

Abduction Dominates Intervention: Isolating the Steps of Counterfactual Reasoning in a Small Neuro-Symbolic System

Miki
RatrektLabs
miki@ratrektlabs.com

Abstract—Counterfactual inference is conventionally described as three steps — abduction, action, prediction — but the steps are rarely priced separately. We isolate abduction by holding knowledge and intervention fixed and varying only whether the observed evidence is used. On 25,978 counterfactual queries over 200 held-out structural causal models, the system that abducts scores 63.3% against 15.9% for the identical system that discards the evidence, a gap of 47.4 points on the queries where abduction changes the answer. For scale, the same isolation applied to the interventional operator in earlier work yielded 7.6 points. We further find that discarding the evidence is worse than not reasoning at all: that condition scores 48.8%, below the 50.5% majority-class baseline, so a solver with good knowledge and correct interventional machinery is beaten by a constant predictor when it answers the wrong question precisely. A transformer trained end to end partially recovers abduction, reaching 46.6% where the evidence-discarding solver reaches 15.9%, but remains 16.7 points behind executing the procedure. Knowledge-base completion, which was worth 13.0 points at the interventional rung, is worth only 3.6 here: inferring which world one was in dominates recovering additional rules. The system is a transformer of 819,968 parameters with a 40-token vocabulary containing no natural language, paired with a Datalog solver, and trains from scratch on CPU. We identify one reported figure that is definitional rather than empirical, and release all code, seeds, and metrics.

Index Terms—counterfactual inference, abduction, causal hierarchy, neuro-symbolic systems, logic programming, structural causal models

I. SCOPE AND CONTRIBUTION

We claim no novelty of mechanism. The three-step characterisation of counterfactual inference is Pearl’s [1], the neuro-symbolic pairing of a learned proposer with a symbolic reasoner is established [2], [3], and the completion component is deliberately unremarkable [4], [5].

Our contribution is a price list. Counterfactual inference is normally presented as an indivisible procedure, and systems are evaluated as wholes. We instead hold every component fixed except one and measure what each is worth, which makes it possible to say that abduction dominates intervention by roughly a factor of six, that omitting it is worse than declining to reason, and that neural completion — valuable one rung down — contributes little here. To our knowledge these steps have not previously been priced against each other on a common benchmark.

Two design choices make the pricing credible: every condition differs from its control in exactly one respect, and the system is small enough that its behaviour is fully inspectable.

Relationship to prior work by the same author. An earlier study [6] applied the same isolation methodology one rung down, to intervention. **It uses a different world configuration (10 variables, 2 exogenous roots) and a different checkpoint.** No number in the present paper is comparable to it, and none is presented as such; the 7.6-point figure quoted for scale is cited as a separate measurement under separate conditions.

II. INTRODUCTION

Pearl’s causal hierarchy separates association, intervention, and counterfactuals [1]. The third rung is distinguished by requiring information the first two discard: a counterfactual asks what would have happened given what actually did happen, so the reasoner must first infer which world it was in.

That inference — abduction — is the step no purely interventional system performs. Systems are typically evaluated end to end, so its contribution is not separately visible. This paper measures it.

The instrument is deliberately blunt. We take a working interventional system, add abduction ahead of it, and construct a control that is identical in every respect except that it discards the evidence. Both conditions read the same knowledge base, apply the same intervention, and run the same solver. Whatever separates them is abduction.

A. Contributions

- An isolation of abduction under fixed knowledge and fixed intervention, measured at 47.4 accuracy points on the queries where it changes the answer (Section V-A).
- The finding that omitting abduction is worse than not reasoning: the evidence-discarding condition falls below the majority-class baseline (Section V-B).
- A measurement of how much of abduction a transformer recovers from training alone, and how much remains (Section V-C).
- Evidence that knowledge-base completion, worth 13.0 points at the interventional rung, is worth 3.6 at the counterfactual one (Section V-D).

TABLE I
THE THREE STEPS, AND WHICH ARE NEW.

Step	Operation	Status
Abduction	retain root worlds matching evidence	new
Action	delete rules concluding X ; assert X	existing
Prediction	forward-chain to fixpoint	existing

- An explicit account of one reported figure that is a structural artefact rather than a result (Section VI).

III. METHOD

A. Worlds

We generate random structural causal models over 12 boolean variables, of which the first 3 are exogenous roots sampled independently as Bernoulli(0.5). Each non-root variable is assigned one or two rules; each rule body is a conjunction of one or two parents drawn from a locality window of the three immediately preceding variables. Rules sharing a head denote disjunction, and acyclicity holds by construction.

With three roots there are $2^3 = 8$ root worlds, so every probability is computed by exhaustive enumeration rather than sampling. Thirty percent of each world’s rules are ablated to form the partial knowledge base, and test worlds are never seen during training.

B. The three steps

A counterfactual query asks $P(Y_x | X = x', Y = y')$ with $x = \neg x'$: *X was x' and Y was y' ; what would Y have been had X been otherwise?* Because the intervention always forces X away from its observed value, every query is contrary to fact. This is the canonical effect-of-treatment-on-the-treated form.

Evidence is evaluated against the *unmutilated* program: the factual world is inferred from what happened, and only then is the intervention applied inside it. Reversing that order yields intervention, not a counterfactual. Writing $\mathcal{W}$ for the root worlds, $\mathcal{R}$ for the rules, E for the evidence, $\mathcal{R}_{\setminus X}$ for the mutilated program, S for the solver, and w_x for a root world with the intervention applied:

$$\mathcal{W}_E = \{ w \in \mathcal{W} : \forall (v, b) \in E, [v \in S(\mathcal{R}, w)] = b \} \quad (1)$$

$$P(Y_x | E) = \frac{1}{|\mathcal{W}_E|} \sum_{w \in \mathcal{W}_E} \mathbb{I}[Y \in S(\mathcal{R}_{\setminus X}, w_x)] \quad (2)$$

The quantity is undefined when $\mathcal{W}_E$ is empty, which means the query describes something that could not have happened; such queries are discarded at generation time. The complete addition to an existing interventional system is fifteen lines:

```
def abduct(rules, scm, evidence):
    return [w for w in root_worlds(scm)
            if all((v in solve(list(rules), w)) == b
                  for v, b in evidence.items())]

def p_counterfactual(rules, scm, dv, dval,
```

```
target, evidence):
    worlds = abduct(rules, scm, evidence)
    if not worlds:
        return float("nan")
    prog, extra = intervene(list(rules), dv, dval)
    hits = sum(1 for w in worlds if target in
               solve(prog, frozenset(w - {dv}) | extra))
    return hits / len(worlds)
```

C. A worked example

Let v_0 be a dry season and v_1 rain, both roots, with v_2 a sprinkler caused by v_0 , v_3 wet ground caused by either v_1 or v_2 , and v_5 dry grass caused by v_0 .

The sprinkler was off and the grass was not dry; would the grass have been dry had the sprinkler been on? Abduction eliminates both worlds containing v_0 , because a dry season would have switched the sprinkler on. Within the survivors the season was not dry, so forcing the sprinkler on cannot dry the grass:

$$\begin{aligned} P(Y_x | E) &= 0.00 && \text{the truth} \\ P(Y | \text{do}(X)) &= 0.50 && \text{intervention, evidence discarded} \end{aligned}$$

The evidence is not context. It is the only thing that determines the answer.

D. Representation and model

Knowledge bases and queries are serialised into a 40-token vocabulary containing no natural language: nine control tokens including [GIVEN], four punctuation symbols, two truth values, twelve variable names, five answer buckets, and eight unused padding slots retained so that the parameter count is unchanged from prior work. A counterfactual example reads

```
[BOS] <rules> [SEP] [GIVEN] v2 = 0 , v5 = 0
[DO] v2 = 1 [QUERY] v5 [ANS] B0 [EOS]
```

Rule order *and* evidence order are shuffled on every encoding. Both are sets, and flattening them into a sequence invents an order the model would otherwise learn to depend on.

The model is a decoder-only transformer with 4 layers, $d_{\text{model}} = 128$, 4 heads, sequence length 128, and 819,968 parameters, trained from scratch. Training from scratch is methodological rather than economic: Pearl’s canonical examples pervade the corpora of pretrained models, so a pretrained model answering correctly could not be distinguished from one recalling a memorised example.

One model trains on three objectives simultaneously — knowledge-base completion, interventional queries, and counterfactual queries — for 6,000 steps at batch size 64 with AdamW at 3×10^{-4} , on 39,623 examples from 2,000 worlds. Test loss falls from 3.51 to 0.57. Sharing one model across objectives is what keeps the direct-answering ablation fair: it has identical capacity and identical training compute to the full pipeline.

IV. EXPERIMENTAL DESIGN

Six conditions vary along two axes: what the system knows, and what it does with the evidence.

TABLE II
CONDITIONS. C AND D DIFFER IN EXACTLY ONE RESPECT.

	Knowledge	Evidence	Operator
CF-A	partial	used	abduct + do
CF-B	partial	learned	end-to-end
CF-C	expanded	used	abduct + do
CF-D	expanded	discarded	do only
CF-E	complete	—	observe
CF-F	expanded	—	observe

TABLE III
MAIN RESULTS. MAJORITY-CLASS BASELINE 50.5%.

Condition	Accuracy	MAE	Abduction matters
CF-A abduct, partial	66.5%	0.254	56.6%
CF-B transformer alone	61.5%	0.323	46.6%
CF-C abduct, expanded	70.1%	0.214	63.3%
CF-D evidence discarded	48.8%	0.272	15.9%
CF-E see, complete	57.6%	0.254	5.0%
CF-F see, expanded	51.8%	0.307	18.8%

TABLE IV
ACCURACY BY REASONING DEPTH.

Condition	1 hop	2 hops	3 hops	4 hops
CF-A abduct, partial	75.2%	66.8%	60.6%	54.6%
CF-B transformer alone	60.0%	60.5%	63.2%	64.6%
CF-C abduct, expanded	74.7%	69.8%	67.3%	64.6%
CF-D evidence discarded	55.6%	47.2%	45.7%	41.5%
CF-E see, complete	64.3%	57.0%	55.0%	47.2%
CF-F see, expanded	59.0%	51.0%	48.1%	42.2%

CF-C against CF-D is the entire experiment. Both read the same expanded knowledge base, apply the same intervention, and run the same solver. The only difference is that D discards the evidence, skipping abduction. Any gap is attributable to abduction alone.

We report exact bucket accuracy and mean absolute error against $P(Y_x | E)$, stratified by reasoning depth and by whether abduction changes the answer — defined as $|P(Y_x | E) - P(Y | \text{do}(X))| > 0.05$.

V. RESULTS

25,978 queries over 200 held-out worlds. The majority-class baseline is 50.5%. Abduction changes the answer on 44.6% of queries.

A. Abduction is worth 47.4 points

On the queries where abduction changes the answer, CF-C scores 63.3% and CF-D scores 15.9%. Overall the gap is 21.3 points; on the subset where the step is load-bearing it is 47.4.

The two conditions are identical apart from the evidence. Neither knowledge nor operator nor solver differs, so the gap is not attributable to anything else.

For scale, applying this same isolation to the interventional operator in earlier work under a different configuration produced 7.6 points [6]. Abduction is the more expensive step

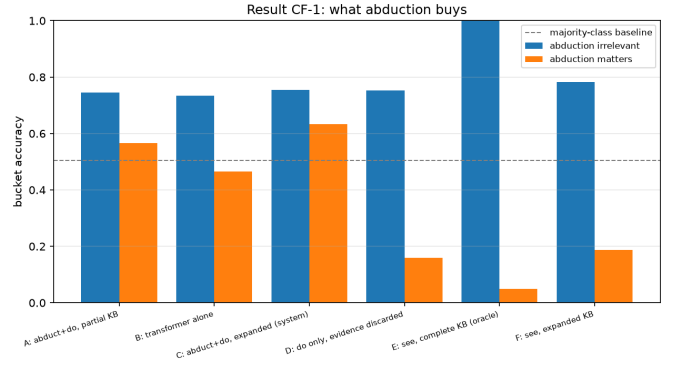


Fig. 1. Accuracy split by whether abduction changes the answer. The dashed line is the majority-class baseline.

by roughly a factor of six — which is consistent with what it does, since intervention alters a graph while abduction discards possible worlds.

B. Discarding the evidence is worse than not reasoning

CF-D scores 48.8% overall, *below* the 50.5% majority-class baseline. A solver holding a good knowledge base, applying textbook-correct interventional surgery, and chaining to fixpoint without error, is outperformed on counterfactual queries by a predictor that always answers with the most common bucket.

The interventional machinery is not broken; it answers its own question correctly. It is answering the wrong question. Precision applied to the wrong question is not merely unhelpful — it is worse than abstention, because it is confidently wrong in a structured way that a constant predictor is not.

C. The transformer partially recovers abduction

CF-B, which answers counterfactuals directly with no solver, reaches 46.6% on the subset where abduction matters. That is far above CF-D’s 15.9%, so the network trained on counterfactual examples did acquire some capacity to condition on evidence, rather than merely performing intervention.

It nonetheless trails CF-C by 16.7 points. A learned approximation of a three-step procedure, executed in a single forward pass, recovers a substantial fraction of the behaviour and not the whole of it.

D. Completion matters less at this rung

Adding neural knowledge-base completion moves accuracy from CF-A’s 66.5% to CF-C’s 70.1%, a gain of 3.6 points. The equivalent step in interventional work was worth 13.0 points [6].

The interpretation is that the binding constraint has moved. At the interventional rung the limiting factor was missing rules; here it is knowing which world one was in. Recovering additional rules helps less when the dominant uncertainty is over exogenous state rather than structure. A system budget allocated by rung-2 intuitions would over-invest in completion.

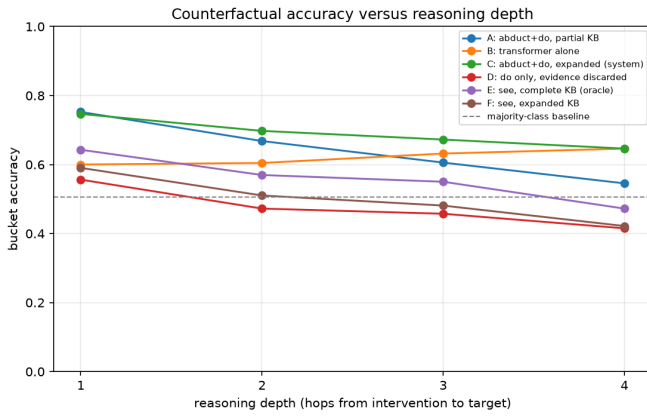


Fig. 2. Counterfactual accuracy versus reasoning depth.

E. Depth

CF-C falls 10.1 points from one hop to four, against 14.1 for CF-D and 20.6 for CF-A. Abduction and completion together keep the full system flattest as reasoning chains lengthen.

VI. LIMITATIONS AND THREATS TO VALIDITY

A. CF-E’s perfect score is definitional

On the subset where abduction does *not* change the answer, CF-E scores 100%. This is a structural artefact, not a result: measured directly, 0 of 4,233 such queries have $\text{bucket}(P(Y | X)) \neq \text{bucket}(P(Y_x | E))$, so the observational baseline is guaranteed correct there by the construction of the query set. It carries no information and should not be read as evidence that observation suffices.

B. CF-B’s depth trend is contaminated

Answer entropy falls as depth increases, because the answer distribution concentrates. CF-B’s accuracy rising with depth should therefore not be read as improved reasoning at depth. The falling curves for CF-A, CF-D, CF-E and CF-F, and the relative flatness of CF-C, are not explained by this artefact.

C. The answer distribution is skewed

The most common bucket accounts for roughly half of all queries, which is why the 50.5% baseline is reported against every condition. CF-D falling below it is meaningful precisely because the baseline is high.

D. No comparability with prior work

The configuration here (12 variables, 3 roots) differs from the earlier interventional study (10 variables, 2 roots), and the checkpoints differ. The 7.6-point and 13.0-point figures quoted from that work are cited for scale as separate measurements under separate conditions, not as controlled comparisons. The earlier code and checkpoint are preserved under a version-control tag and remain reproducible unchanged.

E. Scale and scope

Boolean variables, deterministic mechanisms, no latent confounders beyond the represented roots, no probabilistic evidence, and no natural language. Nested counterfactuals, whose evidence is itself counterfactual, are excluded.

VII. RELATED WORK

Pearl’s three-step characterisation of counterfactual computation [1] is the procedure we implement and decompose. DeepProbLog [2] and Neural LP [3] integrate neural components with logic programming; we avoid differentiable relaxation and train the proposer separately, which costs end-to-end optimisation but permits the component-level attribution that is this paper’s purpose. Knowledge-base completion by embedding [4] and by generative transformers [5] is mature; our completion component is drawn from that line without modification and is evaluated only for its downstream contribution. Prior work by the same author [6] applies the same isolation methodology to intervention.

VIII. FUTURE WORK

The most direct extension is a natural-language front end. The present system reads $\vee 3 : - \vee 0, \vee 1$ and nothing else; a domain knowledge base stated in prose — *rate rises suppress growth valuations* — would require an encoder with linguistic competence. A small pretrained model such as Gemma is appropriate there, and the argument against pretrained weights made in Section IV does not apply: contamination is a hazard when the model might recall the answer, not when it is asked to parse a sentence into structure the solver then evaluates. The invariant to preserve is that structure and labels continue to come from the solver, and only language comes from the model.

Nested counterfactuals and probabilistic evidence are the natural extensions within the symbolic half.

IX. CONCLUSION

Counterfactual inference decomposes into three steps, and they are not equally expensive. Holding knowledge and intervention fixed, abduction is worth 47.4 accuracy points on the queries where it is load-bearing — roughly six times what isolating the interventional operator yielded in comparable earlier work. Omitting it does not degrade a system gracefully: the evidence-discarding condition falls below a constant predictor, because precisely answering the wrong question is worse than not answering.

A transformer trained on counterfactual examples recovers a substantial part of abduction but not all of it, and neural knowledge-base completion — the dominant contributor one rung down — is worth little here. Where a system’s accuracy comes from changes with the question being asked, and component budgets set at one rung do not transfer to the next.

REPRODUCIBILITY

All code, seeds, and metrics are available. The pipeline runs on an 8-vCPU CPU-only machine with no GPU: training takes approximately forty minutes and evaluation twelve. The test suite comprises 101 tests, including oracle values derived by hand from the causal semantics, order-invariance checks over both rule and evidence ordering, and train/test world separation.

REFERENCES

- [1] J. Pearl, *Causality: Models, Reasoning, and Inference*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [2] R. Manhaeve, S. Dumančić, A. Kimmig, T. Demeester, and L. De Raedt, “DeepProbLog: Neural probabilistic logic programming,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, 2018.
- [3] F. Yang, Z. Yang, and W. W. Cohen, “Differentiable learning of logical rules for knowledge base reasoning,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.
- [4] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 26, 2013.
- [5] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Çelikyilmaz, and Y. Choi, “COMET: Commonsense transformers for automatic knowledge graph construction,” in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2019, pp. 4762–4779, arXiv:1906.05317.
- [6] Miki, “Precision is not the objective: Tuning neural knowledge-base completion for downstream interventional reasoning,” RatrektLabs, 2026.