Linear-LLM-SCM: Benchmarking LLMs for Coefficient Elicitation in Linear-Gaussian Causal Models

Kanta Yamaoka [1 2] **Sumantrak Mukherjee** [1] **Thomas Gärtner** [3] **David Selby** [1] **Stefan Konigorski** [3 4] **Eyke Hüllermeier** [1 5 6] **Viktor Bengs** [1] **Sebastian Vollmer** [1 2]

Abstract

Large language models (LLMs) have shown potential in identifying qualitative causal relations, but their ability to perform quantitative causal reasoning—estimating effect sizes that parametrize functional relationships—remains underexplored in continuous domains. We introduce Linear-LLM-SCM, a plug-and-play framework for evaluating LLMs on Linear Gaussian structural causal model parametrization when a directed acyclic graph (DAG) is given. The framework decomposes a DAG into local parent-child sets and prompts an LLM to produce a regression-style structural equation per node, which is aggregated and compared against available ground-truth parameters. Our experiments with seven real-world DAGs effect ground truth illustrate limitations of LLMs as quantitative causal parameterizers. Across most models, we observe variability in coefficient estimates and sensitivity to structural perturbations. We open-sourced the framework to further encourage the community to work on studies toward the use of LLM for causal effect elicitation in safety-critical domain, e.g., healthcare.

1. Introduction

Robust intelligence requires an agent to have an internal "world model" (Ha & Schmidhuber, 2018), that is, an internal causal mechanism to infer causal structures and their effect relationships (Pearl, 2019). Recent work (Richens & Everitt, 2024) suggests that *any agent capable of solving complex decision tasks must effectively learn a causal model of its data-generating process.* Given the advancement of LLMs, they exhibit the ability to encode a corpus of human knowledge, e.g., clinical knowledge (Singhal et al., 2023). One may speculate whether LLMs encode literature with causal information and constructing causal models. Per *LLMs' Causal Hierarchy*, classified by Zhang et al. (2023), LLMs have shown promise in Type 1 tasks (identifying causal relationships using domain knowledge), but they struggle with Type 2 (discovering new knowledge from data) and Type 3 (quantitative estimation of consequences) tasks. Empirically and theoretically, Type 3 aspects in continuous domains remain under-explored. To fill this gap, we propose *Linear-LLM-SCM*, a framework evaluating quantitative causal capabilities of LLMs. It decomposes DAGs into local parent-child structures, assigning LLMs to elicit regression coefficients for linear Gaussian SCMs. In our open-sourced software, researchers can use their own DAGs and LLMs plug-and-play to evaluate in their domains.

In this contribution, we investigate the following **Research Questions (RQs)**: **RQ1:** Given a pre-specified DAG, can LLMs elicit plausible regression coefficients compared to real-world ground truths? **RQ2:** How robust is this parameterization when facing adversarial conditions, such as DAG misspecification (spurious edges) or changes in variable units? **RQ3:** What are the common failure modes encountered when using LLMs for parameterization?

Structural perturbation in RQ2 accounts for observational or modeling limitations. Real-world modeling, including the specification of a directed acyclic graph (DAG) as ground truth, contains uncertainty (Padh et al., 2025; Vowels et al., 2023). If possible, subsequent causal effect estimation should be robust against them. This also mitigates data leakage in the LLM training corpus (Yang et al., 2023).

Contributions. We introduce a framework to evaluate LLMs' ability to estimate linear SCM coefficients. We open-source an evaluation pipeline that supports plug-and-play DAGs, variable metadata, and LLMs, reporting coefficient-

[1]Data Science and its Applications, German Research Centre for Artificial Intelligence (DFKI), Germany [2]Dept. of Computer Science, University of Kaiserslautern–Landau (RPTU), Germany [3]Digital Health - Machine Learning Research Group, Hasso Plattner Institute for Digital Engineering, Germany [4]Hasso Plattner Institute for Digital Health at Mount Sinai, Icahn School of Medicine at Mount Sinai, USA [5]Institute of Informatics, University of Munich (LMU), Germany [6]Munich Center for Machine Learning (MCML), Germany. Correspondence to: Kanta Yamaoka <kanta.yamaoka@dfki.de>.

Proceedings of the Workshop on Structured Data for Health at the 43rd International Conference on Machine Learning, Seoul, South Korea. Copyright 2026 by the author(s).

distance and ordering metrics against ground truth. [1]

2. Background and Related Work

Structural causal models (SCMs) formalize causal systems through a set of structural equations (e.g., in form of a DAG) and distributional assumptions. Recent work explored using LLMs for causal reasoning, focusing on causal discovery and qualitative inference (Long et al., 2022; 2023; Kiciman et al., 2024). Recent studies raise concerns about LLMs' causal capabilities, arguing that they may succeed at identifying causal relationships using prior knowledge, while struggle to discover causal structure from data and quantitatively estimate causal effects (Zhang et al., 2023; Zečević et al., 2023; Yang et al., 2023; Jin et al., 2023).

Existing work on LLM-based causal effect estimation focuses on discrete domains or requires numerical observational data and specialized architectures (Chen et al., 2023; Feder et al., 2023; Zhang et al., 2024). Closest to our setting, Bynum & Cho (2025) combined LLMs with pre-specified causal graphs but estimate effects via sampling-based conditional distributions rather than directly eliciting structural parameters. In contrast, we study whether LLMs can directly estimate *continuous* linear causal effect parameters for pre-specified SCMs using only causal structure and variable semantics, without observational data. Appendix A provides a detailed literature review.

3. Linear-LLM-SCM Framework

We provide pre-specified DAG structures decomposing real-world phenomena into parent-child interactions. The system iterates through the DAG, calling LLMs via prompt templates to elicit functional mappings for each parent-child set, then aggregating results to obtain full effect parameters. This assumes that LLMs have learned quantitative or qualitative information from their large training corpus. For quantitative aspects, LLMs may have encountered scientific literature with effects formalized as SCMs, or linear regressions. For qualitative aspects, LLMs encountered natural language causal statements (e.g., "Coffee consumption positively regulates alertness after 1 hour") in their corpus.

3.1. Overview of the benchmarking framework

Our benchmarking framework requires a DAG structure $\mathcal{G}$ consisting of nodes $\mathcal{V}$ and directed edges $\mathcal{E}$, variable descriptions $\mathcal{D}$ including a short textual description and their unit (e.g., "GC": "Glucose (μM)"), and variable constraints $\mathcal{R}$ (lower and upper bounds). The parameterization task assumes linearity in structural equation, Gaussian noise of target variables while beta coefficients do not have noise assumption. The variable constraints were sourced from original literature or author judgment, as shown in Appendix F Table 4. The DAG structure is given by a yaml file containing that information and effect ground truths. An example including the required input can be found in Appendix B in Figure 1. Given these inputs, Linear-LLM-SCM traverses nodes in topological order and decomposes the graph into local parent-child elicitation tasks. For each target node, it prompts the LLM with domain context, variable descriptions, units, ranges, and output-format instructions, then parses the returned linear equation to extract the intercept and parent coefficients.

Response Parsing: We obtain the functional mapping of direct parents and target variables as textual structural equations from LLMs. While LLMs provide plausibility descriptions for debugging, we utilize only the parameterized equation provided as a string. From this property, the system extracts beta coefficients from the LLM structural equation format, which should be compliant with the demonstration in the prompt template as in Appendix C Figure 3.

3.2. Prompt for Node Level Function Parametrization

Our framework traverses a DAG. For each node, the LLM receives descriptions of its direct parents. While one could pass the entire graph via text representations (e.g., DAGitty by (Textor et al., 2017) or Mermaid by (Sveidqvist & Contributors to Mermaid, 2014)) or feed entire DAGs into visual language models, we chose the parent-child template instead. This focus on local structures aligns with recent work (Bynum & Cho, 2025; Nafar et al., 2025).

Our prompt includes the names and short descriptions of the target variable and its direct parents, alongside formatting instructions. On a high level, the prompt consists of three parts: First, domain expert persona (e.g., expert in consumer behaviour), a summary of the phenomenon of interest (e.g., " Cachexia is a complicated metabolic syndrome related to..."), and variable units are introduced to the LLM. Next, the parameterization task, linear equation template, and variable ranges appear. Finally, the LLM is informed of the output format, which starts with the thought process (*reasoning tokens*) and ends with the parameterization result as a string. Appendix C Algorithm 1 includes details of this process as line 4, and Figure 3 shows an example of a prompt for a parent-child local structure.

Iterative Feedback. The framework elicits functional mapping in parent-child local structure without generating samples. To enforce global consistency as a DAG, it includes an iterative feedback refinement mechanism. This process requires pre-specified hard constraints for each variable, analytically performs sanity check and retry for each elicitation. Details are available in Appendix C Algorithm 2.

[1] https://github.com/datasciapps/parameterize-dag-with-llm

3.3. Metrics for Evaluating Parameterization

Our framework compares ground-truth effect parameters and the effects elicited via LLMs, obtaining metrics (M1)–(M4), which we defined as in Appendix D. For (M1)–(M3), we compute the L2 norm over all linear coefficients across nodes; the contribution aggregation differs by metric. These metrics capture distances between LLM-elicited parameter vectors per node and ground truth vectors per node. (M4) captures relative effect-size ordering per node.

4. Experimental Setup

We assume linear functional elicitation for pre-specified DAGs using LLMs given their nodes and directed edges. We obtained ground-truths in linear settings, including learned parameters as well as DAG structures (Leonelli, 2025) in the real-world settings. In our functional elicitation, we obtain a textual structural equation directly but do not create conditional distributions from LLMs as in (Bynum & Cho, 2025). For DAGs and their parameterization, we focus on continuous variables. The bayesian networks we used for ground-truths have learned effects in continuous linear Gaussian settings. A small subset of DAGs, for example, some variables in *Expenditure* DAG, for example, *Card*, whether card is accepted or not, was also binary but because the ground-truth network learned this effect as continuous, in our prompt we also treat it as continuous.

4.1. LLM Model Selection

We employed general purpose pre-trained LLMs (Gemini 2.5 Flash, Llama 3 family, and GPT-5.4) as in Appendix E Table 3. We tried models with different model sizes and architectures either mixture-of-experts (MoE), where an input is routed into different experts, obtaining output in an ensemble fashion (Jacobs et al., 1991), vs non-MoE ones. Gemini 2.5 Flash consists of sparse MoE (Gemini Team, Google, 2025). Representative models from the open-weights community, Llama 3.1 8B and Llama 3.3 70B have dense transformer, where all parameters are used for inference (Llama Team, Meta, 2024). From frontier model family, we include GPT-5.4 [2] with limited model details disclosed.

4.2. DAGs With Ground Truth Effect Parameters

Our experiment employed real-world DAGs from a Bayesian Network repository, *BnRep* (Leonelli, 2025). We kept the ones with a continuous Linear Gaussian setting which corresponds to the linear SCM setting. For convenience, we included DAGs with at most 15 nodes. We excluded DAGs where variable names consisted of letters and numbered suffixes. Appendix F Figure 5 describes how we selected DAGs and effect ground-truths from the repository, and Table 4 lists the 7 DAGs included in our experiments.

[2] https://openai.com/index/introducing-gpt-5-4/

4.3. Adversarial Conditions

As in RQ2, we introduce two adversarial conditions.

(I) Tweaking Units to Check Robustness. For the first type of adversarial conditions, namely changes in units, we use *Cachexia1* DAG from Table 4 because the DAG comes with units and the DAG structure is relatively simpler among others. We captured aggregated trends on each model with temperature set to zero to make the behavior as deterministic as possible. We employed the following two conditions: (A) the *Cachexia* DAG with original units μM, and (B) the *Cachexia* DAG with tweaked units, nM.

(II) Simulated DAG Misspecification.

We created adversarial mutated examples based on Expenditure DAG (details in Appendix Figure 7). For each, we added a spurious edge between two variables with no actual connections in the ground truths while ensuring acyclicity. The resulting four examples are referred to as (S1)-(S4) as discussed in Appendix F.4. Based on this, we performed parameterization to check how robust each LLM is.

5. Results and Discussion

In our experiments, a *run* consists of a DAG traversal from the first node to the last. For each condition (DAG, LLM), we run 25 independent runs (i.e., $n = 25$) and report mean M1–M4 values. Values after $\pm$ denote 95% CIs.

5.1. RQ1: Direct Estimation Results

Table 1 presents direct parameter estimation results across four LLM models on all DAGs. M1 shows high variability across DAGs because it is not scale-invariant with respect to variable ranges. We therefore focus on M2, M3, and M4. Under M2, the best model varies by DAG. Under M3, Gemini 2.5 Flash and GPT-5.4 each perform best on three DAGs. Under M4, GPT-5.4 performs best on four DAGs and Gemini 2.5 Flash on three, including ties. Comparing Llama 3.1 8B and Llama 3.3 70B, the larger model usually performs better, suggesting a benefit from larger parameter counts for this task. A small model, Llama 3.1 8B failed to generate parsable equations for *algal2*.

While we used temperature zero, larger models (Gemini 2.5 Flash, Llama 3.3 70B, and GPT-5.4) still show substantial stochasticity across metrics, with non-Gaussian distributions at $n = 25$, possibly due to hardware or software factors beyond our control via external APIs. This variability is concerning for safety-critical domains such as healthcare. This result aligns with prior work on LLM non-determinism (Kli-

shevich et al., 2025)—a failure mode in response to RQ3.

Table 1. Direct estimation results in the main text (Averaged $n = 25$, Temp 0) with 95% CIs. We focus discussion on $M3$ and $M4$; the full $M1$–$M4$ table is provided in Appendix G Table 5. For $M3$, lower is better (↓); for $M4$, higher is better (↑). See Table 4 for DAG descriptions.

MODEL	DAG	M3 ↓	M4 ↑
Gemini 2.5 Flash	cachexia1	1.07 ± 0.11	**1.00** ± 0.00
Llama 3.1 8B		1.04 ± 0.00	0.00 ± 0.00
Llama 3.3 70B		1.35 ± 0.07	0.44 ± 0.20
GPT-5.4		**0.98** ± 0.05	0.64 ± 0.19
Gemini 2.5 Flash	expenditure	**1.00** ± 0.23	**7.52** ± 0.20
Llama 3.1 8B		2.05 ± 0.00	7.00 ± 0.00
Llama 3.3 70B		1.55 ± 0.23	6.56 ± 0.20
GPT-5.4		1.96 ± 0.08	6.36 ± 0.19
Gemini 2.5 Flash	foodsecurity	0.24 ± 0.05	0.50 ± 0.21
Llama 3.1 8B		0.45 ± 0.00	0.00 ± 0.00
Llama 3.3 70B		0.40 ± 0.00	0.00 ± 0.00
GPT-5.4		**0.11** ± 0.03	**0.96** ± 0.08
Gemini 2.5 Flash	algal2	**0.51** ± 0.10	**2.00** ± 0.00
Llama 3.1 8B		n/a	
Llama 3.3 70B		0.56 ± 0.05	**2.00** ± 0.00
GPT-5.4		0.52 ± 0.04	**2.00** ± 0.00
Gemini 2.5 Flash	lexical	2.04 ± 0.05	2.04 ± 0.27
Llama 3.1 8B		2.56 ± 0.00	1.00 ± 0.00
Llama 3.3 70B		2.27 ± 0.04	0.78 ± 0.37
GPT-5.4		**1.76** ± 0.07	**2.24** ± 0.20
Gemini 2.5 Flash	liquefaction	**0.84** ± 0.03	2.80 ± 0.16
Llama 3.1 8B		2.01 ± 0.00	**3.00** ± 0.00
Llama 3.3 70B		1.27 ± 0.03	**3.00** ± 0.00
GPT-5.4		0.88 ± 0.02	2.96 ± 0.08
Gemini 2.5 Flash	stocks	1.01 ± 0.06	3.48 ± 0.24
Llama 3.1 8B		1.59 ± 0.00	2.88 ± 0.13
Llama 3.3 70B		**0.93** ± 0.06	2.60 ± 0.23
GPT-5.4		1.18 ± 0.06	**3.80** ± 0.12

5.2. RQ2: Robustness Results

(I) Unit Tweaking Robustness. Appendix G Table 6 evaluates model robustness when unit values for the Cachexia1 DAG are tweaked from μM to nM. Here we only provide overview. In the lens of M2-M4, sometimes, counter-intuitively tweaked units resulted in better parameterization. One possible reason that the metric improved instead of degradation is increased numerical precision in textual formal, e.g., *1* μM $\rightarrow$ *1000* nM. We plan to extend the empirical coverage in the future to see if this is the case.

(II) DAG Misspecification Robustness. Table 2 presents the performance when the Expenditure DAG is intentionally misspecified by adding spurious edges, where O indicates the original DAG while S1-S4 indicate mutated variants; the full $M1$–$M4$ table is provided in Appendix G Table 7. As above, we focus on M3 and M4 in the main text. Under

Table 2. Robustness under DAG misspecification in the main text for the expenditure DAG (Averaged $n = 25$, Temp 0) with 95% CIs. We focus discussion on $M3$ and $M4$; the full $M1$–$M4$ table is provided in Appendix G Table 7. For $M3$, lower is better (↓); for $M4$, higher is better (↑). Misspec: (O)=Original DAG, (S1)-(S4)=Spurious edges added.

MODEL	MISSPEC.	M3 ↓	M4 ↑
Gemini 2.5 Flash	O	**1.00** ± 0.23	**7.52** ± 0.20
	S1	1.37 ± 0.28	6.20 ± 0.30
	S2	1.24 ± 0.27	6.40 ± 0.25
	S3	1.35 ± 0.25	6.56 ± 0.26
	S4	1.99 ± 0.19	7.32 ± 0.27
Llama 3.1 8B	O	**2.05** ± 0.00	**7.00** ± 0.00
	S1	2.48 ± 0.00	6.00 ± 0.00
	S2	2.06 ± 0.00	6.00 ± 0.00
	S3	2.11 ± 0.00	6.00 ± 0.00
	S4	2.41 ± 0.00	**7.00** ± 0.00
Llama 3.3 70B	O	**1.55** ± 0.23	6.56 ± 0.20
	S1	1.70 ± 0.22	5.52 ± 0.23
	S2	1.92 ± 0.19	5.92 ± 0.25
	S3	2.04 ± 0.12	5.36 ± 0.19
	S4	2.22 ± 0.14	**6.68** ± 0.27
GPT-5.4	O	1.96 ± 0.08	6.36 ± 0.19
	S1	**0.90** ± 0.03	6.00 ± 0.00
	S2	1.02 ± 0.08	6.00 ± 0.00
	S3	1.01 ± 0.10	6.12 ± 0.13
	S4	1.74 ± 0.09	**6.96** ± 0.08

M3, the original DAG yields the best result for Gemini 2.5 Flash, Llama 3.1 8B, and Llama 3.3 70B, while GPT-5.4 performs best on S1. Most models therefore degrade under DAG misspecification by M3. Under M4, the original DAG performs best for Gemini 2.5 Flash and Llama 3.1 8B. For Llama 3.3 70B and GPT-5.4, variant S4 is slightly better than the original DAG, which remains second best. Overall, adversarial conditions tend to lower M4, indicating degraded performance under DAG misspecification with spurious edges—a failure mode regarding RQ3.

6. Conclusion

Linear-LLM-SCM benchmarks quantitative causal effect elicitation by decomposing DAGs into local structures, along with four evaluation metrics. Across four LLMs, Gemini 2.5 Flash performs best overall on M3 and M4 across the tested DAGs. These results highlight the importance of scale-invariant metrics such as M4, since M1 is affected by variable ranges. A challenge is the stochasticity of three large LLMs, which produce inconsistent results at temperature zero. In contrast, a small LLM was more consistent but less accurate. Such variance poses risks in safety-critical domains, e.g., healthcare. Robustness tests show that adding spurious edges degrades performance and lowers effect-ordering accuracy, indicating limited robust-

ness to structural uncertainty. Future work should extend it to non-linear functional forms and investigate ways to mitigate structural noise. We did not utilize confidence intervals from LLMs, but plan to examine whether such textual uncertainty outputs are calibrated.

Impact Statement

We foresee our work being useful to benchmark LLMs for causal effect elicitation in safety-critical domains, such as healthcare, where quantitative causal expert knowledge is needed. There are several downstream societal implications of using LLMs but we do not specify all such implications in this manuscript. Given the current results from the manuscript, we need to clarify that we should not use LLMs for quantitative causal effect estimation for real healthcare or clinical decision support, but still we need further research on safety testing and improvements.

Acknowledgement

We acknowledge funding for the project AI4Nof1 by the state of Rhineland Palatinate, Germany. We would like to thank Valentin Margraf, Jonas Hanselle, Serafima Lebedeva and Niklas Nertinger for their valuable feedback during weekly research meeting.

References

Barons, M. J., Wright, S. K., and Smith, J. Q. Eliciting probabilistic judgements for integrating decision support systems. *International Series in Operations Research & Management Science*, pp. 445–478, 2018. URL https://ideas.repec.org//h/spr/isochp/978-3-319-65052-4_17.html.

Baumann, A. and Sekanina, K. Accounting for the relationship between lexical prevalence and acquisition with Bayesian networks and population dynamics. *Linguistics Vanguard*, 8(1):209–224, December 2022. ISSN 2199-174X. doi: 10.1515/lingvan-2021-0038. URL https://www.degruyterbrill.com/document/doi/10.1515/lingvan-2021-0038/html?srsltid=AfmBOooG7SLjI542EXDJKooxAgjLOJENZTNrDwLgkg9GF_2uF6Ab6-2f.

Bynum, L. E. and Cho, K. Language models as causal effect generators. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2096–2115, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.107. URL https://aclanthology.org/2025.emnlp-main.107/.

Chen, Z., Gao, Q., Bosselut, A., Sabharwal, A., and Richardson, K. DISCO: Distilling counterfactuals with large language models. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5514–5528, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.302. URL https://aclanthology.org/2023.acl-long.302/.

Eisner, R., Stretch, C., Eastman, T., Xia, J., Hau, D., Damaraju, S., Greiner, R., Wishart, D. S., and Baracos, V. E. Learning to predict cancer-associated skeletal muscle wasting from 1H-NMR profiles of urinary metabolites. *Metabolomics*, 7(1):25–34, March 2011. ISSN 1573-3890. doi: 10.1007/s11306-010-0232-9. URL https://doi.org/10.1007/s11306-010-0232-9.

Feder, A., Wald, Y., Shi, C., Saria, S., and Blei, D. Data augmentations for improved (large) language model generalization. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 70638–70653. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/df88b275bef31ac96c85f0c4013734fc-Paper-Conference.pdf.

Gemini Team, Google. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

Greene, W. H. Econometric analysis. *Pretence Hall*, 2003.

Görgen, C. and Leonelli, M. Model-preserving sensitivity analysis for families of Gaussian distributions. *Journal of Machine Learning Research*, 21(84):1–32, 2020. ISSN 1533-7928. URL http://jmlr.org/papers/v21/18-668.html.

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Hu, J., Xiong, B., Zhang, Z., and Wang, J. A continuous Bayesian network regression model for estimating seismic liquefaction-induced settlement of the

free-field ground. *Earthquake Engineering & Structural Dynamics*, 52(11):3216–3237, September 2023. ISSN 0098-8847, 1096-9845. doi: 10.1002/eqe.3804. URL https://onlinelibrary.wiley.com/doi/10.1002/eqe.3804.

Jackson-Blake, L. A., Clayer, F., Haande, S., Sample, J. E., and Moe, S. J. Seasonal forecasting of lake water quality and algal bloom risk using a continuous Gaussian Bayesian network. *Hydrology and Earth System Sciences*, 26(12):3103–3124, June 2022. ISSN 1027-5606. doi: 10.5194/hess-26-3103-2022. URL https://hess.copernicus.org/articles/26/3103/2022/.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Jin, Z., Chen, Y., Leeb, F., Gresele, L., Kamal, O., LYU, Z., Blin, K., Gonzalez Adauto, F., Kleiman-Weiner, M., Sachan, M., and Schölkopf, B. CLadder: Assessing causal reasoning in language models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 31038–31065. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/631bb9434d718ea309af82566347d607-Paper-Conference.pdf.

Kiciman, E., Ness, R., Sharma, A., and Tan, C. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=mqoxLkX210. Featured Certification.

Klishevich, E., Denisov-Blanch, Y., Obstbaum, S., Ciobanu, I., and Kosinski, M. Measuring determinism in large language models for software code review, 2025. URL https://arxiv.org/abs/2502.20747.

Leonelli, M. bnRep: A repository of Bayesian networks from the academic literature. *Neurocomputing*, 624:129502, 2025. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2025.129502. URL https://www.sciencedirect.com/science/article/pii/S0925231225001742.

Leonelli, M., Riccomagno, E., and Smith, J. Q. Coherent combination of probabilistic outputs for group decision making: an algebraic approach. *OR Spectrum*, 42(2): 499–528, June 2020. ISSN 1436-6304. doi: 10.1007/s00291-020-00588-8. URL https://doi.org/10.1007/s00291-020-00588-8.

Liu, X., Xu, P., Wu, J., Yuan, J., Yang, Y., Zhou, Y., Liu, F., Guan, T., Wang, H., Yu, T., McAuley, J., Ai, W., and Huang, F. Large language models and causal inference in collaboration: A comprehensive survey. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7668–7684, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.427. URL https://aclanthology.org/2025.findings-naacl.427/.

Llama Team, Meta. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Long, S., Schuster, T., and Piché, A. Can large language models build causal graphs? In *NeurIPS 2022 Workshop on Causality for Real-world Impact*, 2022. URL https://openreview.net/forum?id=LQQoJGw8JD1.

Long, S., Piché, A., Zantedeschi, V., Schuster, T., and Drouin, A. Causal discovery with language models as imperfect experts. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. URL https://openreview.net/forum?id=RXlvYZAE49.

Mohan, S. and Karaletsos, T. How well do LLMs understand drug mechanisms? a knowledge + reasoning evaluation dataset, 2025. URL https://arxiv.org/abs/2511.06418.

Nafar, A., Venable, K. B., Cui, Z., and Kordjamshidi, P. Extracting probabilistic knowledge from large language models for Bayesian network parameterization, 2025. URL https://arxiv.org/abs/2505.15918.

Padh, K., Li, Z., Casolo, C., and Kilbertus, N. Your assumed DAG is wrong and here's how to deal with it. In Huang, B. and Drton, M. (eds.), *Proceedings of the Fourth Conference on Causal Learning and Reasoning*, volume 275 of *Proceedings of Machine Learning Research*, pp. 1239–1267. PMLR, 07–09 May 2025. URL https://proceedings.mlr.press/v275/padh25a.html.

Pearl, J. The seven tools of causal inference, with reflections on machine learning. *Commun. ACM*, 62(3):54–60, February 2019. ISSN 0001-0782. doi: 10.1145/3241036. URL https://doi.org/10.1145/3241036.

Richens, J. and Everitt, T. Robust agents learn causal world models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pOoKI3ouv1.

Sclar, M., Choi, Y., Tsvetkov, Y., and Suhr, A. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RIu5lyNXjT.

Sener, E. and Demir, I. Gaussian Bayesian network model of healthcare, food and energy sectors in the pandemic: Türkiye case. *Heliyon*, 10(1):e23798, January 2024. ISSN 2405-8440. doi: 10.1016/j.heliyon.2023.e23798. URL https://www.sciencedirect.com/science/article/pii/S2405844023110061.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., Payne, P., Seneviratne, M., Gamble, P., Kelly, C., Babiker, A., Schärli, N., Chowdhery, A., Mansfield, P., Demner-Fushman, D., Agüera y Arcas, B., Webster, D., Corrado, G. S., Matias, Y., Chou, K., Gottweis, J., Tomasev, N., Liu, Y., Rajkomar, A., Barral, J., Semturs, C., Karthikesalingam, A., and Natarajan, V. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, Aug 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06291-2. URL https://doi.org/10.1038/s41586-023-06291-2.

Sveidqvist, K. and Contributors to Mermaid. *Mermaid: Generate diagrams from markdown-like text*, December 2014. URL https://github.com/mermaid-js/mermaid.

Textor, J., van der Zander, B., Gilthorpe, M. S., Liśkiewicz, M., and Ellison, G. T. Robust causal inference using directed acyclic graphs: the R package 'dagitty'. *International Journal of Epidemiology*, 45(6):1887–1894, 01 2017. ISSN 0300-5771. doi: 10.1093/ije/dyw341. URL https://doi.org/10.1093/ije/dyw341.

Tsagris, M. The FEDHC Bayesian network learning algorithm. *Mathematics*, 10(15):2604, July 2022. ISSN 2227-7390. doi: 10.3390/math10152604. URL https://www.mdpi.com/2227-7390/10/15/2604.

Vowels, M. J., Camgoz, N. C., and Bowden, R. D'ya like DAGs? A survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4):1–36, April 2023. ISSN 0360-0300, 1557-7341. doi: 10.1145/3527154. URL https://dl.acm.org/doi/10.1145/3527154.

Wang, Z. CausalBench: A comprehensive benchmark for evaluating causal reasoning capabilities of large language models. In Wong, K.-F., Zhang, M., Xu, R., Li, J., Wei, Z., Gui, L., Liang, B., and Zhao, R. (eds.), *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pp. 143–151, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.sighan-1.17/.

Wright, S. Correlation and causation. *Journal of Agricultural Research*, 20(7):557–585, 1921.

Yang, L., Clivio, O., Shirvaikar, V., and Falck, F. A critical review of causal inference benchmarks for large language models. In *AAAI 2024 Workshop on "Are Large Language Models Simply Causal Parrots?"*, 2023. URL https://openreview.net/forum?id=mRwgczYZFJ.

Zečević, M., Willig, M., Dhami, D. S., and Kersting, K. Causal parrots: Large language models may talk causality but are not causal. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=tv46tCzs83.

Zhang, C., Bauer, S., Bennett, P., Gao, J., Gong, W., Hilmkil, A., Jennings, J., Ma, C., Minka, T., Pawlowski, N., and Vaughan, J. Understanding causality with large language models: Feasibility and opportunities, 2023. URL https://arxiv.org/abs/2304.05524.

Zhang, J., Jennings, J., Hilmkil, A., Pawlowski, N., Zhang, C., and Ma, C. Towards causal foundation model: on duality between causal inference and attention, 2024. URL https://arxiv.org/abs/2310.00809.

Appendix

A. Background and Related Work

The section first describes the theoretical foundation of structural causal models. Then, we will examine general LLMs' abilities and inabilities reported in the causality, which is a broad area of study including causal discovery and causal effect estimation. Then we will focus on existing efforts of using LLMs for causal effect estimation in the causality literature.

A.1. Structural Causal Models and Effect Parameterization

A Structural Causal Model is formally defined by a pair $(\mathcal{G}, \mathbf{F})$, where $\mathcal{G}$ is the Directed Acyclic Graph (DAG) representing the causal structure, and $\mathbf{F} = \{f_i\}$ is the collection of structural equations. Each endogenous variable X_i is determined by a function f_i of its direct causes (parents) $\text{Pa}(X_i)$ and an independent exogenous noise term E_i, such that $X_i = f_i(\text{Pa}(X_i), E_i)$. In our problem settings, we focus on linear causal effects to narrow down the problem space, and therefore, parameterization here refers to finding coefficients of these linear functions $\mathbf{F}$.

Conventional causal inference tasks include finding causal relationships, direction of edges and identifying such parameters using data and intervention in the real-world. However, in our work, we only aim to estimate such parameters for linear causal effects using potentially encoded knowledge from large language models (LLMs) for each variable in a *pre-specified* DAG for a real-world phenomenon.

A.1.1. LLMs General Potential and Criticism for Causality

For causal discovery, (Long et al., 2022; 2023) used LLMs to identify causal connections between node pairs in DAGs, reporting opportunities despite inconsistencies and prompt sensitivities. (Kiciman et al., 2024) found LLM-based methods outperform covariance-based algorithms in pairwise causal discovery and excel at natural language counterfactual reasoning.

There is also work raising limitations of the current LLMs in this regard. For example, Zhang et al. (2023) proposed LLMs' Causal Hierarchy, which consists of the three types: *Type 1: Identifying causal relationships using domain knowledge*, *Type 2: Discovering new knowledge from data*, *Type 3: Quantitative estimation of consequences of actions*. They claim LLMs can perform Type 1 tasks via training data but not Type 2 and Type 3 due to token generation limitations. Our work evaluates LLMs' capability for Type 3 tasks. Another study by Zečević et al. (2023) raises doubts about LLMs' causal capabilities, conjecturing they merely learned causal facts from training data rather than causal mechanisms, *Causal Parrots*. They note LLMs are not explicitly trained for causal tasks and may simply parrot causal statements without true reasoning. Their criticism focuses on causal discovery; since we assume given causal structures and only estimate effects, this may not directly apply to our settings. Criticisms also include ground truth leakage in causal discovery (Yang et al., 2023). In the broader context of causal inference, Jin et al. (2023) introduced CausalCOT, a prompting strategy for the whole causal reasoning lifecycle, concluding this task is highly challenging for LLMs.

A.1.2. LLMs for Causal Effect Estimation

While some literature indicates initial success reports for treatment effect estimation in **discrete** domains, including counterfactual generation, causal effect estimation in **continuous** domains (e.g., SCMs) remains largely unexplored. Most existing approaches for continuous causal estimation provide numerical observational data to the model, which often requires specialized architectures or tokenization schemes; by contrast, we study coefficient elicitation without observational data, using only DAG structure and variable semantics.

Bynum & Cho (2025) combined LLMs with structural causal modeling for pre-specified DAGs. However, they use sampling-based approaches to estimate effects rather than directly eliciting functional mappings. Their SD-SCMs represent effects as Conditional Probability Distributions rather than linear structural equations (Wright, 1921), focus on discrete-domain counterfactuals, while our work addresses continuous-domain parametrization for simulating real-world phenomena. Liu et al. (2025) surveys LLM-based causal inference, finding only three papers on treatment effect estimation—two in discrete domains (Chen et al., 2023; Feder et al., 2023) and one (Zhang et al., 2024) addressing both domains but requiring numerical observational data with specialized attention mechanisms. Unlike these approaches, we assume pre-specified causal DAG structures without observational data. Nafar et al. (2025) similarly benchmark *effect* estimation from pre-specified DAGs but assume discrete domains with Conditional Probability and sampling distributions, using eighty public DAGs in finance and health. Other benchmarks (Wang, 2024; Mohan & Karaletsos, 2025) examine LLMs on causality tasks but focus on causal

relationship identification rather than quantitative effect estimation in continuous SCMs.

B. Benchmarking Framework Implementation Details

B.1. Plug-and-play Configuration Example

Our open-source tooling requires a very simple yaml format to specify DAG structures and other inputs required as shown in Figure 1.

```
name: cachexia1_disease_informed_real_bounds_and_units
all_nodes:
- A
- B
- F
- GC
- GM
- V
raw_edges:
- - A
  - GC
- - B
  - V
- - F
  - GM
- - GC
  - B
- - GC
  - V
- - GM
  - A
- - GM
  - B
- - GM
  - V
node_descriptions:
  A: "Adipate (micromolar, μM)"
  B: "Betaine  (micromolar, μM)"
  F: "Fumarate  (micromolar, μM)"
  GC: "Glucose  (micromolar, μM)"
  GM: "Glutamine  (micromolar, μM)"
  V: "Valine  (micromolar, μM)"
primary_domain_name: metabolic systems
secondary_domain_name: biochemistry
node_lower_bound:
  A: 1.6
  B: 2.3
  F: 0.8
  GC: 26.9
  GM: 15.1
  V: 4.1
node_upper_bound:
  A: 325.6
  B: 788.8
  F: 96.6
  GC: 8724.8
  GM: 1684.0
  V: 160.1
ground_truth_effect_sizes:
  A->GC: 13.4753
  B->V: 0.1104
  F->GM: 10.7348
  GC->B: 0.0181
  GC->V: 0.0068
  GM->A: 0.0674
  GM->B: 0.1104
  GM->V: 0.0436
phenomenon_overview: You are going to identify internal dynamics of a
phenomena, Cachexia.
  Cachexia is a complicated metabolic syndrome related to underlying illness and
characterized
  by muscle mass loss with or without fat mass loss that is often associated with
  anorexia, an inflammatory process, insulin resistance, and increased protein
turnover.
include_parent_relationships: false
```

Figure 1. An example configuration of plug-and-play DAG yaml for our open-source framework.

B.2. Example Output Response from an LLM to a Prompt

Please refer to Figure 2.

C. Benchmarking Framework Details

C.1. Algorithm Details

Algorithm 1 provides details of the main parts of the framework.

C.2. Prompt Example for Parent-Child Elicitation

An example is shown in Figure 3.

C.3. The iterative hard-constraint loop overview

In addition to the Algorithm 2, here we provide Figure 4 to show a concrete example of (i) how feedback on failure is incooperated to the next iteration's prompt, and (ii) an example of iterative refinement history from our system's log. As in

Algorithm 1 Linear-LLM-SCM Benchmarking Framework

Require: DAG structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, Variable descriptions $\mathcal{D}$, Value ranges $\mathcal{R}$, Phenomenon overview $\mathcal{P}$, LLM $\mathcal{M}$
Ensure: Aggregated set of linear coefficients $\hat{\boldsymbol{\beta}}$

Initialize $\hat{\boldsymbol{\beta}} \leftarrow \emptyset$
for each target variable $X_j \in \mathcal{V}$ in topological order **do**
  1. Identify direct parents $Pa(X_j) \subset \mathcal{V}$ using edges $\mathcal{E}$ (decompose into local structures)
  2. Construct prompt S_j incorporating:
    - Domain expert persona and Phenomenon overview $\mathcal{P}$
    - Short descriptions $\mathcal{D}$ and Units for X_j and $Pa(X_j)$
    - Hard constraints/Value ranges $\mathcal{R}$
  3. Call LLM: $Response \leftarrow \mathcal{M}(S_j)$ requesting JSON format
  4. Parse $Response$: Extract numerical $\hat{\beta}$ for $Pa(X_j)$ and intercept β_0 (with iterative feedback, see Algorithm 2)
  5. $\hat{\boldsymbol{\beta}} \leftarrow \hat{\boldsymbol{\beta}} \cup \{\hat{\beta}_{i,j}\}$
end for
return aggregated parameterized SCM $\hat{\boldsymbol{\beta}}$ for graph structure $\mathcal{G}$

Algorithm 2 Iterative Feedback for Refinement with Hard Constraints

Require: Target variable X_j, Parent hard constraints $R_{Pa(X_j)}$, Node hard constraints C_2, Loop budget n
Ensure: Accepted linear coefficients $\hat{\beta}$

for *iteration* $= 1$ **to** n **do**
  Call LLM $\mathcal{M}$ with prompt S_j to get proposal P (includes parameterization $\hat{\beta}$)
  Calculate possible value range C_1 of X_j based on $R_{Pa(X_j)}$ and P
  if C_2 includes C_1 **then**
    return $\hat{\beta}$ (accept proposal)
  else
    Reject proposal
    Update prompt S_j with previous proposal and validation results
  end if
end for
return last available $\hat{\beta}$ (budget n reached)

(i) example, the previously proposed but failed proposal and its failure reason is inserted at the last position of the standard prompt template in the next iteration. Please note that, if the proposal fails multiple times, only the last proposal was provided to the next iteration. The example (ii) demonstrates an example, where a variable *lnamax* violated constraint in the first proposal, iteratively refineing the proposal at the end after this process.

C.4. The iterative hard-constraint loop details

In our parameterization, the LLM aims to obtain the following structural equation for each node. In practice, while our system prompts up to the noise ε, but we do not utilize or examine the values, which is future work.

$$Y = \beta_0 + \sum_{i=1}^{n} \beta_i X_i + \varepsilon \tag{1}$$

Given the elicited coefficients $\beta_0, \beta_1, \ldots, \beta_n$ and the known bounds $[lb_i,\, ub_i]$ for each parent variable X_i, the validator computes the worst-case deterministic range $[\hat{Y}_{\min},\, \hat{Y}_{\max}]$ of Y using interval arithmetic.

C.4.1. Initialization

Both extremes are initialised with the intercept:

$$\hat{Y}_{\min} = \beta_0, \qquad \hat{Y}_{\max} = \beta_0 \tag{2}$$

C.4.2. PARENT CONTRIBUTION

For each parent variable X_i with coefficient β_i and bounds $[lb_i,\ ub_i]$:

$$\hat{Y}_{\min} \mathrel{+}= \begin{cases} \beta_i \cdot lb_i & \text{if } \beta_i \geq 0 \\ \beta_i \cdot ub_i & \text{if } \beta_i < 0 \end{cases} \qquad \hat{Y}_{\max} \mathrel{+}= \begin{cases} \beta_i \cdot ub_i & \text{if } \beta_i \geq 0 \\ \beta_i \cdot lb_i & \text{if } \beta_i < 0 \end{cases} \tag{3}$$

C.4.3. VALIDATION CHECK

Let $[lb_Y,\ ub_Y]$ be the hard bounds on the target variable Y. The proposed equation *passes* validation if and only if:

$$lb_Y \leq \hat{Y}_{\min} \quad \text{and} \quad \hat{Y}_{\max} \leq ub_Y \tag{4}$$

If either condition is violated the validator returns a human-readable failure message, e.g. *"Predicted minimum (-0.30) for 'Y' is below its hard lower bound (0.01)."*

The noise term ε is excluded from the interval calculation because at this point, we do not know how reliably LLMs can perform uncertainty quantification in a symbolic fashion, which is future work.

D. Mathematical Definitions for Our Metrics

These metrics capture distances between LLM-elicited parameter vectors per node $\boldsymbol{\beta}_{LLM,j}$ and ground truth vectors per node $\boldsymbol{\beta}_{GT,j}$. For each vector, we denote j's each direct parents' edges effect size (scalar) using index i: $\beta_{LLM,j,i}$ and $\beta_{GT,j,i}$. The metrics are summarized as follows:

- **(M1) L2 Norm distance** between LLM-elicited vs GT:

$$M_1 = \sqrt{\sum_{j \in V} \sum_{i \in Pa(j)} (\beta_{LLM,j,i} - \beta_{GT,j,i})^2} \tag{5}$$

- **(M2) L2 Norm distance** with node-wise effect normalization:

$$M_2 = \sqrt{\sum_{j \in V} \sum_{i \in Pa(j)} \left(\frac{\beta_{LLM,j,i}}{\|\boldsymbol{\beta}_{LLM,j}\|_2} - \frac{\beta_{GT,j,i}}{\|\boldsymbol{\beta}_{GT,j}\|_2} \right)^2} \tag{6}$$

- **(M3) L2 Norm distance** excluding edges with single parents:

$$M_3 = \sqrt{\sum_{j \in \{V : |Pa(X_j)| > 1\}} \sum_{i \in Pa(X_j)} \left(\frac{\beta_{LLM,j,i}}{\|\boldsymbol{\beta}_{LLM,j}\|_2} - \frac{\beta_{GT,j,i}}{\|\boldsymbol{\beta}_{GT,j}\|_2} \right)^2} \tag{7}$$

- **(M4) Effect size relative ordering** matches per target variable:

$$M_4 = \sum_{j \in \{V : |Pa(X_j)| > 1\}} \mathbb{I}\left(\text{ordering of } \{\beta_{\text{LLM},i,j}\}_{i \in Pa(X_j)} = \text{ordering of } \{\beta_{\text{GT},i,j}\}_{i \in Pa(X_j)}\right) \tag{8}$$

(M4) captures relative effect-size ordering per node. If a node j has parents a and b and the effect sizes are $\beta_{LLM,j,a} = -0.8 < \beta_{LLM,j,b} = 0.5$ while $\beta_{GT,j,a} = -2 < \beta_{GT,j,b} = 3$, for this node we increment the sum by one before moving to the next node, finally obtaining (M4).

E. Experimental Details

E.1. LLM Model Details

Please refer to Table 3.

Table 3. Benchmark model specifications for our study. MoE stands for mixture of experts. Dense means dense-transformer and this also indicates the model is non-MoE-based. The tick ✓indicates yes, and the cross × indicates no.

MODEL NAME	MODEL SIZE	ARCHITECTURE	OPEN WEIGHTS
Gemini 2.5 Flash	Unknown	MoE	×
Llama 3.1 8B	8B	Dense	✓
Llama 3.3 70B	70B	Dense	✓
GPT-5.4	Unknown	Unknown	×

E.2. LLM Interaction Protocol

While the full experimental pipeline is publicly available and open-source in [3], here we describe key details for reproducibility.

In all models, we set temperature to zero while we did not change top k / top p sampling because generally when setting temperature, it is recommended to fall back to default top k and top p values. For Groq inference API, there is an option to explicitly set seed, but we do not explicitly set this value. For Gemini API, there is also a similar entry to specify the seed but we did not explicitly set this value. For both providers, setting this value does not guarantee actual seed being set to the specified value because they describe this as best-effort [4] [5], and non-deterministic also seem issues acknowledged by the model providers. For OpenAI GPT-5.4 models, we set `reasoning_effort` to `none` to enable setting temperature to zero.

For Gemini models, via Google AI Studio's APIs. For Llama models via Groq APIs. All the APIs are called via a python package called instructor [6]. Rate limiting was handled using the instructor package with its default exponential backoff. The JSON format was also enforced via the instructor package.

F. DAG Descriptions

F.1. DAGs used for experiment

Please refer to Table 4.

F.2. DAGs selection flowchart

Please refer to Figure 5.

[3] [URL reducted for double-blind review]
[4] https://console.groq.com/docs/api-reference#chat-create
[5] https://docs.cloud.google.com/vertex-ai/generative-ai/docs/model-reference/inference
[6] https://python.useinstructor.com/concepts/validation/#3-custom-validators

Table 4. DAGs with ground-truths included for our study from BnRep repository. **Literature** indicates sources introducing either the DAG structure or DAG effect parameterization. **VR.** (Value Ranges) indicates whether value ranges are available in the original literature.

NAME	LITERATURE	VR.	NODES	DOMAIN
cachexia1	Görgen & Leonelli (2020); Eisner et al. (2011)	Y	6	Genetics
expenditure	Tsagris (2022); Greene (2003)	N	12	Economics
foodsecurity	Leonelli et al. (2020); Barons et al. (2018)	Y	4	Social Sciences
algal2	Jackson-Blake et al. (2022)	Y	9	Env. Science
lexical	Baumann & Sekanina (2022)	Y	8	Social Sciences
liquefaction	Hu et al. (2023)	Y	10	Earth Sciences
stocks	Sener & Demir (2024)	Y	13	Economics

Table 5. Full direct estimation results (Averaged $n = 25$, Temp 0) with 95% CIs. $M1$: L2 distance; $M2$: normalized L2; $M3$: normalized L2 excluding single-parent edges; $M4$: Effect relative order count. For $M1$–$M3$, lower is better (↓); for $M4$, higher is better (↑). See Table 4 for DAG descriptions.

MODEL	DAG	M1 ↓	M2 ↓	M3 ↓	M4 ↑
Gemini 2.5 Flash	cachexia1	**13.783** $\pm$ 1.826	2.449 $\pm$ 0.240	1.074 $\pm$ 0.114	**1.000** $\pm$ 0.000
Llama 3.1 8B	cachexia1	16.527 $\pm$ 0.000	**1.036** $\pm$ 0.000	1.036 $\pm$ 0.000	0.000 $\pm$ 0.000
Llama 3.3 70B	cachexia1	13.843 $\pm$ 0.938	1.994 $\pm$ 0.213	1.352 $\pm$ 0.071	0.440 $\pm$ 0.199
GPT-5.4	cachexia1	17.043 $\pm$ 0.069	1.479 $\pm$ 0.295	**0.976** $\pm$ 0.053	0.640 $\pm$ 0.192
Gemini 2.5 Flash	expenditure	148084.559 $\pm$ 22359.281	**0.998** $\pm$ 0.226	**0.998** $\pm$ 0.226	**7.520** $\pm$ 0.200
Llama 3.1 8B	expenditure	**2463.377** $\pm$ 0.000	2.053 $\pm$ 0.000	2.053 $\pm$ 0.000	7.000 $\pm$ 0.000
Llama 3.3 70B	expenditure	27137.540 $\pm$ 12574.354	1.548 $\pm$ 0.234	1.548 $\pm$ 0.234	6.560 $\pm$ 0.199
GPT-5.4	expenditure	83376.520 $\pm$ 26482.403	1.960 $\pm$ 0.076	1.960 $\pm$ 0.076	6.360 $\pm$ 0.192
Gemini 2.5 Flash	foodsecurity	22.801 $\pm$ 0.038	2.017 $\pm$ 0.006	0.236 $\pm$ 0.050	0.500 $\pm$ 0.214
Llama 3.1 8B	foodsecurity	22.727 $\pm$ 0.006	**0.447** $\pm$ 0.000	0.447 $\pm$ 0.000	0.000 $\pm$ 0.000
Llama 3.3 70B	foodsecurity	22.914 $\pm$ 0.009	2.039 $\pm$ 0.001	0.395 $\pm$ 0.004	0.000 $\pm$ 0.000
GPT-5.4	foodsecurity	**22.587** $\pm$ 0.044	1.248 $\pm$ 0.371	**0.110** $\pm$ 0.028	**0.960** $\pm$ 0.078
Gemini 2.5 Flash	algal2	**4.094** $\pm$ 0.256	**0.514** $\pm$ 0.098	**0.514** $\pm$ 0.098	**2.000** $\pm$ 0.000
Llama 3.1 8B	algal2	*Model output equations not parsable by the program.*			
Llama 3.3 70B	algal2	4.657 $\pm$ 0.004	0.559 $\pm$ 0.054	0.559 $\pm$ 0.054	**2.000** $\pm$ 0.000
GPT-5.4	algal2	4.376 $\pm$ 0.058	0.523 $\pm$ 0.039	0.523 $\pm$ 0.039	**2.000** $\pm$ 0.000
Gemini 2.5 Flash	lexical	42.246 $\pm$ 0.650	2.036 $\pm$ 0.053	2.036 $\pm$ 0.053	2.040 $\pm$ 0.265
Llama 3.1 8B	lexical	35.854 $\pm$ 0.000	3.252 $\pm$ 0.000	2.565 $\pm$ 0.000	1.000 $\pm$ 0.000
Llama 3.3 70B	lexical	35.847 $\pm$ 0.001	2.274 $\pm$ 0.037	2.274 $\pm$ 0.037	0.778 $\pm$ 0.374
GPT-5.4	lexical	**35.122** $\pm$ 2.235	**1.759** $\pm$ 0.072	**1.759** $\pm$ 0.072	**2.240** $\pm$ 0.205
Gemini 2.5 Flash	liquefaction	12.294 $\pm$ 5.556	**0.844** $\pm$ 0.034	**0.844** $\pm$ 0.034	2.800 $\pm$ 0.160
Llama 3.1 8B	liquefaction	9.999 $\pm$ 0.000	2.012 $\pm$ 0.002	2.012 $\pm$ 0.002	**3.000** $\pm$ 0.000
Llama 3.3 70B	liquefaction	12.093 $\pm$ 0.221	1.271 $\pm$ 0.026	1.271 $\pm$ 0.026	**3.000** $\pm$ 0.000
GPT-5.4	liquefaction	**4.400** $\pm$ 0.453	0.881 $\pm$ 0.022	0.881 $\pm$ 0.022	2.960 $\pm$ 0.078
Gemini 2.5 Flash	stocks	0.829 $\pm$ 0.048	1.059 $\pm$ 0.122	1.006 $\pm$ 0.063	3.478 $\pm$ 0.242
Llama 3.1 8B	stocks	0.893 $\pm$ 0.005	1.589 $\pm$ 0.003	1.589 $\pm$ 0.003	2.880 $\pm$ 0.130
Llama 3.3 70B	stocks	1.060 $\pm$ 0.026	**0.934** $\pm$ 0.059	**0.934** $\pm$ 0.059	2.600 $\pm$ 0.226
GPT-5.4	stocks	**0.566** $\pm$ 0.014	0.936 $\pm$ 0.021	0.936 $\pm$ 0.021	**3.680** $\pm$ 0.187

F.3. Samples of DAGs Used in the Experiments

One may easily find DAGs we used in our experiments from references, but to save time of the readers, we attached the two example DAGs, Figure 6 and Figure 7.

F.4. DAG misspecification samplers

For the DAG misspecification experiment in the main text, the four mutated variants of the original Expenditure DAG add the following spurious edges: (S1) *Owner* → *Expenditure*, (S2) *Majorcards* → *Dependents*, (S3) *Owner* → *Share*, and (S4) *Majorcards* → *Selfemp*.

G. Main Experimental Results Details

Table 5 reports the full direct-estimation results ($M1$–$M4$) corresponding to the simplified main-text Table 1.

Table 6 reports the full unit-tweaking robustness results referenced in the main text.

Table 7 reports the full DAG-misspecification robustness results ($M1$–$M4$) corresponding to the simplified main-text Table 2.

Table 6. Robustness under unit tweak (Averaged $n = 25$, Temp 0) with 95% CIs. $M1$: L2 distance; $M2$: normalized L2; $M3$: normalized L2 excluding single-parent edges; $M4$: Effect relative order count. For $M1$–$M3$, lower is better (↓); for $M4$, higher is better (↑). Units: (L)=μM, (T)=nM.

MODEL	CND.	UNITS	M1 ↓	M2 ↓	M3 ↓	M4 ↑
Gemini 2.5 Flash	A	L	13.783 ± 1.826	2.449 ± 0.240	1.074 ± 0.114	1.000 ± 0.000
	B	T	**12.399** ± 1.774	**1.769** ± 0.372	**0.917** ± 0.136	**1.120** ± 0.206
Llama 3.1 8B	A	L	**16.527** ± 0.000	1.036 ± 0.000	1.036 ± 0.000	**0.000** ± 0.000
	B	T	17.198 ± 0.000	**0.905** ± 0.002	**0.905** ± 0.002	**0.000** ± 0.000
Llama 3.3 70B	A	L	13.843 ± 0.938	**1.994** ± 0.213	1.352 ± 0.071	0.440 ± 0.199
	B	T	**13.441** ± 0.199	2.439 ± 0.160	**1.107** ± 0.115	**0.640** ± 0.250
GPT-5.4	A	L	17.043 ± 0.069	1.479 ± 0.295	0.976 ± 0.053	0.640 ± 0.192
	B	T	**8.086** ± 0.763	**0.952** ± 0.129	**0.903** ± 0.073	**1.000** ± 0.000

H. Prompt Sensitivity Analysis for Structural Awareness

In our framework, prompt has the highest degree of freedom. While it is costly to try all the ideas, because LLMs are very sensitive prompts, for example, even new lines or separators affects the next token prediction (Sclar et al., 2024), it makes sense to revisit our prompts. For our additional experiments, due to budget constraints, we only performed experiments with three models, Gemini 2.5 Flash, Llama 3.1 8B and Llama 3.3 70B.

Local structure awareness in prompt:

The default prompt we introduced is a local parent-child set with only visibility on parent-child effect relationship, which is miopic structure awareness. We also devised another variant of prompt, "parent-parent-effect informed", which includes parent-parent relationship in addition to the existing effect information in prompts. Figure 8 visualizes a simple example of (i) the standard prompt and (ii) parent-parent effect informed prompt. Regardless of this additional information, in the Linear SCM formulation, the direct effect should be the same in theory. With this prompt with slightly less miopic example, we would like to see the reaction of LLMs in terms of direct effect estimation.

We summarized the result for parent-parent-effect informed prompt in Table 8. Overall, compared to the standard prompt, in most cases, parent-parent-effect informed prompt yielded worse performances, justifying the standard prompt we used in our main experiments.

Table 7. Full robustness under DAG misspecification for the expenditure DAG (Averaged $n = 25$, Temp 0) with 95% CIs. $M1$: L2 distance; $M2$: normalized L2; $M3$: normalized L2 excluding single-parent edges; $M4$: Effect relative order count. For $M1$–$M3$, lower is better (↓); for $M4$, higher is better (↑). Misspec: (O)=Original DAG, (S1)-(S4)=Spurious edges added.

MODEL	MISSPEC.	M1 ↓	M2 ↓	M3 ↓	M4 ↑
Gemini 2.5 Flash	O	148084.559 ± 22359.281	$\mathbf{0.998} \pm 0.226$	$\mathbf{0.998} \pm 0.226$	$\mathbf{7.520} \pm 0.200$
	S1	$\mathbf{81587.682} \pm 20626.822$	1.372 ± 0.282	1.372 ± 0.282	6.200 ± 0.299
	S2	138770.541 ± 21002.249	1.235 ± 0.267	1.235 ± 0.267	6.400 ± 0.253
	S3	120240.669 ± 20548.802	1.355 ± 0.253	1.355 ± 0.253	6.560 ± 0.255
	S4	106012.716 ± 23004.372	1.988 ± 0.189	1.988 ± 0.189	7.320 ± 0.271
Llama 3.1 8B	O	$\mathbf{2463.377} \pm 0.000$	$\mathbf{2.053} \pm 0.000$	$\mathbf{2.053} \pm 0.000$	$\mathbf{7.000} \pm 0.000$
	S1	20151.134 ± 0.000	2.483 ± 0.000	2.483 ± 0.000	6.000 ± 0.000
	S2	$\mathbf{2463.377} \pm 0.000$	2.063 ± 0.001	2.063 ± 0.001	6.000 ± 0.000
	S3	$\mathbf{2463.377} \pm 0.000$	2.106 ± 0.000	2.106 ± 0.000	6.000 ± 0.000
	S4	$\mathbf{2463.377} \pm 0.000$	2.413 ± 0.000	2.413 ± 0.000	$\mathbf{7.000} \pm 0.000$
Llama 3.3 70B	O	27137.540 ± 12574.354	$\mathbf{1.548} \pm 0.234$	$\mathbf{1.548} \pm 0.234$	6.560 ± 0.199
	S1	25843.000 ± 13086.141	1.703 ± 0.217	1.703 ± 0.217	5.520 ± 0.230
	S2	29202.638 ± 13299.788	1.918 ± 0.186	1.918 ± 0.186	5.920 ± 0.251
	S3	24257.613 ± 11979.590	2.038 ± 0.118	2.038 ± 0.118	5.360 ± 0.192
	S4	$\mathbf{12524.332} \pm 7698.439$	2.222 ± 0.138	2.222 ± 0.138	$\mathbf{6.680} \pm 0.271$
GPT-5.4	O	$\mathbf{83376.520} \pm 26482.403$	1.960 ± 0.076	1.960 ± 0.076	6.360 ± 0.192
	S1	132030.159 ± 25271.455	$\mathbf{0.902} \pm 0.025$	$\mathbf{0.902} \pm 0.025$	6.000 ± 0.000
	S2	107136.488 ± 28206.749	1.020 ± 0.077	1.020 ± 0.077	6.000 ± 0.000
	S3	91970.954 ± 28957.569	1.010 ± 0.095	1.010 ± 0.095	6.120 ± 0.130
	S4	100496.485 ± 26558.895	1.738 ± 0.085	1.738 ± 0.085	$\mathbf{6.960} \pm 0.078$

Table 8. Direct estimation results with parent-parent effect prompting (Averaged $n = 25$, Temp 0). Values after ± indicate 95% CIs. $M1$: L2 distance; $M2$: normalized L2; $M3$: normalized L2 excluding single-parent edges; $M4$: Effect relative order count. For $M1$–$M3$, lower is better (↓); for $M4$, higher is better (↑). See Table 4 for DAG descriptions.

MODEL	DAG	M1 ↓	M2 ↓	M3 ↓	M4 ↑
Gemini 2.5 Flash	cachexia	$\mathbf{13.100} \pm 2.247$	2.059 ± 0.339	$\mathbf{0.803} \pm 0.224$	$\mathbf{1.440} \pm 0.199$
Llama 3.1 8B	cachexia	16.527 ± 0.000	$\mathbf{1.047} \pm 0.000$	1.047 ± 0.000	0.000 ± 0.000
Llama 3.3 70B	cachexia	14.424 ± 0.786	1.774 ± 0.262	1.200 ± 0.092	0.400 ± 0.196
Gemini 2.5 Flash	expenditure	125421.804 ± 24860.693	1.729 ± 0.216	1.729 ± 0.216	6.560 ± 0.279
Llama 3.1 8B	expenditure	$\mathbf{2463.302} \pm 0.000$	2.099 ± 0.001	2.099 ± 0.001	6.000 ± 0.000
Llama 3.3 70B	expenditure	19137.678 ± 10242.559	$\mathbf{1.711} \pm 0.208$	$\mathbf{1.711} \pm 0.208$	$\mathbf{6.680} \pm 0.246$
Gemini 2.5 Flash	foodsecurity	22.852 ± 0.035	2.049 ± 0.011	$\mathbf{0.422} \pm 0.058$	$\mathbf{0.080} \pm 0.108$
Llama 3.1 8B	foodsecurity	$\mathbf{22.700} \pm 0.029$	$\mathbf{0.628} \pm 0.013$	0.628 ± 0.013	0.000 ± 0.000
Llama 3.3 70B	foodsecurity	22.982 ± 0.025	2.049 ± 0.000	0.447 ± 0.000	0.000 ± 0.000
Gemini 2.5 Flash	algal2	$\mathbf{4.203} \pm 0.307$	$\mathbf{0.572} \pm 0.172$	$\mathbf{0.572} \pm 0.172$	$\mathbf{2.000} \pm 0.000$
Llama 3.1 8B	algal2	4.961 ± 0.000	1.079 ± 0.001	1.079 ± 0.001	$\mathbf{2.000} \pm 0.000$
Llama 3.3 70B	algal2	4.635 ± 0.039	0.614 ± 0.036	0.614 ± 0.036	$\mathbf{2.000} \pm 0.000$
Gemini 2.5 Flash	lexical	41.733 ± 2.350	$\mathbf{1.963} \pm 0.061$	$\mathbf{1.963} \pm 0.061$	$\mathbf{2.480} \pm 0.230$
Llama 3.1 8B	lexical	$\mathbf{1.609} \pm 0.000$	2.923 ± 0.000	2.131 ± 0.000	2.000 ± 0.000
Llama 3.3 70B	lexical	10.442 ± 0.023	2.232 ± 0.053	2.232 ± 0.053	2.040 ± 0.211
Gemini 2.5 Flash	liquefaction	13.087 ± 4.536	$\mathbf{0.850} \pm 0.028$	$\mathbf{0.850} \pm 0.028$	2.840 ± 0.147
Llama 3.1 8B	liquefaction	$\mathbf{9.998} \pm 0.000$	0.985 ± 0.002	0.985 ± 0.002	$\mathbf{3.000} \pm 0.000$
Llama 3.3 70B	liquefaction	11.745 ± 0.079	1.302 ± 0.036	1.302 ± 0.036	$\mathbf{3.000} \pm 0.000$
Gemini 2.5 Flash	stocks	$\mathbf{0.786} \pm 0.043$	1.194 ± 0.111	1.145 ± 0.070	3.040 ± 0.309
Llama 3.1 8B	stocks	0.831 ± 0.000	0.945 ± 0.001	0.945 ± 0.001	$\mathbf{3.960} \pm 0.078$
Llama 3.3 70B	stocks	1.060 ± 0.023	$\mathbf{0.935} \pm 0.037$	$\mathbf{0.935} \pm 0.037$	2.680 ± 0.187

```
*****ELICITATION PROMPT*****
You are a leading Finance researcher and an Structural Causal Model (SCM) expert in Consumer Behavior.
This dataset focuses on factors influencing credit card behavior and expenditure patterns, providing insights into consumer finance
decisions.
Given the direct causes, you must propose a linear structural equation for the target variable $Y$. Do not use non-linear functions (e.g.,
exponential, sigmoid). The coefficients of the linear equation are continuous variables in space $\mathbb{R}$.
The target variable is $Y =$ "Income": "The yearly income in $10,000 (Unit: 10,000 USD).".
The direct causes (Parents) are: "Age": "The age in years plus twelfths of a year (Unit: Years + fractions).", "Owner": "Whether the person
owns their home or they rent (Categorical/Binary).".
Propose the complete linear equation: $Y = \beta_0*1 + \beta_Age*Age + \beta_Owner*Owner + E_Y$. You *must* use the actual raw
variable names (e.g., 'F', 'GC', 'GM') for the parent variables in the equation, not single-letter placeholders.

The following hard constraints (value ranges) are known and must be respected:
- Variable Income is bounded within [0.0, 500.0].
- Variable Age is bounded within [18, 100].
- Variable Owner is bounded within [0, 1].

1.  Explicitly define the error term $E_Y$ (e.g., standard normal noise, $E_Y \sim N(0, \sigma^2)$).

2.  For each coefficient (\beta_0, \beta_Age, \beta_Owner), explain its finance plausibility, its expected sign (positive/negative), and justify
your chosen magnitude (unit-effect).

Given the DAG for variable "Income", please provide a plausible linear parameterisation.

Output format: Only respond in JSON format, with the following keys:
- plausibility: str (Finance Plausibility)
- proposed_lin_str_eq: str (Proposed Linear Structural Equation, *do not use placeholder betas like \\beta_0, use concrete numerical
values*)

[Run LLM Elicitation] Proposed Linear String Equation (early validation):
Income = -10 * 1 + 1.5 * Age + 8 * Owner + E_Y, where E_Y ~ N(0, 15^2)
*****MODEL (parsed by instructor)*****
{
  "plausibility": "The proposed linear structural equation for Income is designed based on established financial and economic principles
relating age and homeownership to earning potential. The intercept $\\beta_0 = -10$ provides a baseline adjustment; while it implies a
hypothetical negative income at age zero, it ensures that for the practical age range (18-100), the base income levels are realistic and
positive. The coefficient for Age ($\\beta_{Age} = 1.5$) is positive, reflecting the common trend of increasing income with accumulated
experience and career progression over time. A value of 1.5 suggests an average increase of $15,000 USD per year, which is a plausible
rate of income growth in many careers. The coefficient for Owner ($\\beta_{Owner} = 8$) is also positive, indicating that homeowners
generally have higher incomes than renters, controlling for age. This aligns with empirical observations where homeownership often serves
as an indicator of greater financial stability, wealth, and higher earning capacity. A difference of $80,000 USD between homeowners and
renters is substantial and reflects the significant financial disparities associated with housing status. The error term $E_Y \\sim N(0, 15^2)$
accounts for all unobserved factors, such as education, industry, job performance, economic conditions, and other individual-specific
attributes, which contribute to the variability in income. A standard deviation of $150,000 USD reflects the considerable spread of income
levels not explained by age and homeownership alone, acknowledging the complexity of income determination in consumer finance.",
  "proposed_lin_str_eq": "Income = -10 * 1 + 1.5 * Age + 8 * Owner + E_Y, where E_Y ~ N(0, 15^2)"
}
```

Figure 2. An example prompt and its LLM response structure at the Expenditure DAG at *Income* target variable. The output can be found at the bottom as JSON structured format.

Persona
Phenomenon
Variables & units
(only parent-child)

You are a leading Finance researcher and an Structural Causal Model (SCM) expert in Consumer Behavior.
This dataset focuses on factors influencing credit card behavior and expenditure patterns, providing insights into consumer finance decisions.
Given the direct causes, you must propose a linear structural equation for the target variable Y. Do not use non-linear functions (e.g., exponential, sigmoid). The coefficients of the linear equation are continuous variables in space $\mathbb{R}$.
The target variable is $Y =$ "Share": "The ratio of monthly credit card expenditure to yearly income (Ratio).".
The direct causes (Parents) are: "Card": "Whether the application for credit card was accepted or not (Categorical/Binary).", "Age": "The age in years plus twelfths of a year (Unit: Years + fractions).".

Task instruction
Range constraints
(only parent-child)

Propose the complete linear equation: $Y = \beta_0*1 + \beta_Card*Card + \beta_Age*Age + E_Y$. You *must* use the actual raw variable names (e.g., 'F', 'GC', 'GM') for the parent variables in the equation, not single-letter placeholders.

The following hard constraints (value ranges) are known and must be respected:
- Variable Share is bounded within [0.0, 1.0].
- Variable Card is bounded within [0, 1].
- Variable Age is bounded within [18, 100].

Output format
instruction

1. Explicitly define the error term E_Y (e.g., standard normal noise, $E_Y \sim N(0, \sigma^2)$).

2. For each coefficient (\beta_0, \beta_Card, \beta_Age), explain its finance plausibility, its expected sign (positive/negative), and justify your chosen magnitude (unit-effect).

Given the DAG for variable "Share", please provide a plausible linear parameterisation.

Output format: Only respond in JSON format, with the following keys:
- plausibility: str (Finance Plausibility)
- proposed_lin_str_eq: str (Proposed Linear Structural Equation, *do not use placeholder betas like \\beta_0, use concrete numerical values*)

Figure 3. An example of a prompt for a local parent-child structure in a DAG.

(i) Example: Inserted iterative refinmenet instruction after validation failure to our prompt template

PREVIOUS ATTEMPT FEEDBACK
Your previous proposal failed validation with the following issue: [VALIDATOR] INCONSISTENCY for 'lnamax': Predicted maximum (0.86) for 'lnamax' is above its hard upper bound (0.836).
Proposed Equation: $lnamax = 0.18 + (-0.0005)*lnR + 0.075*Mw + E_Y, where E_Y ~ N(0, 0.02^2)$. Please adjust your new proposal to resolve this issue and respect all given constraints.
Output format: Only respond in JSON format, with the following keys:
- plausibility: str (Geotechnical engineering Plausibility)
- proposed_lin_str_eq: str (Proposed Linear Structural Equation, *do not use placeholder betas like \\beta_0, use concrete numerical values*)

(ii) Example: System's log for iterative refinement history. (This is not an input/output for LLM)

[SCENARIO 0] Iteration history for this scenario:
Iteration 1: Validated = False, Summary = [VALIDATOR] INCONSISTENCY for 'lnamax': Predicted maximum (1.27) for 'lnamax' is above its hard upper bound (0.836).
Proposed Equation: lnamax = 0.18 + (-0.0012)*lnR + 0.12*Mw + E_Y, where E_Y ~ N(0, 0.05^2)
Iteration 2: Validated = False, Summary = [VALIDATOR] INCONSISTENCY for 'lnamax': Predicted maximum (1.01) for 'lnamax' is above its hard upper bound (0.836).
Proposed Equation: lnamax = 0.10 + (-0.0008)*lnR + 0.10*Mw + E_Y, where E_Y ~ N(0, 0.03^2)
Iteration 3: Validated = False, Summary = [VALIDATOR] INCONSISTENCY for 'lnamax': Predicted maximum (0.86) for 'lnamax' is above its hard upper bound (0.836).
Proposed Equation: lnamax = 0.18 + (-0.0005)*lnR + 0.075*Mw + E_Y, where E_Y ~ N(0, 0.02^2)
Iteration 4: Validated = True, Summary = [VALIDATOR] CONSISTENT: Predicted range [0.43, 0.76] for 'lnamax' is within its hard constraints.
Proposed Equation: lnamax = 0.12 + (-0.0004)*lnR + 0.07*Mw + E_Y, E_Y ~ N(0, 0.015^2)

Figure 4. (i) Example of refinement instruction inserted to the next iteration's prompt and (ii) iterative refinement history over time from the system's log.

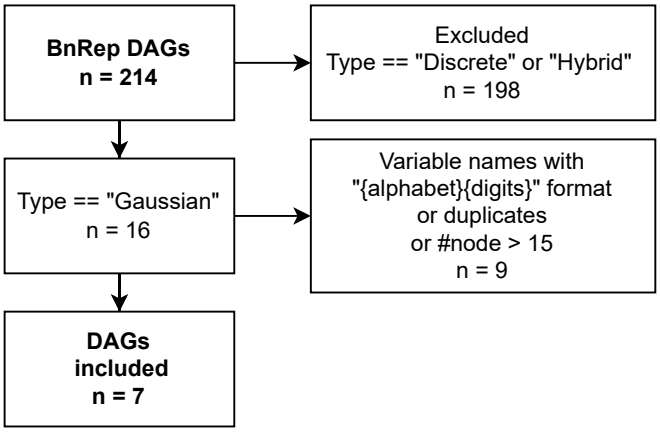

Figure 5. Inclusion and exclusion flowchart for DAG ground-truths from BnRep DAG repository

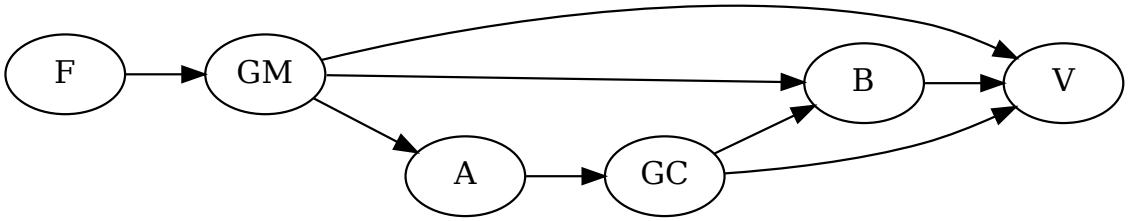

Figure 6. The DAG structure of cachexia1 from BnRep repository.

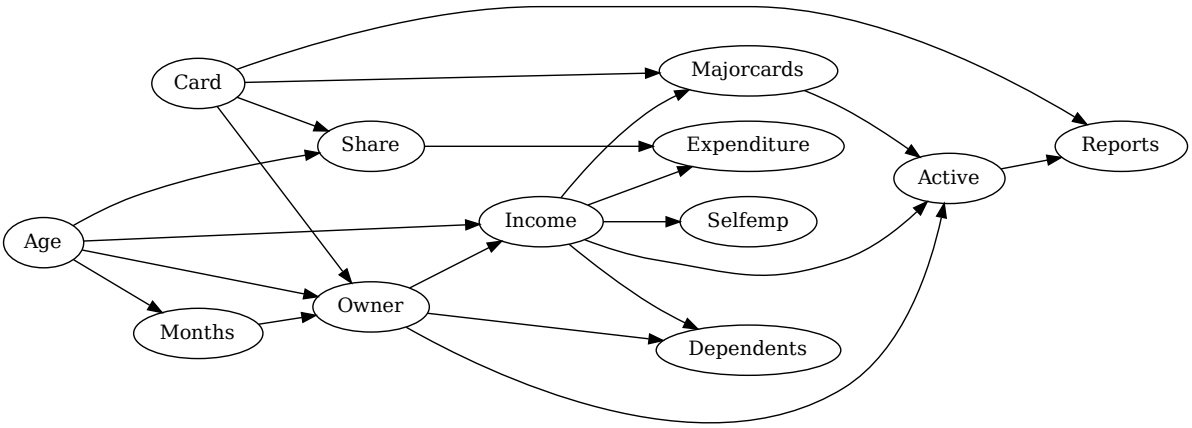

Figure 7. The DAG structure of expenditure from BnRep repository.

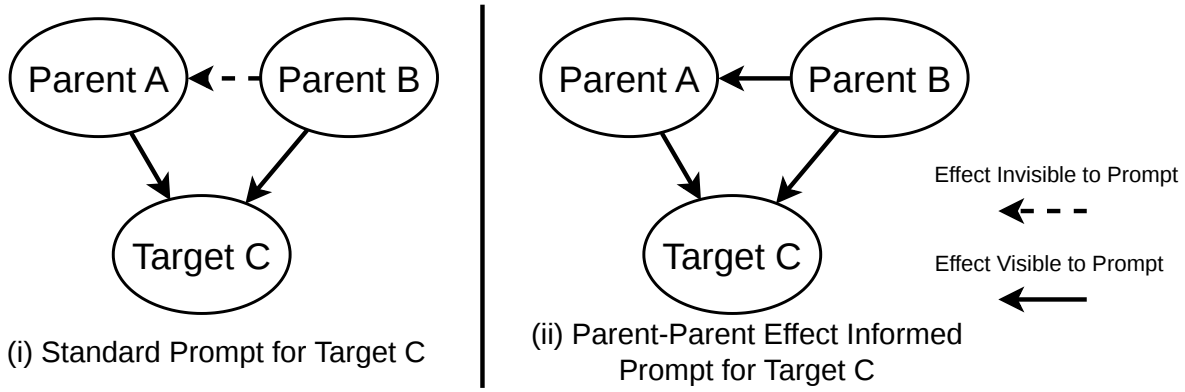

Figure 8. Comparison of (i) standard prompt and (ii) parent-parent effect informed with the simplest structural example where parents relate to each other.