EICR: EXECUTABLE IDENTIFIABILITY COUNTERFACTUAL REASONING FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual reasoning requires more than executing an alternative action. A model must first infer which hidden explanations remain compatible with the factual observation, and then determine whether those explanations agree after intervention. We identify a failure mode in which a model selects one factually consistent explanation, obtains a locally valid counterfactual value, and reports it as a uniquely supported answer even though another equally valid explanation gives a different result. We call this phenomenon *counterfactual identifiability collapse*. We introduce EICR, an executable completion-set reasoning framework that represents a counterfactual answer by all observation-consistent structural completions, their branch-wise rollouts, the resulting answer set, and an instance-level identification label. We construct EIBENCH as a unified benchmark with an oracle-driven construction protocol over a common symbolic task space rendered as executable code, structural causal models (SCMs/DAGs), and natural language. It distinguishes point-identified, set-identified, interval-partially-identified, and unresolved cases. Under an explicit and complete candidate mechanism family, we establish soundness and completeness properties for the oracle certificate, define a training objective for learning task-dependent certificates, specify representation-aware dataset splits, and introduce metrics for false certainty, false uncertainty, set coverage, interval coverage, bound error, and certificate validity. Experiments across five model backbones and three representations show that completion-set supervision reduces false certainty on executable tasks and improves recovery of the supported answer set, while natural-language parsing remains more challenging than structured representations. These results support preserving the evidence-compatible completion set rather than committing prematurely to a single abductive explanation.

1 INTRODUCTION

Counterfactual questions ask what would have happened under an alternative action while conditioning on what actually happened. They arise in treatment attribution, recommendation analysis, incident diagnosis, policy evaluation, and program understanding (Athey & Imbens, 2017; Richens et al., 2020). Unlike ordinary prediction, a counterfactual requires an abduction–action–prediction process: infer hidden state from the factual world, intervene on the mechanism, and predict the target in the alternative world (Pearl, 2002; Pearl et al., 2003).

The abductive step is often underdetermined. Consider a user who receives a coupon and buys an item. The observation is compatible with two explanations: the coupon caused the purchase, or the user would have purchased without it. If the coupon is removed, the two explanations can predict different outcomes. A model that chooses one explanation and returns one value may have executed that explanation correctly, but the observation does not justify presenting that value as uniquely determined. Figure 1 illustrates this distinction. Although both completions are compatible with the factual observation, a single-completion method selects one branch and reports a point answer, whereas EICR evaluates all compatible completions and returns the supported answer set.

This distinction is not merely a formatting issue. A point answer can be locally correct under one valid completion while remaining unsupported as a unique conclusion by the available evidence. We call this error *counterfactual identifiability collapse*: the model compresses multiple factually

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.

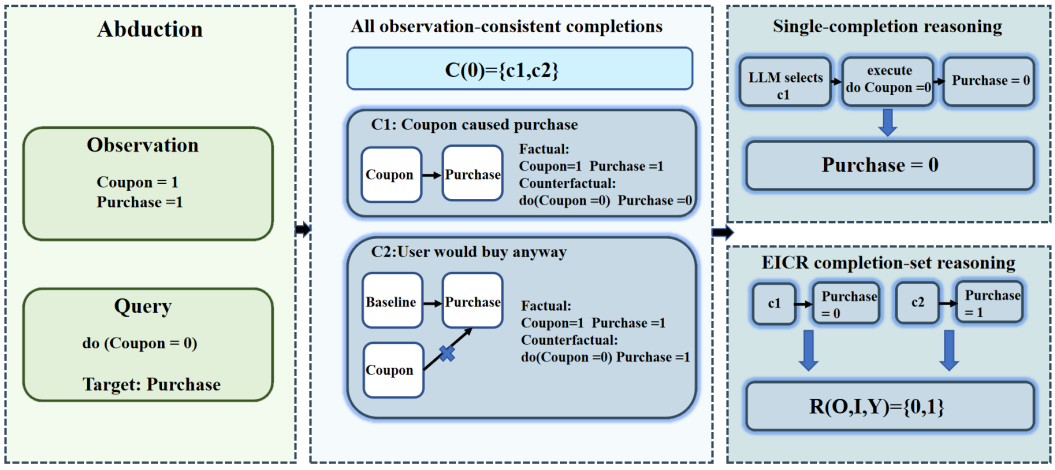


Figure 1: Motivation for completion-set counterfactual reasoning. The factual observation is compatible with two admissible structural completions: the coupon caused the purchase (c_1) or a baseline response explains the purchase without a direct coupon effect (c_2). A single-completion method selects c_1 and reports $\text{Purchase} = 0$, which is locally valid but not uniquely identified. EICR retains both observation-consistent completions, evaluates the intervention on each branch, and returns $R(O, I, Y) = \{0, 1\}$, indicating a set-identified counterfactual answer.

consistent completions into one before checking whether the query result is invariant across them. This failure is distinct from ordinary hallucination and numerical execution error because it concerns the preservation of alternative explanations relevant to the query.

We study an instance-level problem. Given a candidate mechanism family, factual observation, intervention, target, and explicit admissible completion space, we retain every completion compatible with the observation and compare their counterfactual target values. The answer is point-identified only when all compatible explanations agree; otherwise, the evidence supports a set, a proven interval, or an unresolved result rather than an arbitrary branch.

We propose EICR, an executable identifiability-aware training framework. Its central object is a structured certificate

$$\tau = (C, V, B, \rho, a, h), \tag{1}$$

where C contains candidate completions, V records factual verification, B records branch-wise counterfactual rollouts, ρ stores the applicable result payload, a is the identification label, and h records completeness status or an unresolved reason. The certificate is generated by an executable oracle and used to supervise a language model. The same symbolic task is rendered in three representations: Code, SCM/DAG, and natural language. This makes it possible to distinguish representation transfer from causal-structure transfer.

Our contributions are fourfold. (1) We formalize observation-relative counterfactual answer identification and define false certainty as a measurable failure mode. (2) We introduce completion-set reasoning and a verifiable certificate that separates hidden-state ambiguity from target ambiguity. (3) We construct EIBENCH as a shared symbolic task space with hidden-irrelevant, hidden-divergent, bounded, query-dependent, and unresolved families rendered across Code, SCM, and natural-language representations. (4) We provide an oracle-driven data-construction protocol, a unified evaluation design, and an empirical study covering multiple model backbones and representations.

Accordingly, we formulate counterfactual identification at the instance level: given a candidate mechanism family, factual observation, intervention, target, and explicit admissible completion space, EICR determines whether the target is point-identified, set-identified, interval-identified, or unresolved under the available evidence.

2 Related Work

Abduction–action–prediction. Structural causal models (SCMs) provide a language for counterfactual reasoning. Factual evidence constrains exogenous variables or latent states, an intervention replaces structural equations, and prediction evaluates the modified model (Pearl, 2002; Pearl et al., 2003). If the latent state is fully specified, the problem reduces to intervention.

Executable counterfactuals. Vashishtha et al. (Vashishtha et al., 2026) use executable programs with hidden random variables to make abduction explicit and evaluate multiple valid counterfactual answers. EICR builds on this setting by supervising the completion set, factual verification, branch-wise rollouts, and instance-level identification label, so that the model reports not only possible answers but also the evidence supporting them.

Several recent studies evaluate causal or counterfactual reasoning in language models. Corr2Cause and CLadder test formal causal inference from symbolic or graph-generated questions (Jin et al., 2024; 2023); other benchmarks focus directly on counterfactual tasks and counterfactual generation or evaluation (Wu et al., 2024; Youssef et al., 2024; Chen et al., 2026). EICR instead makes a completion-grounded certificate the primary supervised object.

Counterfactual language has also been used for story reasoning, controllable generation, and retrieval-guided question answering (Qin et al., 2019; Wu et al., 2021; Paranjape et al., 2022). These settings generally target generation or robustness rather than enumerating all observation-consistent structural completions and certifying whether the target is identified. Related training work uses axiomatic supervision to teach causal behavior (Vashishtha et al., 2024); EICR instead makes the evidence-compatible completion set the supervised object.

In language processing, counterfactual augmentation and controlled generation have been used to test robustness, reduce spurious correlations, and construct alternative questions or narratives (Epstude & Roese, 2008; Gerstenberg, 2022; Kaushik et al., 2019; Joshi & He, 2022; Madaan et al., 2021; Li et al., 2024; Huang et al., 2024; Mishra et al., 2024; Yu et al., 2023; Ye et al., 2021).

Counterfactual prediction has also been studied in sequential and policy settings, where the goal is to forecast an alternative trajectory or outcome under a changed history (Poulos & Zeng, 2021). EICR addresses a different source of uncertainty: multiple latent completions can agree with the same factual record while disagreeing about the queried intervention outcome.

Recent causal benchmarks broaden evaluation from individual questions to graph structure and decomposed reasoning. LLM-assisted causal-discovery systems use language models as imperfect structural experts (Ban et al., 2025), while CausalBN-Bench evaluates causal learning capabilities over correlation, skeleton, and causal-identification tasks (Zhou et al., 2026). Decompositional studies analyze whether a model's counterfactual errors arise from parsing, abduction, intervention, or answer selection (Yang et al., 2026; Yamin et al., 2025).

Comparisons of supervised fine-tuning and reinforcement learning report different memorization and transfer behavior (Chu et al., 2025; Wu et al., 2026); studies of RLHF further examine generalization and diversity under preference-based training (Kirk et al., 2024). Large-scale reasoning systems such as DeepSeekMath and DeepSeek-R1 use verifiable or outcome-based signals to improve multi-step reasoning (Shao et al., 2024; Guo et al., 2025). Causal-order supervision and counterfactual feedback provide more task-specific alternatives (Vashishtha et al., 2025; Hüyük et al., 2025). Our training target is complementary: it supplies a verifiable, instance-level certificate whose validity can be checked independently of the model's prose.

Causal identification and partial identification. Classical identifiability asks whether a causal quantity is uniquely determined by an observational distribution. Partial-identification methods derive sharp bounds when point identification is impossible, including bounds for treatment effects under imperfect compliance (Balke & Pearl, 1997). We use these ideas at the instance level and call our target *observation-relative counterfactual answer identification*.

Language-model reasoning and uncertainty. Chain-of-thought prompting and self-consistency can improve computation by eliciting or aggregating multiple reasoning paths (Wei et al., 2022; Wang et al., 2022). However, sampling several answers does not guarantee that all factually con-

sistent completions have been represented. Set-valued prediction and calibration methods measure coverage and width (Romano et al., 2020) but do not necessarily provide a mechanism-grounded certificate. EICR combines executable supervision with structured output so that a prediction can be checked branch by branch.

3 Problem Definition

3.1 Instance and Completion Set

An instance is $q = (\mathcal{M}, O, I, Y)$, where $\mathcal{M}$ is a task-level candidate mechanism family, O is the factual observation, I is an intervention, and Y is the target. A completion $c \in \mathcal{C}(\mathcal{M})$ selects one admissible structural explanation in $\mathcal{M}$. It may specify hidden variables, structural parameters, or, when allowed by the task, causal relations. The instantiated mechanism is denoted by M_c. Write

$$s_f(c) = \mathrm{Exec}(M_c, \mathrm{factual}), \qquad s_{cf}(c) = \mathrm{Exec}(M_c, \mathrm{do}(I)). \tag{2}$$

The predicate $\mathrm{Agree}(s_f(c), O)$ checks agreement on observed fields. The task supplies an admissible completion space $\mathcal{C}(\mathcal{M})$; it is finite for enumerative tasks and is accompanied by explicit constraints for bounded tasks. Different completions may differ either in hidden values or structural parameters under a shared graph, or in the causal relations of the candidate mechanism when structural variation is allowed by the task. The factually consistent completion set and its counterfactual image are

$$\begin{aligned} C(O) &= \{c \in \mathcal{C}(\mathcal{M}) : \mathrm{Agree}(\mathrm{Exec}(M_c, \mathrm{factual}), O)\}, \\ R(O, I, Y) &= \{\mathrm{Exec}(M_c, \mathrm{do}(I))[Y] : c \in C(O)\}. \end{aligned} \tag{3}$$

The target, rather than the completion itself, is identified when all valid completions agree on its counterfactual value. The figures use structural differences to make this ambiguity visually explicit, while the formalism also covers completions that differ only in hidden values or parameters.

3.2 Identification Labels and Certificates

We use four mutually exclusive labels:

- **Point-identified**: $R(O, I, Y) = \{y\}$.
- **Set-identified**: R is completely enumerable and $|R| > 1$.
- **Interval-partially-identified**: the complete result set is not enumerated, but the mechanism proves sharp bounds $[\ell, u]$.
- **Unresolved**: the candidate space is incomplete, no factual completion exists, or neither a complete result set nor sharp bounds can be certified.

Multiple valid hidden states do not by themselves imply a set-identified target: if they all yield the same target value, the answer is point-identified.

A certificate is $\tau = (C, V, B, \rho, a, h)$ with a task-dependent payload. Here C lists candidate completions when the task is enumerable, V records factual states and consistency decisions, B records branch-wise counterfactual executions when branches are explicitly evaluated, ρ is the applicable result payload when available, namely a deduplicated result set or a certified interval, a is the identification label, and h records completeness status or an unresolved reason. A certificate is valid only when its task-specific soundness and completeness conditions hold: retained completions must satisfy the factual observation, branch values must follow from the instantiated mechanisms, result sets or interval bounds must match the oracle, and unresolved reasons must be supported by the task specification.

3.3 Counterfactual Identifiability Collapse

A single-completion method chooses $\hat{c} = A_\theta(\mathcal{M}, O)$ and reports

$$\hat{y} = \mathrm{Exec}(M_{\hat{c}}, \mathrm{do}(I))[Y]. \tag{4}$$

When $\hat{c} \in C(O)$, this value can be correct under one valid explanation. We call the output a *counterfactual identifiability collapse* if

$$|R(O, I, Y)| > 1, \qquad \hat{y} \in R(O, I, Y), \tag{5}$$

but the method presents $\hat{y}$ as uniquely identified. The error is not necessarily a hallucination or an execution error: it is a locally valid branch value incorrectly promoted to a uniquely supported conclusion.

Proposition 1 (single-completion incompleteness). If two completions in $C(O)$ yield different counterfactual target values, a method that retains only one completion cannot in general guarantee the correct identification label.

Proof sketch. The observed input is compatible with both completions, but a one-completion representation discards their disagreement. It therefore lacks the information required to certify that the target is not point-identified. □

Proposition 2 (oracle soundness and completeness). If $\mathcal{C}(\mathcal{M})$ is complete, Agree is correct, and execution is deterministic, enumerating $C(O)$ and evaluating every valid branch returns exactly the counterfactual values supported by the task evidence.

Proof sketch. Every retained completion agrees with the observation and every returned value is produced by the specified intervention, giving soundness. Since every candidate is tested, every factually consistent completion is retained, giving completeness. □

Proposition 3 (query-equivalent compression). For an enumerable task, replacing valid completions that yield the same target value for the query by one representative does not change $R(O, I, Y)$. Query-equivalent compression is used only for internal result computation. For enumerable tasks, a model-generated certificate is still required to enumerate the complete gold observation-consistent completion set; compression must not be used to shorten the certificate itself.

For the explicit executable task, naive enumeration costs $O(|\mathcal{C}(\mathcal{M})|T_{\mathrm{exec}})$; factual pruning, query-equivalent compression, and symbolic bounds can reduce the cost.

4 EICR Framework and EIBench

4.1 Oracle and Certificate Construction

The oracle (i) enumerates or symbolically represents candidate completions in $\mathcal{C}(\mathcal{M})$, (ii) executes each instantiated mechanism in the factual world, (iii) retains completions consistent with the observation, (iv) executes the intervention under every retained completion when the task is enumerable, or computes certified bounds when the task is bounded, and (v) aggregates the target values and assigns an identification label. This order makes the certificate a consequence of the task specification rather than a language-model judgment.

Figure 2 summarizes the overall workflow. Given the factual observation O and an intervention query I for target Y, EICR first obtains the observation-consistent completion set $C(O)$. Each completion c_i instantiates one candidate structural explanation of the factual observation. The intervention is then executed separately under every retained completion, producing branch-wise target values y_i. These values are aggregated into $R(O, I, Y)$, with repeated values deduplicated because the answer is a set. The resulting set is used together with the completeness status of the candidate space to determine the instance-level identification label. In Figure 2, C, P, and Y are schematic node labels used to illustrate candidate mechanisms; the concrete coupon–purchase interpretation is given in Figure 1. The different small graphs are representative candidate mechanisms; they do not imply that all benchmark instances share the same graph structure.

4.2 Shared Task Space and Representations

EIBench is defined over five task families: (1) hidden-irrelevant tasks have multiple valid completions but an invariant target; (2) hidden-divergent tasks have valid completions that yield different target values; (3) bounded tasks admit sharp intervals over continuous or large latent spaces; (4) query-dependent tasks change identification under a different intervention or target; and (5) un-

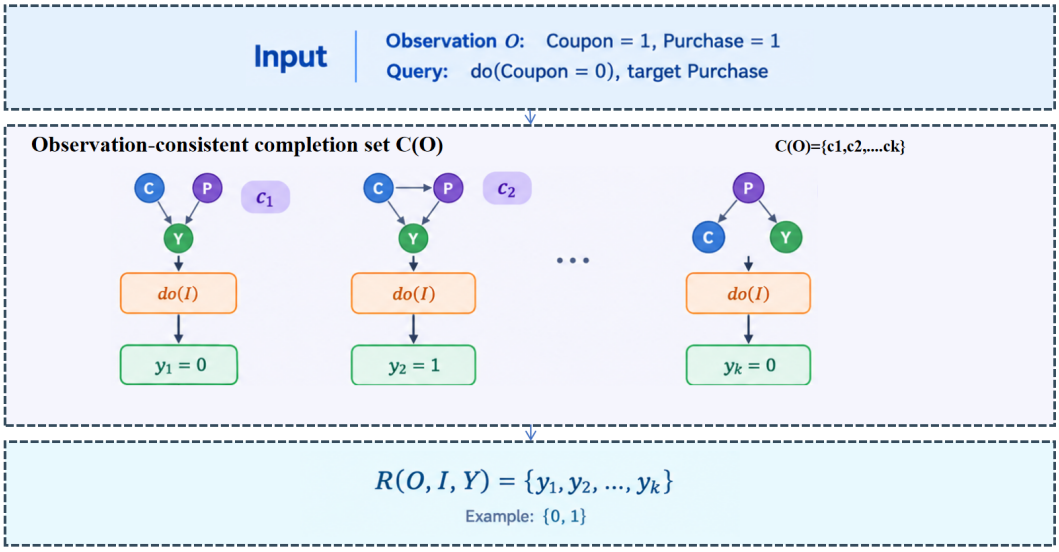


Figure 2: Overview of EICR completion-set reasoning. Given a factual observation O and an intervention query I targeting Y, EICR retains all observation-consistent structural completions $C(O) = \{c_1, c_2, \ldots, c_k\}$ from the candidate mechanism family, executes the intervention under each completion, and aggregates the branch-wise target values into the result set $R(O, I, Y) = \{y_1, y_2, \ldots, y_k\}$. The displayed graphs are representative admissible candidate mechanisms rather than a single fixed DAG shared by all branches. Factual verification, identification-label assignment, and certificate serialization are described in the surrounding text.

resolved tasks have an incomplete candidate mechanism family or insufficient information for a complete set or sharp bounds.

Each symbolic task is rendered as executable Code, an SCM/DAG with structural equations, and natural language. The underlying candidate mechanism family, observation, intervention, target, admissible completion space, and oracle certificate are held fixed across renderings. In Code, hidden variables occur as function arguments or branch values; in an SCM/DAG, the task specifies nodes, edges, and equations; in natural language, the same task-level mechanism family is stated without code or a graph. This paired design separates representation transfer from changes in task semantics and follows the broader view that stable mechanisms should be distinguished from surface variables (Schölkopf et al., 2021).

For example, consider a binary purchase task with an observed coupon variable X, a purchase outcome Y, and two admissible structural completions. Under c_1, the coupon directly causes the purchase, so the factual observation $X = 1, Y = 1$ is explained by the coupon effect and the intervention $X \leftarrow 0$ yields $Y = 0$. Under c_2, the purchase is explained by a baseline response while the coupon has no direct effect; the same factual observation is therefore also compatible with c_2, but $X \leftarrow 0$ yields $Y = 1$. Hence, the observation supports the result set $\{0, 1\}$ and the target is set-identified. The two completions are different admissible structural explanations of the same task-level candidate mechanism family, matching the structural contrast illustrated in Figure 1.

4.3 Data Generation and Quality Control

The generator samples a task-level candidate mechanism family, an admissible completion space, a factual observation, an intervention, and a target. It then computes the oracle certificate before producing any surface rendering. Every rendered item is checked against the same canonical record, and test labels are generated independently of model outputs. For bounded tasks, a symbolic interval procedure computes endpoint extrema; finite sampling is not treated as proof of sharp bounds. For unresolved tasks, the record stores the reason that completeness cannot be certified. This procedural approach provides controlled task variation and scalable evaluation (Cobbe et al., 2020).

Table 1: Full-scale EIBENCH configuration. The three datasets share the same task-family composition and evaluation protocol.

Dataset	Irrelevant	Divergent	Bounded	Query dep.	Unresolved
Code-EIBench	2,000	2,000	1,500	1,500	1,000
SCM-EIBench	2,000	2,000	1,500	1,500	1,000
NL-EIBench	2,000	2,000	1,500	1,500	1,000

4.4 TRAINING OBJECTIVE

For an input $x = (\mathcal{M}, O, I, Y)$ and oracle certificate τ^*, supervised fine-tuning minimizes

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\sum_{(x,\tau^*)\in D} \log p_\theta(\tau^* \mid x). \tag{6}$$

Certificates are serialized with a task-dependent constrained schema. Enumerative tasks use valid completions, factual verification records, branch-wise outcomes, a deduplicated result set, and an identification label. Bounded tasks use certified interval endpoints in place of a fully enumerated result set, while unresolved tasks include an incompleteness status and reason. Natural-language explanations are optional; the structured fields are evaluated. A teacher may verbalize an oracle trace but does not determine the completion set, bounds, or label.

5 EXPERIMENTS

5.1 SETUP, BASELINES, AND METRICS

Dataset configuration. EIBENCH contains three parallel datasets: Code-EIBench, SCM-EIBench, and NL-EIBench. They share symbolic tasks and oracle certificates but use different surface representations: executable mechanisms, structural equations with DAGs, and natural language, respectively.

Each dataset contains the same five task families: hidden-irrelevant, hidden-divergent, bounded, query-dependent, and unresolved. We first split canonical task IDs, together with their generator templates, into train, validation, and test partitions. Each partition is then rendered independently as Code, SCM/DAG, and natural language, with task families represented in every split. Thus, the same canonical task may appear across representations within one partition, but no canonical task or complete mechanism template crosses partitions.

Each dataset contains 8,000 examples. The three datasets use matched task counts so that representation-level comparisons are not confounded by dataset size.

The evaluation includes a collapse test, a unified multi-backbone comparison, and a component ablation. We use Qwen2.5-2.5B, Qwen2.5-7B, Qwen2.5-14B, DeepSeek-14B, and Llama3-8B. Methods within each backbone use matched training and decoding settings, and results are averaged over three seeds.

The comparison includes CoT (Wei et al., 2022), CoT-SC (Wang et al., 2022), an executable-counterfactual baseline aligned with Vashishtha et al. (2026), Answer-Set SFT, and EICR-SFT. Answer-Set SFT supervises the final result set and identification label without supervising completion provenance, while EICR-SFT supervises the complete certificate. The oracle set solver is used as an execution reference. This design separates the effect of answer-set supervision from the additional effect of preserving completion provenance and branch-level evidence.

Point accuracy is evaluated on point-identified items, and identification accuracy scores the four labels. False Certainty Rate (FCR) measures non-point items predicted as point-identified, while False Uncertainty Rate (FUR) measures point items predicted as non-point. For enumerable discrete tasks with a gold set R and predicted set $\hat{R}$, we report

$$\mathrm{Coverage} = \frac{|\hat{R} \cap R|}{|R|}, \qquad \mathrm{Tightness} = \frac{|\hat{R} \cap R|}{|\hat{R}|}. \tag{7}$$

Table 2: False Certainty Rate (FCR, %) on hidden-divergent subsets from the three EIBench test datasets using EICR-SFT. The comparison varies only the model backbone. Lower is better.

Backbone	Code-EIBench	SCM-EIBench	NL-EIBench
Qwen2.5-2.5B	13.5	15.6	18.2
Qwen2.5-7B	11.8	13.2	16.0
Qwen2.5-14B	8.9	10.4	12.7
DeepSeek-14B	7.8	9.0	11.0
Llama3-8B	12.4	14.0	16.5

Coverage and tightness are computed only when the complete result set is enumerable. For a gold interval $[\ell, u]$ and predicted interval $[\hat{\ell}, \hat{u}]$, we use

$$\text{IntervalCoverage} = \mathbb{1}[\hat{\ell} \le \ell \ \wedge \ u \le \hat{u}], \qquad \text{BoundError} = \frac{|\hat{\ell} - \ell| + |\hat{u} - u|}{2}. \tag{8}$$

Bounded tasks are evaluated with these interval metrics, while unresolved tasks are evaluated with identification-label accuracy and incompleteness-reason validity. We additionally report exact set match and certificate validity. The main table reports PointAcc and FCR on their respective applicable subsets of the full test sets: PointAcc is computed on point-identified items, and FCR is computed on all non-point items, including hidden-divergent, bounded, query-dependent, and unresolved cases. SetCov. and Tightness are reported on the enumerable-result subset. All metrics are computed per example and macro-averaged within each representation.

5.2 Experiment 1: Simple Test of Single-Point Collapse

To test single-point collapse directly, we construct a hidden-divergent subset from each of the three test datasets. Every item contains at least two factually consistent completions with different intervention outcomes. We evaluate the full EICR-SFT configuration across the five model backbones, keeping the training target and evaluation protocol fixed so that only the backbone changes. The reported metric is FCR, expressed as a percentage; lower values indicate fewer non-point cases incorrectly reported as point-identified. This construction isolates whether the backbone can preserve disagreement between valid explanations rather than selecting a single locally consistent branch.

The hidden-divergent test shows a consistent representation effect: FCR is higher on NL-EIBench than on Code-EIBench and SCM-EIBench for every backbone. Larger backbones generally reduce FCR within the Qwen2.5 family, while the DeepSeek-14B configuration obtains the lowest values in this comparison. This experiment isolates backbone sensitivity within the same completion-set supervision scheme rather than comparing different supervision targets. The result pattern suggests that structured representations make completion-level evidence easier to retain during counterfactual evaluation.

5.3 Experiment 2: Unified Main Comparison

Main multi-backbone comparison. The main experiment compares all methods within each backbone block on the full Code-EIBench, SCM-EIBench, and NL-EIBench test sets. The grouped-column format follows the original paper design: the three datasets are parallel representation-level benchmarks, while the backbone blocks expose how the comparison changes with model capacity. We report PointAcc on point-identified items and FCR on all non-point items from the full test sets, together with SetCov. and Tightness on the enumerable-result subset. Identification accuracy, FUR, set exact match, interval coverage, bound error, incompleteness-reason validity, and certificate validity are reported as secondary analyses because they diagnose different failure modes and would make the multi-backbone table unnecessarily wide.

The main results compare all listed methods under matched budgets and decoding settings. The executable-counterfactual baseline follows the task setting of Vashishtha et al. (Vashishtha et al., 2026). We do not transfer numerical results from that work. The grouped layout makes it possible to inspect both the effect of supervision and the effect of model capacity.

Table 3: Main multi-backbone comparison on the full Code-EIBench, SCM-EIBench, and NL-EIBench test sets. PointAcc and FCR use their respective applicable test subsets; SetCov. and Tightness are computed on enumerable-result items. Higher is better for PointAcc, SetCov., and Tightness; lower is better for FCR.

	Code-EIBench				SCM-EIBench				NL-EIBench			
Method	PointAcc	FCR	SetCov.	Tight.	PointAcc	FCR	SetCov.	Tight.	PointAcc	FCR	SetCov.	Tight.
Qwen2.5-2.5B												
CoT	56.9	44.6	44.9	46.1	53.8	47.2	42.4	44.0	50.3	50.1	39.2	41.4
CoT-SC	60.1	40.8	49.0	50.1	56.1	43.2	46.0	47.4	53.2	46.4	42.7	44.7
Executable CF	68.0	28.9	61.0	62.7	63.7	32.2	58.0	59.7	60.5	35.8	54.4	56.1
Answer-Set SFT	74.8	23.5	67.4	72.6	71.6	26.8	63.5	69.6	67.8	32.7	56.9	64.8
EICR-SFT	**80.2**	**16.8**	**76.1**	**78.4**	**77.1**	**18.9**	**72.5**	**74.0**	**74.0**	**22.4**	**68.7**	**70.5**
Qwen2.5-7B												
CoT	62.7	39.6	50.4	51.7	60.4	42.3	47.3	48.8	56.8	45.5	43.7	45.9
CoT-SC	65.4	36.6	54.0	55.3	63.0	39.0	50.6	52.0	60.1	42.0	47.0	48.8
Executable CF	72.8	26.4	64.8	66.2	70.7	29.2	61.7	63.0	67.6	32.0	57.4	59.0
Answer-Set SFT	79.8	19.8	71.6	76.8	76.7	22.6	67.6	73.7	72.9	28.7	61.8	68.9
EICR-SFT	**83.4**	**14.6**	**79.1**	**81.0**	**81.8**	**16.3**	**75.4**	**77.0**	**78.4**	**20.1**	**70.9**	**72.4**
Qwen2.5-14B												
CoT	68.6	34.8	56.2	57.8	66.7	36.9	54.4	55.8	63.5	39.9	50.8	52.4
CoT-SC	70.6	31.8	59.7	61.1	68.9	33.5	57.8	59.2	65.9	36.6	54.0	55.5
Executable CF	77.6	21.8	69.8	71.4	75.5	24.3	67.2	68.8	72.0	27.0	63.4	65.0
Answer-Set SFT	83.9	15.9	76.9	81.9	80.8	18.6	72.8	78.7	77.9	24.6	66.8	72.8
EICR-SFT	**86.1**	**11.7**	**82.8**	**84.5**	**84.5**	**13.9**	**79.8**	**81.2**	**81.1**	**17.6**	**75.5**	**77.1**
DeepSeek-14B												
CoT	70.3	32.1	57.9	59.4	68.6	33.8	56.5	57.9	65.7	37.0	53.1	54.7
CoT-SC	72.4	29.5	61.4	62.9	70.7	31.5	59.8	61.0	68.0	34.0	56.4	58.0
Executable CF	79.2	19.7	71.6	73.1	77.5	22.1	69.0	70.4	74.1	24.7	65.4	67.0
Answer-Set SFT	84.9	13.7	79.8	84.7	82.8	16.6	75.8	81.7	79.8	21.6	69.9	75.8
EICR-SFT	**87.2**	**10.5**	**84.6**	**86.1**	**85.6**	**12.6**	**81.6**	**83.2**	**82.0**	**15.8**	**77.0**	**78.5**
Llama3-8B												
CoT	61.4	41.0	49.0	50.5	58.9	43.9	46.5	48.1	55.6	47.0	42.8	44.6
CoT-SC	64.0	37.9	52.5	53.8	61.4	40.6	49.8	51.3	58.6	43.8	46.1	47.8
Executable CF	71.0	27.4	62.4	63.9	68.4	30.0	59.8	61.3	65.2	32.7	55.7	57.0
Answer-Set SFT	77.8	21.7	69.8	74.7	74.8	24.6	64.8	71.6	70.7	30.6	58.8	67.6
EICR-SFT	**81.5**	**15.2**	**77.8**	**79.6**	**79.2**	**17.8**	**74.4**	**75.8**	**75.8**	**21.4**	**69.1**	**70.8**
Oracle Set Solver	100.0	0.0	100.0	100.0	100.0	0.0	100.0	100.0	100.0	0.0	100.0	100.0

Across the main comparison, EICR-SFT improves both point accuracy and set recovery relative to the answer-only and executable baselines across all three representations. The advantage is accompanied by lower FCR, indicating that the improvement is not obtained by simply predicting broader answer sets. The remaining gap on NL-EIBench is consistent with additional errors in parsing the mechanism and identifying the relevant completion constraints.

5.4 Experiment 3: Component Ablation

We conduct the ablation on Qwen2.5-14B with a fixed training budget, removing one EICR component at a time. Complete results are reported in Appendix H.

6 Conclusion

We introduced EICR, which represents counterfactual answers through all factually consistent completions and their intervention outcomes, and constructed EIBench to evaluate this approach across Code, SCM/DAG, and natural-language representations. Experiments across five model backbones show that completion-set supervision reduces false certainty and improves recovery of supported answer sets, although natural-language tasks remain more challenging. EICR therefore reports a point, a set, an interval, or an unresolved result according to the evidence available for each instance.

AI USE STATEMENT

Generative AI tools were used to assist with drafting, language polishing, and stylistic refinement of this manuscript. They were not used to replace the authors' responsibility for the problem formulation, technical claims, experimental design, result interpretation, or final verification. All AI-assisted text was reviewed and revised by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses synthetic tasks and does not involve human participants, private data, or personally identifiable information. The framework is intended to improve uncertainty communication in counterfactual reasoning, but performance on synthetic tasks should not be interpreted as evidence for deployment in high-stakes domains. Any practical deployment should use expert-defined mechanisms, independent validation, and human review of non-point conclusions.

REPRODUCIBILITY STATEMENT

The paper specifies the task definition and oracle assumptions in Sections 3 and 4, the three parallel datasets and evaluation metrics in Section 5.1, and the data-generation, output-verification, representation-consistent evaluation, scoring, additional experiments, and theoretical properties in Appendices A–I. These sections provide the schemas, task families, certificate fields, comparison settings, evaluation criteria, and analysis protocols needed to reproduce the benchmark and reported analyses. The accompanying implementation should follow these specifications when regenerating the data and evaluation outputs.

REFERENCES

Susan Athey and Guido W Imbens. The state of applied econometrics: Causality and policy evaluation. *Journal of Economic perspectives*, 31(2):3–32, 2017.

Alexander Balke and Judea Pearl. Bounds on treatment effects from studies with imperfect compliance. *Journal of the American statistical Association*, 92(439):1171–1176, 1997.

Taiyu Ban, Lyuzhou Chen, Derui Lyu, Xiangyu Wang, Qinrui Zhu, Qiang Tu, and Huanhuan Chen. Integrating large language model for improved causal discovery. *IEEE Transactions on Artificial Intelligence*, 6(11):3030–3042, 2025.

Yuefei Chen, Vivek K Singh, Jing Ma, and Ruixiang Tang. Counterbench: Evaluating and improving counterfactual reasoning in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30350–30358, 2026.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pp. 2048–2056. PMLR, 2020.

Kai Epstude and Neal J Roese. The functional theory of counterfactual thinking. *Personality and social psychology review*, 12(2):168–192, 2008.

Tobias Gerstenberg. What would have happened? counterfactuals, hypotheticals and causal judgements. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 377(1866): 20210339, 2022.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie

Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, Sept 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Yinya Huang, Ruixin Hong, Hongming Zhang, Wei Shao, Zhicheng Yang, Dong Yu, Changshui Zhang, Xiaodan Liang, and Linqi Song. Clomo: Counterfactual logical modification with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11012–11034, 2024.

Alihan Hüyük, Xinnuo Xu, Jacqueline Maasch, Aditya Nori, and Javier Hernandez. Reasoning elicitation in language models via counterfactual feedback. In *International Conference on Learning Representations*, volume 2025, pp. 76804–76832, 2025.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. Cladder: Assessing causal reasoning in language models. *Advances in Neural Information Processing Systems*, 36: 31038–31065, 2023.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, David Ha, et al. Can large language models infer causation from correlation? In *International Conference on Learning Representations*, volume 2024, pp. 28663–28679, 2024.

Nitish Joshi and He He. An investigation of the (in) effectiveness of counterfactually augmented data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3668–3681, 2022.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. In *International Conference on Learning Representations*, volume 2024, pp. 20620–20653, 2024.

Yongqi Li, Mayi Xu, Xin Miao, Shen Zhou, and Tieyun Qian. Prompting large language models for counterfactual generation: An empirical study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 13201–13221, 2024.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 13516–13524, 2021.

Ashish Mishra, Gyanaranjan Nayak, Suparna Bhattacharya, Tarun Kumar, Arpit Shah, and Martin Foltin. Llm-guided counterfactual data generation for fairer ai. In *Companion Proceedings of the ACM Web Conference 2024*, pp. 1538–1545, 2024.

Bhargavi Paranjape, Matthew Lamm, and Ian Tenney. Retrieval-guided counterfactual generation for qa. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1670–1686, 2022.

Judea Pearl. Reasoning with cause and effect. *AI Magazine*, 23(1):95–95, 2002.

Judea Pearl et al. Causality: models, reasoning, and inference. *Econometric Theory*, 19(675-685): 46, 2003.

Jason Poulos and Shuxi Zeng. Rnn-based counterfactual prediction, with an application to homestead policy and public schooling. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 70(4):1124–1139, 2021.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. Counterfactual story reasoning and generation. 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)*, pp. 5043–5053, 2019.

Jonathan G Richens, Ciarán M Lee, and Saurabh Johri. Improving the accuracy of medical diagnosis with causal machine learning. *Nature communications*, 11(1):3923, 2020.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in neural information processing systems*, 33:3581–3591, 2020.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Aniket Vashishtha, Abhinav Kumar, Atharva Pandey, Abbavaram Gowtham Reddy, Kabir Ahuja, Vineeth N Balasubramanian, and Amit Sharma. Teaching transformers causal reasoning through axiomatic training. *arXiv preprint arXiv:2407.07612*, 2024.

Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth Balasubramanian, and Amit Sharma. Causal order: The key to leveraging imperfect experts in causal inference. In *International Conference on Learning Representations*, volume 2025, pp. 95330–95371, 2025.

Aniket Vashishtha, Qirun Dai, Hongyuan Mei, Amit Sharma, Chenhao Tan, and Hao Peng. Executable counterfactuals: Improving llms' causal reasoning through code. In *International Conference on Learning Representations*, volume 2026, pp. 101098–101123, 2026.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6707–6723, 2021.

Yongliang Wu, Yizhou Zhou, Ziheng Zhou, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, et al. On the generalization of sft: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations*, volume 2026, pp. 27550–27571, 2026.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 1819–1862, 2024.

Khurram Yamin, Gaurav Ghosal, and Bryan Wilder. Can llms reconcile knowledge conflicts in counterfactual reasoning. *arXiv preprint arXiv:2506.15732*, 2025.

Shuai Yang, Qi Yang, Luoxi Tang, Yuqiao Meng, Nancy Guo, Jeremy Blackburn, and Zhaohan Xi. On the eligibility of llms for counterfactual reasoning: a decompositional study. In *International Conference on Learning Representations*, volume 2026, pp. 51251–51283, 2026.

Xi Ye, Rohan Nair, and Greg Durrett. Connecting attributions and qa model behavior on realistic counterfactuals. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5496–5512, 2021.

Paul Youssef, Christin Seifert, Jörg Schlötterer, et al. Llms for generating and evaluating counterfactuals: A comprehensive study. In *Findings of the association for computational linguistics: EMNLP 2024*, pp. 14809–14824, 2024.

Wenhao Yu, Meng Jiang, Peter Clark, and Ashish Sabharwal. Ifqa: A dataset for open-domain question answering under counterfactual presuppositions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8276–8288, 2023.

Yu Zhou, Xingyu Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. Causalbn-bench: A comprehensive benchmark for causal learning capability of llms. *IEEE Transactions on Artificial Intelligence*, 2026.

A Data Generation Details

A.1 Canonical Record

Each generated record contains a task identifier, a surface representation label, a candidate mechanism family, the factual observation, the intervention, the target variable, and the declared admissible completion space. The record also stores the task-dependent oracle certificate, including observation-consistent completions and branch values for enumerable tasks, certified interval endpoints for bounded tasks, or an incompleteness reason for unresolved tasks.

The record is generated before any language-model response. The same record can be serialized as Code, SCM/DAG, or natural language. This paired representation is important for measuring whether a model transfers the certificate structure rather than memorizing a surface template.

A.2 Task Generator

The generator samples a task family, a candidate mechanism family, admissible latent or structural support, an observation, an intervention, and a target. For enumerable tasks, it executes every candidate completion on the factual input, filters by agreement, and then executes every surviving completion under intervention. For bounded tasks, a symbolic interval solver computes endpoint extrema under the declared constraints. For unresolved tasks, the generator stores an explicit reason, such as unbounded support or incomplete candidate topology.

B Output Schema and Verification

The recommended model output is a task-dependent structured certificate. Enumerative tasks record the retained observation-consistent completions, factual verification decisions, branch-wise target values, the deduplicated result set, and the identification label. Bounded tasks record the certified lower and upper endpoints together with the constraints used to establish them. Unresolved tasks record the incompleteness status, the identification label, and an explicit reason why a complete result set or sharp bounds cannot be certified.

Evaluation parses the applicable fields and re-executes every claimed branch when branches are enumerable. For finite candidate spaces, certificate evaluation separately checks soundness and completeness. A certificate is sound when every claimed completion is valid and every claimed branch is correctly executed. It is complete only when no valid observation-consistent completion is omitted. Omitting a valid completion makes the certificate incomplete, even if all retained branches are locally correct. For bounded tasks, validity requires matching the certified interval endpoints; for unresolved tasks, validity requires a correct unresolved label and a supported incompleteness reason.

C Additional Evaluation Protocol

C.1 Training-Controlled Comparisons

The main comparison uses the same prompts, sample counts, context budget, and optimizer settings within each backbone. CoT and CoT-SC receive answer-oriented prompts; the executable-counterfactual baseline receives executable counterfactual tasks; Answer-Set SFT receives the task-specific answer payload and identification label but not completion provenance; and full EICR receives the complete task-dependent certificate. This controls the comparison across generic reasoning, executable counterfactual training, answer-set supervision, and evidence-preserving certificate supervision.

C.2 Representation-Consistent Evaluation

The three datasets use the same canonical task records and oracle labels within corresponding partitions. Canonical task IDs and generator-template groups are split before rendering, so no canonical task or complete mechanism template appears across train, validation, and test partitions. Each split is then rendered independently as Code, SCM/DAG, and natural language. The test partitions are balanced across the task families and are evaluated independently for Code-EIBench, SCM-EIBench, and NL-EIBench. This design measures how the same identification problem is expressed and solved in the three representations without introducing additional benchmark categories.

D Additional Scoring Details

D.1 Certificate Matching

Evaluation first parses each model output into the task-dependent certificate schema. Completion and branch records are matched by their canonical structural values rather than by their textual order. Result sets are deduplicated before comparison, so repeated predictions do not improve coverage. For discrete targets, branch values are compared by exact equality. For finite candidate spaces, the predicted valid-completion set is compared with the complete gold completion set in addition to checking each local branch. For bounded tasks, predicted endpoints are compared with the oracle interval; for unresolved tasks, the reason is checked against the recorded incompleteness condition. The identification label is scored separately from the result payload, which allows value errors and diagnosis errors to be analyzed independently.

D.2 Invalid and Incomplete Outputs

Outputs that cannot be parsed, omit required fields, contain inconsistent completion and branch records, or omit a valid observation-consistent completion in an enumerative task are marked as

invalid or incomplete certificates. Invalid or incomplete certificates receive zero certificate validity and do not receive credit for an exact certificate match. The final-answer metrics are computed from the fields that can be parsed without manually repairing the model output. This keeps the scoring procedure deterministic across methods and representations.

D.3 Aggregation Protocol

Metrics are computed per example and then macro-averaged within each dataset. PointAcc and FUR use point-identified items; FCR uses all non-point items in the full test set for the main comparison and the hidden-divergent subset for the collapse experiment. Set coverage, tightness, and exact set match are computed only on enumerable discrete tasks. Bounded tasks use interval coverage and bound error, while unresolved tasks use label accuracy and incompleteness-reason validity. The main table reports PointAcc on point-identified items, FCR on all non-point items in the full test set, and SetCov. and Tightness on enumerable-result items; the ablation table averages the three dataset-level scores after computing them independently. Results from different random seeds are averaged only after per-seed metrics have been computed. No test-set examples are used to select prompts, task templates, or evaluation thresholds.

E Error Taxonomy

Certificate errors are divided into enumeration omissions, invalid factual completions, branch execution errors, result aggregation errors, interval-bound errors, and diagnosis or incompleteness-reason errors. This taxonomy supports the following interpretation. If removing enumeration increases FCR, enumeration prevents collapse. If removing factual verification lowers certificate validity, filtering is doing semantic work. If removing branch rollout lowers coverage, the model is not merely naming completions. If removing diagnosis lowers label accuracy while preserving values, diagnosis is a separate capability rather than a formatting artifact.

F Failure Cases

The natural-language evaluation illustrates two complementary failures. A model may choose one latent value and output a point answer for set, bounded, and unresolved cases, or it may list multiple branches correctly but over-predict uncertainty on hidden-irrelevant point cases. The first failure is false certainty; the second is false uncertainty caused by imperfect rule parsing. Reporting both is important because a system that avoids certainty by declaring every task unresolved is not solving the identification problem.

G Computational Considerations

Oracle generation is embarrassingly parallel across examples. The principal cost is the number of candidate completions times execution cost. Three optimizations are recommended: prune candidates using factual constraints before full counterfactual execution; cache factual states for repeated interventions; and merge completions that are equivalent for the target query. For continuous tasks, symbolic interval propagation avoids dense sampling. The model certificate can be lengthier than a single answer, so evaluations should report token count and wall-clock latency alongside accuracy.

H Additional Experimental Results

H.1 Component Ablation

The ablation uses Qwen2.5-14B and removes one EICR component at a time. Scores are macro-averaged across Code-EIBench, SCM-EIBench, and NL-EIBench.

Table 4: Component ablation on Qwen2.5-14B, averaged across the three EIBench representations. FCR uses non-point items; SetCov. and Tight. use enumerable-result items. Values are percentages (%).

Variant	PointAcc	IdentAcc	FCR	FUR	SetCov.	Tight.	CertValid
Full EICR-SFT	83.9	79.6	14.4	13.8	79.4	80.9	84.7
w/o completion enumeration	70.8	47.6	68.7	16.4	25.8	32.6	36.9
w/o factual verification	77.6	64.3	29.8	18.7	61.7	54.6	42.8
w/o branch-wise rollout	78.7	68.4	24.6	17.3	41.8	47.5	48.3
w/o identification diagnosis	81.8	51.7	45.9	15.2	68.5	66.8	67.4

Table 5: Seed stability of EICR-SFT on Qwen2.5-14B on the complete test sets. Values are percentages, reported as mean $\pm$ standard deviation over three random seeds.

Dataset	PointAcc	FCR	SetCov.	CertValid
Code-EIBench	86.1 ± 0.5	11.7 ± 0.6	82.8 ± 0.7	85.4 ± 0.9
SCM-EIBench	84.5 ± 0.7	13.9 ± 0.8	79.8 ± 0.9	84.8 ± 1.0
NL-EIBench	81.1 ± 0.9	17.6 ± 1.0	75.5 ± 1.1	83.9 ± 1.2

H.2 Seed Stability

We assess result stability by repeating the full EICR-SFT configuration with three random seeds on the Qwen2.5-14B backbone and evaluating the complete test sets. The table reports the mean and standard deviation for each dataset. The variation is small relative to the gaps between the complete certificate model and the main comparison methods, and the ordering of the three representations is unchanged across seeds.

H.3 Performance by Task Family

We further break down the Qwen2.5-14B EICR-SFT results by task family after averaging across the three representations. Hidden-irrelevant cases are relatively easy because the retained completions agree on the queried value. Hidden-divergent and bounded cases require explicit preservation of alternative outcomes, while unresolved cases additionally require the model to recognize that a complete certificate cannot be formed. The resulting pattern is consistent with the component ablation: completion enumeration and identification diagnosis are most important when the task contains genuine ambiguity or incomplete support.

The family-level results indicate that false certainty is concentrated in cases where the answer requires either multiple branch outcomes or an explicit completeness diagnosis. In contrast, hidden-irrelevant cases mainly test whether the model can preserve redundant completions without introducing unnecessary uncertainty.

I Theoretical Supplement

I.1 Certificate Soundness

Proposition 4 (certificate soundness). Suppose execution is deterministic and a certificate retains only completions that agree with the factual observation. If every retained branch is re-executed under the intervention and the reported result set is the deduplicated set of branch values, then every reported value is supported by at least one observation-consistent completion.

Proof sketch. By the factual verification condition, each retained completion belongs to $C(O)$. By the branch verification condition, each reported branch value equals $\mathrm{Exec}(M_c, \mathrm{do}(I))[Y]$ for its retained completion c. Therefore every element of the reported result set belongs to $R(O, I, Y)$. Deduplication changes multiplicity but not set membership. □

Table 6: Task-family breakdown for EICR-SFT on Qwen2.5-14B, averaged across the three EIBench representations. Values are percentages. SetCov. is reported only for enumerable-result families; bounded and unresolved families require the task-specific interval or reason metrics defined in the text.

Task family	IdentAcc	FCR	SetCov.	CertValid
Hidden-irrelevant	88.6	–	88.7	91.4
Hidden-divergent	82.3	9.8	78.5	84.1
Bounded	76.9	12.6	–	80.6
Query-dependent	78.1	11.7	75.9	81.9
Unresolved	71.5	18.9	–	75.7

I.2 Certificate Completeness

Proposition 5 (certificate completeness). Suppose the declared candidate space is complete, factual agreement is evaluated exactly, and the oracle enumerates every candidate completion before intervention execution. The resulting certificate contains every evidence-compatible branch and no unsupported branch.

Proof sketch. Exhaustive enumeration exposes every element of $\mathcal{C}(\mathcal{M})$. Exact factual verification retains precisely the elements satisfying $\mathrm{Agree}(\mathrm{Exec}(M_c, \mathrm{factual}), O)$, which is the definition of $C(O)$. Executing the intervention for each retained completion therefore produces exactly the image $R(O, I, Y)$. □

I.3 Identification Equivalence

Corollary 1 (result-set criterion). For a finite complete candidate space with at least one factual completion, the target is point-identified if and only if $|R(O, I, Y)| = 1$. If $|R(O, I, Y)| > 1$, the target is set-identified.

Proof sketch. If the result set is a singleton, every evidence-compatible completion yields the same target value, which is precisely point identification. If it contains more than one value, two compatible completions disagree on the target, so a unique point conclusion is unsupported while the enumerated set remains the exact answer. □

I.4 Candidate-Space Refinement

Proposition 6 (monotonicity under refinement). Let $\mathcal{C}_1(\mathcal{M}) \subseteq \mathcal{C}_2(\mathcal{M})$ be two candidate spaces for the same mechanism family, observation, intervention, and target. When both factual completion sets are nonempty, their result sets satisfy $R_1(O, I, Y) \subseteq R_2(O, I, Y)$. Consequently, narrowing the candidate space can remove supported outcomes, while broadening it cannot remove an existing disagreement.

Proof sketch. Every completion retained under $\mathcal{C}_1(\mathcal{M})$ is also available under $\mathcal{C}_2(\mathcal{M})$ and passes the same factual agreement test. Its intervention value therefore appears in both result sets. The stated consequences follow directly from set inclusion: a singleton result under the broader space remains the same singleton under any nonempty refinement, whereas a non-singleton result under the narrower space remains non-singleton after broadening. □