IGDA: Interactive Graph Discovery through Large Language Model Agents

Anonymous ACL submission

Abstract

Large language models have emerged as a powerful tool for accelerating science and decision making. Towards further improving LLM utility in these domains we study the application of LLMs to the novel task of *interactive graph discovery*: given a ground truth graph G^* capturing variable relationships and a budget of I edge experiments over R rounds, minimize the distance between the predicted graph $\hat{G}_R$ and G^* at the end of the R-th round. To solve this task we propose **IGDA**, a LLM-based pipeline incorporating two key components: 1) an LLM uncertainty-driven method for edge experiment selection 2) a local graph update strategy utilizing binary feedback from experiments to improve predictions for unselected neighboring edges. Experiments on eight different real-world graphs show our approach often outperforms all baselines including a state-of-the-art numerical method for interactive graph discovery. Further, we conduct a rigorous series of ablations dissecting the impact of each pipeline component. Overall, our results show IGDA to be a powerful method for graph discovery complementary to existing numerically driven approaches.

1 Introduction

The research process can vary widely across different domains ranging from medicine to ML. One common phase shared between all disciplines is the *experimental design* process during which researchers read relevant literature and then propose high-priority experiments to carry out. Based on experimental outcomes researchers can update their understanding of the problem of interest, leading to future rounds of research and discovery.

We can formalize this process as the following *graph discovery* task: given a set of variables $X_1, ..., X_n$ find a graph G^* on the nodes $X_1, ..., X_n$ whose edges capture causal relationships between the *parent* (source) and *child* (target). Often, observational data can be collected for the variables $X_1, ..., X_n$. This data can then be used to predict an initial graph G_0 using statistical causal discovery techniques (Spirtes and Zhang, 2016). Recently, large language models (**LLMs**) have emerged as a competitive alternative method for predicting causal graphs (Kıcıman et al., 2024; Abdulaal et al., 2024; Chen et al., 2024). Unlike pre-existing statistical methods, LLMs require no observational data (Kıcıman et al., 2024), instead relying purely on semantic metadata such as variable names and descriptions. Another related line a work (Yang et al., 2024) investigates the abilities of LLMs to act as in-context black-box optimizers. Given an objective function f and an evaluation budget B, the LLM is tasked with finding a maximizer x^* of f by sequentially proposing queries $\{x_i\}_{i=1}^B$ and observing their associated values $\{f(x_i)\}_{i=1}^B$. Taken together, these directions suggest a powerful new application of LLMs: *interactive graph discovery*.

Given an initial predicted graph $\hat{G}_0$ and a series of experiment rounds $1, ..., R$, the interactive graph discovery problem involves minimizing some distance $d(\hat{G}_k, G^*)$ between the predicted graph $\hat{G}_k$ at round k and the true graph G^* (unknown to the learner) through a sequence of targeted experiments on edges $e = (X, Y)$ testing the effect of the parent variable X on the child variable Y. The edge experiment operation is kept purposefully abstract, requiring only that binary feedback be given indicating the presence or absence of an edge. In practice this operation can be implemented via any number of experimental procedures (e.g. via hard interventions in the formal causal sense (Pearl, 2009) or empirical methods such as randomized controlled trials (Sibbald and Roland, 1998)). The IGD problem setup captures the process researchers go through everyday when designing and prioritizing experiments, guided by their prior experience, to study numerous potential relationships

between any number of variables.

The interactive graph discovery problem requires the agent to solve two key sub-tasks:

1. **Experiment selection**: Selecting which edges (X_i, X_j) to target for experimentation in the next round.
2. **Graph updates**: Updating the predicted graph from $\hat{G}_{k-1}$ to $\hat{G}_k$ given binary feedback based on the outcome of the previous experiments.

We propose to solve this task with the Interactive Graph Discovery Agent (**IGDA**): a novel LLM agent uncertainty-driven approach as an alternative to existing statistical methods (Olko et al., 2024; Scherrer et al., 2022). While statistical models can work well in some settings, they crucially rely on the abundance of domain specific observational and interventional numerical data. For many problems, such data might be hard or impossible acquire. LLMs, however, potentially contain relevant latent knowledge derived from vast amounts of variable semantic metadata contained in their pre-training or internet corpora. Further, we find that, via a combination of broad background knowledge and reasoning abilities, advanced LLMs (Grattafiori et al., 2024) are capable of updating their predictions and confidences when presented with experimental feedback revealing unexpected relationships between a subset of edges. This makes LLM based approaches a powerful alternative to statistical methods when numerical data is not available.

In particular, IGDA predicts and maintains uncertainty estimates for each unknown edge $e \in \hat{G}_k$. Edges are then selected for experimentation by prioritizing those with the highest uncertainty. When feedback is received on the selected edges, pairwise-local updates on both edge predictions and uncertainty estimates are performed for each edge in $\hat{G}$ sharing a parent or child variable with an experimented edge. This process continues for R rounds with I edges selected for experimentation each round. We benchmark IGDA on eight real world graphs, finding uncertainty driven selection with local updates outperforms baselines. In summary, we make the following contributions:

- The interactive graph discovery problem as a novel setting for evaluating LLM capabilities.
- LLM-based uncertainty-guided edge experiment selection as a policy for prioritizing edge experimentation.
- A local update strategy for robustly updating the predicted graph G_k with binary experiment feedback.
- Ablations rigorously evaluating the contribution of each pipeline component and other discovery strategies.

2 Background and Related Work

LLMs as Agents Recently LLMs have been applied to across a variety of domains including math, science, coding, writing and more (Yao et al., 2023; DeepSeek-AI et al., 2025; Jiang et al., 2025; Veličković et al., 2024). For example, Tree of Thoughts (**ToT**) (Yao et al., 2023) applies LLMs to solve crosswords augmented with the ToT algorithm. Deepseek R1 DeepSeek-AI et al. (2025) deploys LLMs to solve hard math problems augmented with improved "thinking" abilities using RL. Veličković et al. (2024) combines LLMs with evolutionary algorithms to generate competitive and diverse solutions to competitive coding problems. These examples underscore the broad applicability of LLM capability supported by domain specfic algorithms.

Causal Discovery and LLMs. The causal discovery task involves learning causal relationships from observed empirical data (Peters et al., 2017; Spirtes and Zhang, 2016). Many proposed algorithms exist (Spirtes et al., 1993; Yu et al., 2019; Nauta et al., 2019; Zheng et al., 2018; Chickering, 2002) attempting to solve the causal discovery problem. However, these methods are known to struggle on real world graphs where observations are noisy or common structural assumptions are violated (Chevalley et al., 2023; Tu et al., 2019).

Recently, LLMs have emerged as an alternative approach to causal discovery (Kıcıman et al., 2024; Abdulaal et al., 2024; Vashishtha et al., 2023; Li et al., 2024; Lampinen et al., 2023). Kıcıman et al. (2024) first investigated the capability of LLMs to act as zero-shot causal discovery agents using only semantic information and pairwise prompting on each variable pair. Follow-up work (Abdulaal et al., 2024) further improves LLM predictions with observational data by selecting for predictions which maximize data likelihood. Vashishtha et al. (2023) utilize *triplet prompting* to prevent cycles when the causal graph is acyclic. They show only a topological ordering on variables is required for many common causal reasoning tasks (Chu et al., 2023). Other works (Zhou et al., 2024; Chen et al., 2024) benchmark LLMs across a range of causality related tasks including causal discovery and causal

inference confirming that LLMs struggle with integrating numerical data.

Another line of work more related to our proposed interactive causal discovery problem studies how to incorporate background knowledge into causal discovery algorithms (Meek, 2013). Define a set of *background knowledge* as the tuple $\mathcal{K} = (F, R)$, where F specifies a set of "forbidden" graph edges and R specifies a set of "required" graph edges. Meek (2013) presents an algorithm for constructing a causal graph consistent with $\mathcal{K}$ by leveraging an assumed structural directed acyclic graph (DAG) property. Building on Meek (2013), Chickering (2002) proposes a greedy search algorithm that performs well in practice.

Most related are statistical methods from the causal discovery literature which aim to efficiently choose a sequence of interventions to discover causal structure (Scherrer et al., 2022; Olko et al., 2024). In particular, Gradient based Interventional Targeting (**GIT**) (Olko et al., 2024) utilizes existing neural causal discovery methods (Lippe et al., 2022) to learn a distribution over possible graph structures and variable assignments. For each round of intervention, GIT prioritizes variables whose simulated interventional distribution have large gradient with respect to the structural training loss.

In contrast to these works, our proposed algorithm utilizes LLMs to reason about the semantic/physical, as opposed to formal/structural, relationships between variables and edges in causal graphs. For this reason we are not required to make any structural assumptions on an underlying DAG, as is common in the causal discovery literature. This is desirable as in practice many real-world causal graphs are cyclic and poorly structured (Zhu et al., 2024; Huang et al., 2021). Additionally our method does not rely on observational or interventional data for real world graphs which may be expensive to acquire but crucial for good performance with statistical methods. Further, we note we succeed at designing a competitive graph discovery agent despite the difficulty LLMs have understanding graph data when applied naively (Guo et al., 2023).

LLMs as Optimizers. Another growing line of work utilizes LLMs as black-box optimizers (Yang et al., 2024; Roohani et al., 2024). Yang et al. (2024) introduce the notion of an LLM as a generic optimizer and use it to optimize performance objectives stemming from a range of tasks including linear regression and mathematical word problems (Cobbe et al., 2021). Other works (Madaan et al., 2023; Havrilla et al., 2024) examine the self-refinement capabilities of LLMs where the LLM must reason and self-improve on earlier responses. A growing number of papers apply LLMs to optimal experiment design and discovery (Roohani et al., 2024; AI4Science and Quantum, 2023; Gao et al., 2024; Majumder et al., 2024; Jansen et al., 2024). Roohani et al. (2024) apply LLMs to gene discovery tasks which aim to find highly-influential parent genes affecting the regulation of a downstream target gene. Majumder et al. (2024); Jansen et al. (2024) both present benchmarks evaluating the ability of LLMs to perform real-world and synthesized discovery tasks.

3 Method

Setup. As input we are given a set of variables $X_1, ..., X_n$ with associated metadata including variable names and variable descriptions. We use the notation $Y \to X$ to indicate when variable Y has a direct effect on variable X and the set of parents of a variable X as $Pa(X) = \{X_i : X_i \to X\}$. We can then consider the directed ground truth graph $G^* = \{(X_i, X_j) : X_i \in Pa(X_j)\}$ with unlabeled and unweighted edges. The only assumed graph structure is simplicity i.e. no self-edges or multi-edges. No additional structure on the graph (such as acyclicity) is assumed. We can frame the prediction of G^* as an edge-wise binary classification problem over the complete graph K_n, where an edge (X_i, X_j) has the label $l_{ij} = 1$ if $X_i \to X_j$ and $l_{ij} = 0$ otherwise. G^* can then be written as a collection of ground truth labelings $G^* = \{(X_i, X_j, l_{ij}) : 1 \leq i \neq j \leq n\}$.

The *interactive graph discovery* task then aims to learn G^* by interacting with the discovery environment via *experiments* on each edge (X_i, X_j). We define an *experiment* on an edge (X_i, X_j) as an operation revealing the ground truth label $l_{i,j}$. This experiment operation is purposefully kept abstract for generality and could correspond to any number of real-world experimental experimental strategies including formal do operations (Pearl, 2009) or empirical randomized control trials (Sibbald and Roland, 1998). Interactive graph discovery then proceeds in two phases:

Phase 1 (Zero-shot prediction): Produce an initial graph prediction $\hat{G}_0$ using available

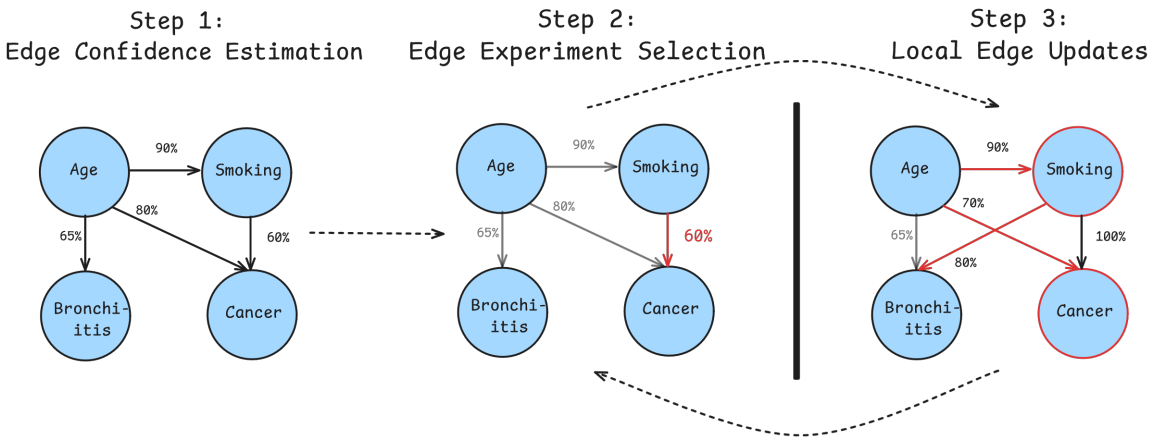

Figure 1: Diagram of the interactive graph discovery process through LLMs. The process begins by predicting edges and confidences for each edge. Interactive discovery then proceeds by selecting the most uncertain edges for experimentation. The LLM then updates its predictions and confidences for edges adjacent to the selected edge. Note: only edges predicted as present are shown.

variables $X_1, ..., X_n$ plus semantic metadata.

Phase 2 (Interactive Discovery): Over a series of R rounds, propose I edge experiments on (X_i, X_j) each round and receive binary feedback on l_{ij}. Use this to produce an updated prediction $\hat{G}_{r-1} \to \hat{G}_r$

We evaluate the accuracy of a prediction $\hat{G}$ using the F1 objective, i.e.

$$F1(G^*, \hat{G}) = \frac{2 \cdot \texttt{Precision}_{\hat{G}} \cdot \texttt{Recall}_{\hat{G}}}{\texttt{Precision}_{\hat{G}} + \texttt{Recall}_{\hat{G}}}$$

where $\texttt{Precision}_{\hat{G}}$ and $\texttt{Recall}_{\hat{G}}$ are computed with the label predictions $(X_i, X_j, \hat{l}_{ij}) \in \hat{G}$ and l_{ij} as ground truth. The goal of the interactive discovery process is then to maximize $F1(G^*, \hat{G}_R)$.

Method. Our proposed method IGDA begins by generating a zero-shot graph prediction $\hat{G}_0$. A prediction for each variable pair (X_i, X_j), $1 \leq i \neq j \leq n$, is generated by prompting an LLM to reason about $X_i \to X_j$ in a manner similar to the pairwise-prompting strategy utilized in Kıcıman et al. (2024). In addition, we prompt the LLM to reason about its confidence in the prediction and output a confidence score from 1 - 100. Section D shows the exact prompt used. To obtain a reliable confidence estimate we sample the LLM $K = 16$ times. We denote the initial confidence for (X_i, X_j) as c^0_{ij} and set it to be the (signed) average over $K = 16$ output confidences. The initial edge label l^0_{ij} is then taken as the boolean $l^0_{ij} = \mathbf{1}_{c^0_{ij} \geq 0}$. This gives us the initial prediction $\hat{G}_0$.

Next, in each experimentation round $r \leq R$, we sort the confidence scores $\{c^r_{ij} : 1 \leq i, j \leq n\}$ by absolute value and experiment on the I edges with the lowest absolute confidence (and highest uncertainty). This reveals the ground truth labels l_{ij} for for each experimented edge (X_i, X_j). Using this feedback, we update the predicted edge labels for experimented edges to $l^{r+1}_{ij} = l_{ij}$ and the confidences to $c^{r+1}_{ij} = 100$. Additionally, we prompt the LLM, conditioned on the ground truth label l_{ij}, to update its prediction and confidence for each edge (X_i, X_k) or (X_l, X_j), $1 \leq k, l \leq n$ which shares a node with (X_i, X_j) and has absolute confidence less than 100. We call each update to an edge (X_l, X_k) a *local update*. It may be that an edge (X_l, X_k) is adjacent to multiple experimented edges $(X_{i_1}, X_{j_1}), (X_{i_2}, X_{j_2})$ in a single round and thus receives multiple local updates. To manage these cases we set the next confidence c^{r+1}_{lk} to the (signed) average of all individual local updates to c^r_{lk}. Then we set $l^{r+1}_{lk} = \mathbf{1}_{c_{lk} \geq 0}$ as before. This continues until the final round R is reached.

We call the complete discovery pipeline the *Interactive Graph Discovery Agent* (IGDA). A diagram of the full pipeline is shown in Figure 1. We report all prompts in D.

4 Results

We evaluate our approach on seven real-world graphs. The graphs range in size from 8 to 30 nodes (variables) and vary widely in structure (some are acyclic while others are cyclic). Details for each graph can be found in Appendix E. To produce ini-

tial zero-shot graph predictions $\hat{G}_0$ for all graphs we utilize pairwise causal prompting as in Kıcıman et al. (2024) with `Meta-Llama-3-70B-Instruct` (Grattafiori et al., 2024) as the base LLM. We chose `Meta-Llama-3-70B-Instruct` as at the time of our experiments it was the best open-source model with advanced reasoning and instruction following capabilities For the interactive discovery phase we then initialize all methods using $\hat{G}_0$. We compare our method against several baselines:

Random selection: Starting from $\hat{G}_0$ we randomly select edges for experimentation. After receiving binary feedback we update incorrect predictions on experiment edges for the next round. We do not allow edges to be selected for experimentation twice.

Static confidence selection: We select edges for experimentation based on the initial confidence scores c_{ij}. No updates are performed beyond fixing incorrect predictions in the experimentation set.

Gradient-based Intervention Targeting (GIT): We adapt the statistical GIT method (Olko et al., 2024) by selecting the node at each round which has a) not already been selected and b) has the largest loss gradient under a neural causal model (Lippe et al., 2022) trained with all available observational and interventional training data. We initially train the model with 5000 observational datapoints sampled from the ground-truth graph. 100 additional interventional datapoints on the experiment node are sampled from the ground-truth graph and added to the training set after each round of experimentation.

`Meta-Llama-3-70B-Instruct` is used as the base LLM when applicable. To assess performance, we plot the mean F1 score, averaged over five independent runs, against the percentage of edges selected in each graph. Results are shown in Figure 2.

Uncertainty driven experiment selection with local updates performs best. Uncertainty driven experiment selection with the LLM utilizing experimental feedback for local updates performs best on nearly all graphs. Further, it outperforms the random selection baselines at nearly every round on every graph, at times by up to 0.5 absolute F1 score. The only exception to this is the Arctic sea ice graph where local updates initially perform poorly. We attribute this to the highly cyclic and thus harder-to-predict graph structure. Additionally, the method significantly outperforms the statistical GIT baseline on both Az and Covid graphs and remains competitive on the rest. Figure 3 plots the average rank of all methods over all timesteps, confirming IGDA's strong performance. Notably, even on graphs where the LLM proposes a poor zero-shot initial prediction, the LLM is able to recover quickly, converging to the correct structure with local updates. This suggests the LLM is able to effectively utilize experiment feedback even when lacking detailed domain knowledge.

Local updates can outperform random selection even with few experiments. Allowing the LLM to make local edge updates using experiment feedback quickly improves the predicted graph even when relatively few edges are selected. This behavior is particularly desirable, as in practice it may be expensive to experiment on even a small fraction of all edges. On some graphs, where the initial LLM confidence estimates are good, the static confidence selection baseline without local updates is also able to quickly outperform random selection. Yet, even when the initial confidence estimates are subpar, local updates compensate and allow for the prediction to quickly improve with just a few edge experiments. This again demonstrates the broad effectiveness of local updates even when initial predictions are poor.

Static uncertainty driven selection performs better than random selection. Despite not fully utilizing experimental feedback, static uncertainty driven selection still outperforms the random selection baseline on five out of seven graphs. This method performs particularly well on AZ and Covid graphs where the initial LLM predictions are already reasonably good. On these graphs static uncertainty selection quickly outperforms randomly selection and is competitive even with local updates. This shows that, on a subset of the graphs, the LLM's confidence in its predictions are well-calibrated, allowing our selection policy to prevent wasting experiments on edges which are most likely already correct. However, we also see the LLM's confidence estimates can be poorly calibrated on graphs for which the initial predictions are inaccurate. See for example the Asphyxia and Neuropathic pain graphs, which start with initial F1 score less than 0.2. On these graphs the static confidence

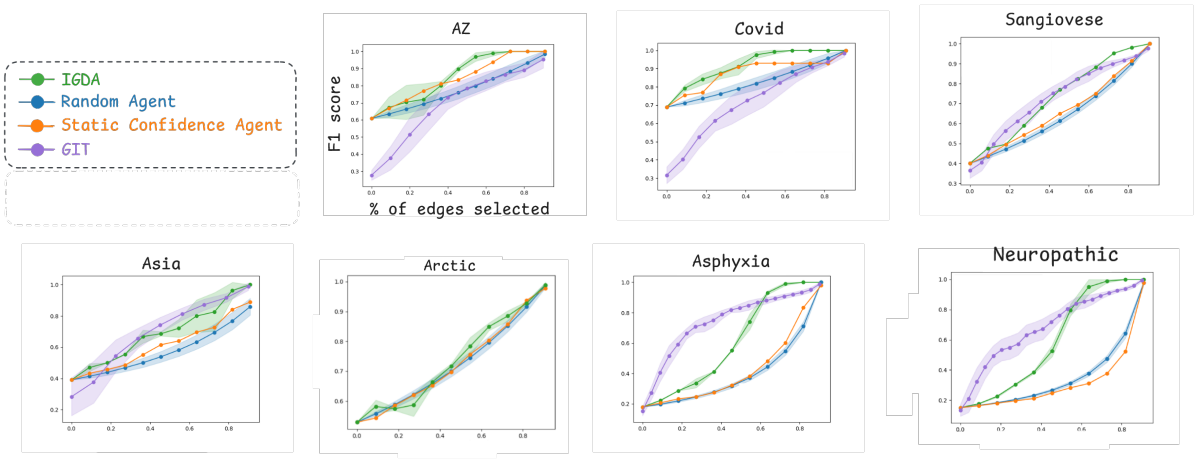

Figure 2: Results on real world graphs showing F1 score of the predicted graph against percentage of edges in the graph selected. IGDA almost always outperforms both the random baseline and static selection via uncertainty. Note: static confidence selection without local updates is deterministic and thus has no confidence intervals. Additionally, GIT is not reported on the Arctic graph because the graph is cyclic. **Note:** GIT uses synthetically generated numerical observational/interventional data. IGDA receies only binary edge feedback.

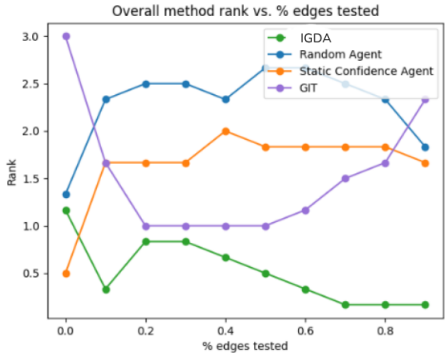

Figure 3: Average rank of each method when numbered from 0 to 2 across each timestep on each graph. The full LLM driven update agent consistently achieves rank 0 across all timesteps. Note: **lower is better**.

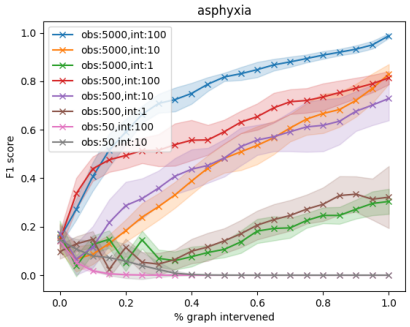

Figure 4: GIT with varying amounts of observational and interventional data. Decreasing either observational or interventional sample sizes can decrease performance by over 0.2 F1 score.

selection component struggles to outperform the random baseline.

GIT performance heavily depends on availability of both observational and interventional data With ample data (5000 observational samples and 100 interventional samples per node) the statistical GIT methods performs well on most graphs where it is applicable (i.e. the graph is acyclic). However, we find this good performance heavily depends on the availability of such data, with decreases in both observational and interventional sample sizes significantly impacting results. In Figure 4 we plot the performance of GIT on the Asphyxia graph with varying amounts of data demonstrating this effect. Results on more graphs are presented in the appendix. In contrast, IGDA does not depend at all on the availability of numerical observational or interventional data. Instead, IGDA relies on the complementary availability of semantic metadata of graph variables within either its pretraining dataset or on the internet. This gives **IGDA a clear advantage over GIT in low-data regimes**.

In an effort to better understand the factors behind IGDA's success we conduct a number of ablations in the following section.

4.1 Ablations

Impact of experiment improvements versus update improvements As a starting point we define the *net graph improvement* in a round r as the difference between the number of edges correctly classified in in $\hat{G}_r$ versus in $\hat{G}_{r-1}$. If an

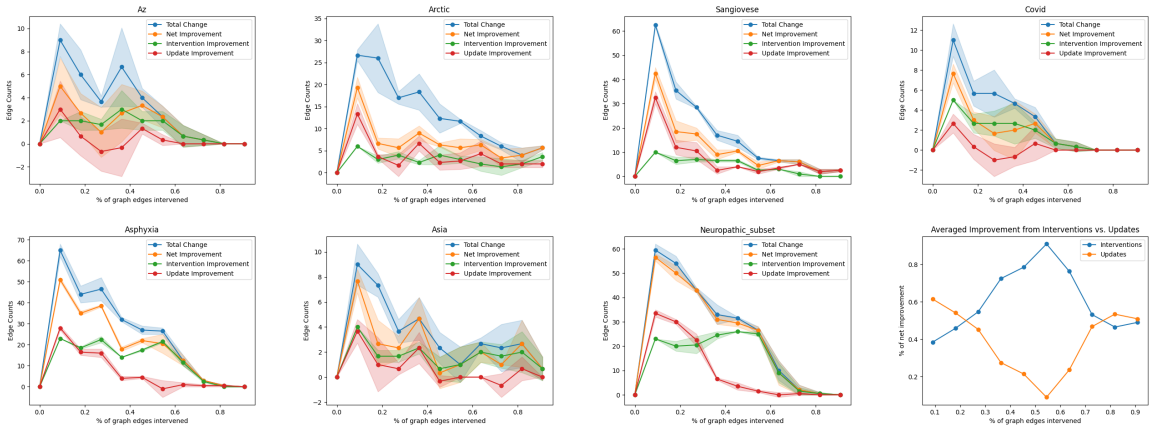

Figure 5: % Improvement from experiments vs. LLM prediction updates across timesteps. Improvement directly from LLM updates peaks early but then falls off. Improvement from experiments stays constant or improves with more experiments as confidence scores become better calibrated.

edge (X_i, X_j) is correctly classified in $\hat{G}_r$ but not in $\hat{G}_{r-1}$ we say it has been *improved*. Recall there are two potential mechanisms of improvement for (X_i, X_j): 1) (X_i, X_j) was selected for experimentation in the previous round $r-1$ and feedback on the experiment was received at the start of round r 2) The prediction for (X_i, X_j) was updated by the LLM after receiving experiment feedback for an adjacent edge (X_k, X_l). We call the former improvements *experiment improvements* and the latter *update improvements*. In a given round r we are interested in how much of the net improvement for a graph is due to experiment improvements versus update improvements. To examine this, we plot both quantities in Figure 5 for the discovery processes discussed in the previous section. In addition, we plot the net graph improvement and total number of edges changed from each round.

In all seven graphs we see both the total number of changed edges and the net improved edges peak at the first round and then decay towards zero. Notably, on some graphs there is a significant gap between net improvement and total change, indicating many edges changed during dynamic updates are misclassified after previously being correctly classified. This decline in total and net change is reflected in the number of update improvements which peak early and sharply decline to zero. This observation supports our intuition above that allowing the LLM to dynamically update edge predictions without direct experimental feedback on the edge can dramatically improve performance at small percentages of experiments. In contrast, experiment improvement accounts for a smaller percentage (less than 40%) of edge improvements early on. However, in most graphs the number of experiment improvements stays nearly constant until at least 50% of edges are already selected. As a result, improvement from experiments grows to account for 90% of all edge improvements for rounds performed during this period. This demonstrates improvements from experiment and updates complement each other, with **update improvement driving net improvement early and experiment improvement driving net improvement later on**.

Our analysis here also confirms the effectiveness of allowing the LLM agent to update both the prediction **and** confidence for an edge. Even when only considering improvements from experiments when doing local updates, we see a major improvement over the static confidence baseline. This suggests the **updates made to edge confidence scores are equally important in achieving good performance**, allowing for sustained experiment improvement throughout the discovery process. See Section B for an ablation investigating the effect of selecting edges using LLM generated confidence scores.

Impact of the LLM Model Size The above experiments exclusively use a single base LLM (`Meta-Llama-3-70B-Instruct`) to perform both the initial round of zero-shot edge predictions and dynamically update edge predictions/confidences using experiment feedback. Now, we examine the impact of changing both the base model size and type. In Figure 6 we initialize the discovery process with zero-shot predictions made by `Meta-Llama-3-70B-Instruct` and run local updates using the smaller

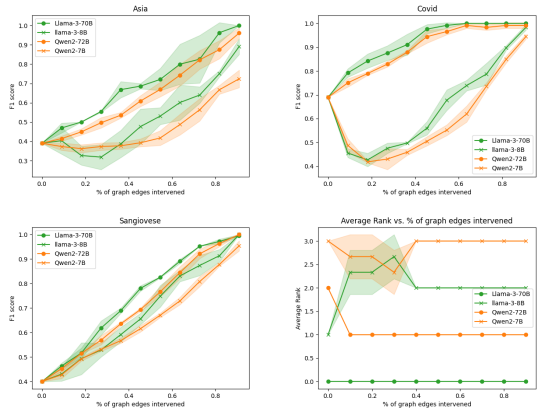

Figure 6: Performance of LLM driven interactive discovery on different sized models. Small LLMs (8B params) underperform the random baseline.

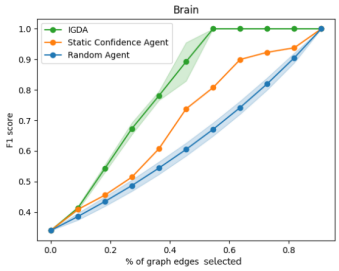

Figure 7: Performance curves of uncertainty driven selection + local prompting vs. baselines on the Brain graph (Zhu et al., 2024) recently published in July 2024.

`Meta-Llama-3-8B-Instruct` as well as two models from the Qwen2 series.

We find the original `Meta-Llama-3-70B-Instruct` consistently performs best on all graphs at every time step. The other 70B model, `Qwen2-72B-Instruct`, performs similarly but consistently worse. In contrast, on the Asia and Covid graphs, both 8B models perform worse than even the random baseline. Surprisingly `Meta-Llama-3-8B-Instruct` performs reasonably well on the Sangiovese graph, performing similarly even to the 9x larger Qwen2 70B model. Overall however these results indicate performance on the interactive graph discovery task can be substantially improved with model scale.

We next investigate the performance of different models on the initial zero-shot edge prediction task. Using the pairwise confidence estimation prompt in Section D we prompt each of four models to produce a zero-shot prediction $\hat{G}_0$ with edge confidence values. Using the predicted confidence estimates we run greedy static confidence selection procedure as in 4. Ranks for each selection procedure averaged over all graphs are plotted in Figure 10. F1 scores in each graph are reported in Figure 9 in the Appendix.

Impact of Memorization The success of LLMs in discovery stems from their immense background knowledge acquired during pre-training. This background knowledge informs the model during edge prediction and confidence calibration, allowing for strong performance even zero-shot. However, if benchmark graphs are contained verbatim in pre-training data, memorization becomes a significant confounding factor. To investigate to what extent memorization impacts performance we find a recently published graph (published in July 2024) from Zhu et al. (2024) modeling the gene regulatory network underlying 29 protein transcription factors. Because `Meta-Llama-3-70B-Instruct` finished training in 2023 this graph is guaranteed to be memorization free. Figure 7 plots the performance of uncertainty driven edge selection + local updates compared to the static selection and random baseline.

Figure 7 shows our confidence driven selection + local update approach performs very well even on graphs with minimal memorization contamination. As previously observed, local prediction updates allow for fast improvement over the random baseline even with a small number of experiments. Surprisingly, the static confidence selection approach also works well here. This indicates zero-shot edge confidence scores can be well calibrated on graphs with no contamination from memorization. We additionally note this graph has a complex structure with many cycles of varying lengths. This shows our method performs well even on graphs which strongly violate often assumed DAG conditions.

5 Conclusions and Future Work

In this work we proposed IGDA as a novel application of LLMs to interactive graph discovery. Our experiments confirm the proposed IGDA method significantly outperforms baselines. Our ablations confirm both uncertainty driven edge selection and local updates using experiment feedback as importantly contributing to the method's good performance. Further, this method is complementary to existing statistical methods which utilize numerical data for experiment design or causal discovery (e.g. GIT (Olko et al., 2024)).

6 Limitations and Broader Impact

Limitations IGDA does not leverage numerical observational/interventional causal data. Instead the agent utilizes available semantic variable metatdata from pre-training. Future work might investigate methods leveraging both numerical and semantic data.

Broader Impact As with any work studying generative models, we note generative modeling can suffer from pre-existing biases in the training data. This behavior may help propagate existing societal biases present today.

References

Ahmed Abdulaal, adamos hadjivasiliou, Nina Montana-Brown, Tiantian He, Ayodeji Ijishakin, Ivana Drobnjak, Daniel C. Castro, and Daniel C. Alexander. 2024. Causal modelling agents: Causal graph discovery through synergising metadata- and data-driven reasoning. In *The Twelfth International Conference on Learning Representations*.

Microsoft Research AI4Science and Microsoft Azure Quantum. 2023. The impact of large language models on scientific discovery: a preliminary study using gpt-4. *Preprint*, arXiv:2311.07361.

Sirui Chen, Mengying Xu, Kun Wang, Xingyu Zeng, Rui Zhao, Shengjie Zhao, and Chaochao Lu. 2024. Clear: Can language models really understand causal graphs? *Preprint*, arXiv:2406.16605.

Mathieu Chevalley, Yusuf Roohani, Arash Mehrjou, Jure Leskovec, and Patrick Schwab. 2023. Causalbench: A large-scale benchmark for network inference from single-cell perturbation data. *Preprint*, arXiv:2210.17283.

David Maxwell Chickering. 2002. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3:507–554.

Zhixuan Chu, Jianmin Huang, Ruopeng Li, Wei Chu, and Sheng Li. 2023. Causal effect estimation: Recent advances, challenges, and opportunities. *Preprint*, arXiv:2302.00848.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Shanghua Gao, Ada Fang, Yepeng Huang, Valentina Giunchiglia, Ayush Noori, Jonathan Richard Schwarz, Yasha Ektefaie, Jovana Kondic, and Marinka Zitnik. 2024. Empowering biomedical discovery with ai agents.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. 2023. Gpt4graph: Can large language models understand graph structured data ? an empirical evaluation and benchmarking. *Preprint*, arXiv:2305.15066.

Alex Havrilla, Sharath Raparthy, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. 2024. Glore: When, where, and how to improve llm reasoning via global and local refinements. *Preprint*, arXiv:2402.10963.

Yiyi Huang, Matthäus Kleindessner, Alexey Munishkin, Debvrat Varshney, Pei Guo, and Jianwu Wang. 2021. Benchmarking of data-driven causality discovery approaches in the interactions of arctic sea ice and atmosphere. *Frontiers in Big Data*, 4.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. 2024. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *Preprint*, arXiv:2406.06769.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. 2025. Aide: Ai-driven exploration in the space of code. *Preprint*, arXiv:2502.13138.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2024. Causal reasoning and large language models: Opening a new frontier for causality. *Preprint*, arXiv:2305.00050.

Andrew Kyle Lampinen, Stephanie C Y Chan, Ishita Dasgupta, Andrew J Nam, and Jane X Wang. 2023. Passive learning of active causal strategies in agents and language models. *Preprint*, arXiv:2305.16183.

Peiwen Li, Xin Wang, Zeyang Zhang, Yuan Meng, Fang Shen, Yue Li, Jialong Wang, Yang Li, and Wenweu Zhu. 2024. Realtcd: Temporal causal discovery from interventional data with large language model. *Preprint*, arXiv:2404.14786.

Phillip Lippe, Taco Cohen, and Efstratios Gavves. 2022. Efficient neural causal discovery without acyclicity constraints. *Preprint*, arXiv:2107.10483.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2024. Discoverybench: Towards data-driven discovery with large language models. *Preprint*, arXiv:2407.01725.

Christopher Meek. 2013. Causal inference and causal explanation with background knowledge. *Preprint*, arXiv:1302.4972.

Meike Nauta, Doina Bucur, and Christin Seifert. 2019. Causal discovery with attention-based convolutional neural networks. *Mach. Learn. Knowl. Extr.*, 1:312–340.

Mateusz Olko, Michał Zając, Aleksandra Nowak, Nino Scherrer, Yashas Annadani, Stefan Bauer, Łukasz Kuciński, and Piotr Miłoś. 2024. Trust your ∇: Gradient-based intervention targeting for causal discovery. *Preprint*, arXiv:2211.13715.

Judea Pearl. 2009. *Causality: Models, Reasoning and Inference*, 2nd edition. Cambridge University Press, USA.

Jonas Peters, Dominik Janzing, and Bernhard Schlkopf. 2017. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press.

Yusuf Roohani, Jian Vora, Qian Huang, Zachary Steinhart, Alexander Marson, Percy Liang, and Jure Leskovec. 2024. Biodiscoveryagent: An ai agent for designing genetic perturbation experiments. *Preprint*, arXiv:2405.17631.

Nino Scherrer, Olexa Bilaniuk, Yashas Annadani, Anirudh Goyal, Patrick Schwab, Bernhard Schölkopf, Michael C. Mozer, Yoshua Bengio, Stefan Bauer, and Nan Rosemary Ke. 2022. Learning neural causal models with active interventions. *Preprint*, arXiv:2109.02429.

B Sibbald and M Roland. 1998. Understanding controlled trials: Why are randomised controlled trials important? *BMJ*, 316(7126):201.

Peter Spirtes, Clark Glymour, and Richard Scheines. 1993. Causation, prediction, and search.

Peter Spirtes and Kun Zhang. 2016. Causal discovery and inference: concepts and recent methodological advances. *Applied Informatics*, 3(1):3.

Ruibo Tu, Kun Zhang, Bo Christer Bertilson, Hedvig Kjellström, and Cheng Zhang. 2019. Neuropathic pain diagnosis simulator for causal discovery algorithm evaluation. *Preprint*, arXiv:1906.01732.

Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N Balasubramanian, and Amit Sharma. 2023. Causal inference using llm-guided discovery. *Preprint*, arXiv:2310.15117.

Petar Veličković, Alex Vitvitskyi, Larisa Markeeva, Borja Ibarz, Lars Buesing, Matej Balog, and Alexander Novikov. 2024. Amplifying human performance in combinatorial competitive programming. *Preprint*, arXiv:2411.19744.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. *Preprint*, arXiv:2309.03409.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. 2019. Dag-gnn: Dag structure learning with graph neural networks. *Preprint*, arXiv:1904.10098.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. 2018. Dags with no tears: Continuous optimization for structure learning. *Preprint*, arXiv:1803.01422.

Yu Zhou, Xingyu Wu, Beicheng Huang, Jibin Wu, Liang Feng, and Kay Chen Tan. 2024. Causalbench: A comprehensive benchmark for causal learning capability of large language models. *Preprint*, arXiv:2404.06349.

Yuehua Zhu, Panayiotis V Benos, and Maria Chikina. 2024. A hybrid constrained continuous optimization approach for optimal causal discovery from biological data. *Bioinformatics*, 40(Supplement$_2$) : $ii87 - -ii97$.

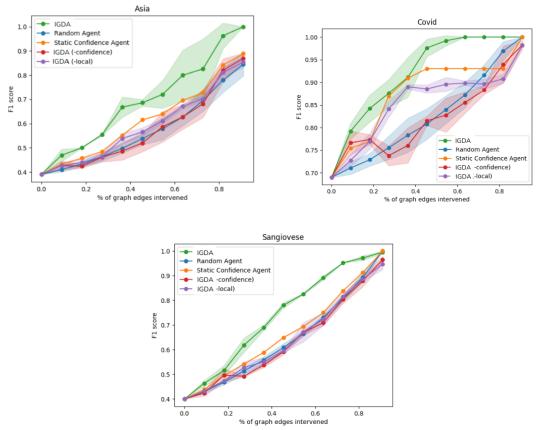

Figure 8: Ablating confidence based edge selection and local update prompting.

A Implementation Details

For a python implementation of IGDA go to https://anonymous.4open.science/r/IGDA-7AFB/README.md.

B Impact of Confidence Based Selection and Local Prompting

We now ablate the impact of two key components of our discovery strategy: 1) confidence based edge selection and 2) local update prompting. To ablate 1) we directly prompt the LLM to generate a list of edges to experiment on instead of selecting via confidence. This requires us to put the entire current predicted graph $\hat{G}_r$ in-context. When dynamically updating $\hat{G}_r$ after receiving experimental feedback we remove all confidence estimates but retain the local prompting strategy. To ablate 2) we retain the same confidence edge selection proposed but replace local update prompts after with a single global update prompt containing the current prediction $\hat{G}_r$ and all recently received experiment feedback. We report the results of running the interactive discovery process with these methods in Figure 8.

We find both ablations struggle to perform better than the random baseline. Local updates without confidence selection perform well early on but fall off quickly. F1 score on the Covid graph even regresses after the initial improvements, likely due to incorrect local updates and a poor experiment selection policy. This suggests in addition to providing a strong experiment selection procedure, maintaining running confidence estimates for each edge reduces the variance of local updates from experiment feedback. Turning to the ablation for local prompting, we again find performance not much better than the random baseline. Surprisingly, even on Covid where the static confidence selection performs well, confidence based selection + global updates still struggles. This indicates the base LLM is not able to correctly update the predicted graph when giving everything in context at once. This further motivate the practical importance of the local prompting procedure, which greatly simplifies the context the LLM must consider in each model call. Additionally, we note that for large enough graphs, putting everything in context is simply not feasible. By contrast, local prompting is easily scalable to larger graphs, albeit at a quadratic cost.

C Static Confidence Selection over Multiple Models

Figure 9 reports the results of applying static confidence experiment selection using various models.

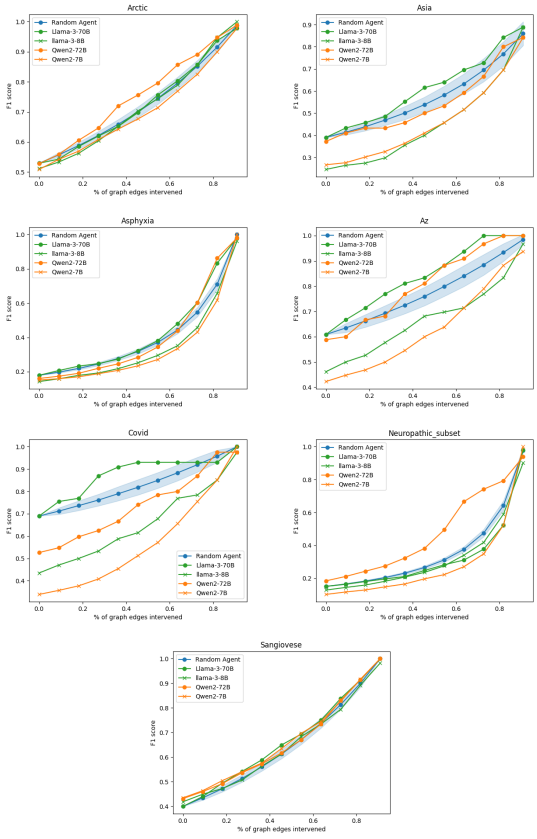

Figure 9: Static confidence selection over multiple models.

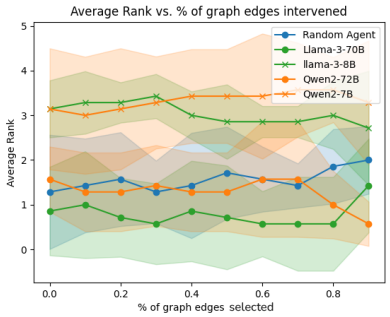

Figure 10: Static confidence based selection ranks for different models averaged across graphs. `Meta-Llama-3-70B-Instruct` is the only model to consistently outperform random guessing. Note: **lower is better**.

D Prompts

Zero-shot Confidence Estimation Prompt

[ht] {task_description} Your goal is to understand the direct causal parents of {target}. Another variable is a direct causal parent of {target} if an experiment on the variable affects {target} and there are no other causal parents between the variable and {target}. Now, you must determine whether {parent} is a causal parent of {target}. Here is a list of all other variables to consider:
{variables_info}
Do some brainstorming, comparing relevant characteristics of both variables and then print your judgment at the end of your response enclosed in the tags decision YES/NO/decision . Print YES if {parent} is causal. Otherwise print NO. You should also print your confidence from a scale from 1 - 100 (with 100 being most confident) in the tags confidence ... /confidence .
Information about {target}: {target_info}
Information about {parent}: {parent_info}

Parent Update Prompt

You are a causal discovery expert. You have been given the following list of variables and tasked with predicting the true causal graph through a sequence of experiments on edges.
{variables_info}
Note: each edge has an associated confidence value from 1 - 100. The presence of an edge is represented as (A− >B,CONFIDENCE) where A is the parent and B is the child. The absence of an edge is represented as (NOT A− >B, CONFIDENCE)
From one experiment you have discovered {experiment_feedback} Previously you predicted {experiment_prediction}
Now you should update your belief about the other edges of {parent} based on the results of the experiment. Consider the predicted edge
{other_edge_prediction}
Now you should reason about how to update your belief about the above edge based on the experiment. This means you can either keep your confidence the same, update your confidence, or change your prediction entirely. At the end of your response give your updated prediction at the end of your response in the format decision PARENT/NOT CAUSAL/decision> confidence CONFIDENCE/confidence .
Print 'PARENT' if the edge should be present and 'NOT CAUSAL' if the edge should be absent.
You should do this in three steps.
Step 1: Brainstorm what physical causal connection there may be, if any.
Step 2: Reason about what the experiment feedback tells you. Think carefully about how similar the new child is to the experimental child.
Step 3: Give your final decision.

Child Update Prompt

You are a causal discovery expert. You have been given the following list of variables and tasked with predicting the true causal graph through a sequence of experiments on edges.
{variables_info}
Note: each edge has an associated confidence value from 1 - 100. The presence of an edge is represented as (A$-$ $>$ B,CONFIDENCE) where A is the parent and B is the child. The absence of an edge is represented as (NOT A$-$ $>$B, CONFIDENCE)
From one experiment you have discovered {experiment_feedback} Previously you predicted {experiment_prediction}
Now you should update your belief about the other edges of {child} based on the results of the experiment. Consider the predicted edge
{other_edge_prediction}
Now you should reason about how to update your belief about the above edge based on the experiment. This means you can either keep your confidence the same, update your confidence, or change your prediction entirely. At the end of your response give your updated prediction at the end of your response in the format decision PARENT/NOT CAUSAL/decision confidence CONFIDENCE/confidence .
Print 'PARENT' if the edge should be present and 'NOT CAUSAL' if the edge should be absent.
You should do this in three steps.
Step 1: Brainstorm what physical causal connection there may be, if any.
Step 2: Reason about what the experiment feedback tells you. Think carefully about how similar the new parent is to the experiment parent.
Step 3: Give your final decision.

E Causal Graphs

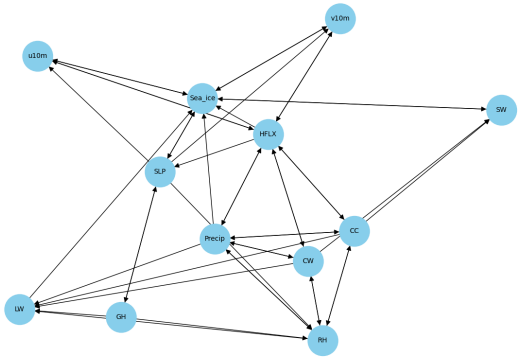

Figure 11: Arctic sea ice causal graph.

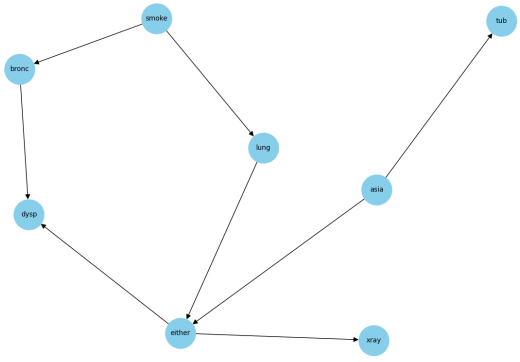

Figure 12: Asia causal graph.

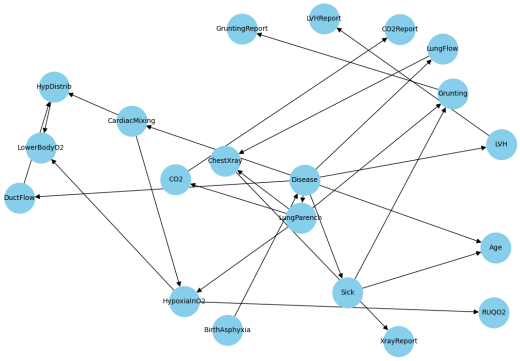

Figure 13: Asphyxia causal graph.

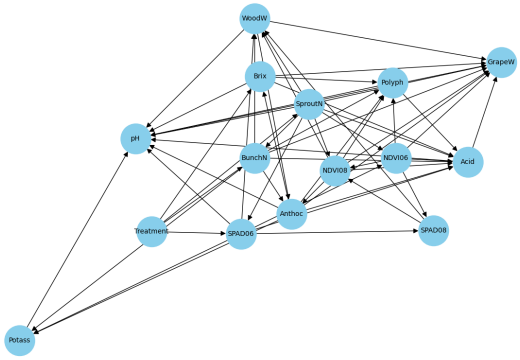

Figure 17: Sangiovese causal graph.

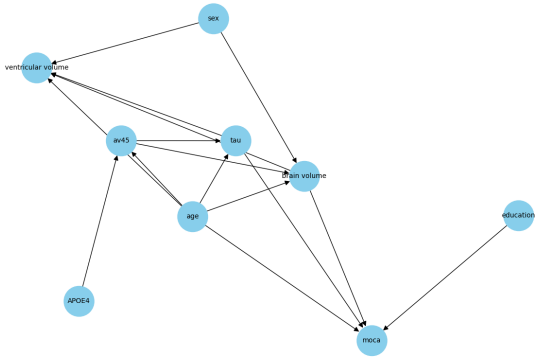

Figure 14: Alzheimers causal graph.

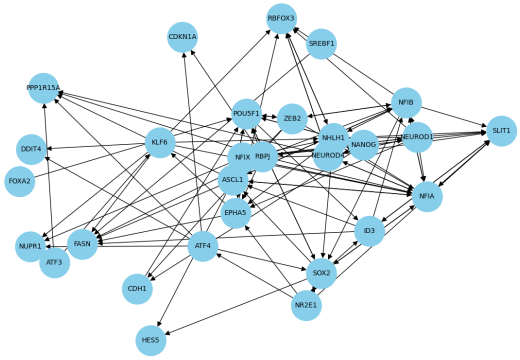

Figure 18: Brain causal graph.

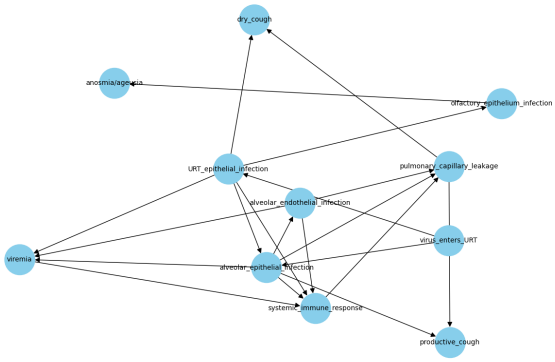

Figure 15: Covid causal graph.

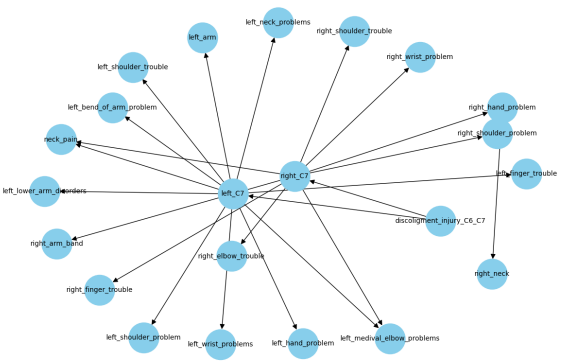

Figure 16: Neuropathic pain causal graph.

F GIT Ablations

Figure 19 plot GIT performance (Olko et al., 2024) over six causal graphs with varying amounts of observational and interventional data.

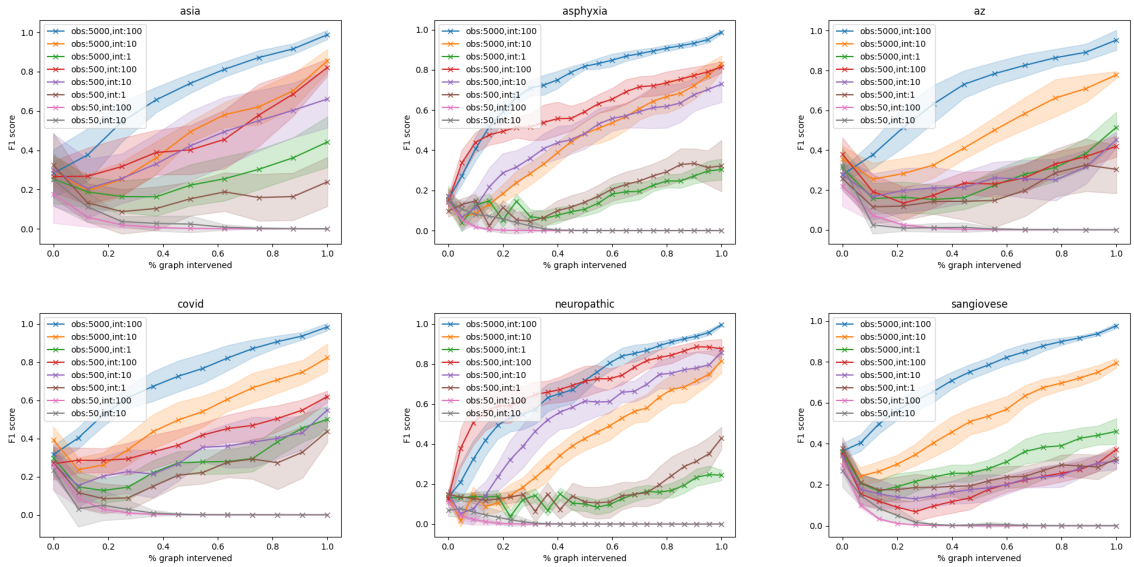

Figure 19: GIT ablations with varying amounts of observational and interventional data.