamini & Hochberg, 1995). The problem is challenging because testing each step separately at level α can lead to an accumulation of false discoveries. Conversely, a Bonferroni correction (Dunn, 1961), which tests each step at level α divided by the total number of steps, can be overly conservative and substantially reduce power.

E-values provide a convenient way to address this problem (Vovk & Wang, 2021; Wang & Ramdas, 2022). In particular, the e-Benjamini–Hochberg (BH) procedure controls the FDR under arbitrary dependence among valid e-values (Wang & Ramdas, 2022). Specifically, let $e_1, \ldots, e_T$ denote the step-level e-values defined in equation 5, and let $e_{(1)} \geq \cdots \geq e_{(T)}$ denote their order statistics. The e-BH procedure defines

$$\widehat{r} = \max \left\{ r \in \{1, \ldots, T\} : e_{(r)} \geq \frac{T}{\alpha r} \right\}, \tag{7}$$

Table 2: Three experimental conditions defining $\Psi_{i,k}(j, \ell)$, which vary the target step and mediator distribution separately. The subscripts i and k are omitted in the table for brevity.

	Target step	Mediator distribution
$\Psi(0,0)$	$a^{(0)}$	$G^{(0)}$
$\Psi(1,0)$	$a^{(1)}$	$G^{(0)}$
$\Psi(1,1)$	$a^{(1)}$	$G^{(1)}$

with $\widehat{r} = 0$ if the set is empty. We then select

$$\mathcal{D} = \left\{ i : e_i \geq \frac{T}{\alpha \widehat{r}} \right\}, \tag{8}$$

with $\mathcal{D} = \emptyset$ when $\widehat{r} = 0$. If the step-level e-values are valid, the resulting discovery set satisfies $\mathrm{FDR}(\mathcal{D}) \leq \alpha$ (Wang & Ramdas, 2022), achieving the desired FDR control.

How do reasoning steps matter? Total effect alone does not reveal how much of the effect is mediated through subsequent reasoning. This motivates us to adopt the interventional mediation framework to decompose the total effect into an interventional direct effect (IDE) and an interventional indirect effect (IIE) (Vansteelandt & Daniel, 2017).

Using the mediator M_i introduced earlier to denote the subsequent reasoning, we similarly define the do operation $\mathrm{do}(M_i = m)$, which sets the subsequent reasoning text to m. Let $G_{i,k}^{(0)}$ and $G_{i,k}^{(1)}$ denote the distributions of subsequent reasoning generated by the same language model under the context C_i, with the ith reasoning step set to $a_{i,k}^{(0)}$ and $a_{i,k}^{(1)}$, respectively. We then define the interventional direct effect (IDE) and interventional indirect effect (IIE) as

$$\mathrm{IDE}_{i,k} = \Psi_{i,k}(1,0) - \Psi_{i,k}(0,0), \qquad \mathrm{IIE}_{i,k} = \Psi_{i,k}(1,1) - \Psi_{i,k}(1,0), \tag{9}$$

where, for $j, \ell \in \{0, 1\}$,

$$\Psi_{i,k}(j,\ell) = \mathbb{E}_{m \sim G_{i,k}^{(\ell)}} \left\{ \mathbb{E}\big[h(A) \mid \mathrm{do}(R_i = a_{i,k}^{(j)}), \mathrm{do}(M_i = m), C_i \big] \right\}. \tag{10}$$

In equation 10, the inner expectation is taken over the distribution of the answer A, while the outer expectation averages over the mediator distribution $G_{i,k}^{(\ell)}$. Table 2 summarizes the intervention conditions for each $\Psi_{i,k}$.

Again, in words, the IDE measures the average effect of switching the target step from its original version to the intervention while holding the distribution of subsequent reasoning fixed at that induced by the original step. The IIE measures the average effect of switching the distribution of subsequent reasoning from that induced by the original step to that induced by the intervention, while holding the target step fixed at its intervened version.

To estimate the IDE and IIE, we similarly use Monte Carlo methods. For each condition in Table 2, we fix the target step and the subsequent reasoning at the corresponding versions, regenerate the final answer, and compute the sample mean of $h(A)$. We then estimate $\mathrm{IDE}_{i,k}$ and $\mathrm{IIE}_{i,k}$ as in equation 2; see Appendix A.3 for details.

4 Experiments

In this section, we conduct experiments to evaluate the proposed approach. We consider three models, *Qwen3.8-27B*, *Qwen3.5-9B*, and *Mistral-Small-3.2-24B*, to examine whether our step-level analyses are consistent across model scales and architectures. Specifically, the two Qwen models allow us to examine how the analysis changes with model scale while holding the model family fixed. In contrast, Mistral-Small-3.2-24B allows us to assess whether the results hold under a different model architecture at a parameter scale comparable to Qwen3.8-27B.

We also consider four benchmark datasets covering different reasoning tasks. *Basic Arithmetic* and *Knights & Knaves* from Reasoning Gym consider relatively simple arithmetic and logical reasoning tasks, respectively. *GSM8K* evaluates more complex mathematical questions and *MMLU-Pro*

Table 3: Summary of benchmark datasets and step-level causal effects. For each model and benchmark, we report the number of problems and the average number of content-bearing, structural, and invalid reasoning steps per problem. Under Discoveries, we report the average number of steps per problem identified as significant by e-BH, the discovery rate, and the mean absolute total effect across significant steps. The discovery rate is the percentage of significant steps among all content-bearing and structural steps, pooled over problems; invalid steps are excluded.

		Average steps per problem			Discoveries		
Benchmark dataset	Number of problems	Content-bearing	Structural	Invalid	Significant per problem	Rate (%)	Mean $\lvert TE \rvert$
Qwen3.8-27B							
Basic Arithmetic	100	3.27	0.00	0.00	2.82	86.2	0.897
Knights & Knaves	100	4.07	0.00	0.03	3.05	74.9	0.508
GSM8K	300	3.38	0.02	0.00	3.03	89.3	0.925
MMLU-Pro	300	3.72	0.01	0.00	1.53	41.0	0.550
Qwen3.5-9B							
Basic Arithmetic	100	3.00	0.00	0.00	2.49	83.0	0.892
Knights & Knaves	100	2.82	0.00	0.01	1.33	47.2	0.461
GSM8K	300	3.11	0.00	0.00	2.80	90.0	0.943
MMLU-Pro	300	2.90	0.00	0.01	1.42	49.1	0.588
Mistral-Small-3.2-24B							
Basic Arithmetic	100	3.72	0.08	0.03	2.78	73.2	0.800
Knights & Knaves	100	4.50	0.11	0.11	2.61	56.6	0.443
GSM8K	300	3.72	0.04	0.00	3.22	85.6	0.892
MMLU-Pro	300	4.72	0.08	0.05	1.50	31.3	0.581

evaluates general knowledge-intensive multiple-choice problems. Table 3 reports the number of problems and reasoning steps in each benchmark dataset. It also reports the number and percentage of reasoning steps identified as significant by e-BH, together with the average absolute total effect across significant steps. Figure 2 further reports the average absolute total effects across different intervention types.

We make a few observations. First, across all three models, the two mathematical benchmarks, Basic Arithmetic and GSM8K, exhibit both higher discovery rates and larger average absolute total effects than Knights & Knaves and MMLU-Pro. This suggests that individual reasoning steps are generally more strongly tied to the final answer in mathematical reasoning tasks. This observation is consistent with prior work showing that reasoning models' performance varies substantially across tasks (Lanham et al., 2023), and that reasoning is particularly useful for multi-step mathematical problems such as GSM8K (Cobbe et al., 2021; Wei et al., 2022). Second, the overall pattern is relatively stable across model families: Basic Arithmetic and GSM8K show strong step-level effects for the three models, although the discovery rates and effect sizes vary across models.

Figure 3 further illustrates why mathematical reasoning can exhibit stronger step-level dependence. Both examples are generated by Qwen3.8-27B. Additional examples from other models and benchmarks are provided in Appendix C. In the GSM8K example, all six steps have significant effects: Steps 1–5 act almost entirely through indirect effects and the final step acts directly on the answer. This pathway is consistent with the example: each intermediate quantity is carried forward into later calculations, so the effect of an early step propagates through the remainder of the reasoning chain. In contrast, in the MMLU-Pro example, only Steps 1 and 4 are significant. Step 1 has a purely indirect effect, consistent with its role in explaining the definition used by the subsequent steps.

5 Discussion

This paper proposes a methodology for assessing the significance of individual reasoning steps for the final answer and for characterizing the causal pathways through which their effects arise. Across our experiments, we find that mathematical reasoning tasks contain a larger proportion of causally relevant reasoning steps than other tasks, and exhibit strong sequential patterns, with earlier steps acting primarily through indirect effects and the last step exerting direct effects on the final answer.

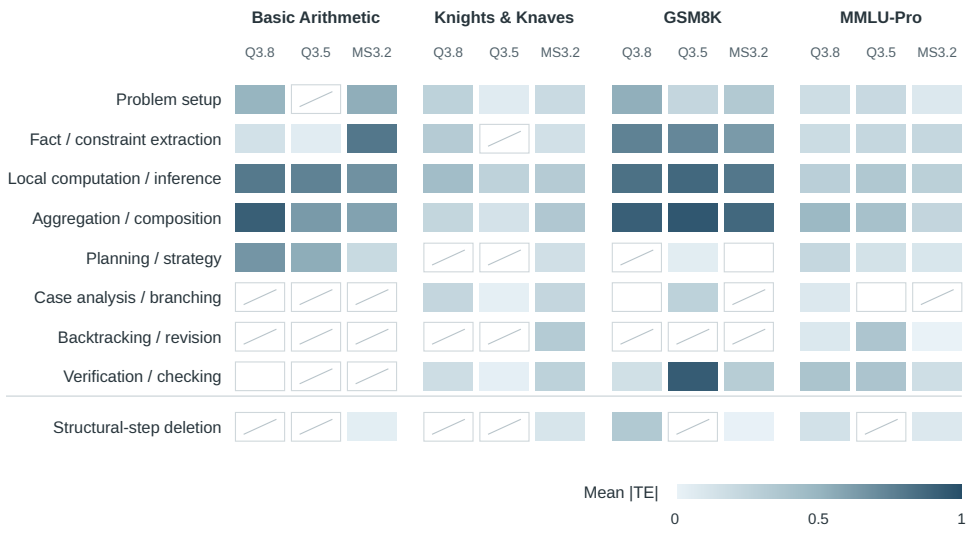

Figure 2: Average absolute total effects across reasoning steps by model, benchmark dataset, and intervention type; see Appendix A.1.1 for detailed definitions of the interventions. Cell color indicates the size of the total effect. A "/" denotes cases in which the corresponding intervention is not applicable to any reasoning step produced under that model–benchmark combination.



Figure 3: Two examples from GSM8K and MMLU-Pro. Each shows the prompt (top), reasoning trace with total, direct, and indirect effects (middle), and final answer (bottom).

AI USE STATEMENT

In this work, generative AI plays two distinct roles. First, LLMs are a component of our experimental protocol: we use them to construct the interventions, classify reasoning steps as content-bearing, structural, or invalid and assign content-bearing steps to functional categories. These uses are part of the method rather than research assistance; their instructions, inputs, and decoding settings are described in Appendix B, and every accepted intervention was fixed before outcome sampling and manually spot-checked by the authors. Second, we used generative AI tools to implement methods, namely to assist in writing the code for our experimental pipeline, including rollout generation, answer extraction, effect estimation, and e-value computation. We also used them to provide feedback on the experimental design and to assist in analyzing experimental results; all design decisions and scientific conclusions were made by the authors. We have not used generative AI tools to develop the causal framework, formulate mathematical claims, write proofs, or propose hypotheses: the research idea, the causal framework, and the statistical methodology are original to the authors. Cleaning datasets and translation are not applicable to this work. Additionally, we used generative AI tools for literature search and summarization (deep research), for drafting parts of the manuscript, and for creating figures. We have reviewed all AI-assisted work: every cited reference was verified against its original source, all AI-assisted code was checked and tested by the authors, and all AI-drafted text and figures were manually revised by the authors, who wrote the final version of the manuscript. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work involves no human subjects, crowdsourcing, or personal data. All experiments use publicly available models and public benchmarks, used in accordance with their licenses. Our interventions insert errors only into reasoning text generated by the models themselves, for the purpose of analysis; no generated content is deployed or presented to users. Because the framework identifies the steps whose corruption most affects the final answer, it could in principle inform targeted manipulation of reasoning traces; we believe this risk is outweighed by its value for locating where verification and monitoring are most needed. Our estimates are conditional on the specified model, prompt, interventions, and outcome function, and do not by themselves certify that a reasoning trace is faithful or safe. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

Code will be publicly released upon publication.

REFERENCES

Amir Asiaee. Certified interventional fidelity: Anytime-valid, adaptive evaluation of causal claims in mechanistic interpretability. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, 2026. arXiv:2607.08349.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which LLM reasoning steps matter?, 2025.

Jannik Brinkmann, Abhay Sheshadri, Victor Levoso, Paul Swoboda, and Christian Bartelt. A mechanistic analysis of a transformer trained on a symbolic multi-step reasoning task, 2024.

Vivien Cabannes, Charles Arnal, Wassim Bouaziz, Xingyu Alice Yang, Francois Charton, and Julia Kempe. Iteration head: A mechanistic study of chain-of-thought. In *Advances in Neural Information Processing Systems*, 2024.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don't always say what they think, 2025.

James Chua and Owain Evans. Are DeepSeek R1 and other reasoning models more faithful?, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Olive Jean Dunn. Multiple comparisons among means. *Journal of the American statistical association*, 56(293):52–64, 1961.

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning, 2024.

Jiarun Fu, Lizhong Ding, Hao Li, Pengqi Li, Qiuning Wei, and Xu Chen. Unveiling and causalizing CoT: A causal perspective, 2025.

Leo Gao. Shapley value attribution in chain of thought, 2023. LessWrong.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks, 2021.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability, 2025.

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does GPT-2 compute greater-than? interpreting mathematical abilities in a pre-trained language model, 2023.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 2021.

Yi Hu, Jiaqi Gu, Ruxin Wang, Zijun Yao, Hao Peng, Xiaobao Wu, Jianhui Chen, Muhan Zhang, and Liangming Pan. Towards a mechanistic understanding of large reasoning models: A survey of training, inference, and failures. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19449–19466, San Diego, California, United States, 2026. Association for Computational Linguistics. URL https://aclanthology.org/2026.acl-long.889/.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning, 2023.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, et al. On the biology of a large language model. *Transformer Circuits Thread*, 2025.

Uzay Macar, Paul C. Bogdan, Senthooran Rajamanoharan, and Neel Nanda. Thought branches: Interpreting LLM reasoning requires resampling. In *International Conference on Learning Representations*, 2026. arXiv:2510.27484.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022. arXiv:2202.05262.

Gouki Minegishi, Hiroki Furuta, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. Topology of reasoning: Understanding large reasoning models through reasoning graph properties. In *Advances in Neural Information Processing Systems*, 2025.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15012–15032, 2024.

Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 411–420, 2001.

James M. Robins and Sander Greenland. Identifiability and exchangeability for direct and indirect effects. *Epidemiology*, 3(2):143–155, 1992.

Yuxi Sun, Aoqi Zuo, Haotian Xie, Wei Gao, Mingming Gong, and Jing Ma. FACT-E: Causality-inspired evaluation for trustworthy chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. arXiv:2604.10693.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, 2023. arXiv:2305.04388.

Tyler J. VanderWeele, Stijn Vansteelandt, and James M. Robins. Effect decomposition in the presence of an exposure-induced mediator-outcome confounder. *Epidemiology*, 25(2):300–306, 2014.

Stijn Vansteelandt and Rhian M. Daniel. Interventional effects for mediation analysis with multiple mediators. *Epidemiology*, 28(2):258–265, 2017.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. In *Workshop on Reasoning and Planning for Large Language Models*, 2025.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural NLP: The case of gender bias, 2020a.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020b.

Sydney Von Arx and Amy Deng. CoT may be highly informative despite "unfaithfulness", 2025. METR blog, August 2025.

Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination, and applications. *The Annals of Statistics*, 2021.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023. arXiv:2211.00593.

Ruodu Wang and Aaditya Ramdas. False discovery rate control with e-values. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(3):822–852, 2022. doi: 10.1111/rssb.12489.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(1):1–27, 2024. doi: 10.1093/jrsssb/qkad009. Discussion paper.

Dennis Wei, Yannis Belkhiter, Erik Miehling, and Radu Marinescu. Local causal attribution of chain-of-thought reasoning, 2026. Mechanistic Interpretability Workshop at ICML 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Zidi Xiong, Shan Chen, Zhenting Qi, and Himabindu Lakkaraju. Measuring the faithfulness of thinking drafts in large reasoning models, 2025.

Xiangning Yu, Zhuohan Wang, Linyi Yang, Haoxuan Li, Anjie Liu, Xiao Xue, Jun Wang, and Mengyue Yang. Causal sufficiency and necessity improves chain-of-thought reasoning, 2025.

Kerem Zaman and Shashank Srivastava. Is chain-of-thought really not explainability? chain-of-thought can be faithful without hint verbalization, 2025.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. In *Conference on Language Modeling*, 2025. arXiv:2504.05419.

Zheng Zhao, Yeskendir Koishekenov, Xianjun Yang, Naila Murray, and Nicola Cancedda. Verifying chain-of-thought reasoning via its computational graph. In *International Conference on Learning Representations*, 2026. Oral. arXiv:2510.09312.

A METHODOLOGY DETAILS

This appendix complements Section 3 and follows its notation: R_i is the target step, C_i consists of the prompt and the preceding steps, and each intervention $k \in \mathcal{K}_i$ defines an original version $a_{i,k}^{(0)}$ and an intervened version $a_{i,k}^{(1)}$ of R_i. The LLMs and prompts used to classify steps and construct interventions are described in Appendix B.

A.1 STEP CLASSIFICATION AND INTERVENTION CONSTRUCTION

The target model writes its reasoning as numbered `[STEP i]` blocks followed by one `[FINAL]` block that contains the answer (Appendix B.2). A deterministic parser splits each output into steps, and outputs that do not follow this format are discarded. Each remaining step is then classified as content-bearing, structural, or invalid (Section 3). A step is content-bearing if it contributes task-relevant reasoning, regardless of whether this reasoning is correct or whether the step affects the answer.

The intervention set $\mathcal{K}_i$ depends on the category of R_i. Each content-bearing step is assigned one or more of the eight functions in Appendix A.1.1, and each assigned function defines one intervention $k \in \mathcal{K}_i$: $a_{i,k}^{(0)}$ is the generated step, and $a_{i,k}^{(1)}$ replaces the principal contribution of the step under that function (defined below) while keeping the rest of the step unchanged. A step with several functions therefore has several interventions. Each structural step has a single deletion intervention (Appendix A.1.2), and invalid steps are not intervened on. All interventions are constructed and validated before any continuation is sampled, and none of the LLMs used for this purpose sees the final answer, the gold answer, or any effect estimate.

A.1.1 EIGHT FUNCTION-SPECIFIC INTERVENTIONS

Each box below defines one function, separates it from similar functions, and specifies the corresponding intervention. We call the content that a step contributes under a given function its *principal contribution*, for example, the new value produced by a local computation or the verdict of a verification. An intervention replaces the principal contribution and keeps everything else fixed, including the function of the step, its other content, and the context C_i. The replacement must also satisfy the criterion stated in the box, which makes it an error with respect to the question and C_i; we refer to this requirement as the *local error criterion*. For background facts and planning choices, correctness cannot always be determined from the question and C_i. In these two cases, any substantive replacement that serves the same role is accepted, which we refer to as the *operational criterion* of our protocol; such a replacement is not guaranteed to be objectively wrong.

Problem Setup

Definition and boundary. A step has this function when it assigns a meaning or role to a variable, quantity, object, target quantity, or other representational element. It establishes a binding; it does not merely extract a premise or select a subsequent operation.

Intervention and criterion. Replace the principal binding with a different binding serving the same setup role. Keep the task, problem facts, available entities, and other setup bindings fixed. The replacement meets the local error criterion only if it conflicts with a role, unit, target quantity, or other representation requirement stated or uniquely determined by the question or preceding context.

Fact or Constraint Extraction

Definition and boundary. A step has this function when it states or invokes an existing fact, quantity, relation, condition, or constraint needed later, either from explicit context or background knowledge invoked in the trace. It does not derive a new result or establish a new representational binding.

Intervention and criterion. Replace the principal fact or constraint with substantively different content serving the same entity and relation role. Keep the question, preceding context, and all other facts and constraints fixed. If the target is stated explicitly in the question or preceding context, the replacement must contradict that statement. For a background-knowledge assertion not stated there, a substantive substitution meets the protocol's operational criterion without independently verifying the assertion's truth. A paraphrase or equivalent value does not qualify.

Local Computation or Inference

Definition and boundary. A step has this function when it applies available operands or premises to produce a new local value, relation, proposition, or conclusion. This differs from stating an existing premise, choosing a future operation, combining several established subresults, or checking an existing result.

Intervention and criterion. Replace only the new principal result. Keep the inputs, premises, upstream information, entities, and computation or inference relation fixed. For a computation or transformation, the replacement must differ from the result of applying the unchanged operation to the unchanged inputs. For an inference, it must not be entailed by the unchanged premises under the step's stated inference relation.

Aggregation or Composition

Definition and boundary. A step has this function when it combines multiple previously established component results into a higher-level aggregate or composite result. Selecting an aggregation operation for a later step is planning; producing a local result directly from operands is local computation.

Intervention and criterion. Replace only the aggregate or composite output. Keep the component inputs, objective, aggregation relation or operation, and involved entities fixed. The replacement meets the local error criterion when it is not semantically equivalent to the result of applying that operation to the unchanged components.

Planning or Strategy

Definition and boundary. A step has this function when it selects a next operation, method, subgoal, information request, or strategy before executing it. An explicit condition-to-branch decision belongs to case analysis; a computed result belongs to local computation.

Intervention and criterion. Replace the principal next-step choice with a substantively different choice serving the same planning role. Keep the task, current state, available information, stated objective, and explicit admissibility constraints fixed. A substitution between choices explicitly allowed or stated to be equally valid for that state and objective does not meet the criterion. Otherwise, a substantive same-role substitution meets the operational criterion; this alone does not establish that the alternative strategy is objectively incorrect.

Case Analysis or Branching

Definition and boundary. A step has this function when it evaluates a condition and selects a branch under an explicit condition-to-branch mapping. A general choice of next method without such a mapping is planning.

Intervention and criterion. Replace the principal branch decision, including the condition verdict when it forms part of that decision. Keep the tested condition, evidence, available branches, mapping, and entities fixed. The replacement meets the local error criterion if its verdict conflicts with the unchanged evidence or its branch is not assigned by the unchanged mapping.

Backtracking or Revision

Definition and boundary. A step has this function when it returns to a previously stated value, assumption, relation, or conclusion and proposes a correction. Evaluating existing content without changing it is verification.

Intervention and criterion. Replace the principal proposed correction while keeping the revision target, correction evidence or reason, and unrelated reasoning history fixed. The replacement meets the local error criterion if it leaves the identified erroneous content unchanged or differs from the correction determined by the unchanged evidence.

Verification or Checking

Definition and boundary. A step has this function when it evaluates an existing value, derivation, candidate solution, or conclusion against a stated condition, constraint, or consistency criterion. Producing a new result is local computation; proposing a correction is revision.

Intervention and criterion. Replace only the principal verification verdict. Keep the checked object, verification evidence, criterion, and checking procedure, when present, fixed. The replacement meets the local error criterion when it differs from the verdict obtained by applying the unchanged criterion to the unchanged object and evidence.

A.1.2 Structural-Step Deletion

For a structural step, $a^{(1)}_{i,k}$ deletes the body of the step but keeps the `[STEP i]` marker and the surrounding whitespace, so that C_i and the step numbering are unchanged. The deletion is deterministic and does not involve the intervention generator or validator. We accept the edited text only if encoding it with the tokenizer and decoding it back reproduces the text exactly.

A.2 Outcome Function

All effects in Section 3 are defined through a prespecified outcome function $h(A)$ of the extracted answer $A = A(Y)$. In our experiments, we use the task score

$$h_{\text{task}}(A) = \begin{cases} \mathbb{I}\{A = A^\star\}, & \text{if the benchmark is scored by exact match,} \\ S(A), & \text{otherwise,} \end{cases} \tag{11}$$

where the gold answer $A^\star$ is processed by the same parser as A, and S is the scoring function provided by the benchmark, taking values in $[0, 1]$. This extends the example $h(A) = \mathbb{I}\{A = A^\star\}$ in Section 3 to benchmarks with their own scoring functions. An unparseable answer, $A = \text{NULL}$, receives $h_{\text{task}}(A) = 0$.

A.3 Estimation of the Direct and Indirect Effects

Fix a step–intervention pair (i, k). For brevity, we write $\Psi(j, \ell)$ for $\Psi_{i,k}(j, \ell)$ in equation 10 and $G^{(\ell)}$ for $G^{(\ell)}_{i,k}$. We call the continuations used to estimate the total effect in equation 2 *free continuations*, since the model generates both the subsequent reasoning and the answer. Estimating $\Psi(j, \ell)$ instead requires holding the subsequent reasoning fixed and regenerating only the answer, which we call *fixed-mediator generation*.

Draws from $G^{(0)}$ and $G^{(1)}$ are already available from the free continuations: they are the subsequent reasoning $M_{i,0,s}$ and $M_{i,1,s}$, $s = 1, \ldots, N$. We store their token sequences and select B_0 draws from arm 0 and B_1 draws from arm 1, using a rule fixed before any outcome is observed. Denote the selected draws by $m_{\ell,b}$, $b = 1, \ldots, B_\ell$, for $\ell \in \{0, 1\}$. For each condition (j, ℓ) in Table 2 and each draw $m_{\ell,b}$, we set the target step to $a_{i,k}^{(j)}$ and the subsequent reasoning to $m_{\ell,b}$, and regenerate the answer L times with the same model and decoding configuration. Conditions $(0, 0)$ and $(1, 0)$ therefore use the same arm-0 draws, and condition $(1, 1)$ uses the arm-1 draws. Let $A_{j\ell,b,r}$ denote the rth regenerated answer, and define the average outcome for draw b as

$$\overline{H}_{j\ell,b} = \frac{1}{L} \sum_{r=1}^{L} h(A_{j\ell,b,r}). \tag{12}$$

We estimate each $\Psi(j, \ell)$ by averaging over the selected draws:

$$\widehat{\Psi}(0,0) = \frac{1}{B_0} \sum_{b=1}^{B_0} \overline{H}_{00,b}, \qquad \widehat{\Psi}(1,0) = \frac{1}{B_0} \sum_{b=1}^{B_0} \overline{H}_{10,b}, \qquad \widehat{\Psi}(1,1) = \frac{1}{B_1} \sum_{b=1}^{B_1} \overline{H}_{11,b}. \tag{13}$$

These equal-weight averages are appropriate when all N draws are used or when the B_ℓ draws are selected uniformly at random; any other selection rule requires corresponding sampling weights. Draws with identical text are kept as separate draws.

Following equation 9, we estimate

$$\widehat{\mathrm{IDE}}_{i,k} = \frac{1}{B_0} \sum_{b=1}^{B_0} \big(\overline{H}_{10,b} - \overline{H}_{00,b}\big), \qquad \widehat{\mathrm{IIE}}_{i,k} = \widehat{\Psi}(1,1) - \widehat{\Psi}(1,0). \tag{14}$$

The IDE estimate is thus a mean of paired differences over the same arm-0 draws. Since $\widehat{\Psi}(1, 0)$ appears in both estimates, the following identity holds exactly:

$$\widehat{\mathrm{IDE}}_{i,k} + \widehat{\mathrm{IIE}}_{i,k} = \widehat{\Psi}(1,1) - \widehat{\Psi}(0,0). \tag{15}$$

A.4 Relation Between the Total Effect and Its Decomposition

By equation 9, $\mathrm{IDE}_{i,k} + \mathrm{IIE}_{i,k} = \Psi(1, 1) - \Psi(0, 0)$. For $j \in \{0, 1\}$, let $\theta_j = \mathbb{E}[h(A) \mid \mathrm{do}(R_i = a_{i,k}^{(j)}), C_i]$ denote the mean outcome under free continuation, so that $\mathrm{TE}_{i,k} = \theta_1 - \theta_0$ by equation 1. The decomposition therefore recovers the total effect if $\Psi(j, j) = \theta_j$ for $j \in \{0, 1\}$. A sufficient condition is that, for $j \in \{0, 1\}$ and $G^{(j)}$-almost every m,

$$\mathbb{E}\big[h(A) \mid \mathrm{do}(R_i = a_{i,k}^{(j)}),\, M_i = m,\, C_i\big] = \mathbb{E}\big[h(A) \mid \mathrm{do}(R_i = a_{i,k}^{(j)}),\, \mathrm{do}(M_i = m),\, C_i\big]. \tag{16}$$

In words, the answer that follows m in a free continuation has the same mean as the answer regenerated after the subsequent reasoning is fixed to m. Averaging both sides over $m \sim G^{(j)}$ gives $\theta_j = \Psi(j, j)$, and hence $\mathrm{TE}_{i,k} = \mathrm{IDE}_{i,k} + \mathrm{IIE}_{i,k}$. Because the answer is generated token by token given the preceding text, equation 16 is expected to hold when fixed-mediator generation uses the same model, decoding configuration, and token sequence as the free continuation and starts the answer at the same position. We check these conditions in our implementation.

Even when equation 16 holds, $\widehat{\mathrm{TE}}_{i,k}$ and $\widehat{\mathrm{IDE}}_{i,k} + \widehat{\mathrm{IIE}}_{i,k}$ differ in finite samples, because they are computed from different generations. We call their difference, $\widehat{\mathrm{TE}}_{i,k} - (\widehat{\mathrm{IDE}}_{i,k} + \widehat{\mathrm{IIE}}_{i,k})$, the *bridge gap*; Appendix C.1 reports it on MMLU-Pro. When equation 16 is not supported, the IDE and IIE still decompose $\Psi(1, 1) - \Psi(0, 0)$, but this quantity need not equal the total effect, and the two should be interpreted separately.

A.5 Uncertainty Intervals

We construct intervals by the bootstrap. For the total effect, we resample free continuations separately within each arm. For the IDE and IIE, we resample the selected draws: arm-0 draws are resampled once and used in both conditions $(0, 0)$ and $(1, 0)$, which preserves their pairing, and arm-1 draws are resampled independently. When the variability of regenerated answers is included, we also resample the L answers within each draw. For summaries across problems, problems are the top-level resampling unit. Since the IDE and IIE are estimated only for steps selected by e-BH, their intervals describe these selected steps and are not used to declare an IDE or IIE significant.

B EXPERIMENTAL SETUP AND PROMPTS

This appendix describes the LLMs and prompts used to classify reasoning steps and construct the interventions in Appendix A.1.

B.1 LLM ROLES

We use LLMs in five roles, summarized in Table 4. The target model generates the reasoning trace and is one of the three models in Section 4. The other four roles classify steps and construct interventions, and all of them use Qwen3.8-27B.

Table 4: LLM roles and decoding settings. Sampling uses temperature 0.7, top-p 0.8, and top-k 20. Where the output limit differs across experiments, it is listed by target model.

Role	Model	Decoding	Maximum output tokens
Target reasoning	Target model	Sampling	1024
Step classification	Qwen3.8-27B	Greedy	256 (Qwen targets); 32 (Mistral target)
Function labeling	Qwen3.8-27B	Greedy	256 (Qwen3.8-27B target); 512 (others)
Intervention generation	Qwen3.8-27B	Sampling	512
Intervention validation	Qwen3.8-27B	Greedy	2048

B.2 PROMPTS

For each role, we give the instruction and describe its input, output, and the demonstrations included in the prompt. We use few-shot in-context learning throughout, with each prompt containing task-specific demonstrations.

Target Reasoning Prompt

Input. The target model receives the problem statement.

Instruction.

> Solve the problem using a sequence of numbered reasoning steps. Each `[STEP i]` block should express one coherent local objective or conclusion. Keep calculations and explanations together when they serve that same objective, but separate conclusions that could stand and be modified independently. Number the steps consecutively. After the last reasoning step, write one `[FINAL]` block and place the answer exactly once between `<answer>` and `</answer>`. Do not add reasoning after the final answer.

The `[FINAL]` block may contain a short concluding sentence around the answer. Empty steps are not allowed; when no further reasoning is needed, the model moves to the `[FINAL]` block. All three target models use the same instruction, followed by seven demonstrations of the step format and step boundaries.

Step Classification Prompt

Input. The step classifier receives the context C_i and the current step R_i. It does not see subsequent steps, the final answer, the gold answer, or any effect estimate.

Instruction.

> First decide whether the generated block is a coherent step-level unit. If it is incomplete or combines independently meaningful reasoning units, label it *invalid*. Otherwise, label it *content-bearing* when it contributes substantive task-relevant reasoning, and *structural* when it only organizes or transitions between reasoning steps. Judge the unit and its contribution, not whether its conclusion is correct. Do not split, rewrite, or repair the step.

A step may contain several operations if they serve one local objective. A step is not invalid merely because its calculation or conclusion is wrong, or because an intervention on it would be hard to construct. A coherent transition is structural rather than invalid.
When the target model is Qwen3.8-27B or Qwen3.5-9B, the classifier returns the category together with a short supporting span and a brief rationale; when the target model is Mistral-Small-3.2-24B, it returns only the category. The prompt includes demonstrations of the three categories and of the intended step boundaries.

Function-Labeling Prompt

Input. The function labeler receives the context C_i and one content-bearing step R_i.

Instruction.

> Assign the function or functions performed by this step, using only the eight definitions above. Assign multiple functions only when the same step actually performs each of them; do not use multiple labels to express uncertainty. Return no label if none applies. Classify the step's contribution without judging its correctness, causal importance, or the eventual answer. Do not inspect subsequent reasoning or alter the step.

The definitions separate a stated fact from a newly derived result, the choice of an operation from its execution, and checking a result from revising it. A step that invokes background knowledge counts as fact extraction even if the fact does not appear in the question. Transitional steps are already classified as structural and receive no function.
When the target model is Qwen3.8-27B or Qwen3.5-9B, the labeler also returns a supporting quotation and a brief rationale for each label; when the target model is Mistral-Small-3.2-24B, it returns only the labels. The demonstrations cover all eight functions, one step with multiple functions, and one step with none.

Intervention-Generation Prompt

Input. The intervention generator receives the context C_i, the text of the original step $a_{i,k}^{(0)}$, and the intervention rule for its assigned function (Appendix A.1.1).

Instruction.

> Propose one replacement for the current step body. Change only the principal contribution named by the function-specific rule; preserve the step's function, all other semantic content, and the preceding context. The change must satisfy that rule's local criterion. Do not merely paraphrase the step, change an upstream premise, add subsequent reasoning, or add a final answer. If no compliant replacement can be produced on this attempt, report that no candidate was constructed.

As noted in Appendix A.1.1, a replacement that satisfies the rule for a background fact or a planning choice is not guaranteed to be objectively wrong. The generator returns only the new step text, without the step marker or the `[FINAL]` marker. Each call includes one demonstration of a valid replacement for the given function; calls for planning steps include a second planning demonstration.

Intervention-Validation Prompt

Input. The intervention validator receives the context C_i, the original step, the candidate replacement, and the same rule given to the generator.

Instruction.

> Decide whether the candidate follows the assigned rule. Accept only if it changes the designated contribution, preserves the step's function and all non-target content, satisfies the function-specific local criterion, remains within the intended step, is semantically complete, and contains no downstream reasoning or final answer. Evaluate the candidate; do not repair or rewrite it.

The validator checks eight conditions: (1) the candidate follows the assigned rule; (2) it changes the principal contribution named by the rule; (3) it preserves the function of the step; (4) it keeps all other content unchanged; (5) it satisfies the local error criterion; (6) it stays within the current step; (7) it is semantically complete; and (8) it contains no subsequent reasoning or final answer. Condition (5) is checked only after conditions (1)–(4) pass, and a candidate is accepted only if all eight pass.

When a rule requires a conflict with an explicit statement, that statement must come from the question or C_i; neither the original nor the candidate step can serve as evidence. The validator does not see final answers, outcomes, effect estimates, or which steps are selected by e-BH. Each call includes accepted and rejected examples for the given function. The validator returns its decision, the result of each check, and a short explanation, and it does not rewrite rejected candidates.

C Additional Results

C.1 Agreement Between the Total Effect and Its Decomposition

Under equation 16, the total effect equals the sum of the IDE and IIE. The estimates, however, come from different generations: the total effect from free continuations, and the IDE and IIE from fixed-mediator generations (Appendix A.3). Figure 4 shows the resulting bridge gap for 571 step–intervention pairs from 228 MMLU-Pro problems for which both estimates are available, with answers rescored by their option label. The gap is exactly zero for 375 pairs (65.7%) and at most 0.05 in absolute value for 533 pairs (93.3%), so the two estimates agree closely for most pairs.

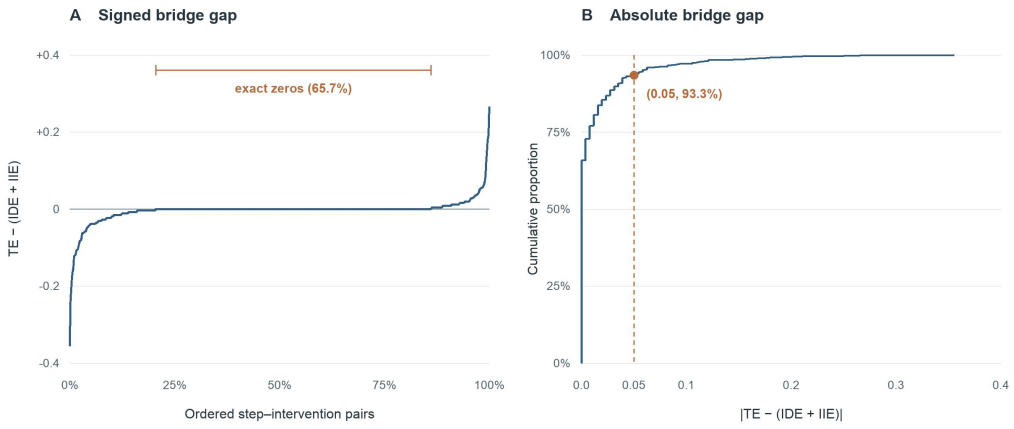

Figure 4: Bridge gap $\widehat{\mathrm{TE}} - (\widehat{\mathrm{IDE}} + \widehat{\mathrm{IIE}})$ for 571 step–intervention pairs from 228 MMLU-Pro problems, with answers rescored by option label. The total effect is estimated from free continuations, and the IDE and IIE from fixed-mediator generations. (A) Signed gaps in increasing order. (B) Cumulative proportion of pairs by absolute gap.

C.2 Cross-Model Comparison on a Fixed MMLU-Pro Case

We fix MMLU-Pro question 219, its Qwen3.8-27B reasoning trace, and three interventions on this trace, and generate continuations with each of the three target models. Table 5 reports the results.

Table 5: Total, direct, and indirect effects of the same three interventions on MMLU-Pro question 219, with continuations generated by each target model and 64 free continuations per arm. For Qwen3.5-9B and Mistral-Small-3.2-24B, IDE and IIE estimates whose 95% cluster-bootstrap intervals include zero are shown as 0.

Step	Target model	TE	IDE	IIE
1	Qwen3.8-27B	0	0	0
1	Qwen3.5-9B	0	0	0
1	Mistral-Small-3.2-24B	0	0	0
2	Qwen3.8-27B	-1	0	-1
2	Qwen3.5-9B	-0.031	0	0
2	Mistral-Small-3.2-24B	-0.281	0	-0.328
3	Qwen3.8-27B	-1	-1	0
3	Qwen3.5-9B	-1	-1	0
3	Mistral-Small-3.2-24B	-1	-1	0

At Step 2, the intervention reverses the priority of preferred and common stock. It yields $\mathrm{TE} = -1$ for Qwen3.8-27B, -0.281 for Mistral-Small-3.2-24B, and -0.031 for Qwen3.5-9B, and the interval for Qwen3.5-9B includes zero. For Qwen3.8-27B and Mistral-Small-3.2-24B, the IIE is negative, consistent with the erroneous premise propagating through the subsequent reasoning. At Step 3, the intervention replaces the stated answer option with an incorrect one and yields $\mathrm{TE} = \mathrm{IDE} = -1$ for all three models. This single case shows that the same intervention can have different effects across models; it does not show how often such differences occur, nor what happens when the original answer is incorrect.

C.3 Additional Step-Level Examples

Figures 5–14 show one example problem for each benchmark and each target model, excluding the two examples in Figure 3. The same problem is used across models within each benchmark. Each figure follows the format of Figure 1: the prompt (top), the reasoning trace with the total, direct, and indirect effect of each step (middle), and the final answer (bottom).

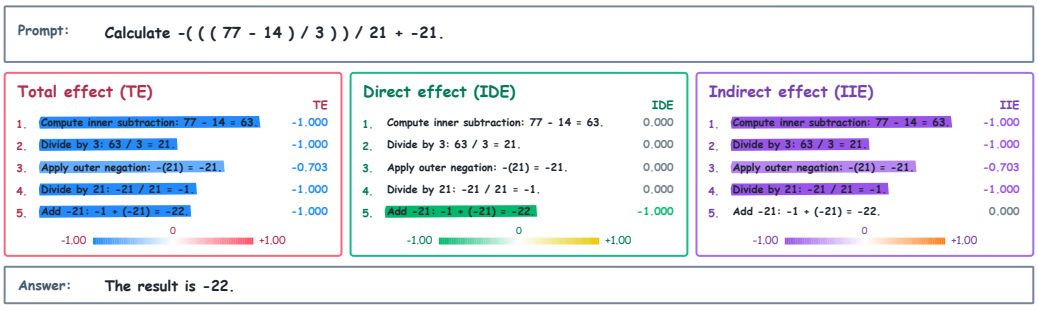

Figure 5: Step-level effects for Basic Arithmetic question 46, Qwen3.8-27B.

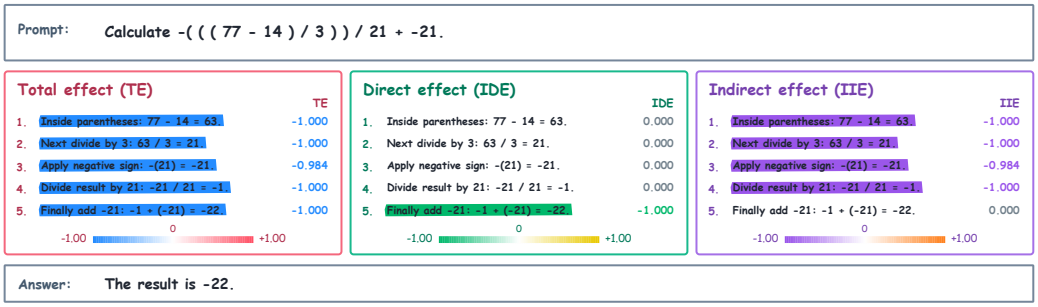

Figure 6: Step-level effects for Basic Arithmetic question 46, Qwen3.5-9B.

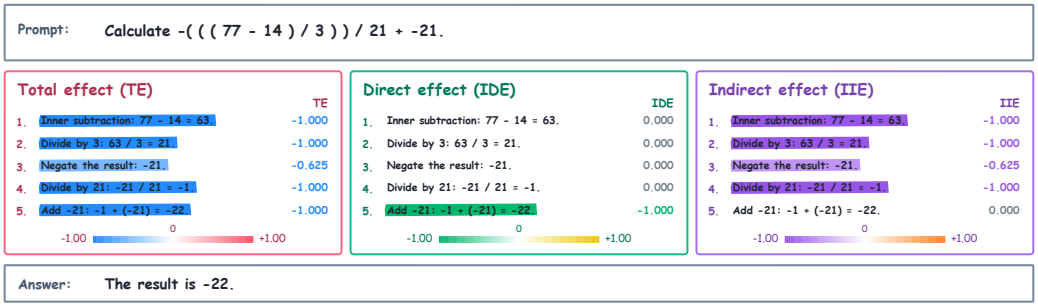

Figure 7: Step-level effects for Basic Arithmetic question 46, Mistral-Small-3.2-24B.

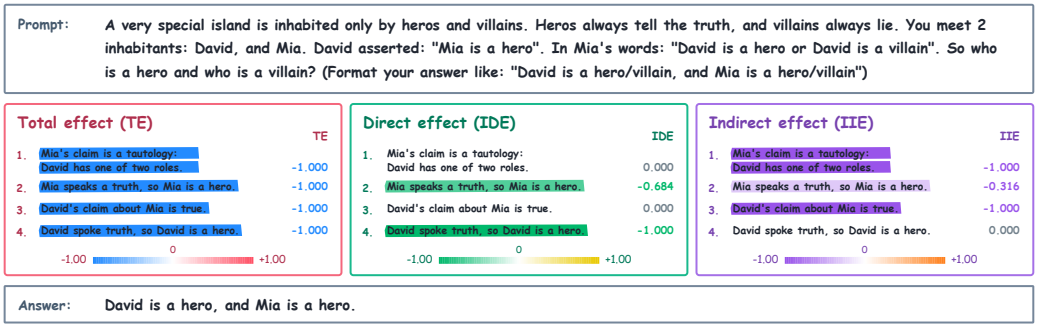

Figure 8: Step-level effects for Knights & Knaves case 17, Qwen3.8-27B.

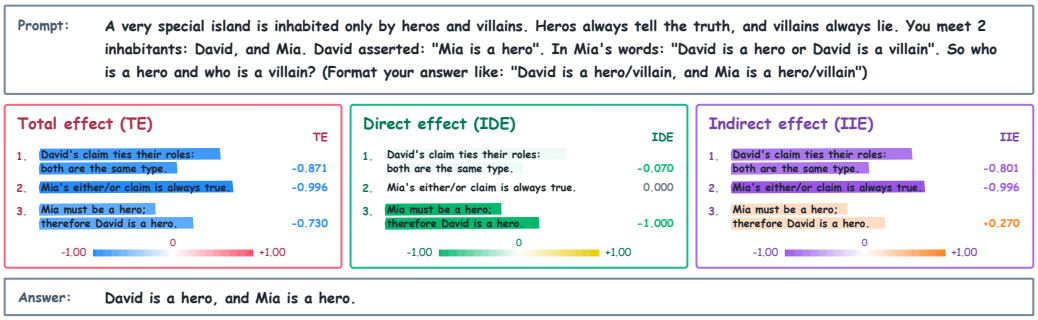

Figure 9: Step-level effects for Knights & Knaves case 17, Qwen3.5-9B.

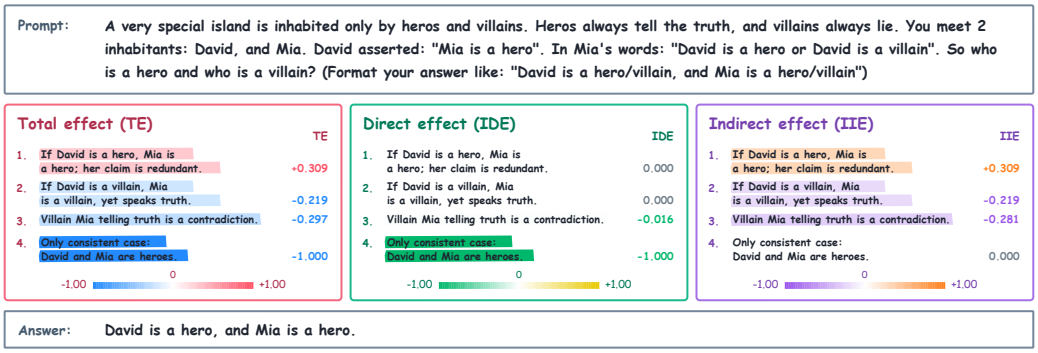

Figure 10: Step-level effects for Knights & Knaves case 17, Mistral-Small-3.2-24B.

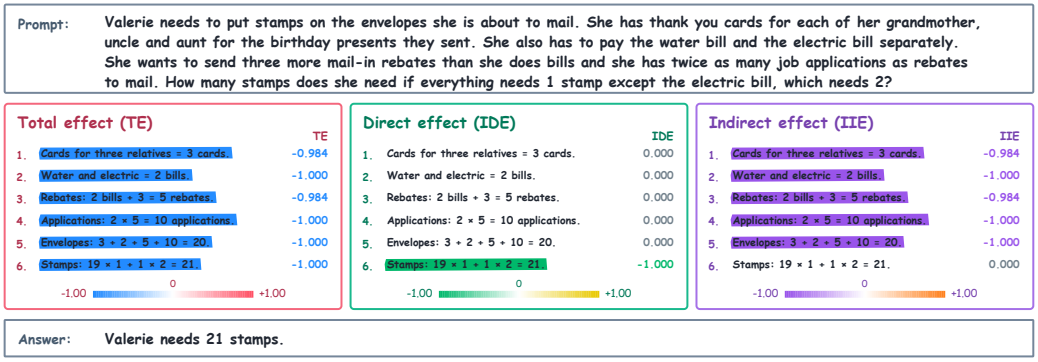

Figure 11: Step-level effects for GSM8K question 116, Qwen3.5-9B.

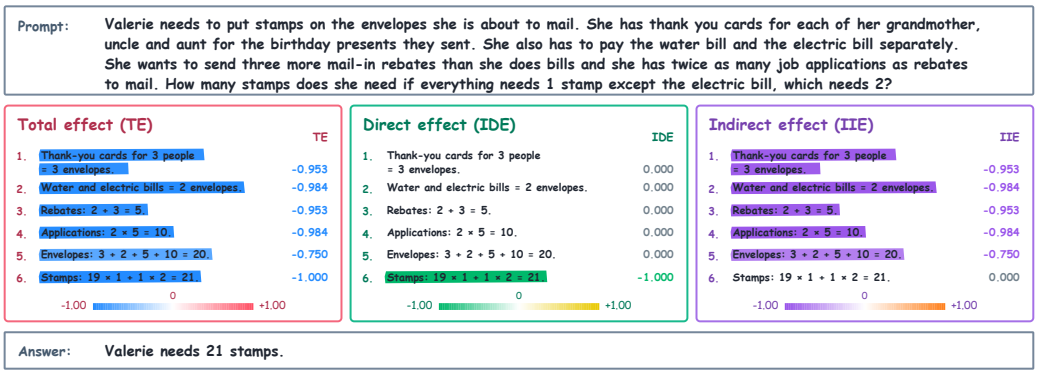

Figure 12: Step-level effects for GSM8K question 116, Mistral-Small-3.2-24B.



Figure 13: Step-level effects for MMLU-Pro question 200, Qwen3.5-9B.

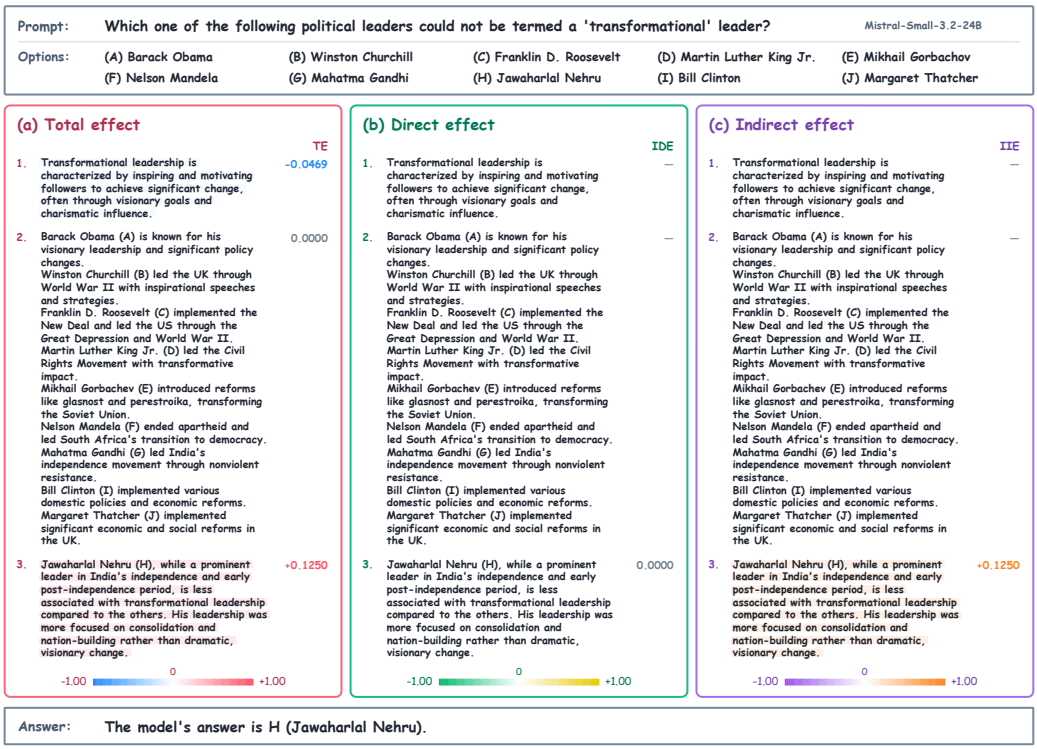

Figure 14: Step-level effects for MMLU-Pro question 200, Mistral-Small-3.2-24B.