Workshop on Causal Discovery
CaDis 2024

ACADEMIA MEXICANA DE COMPUTACIÓN, A. C.

Proceedings of the 2nd Workshop on Causal Discovery CaDis 2024
Editors: Luis Enrique Sucar and Julio César Muñoz–Benítez

In collaboration with Academia Mexicana de Computación.

First Edition 2024.
Academia Mexicana de Computación, A. C.

ISBN:

Style correction: Luis Enrique Sucar.
Cover design: Instituto Nacional de Astrofíscia, Óptica y Electrónica.
Editing: Luis Enrique Sucar.

Printed in Mexico.

Proceedings of the 2nd Workshop on Causal Discovery
CaDis 2024

Workshop Chairs:

Luis Enrique Sucar, INAOE, Mexico
Julio César Muñoz-Benítez, INAOE, Mexico
Billy Peralta, CENIA, Chile

Program Committee:

Adnan Darwiche, University of California, Los Angeles, USA
Hugo Jair Escalante, INAOE, Mexico
Mauricio González, ITESM, Mexico
Samuel Montero, University of Birmingham, UK
Eduardo Morales, INAOE, Mexico
Julio César Muñoz-Benítez, INAOE, Mexico
Billy Peralta, CENIA, Chile
Guillermo Puebla, Universidad de Tarapacá, Chile
Yamisley Salgueiro, Universidad de Talca, Chile
Rubén Sánchez-Romero, Rutgers University-Newark, USA
Luis Enrique Sucar, INAOE, Mexico

Foreword

This volume contains the proceedings of the 2nd Workshop on Causal Discovery (CaDis 2024). The workshop was held at the Facultad de Ingeniería of the Universidad de la República in Montevideo, Uruguay as part of IBERAMIA 20224. This edition was made possible through the collaboration of the National Institute of Astrophysics, Optics, and Electronics of Mexico (INAOE), the National Center for Artificial Intelligence (CENIA) of Chile, and the Iberamia organizing committee. We acknowledge ths support of the Artificial Intelligence Journal, and the Mexican Academy of Computing (AMexComp).

Causal models have many advantages, including the ability to reason about the effects of interventions, as well as the results of different scenarios or counterfactuals. The traditional approach for building causal models is by conducting experiments, however these are often infeasible, unethical or too expensive. Recently there has been a lot of interest in the scientific community to learn causal models from observational data, but this is a great challenge, as just from observations is not possible, in general, to define a unique causal model.

The objective of this workshop was to present recent advances in causal discovery, including different approaches that consider observational and/or interventional data, and also building models with the help of human experts. It is also of interest the combination of causal discovery with other areas of machine learning, such as reinforcement learning and deep learning; as well as real-world applications.

The CaDis 2024 program included invited talks by Prof. Kun Zhang (professor and the acting department chair of machine learning at Mohamed bin Zayed University of Artificial Intelligence, MBZUAI, and professor at Carnegie-Mellon University) and Prof. David Danks (professor of Data Science, Philosophy,

& Policy at University of California, San Diego). Video recordings of these talks are available at the workshop website: `https://cadisworkshop.com.mx/program/`.

After a review by at least three members of the program committee, eleven papers were accepted for publication in these proceedings. In an analogous way as the workshop, this proceedings are divided in three parts: (i) Invited Talks abstracts, (ii) Fundamentals and Algorithms for Causal Discover, and (iii) Applications.

We hope that this workshop will help to increase the interest of the academic community in causal reasoning and discovery, in particular in the Iberoamerican region; and help to foster new collaborations between different research groups.

Luis Enrique Sucar and Julio César Muñoz-Benitez
Workshop Chairs

Contents

Section 1

Invited Talks

Causal Representation Learning: Uncovering the Hidden World

Author: Prof. Kun Zhang, Mohamed bin Zayed University / Carnegie-Mellon University

Abstract: Causal representation learning aims to reveal the underlying high-level hidden causal variables and causal influences, and causal models naturally inform optimal interventions, support decision making, and allow adaptive prediction in nonstationary environments. In this talk, we show how to recover the underlying causal representations from observational data with identifiability guarantees: under appropriate assumptions, the learned representations are consistent with the underlying causal process. Various problem settings are considered, involving independent and identically distributed (i.i.d.) data, temporal data, or data with distribution shift as input. We then demonstrate when identifiable causal representation learning can benefit from flexible deep learning and when suitable parametric assumptions have to be imposed on the causal process. We then show how to learn compact, interpretable state representations, together with the underlying causal model, in deep reinforcement learning and how to make quick adaptations in transfer reinforcement learning.

Causal Discovery from Complicated Data

Author: Prof. David Danks, University of California, San Diego

Abstract: Much of the research on causal discovery methods focuses on methods able to handle increasingly complex systems, whether in terms of number of variables, density of connections, complexity of parametric functions, and more. In contrast, this talk will focus on causal discovery when we have challenges due to complicated measurements. We are often unable to measure exactly what and how we want, and as a result, real-world causal discovery requires methods that are robust to different measurement issues. Specifically, I focus on two different types of challenges. First, we might not be able to measure at the same timescale as the causal connections; for example, fMRI measurements are much slower than the brain's timescale, or economic measures are typically released less frequently than economic activity occurs. I will describe several causal discovery algorithms that we have developed for this situation, demonstrating both theoretical and real-world performance. I also demonstrate that we can sometimes benefit by measuring even more slowly, contra the standard advice that "faster measurement is better measurement." Second, we might not be able to measure all of the relevant variables at the same time; for example, a healthcare organization and financial institution might each measure variables that interest them, but we cannot (for privacy reasons) integrate those into a single dataset. I describe causal discovery methods that can extract information about the global causal structure from these partial, overlapping datasets, and show that the methods can yield novel scientific insights on real-world data.

Section 2

Fundamentals and Algorithms for Causal Discovery

A Framework for Practical Causal Discovery from Observational Data

Gustavo F.V. de Oliveira[1], Fabrício A. Silva[1], and Marcus H.S. Mendes[1]

IEF, Universidade Federal de Viçosa, Florestal, Brazil
{gustavo.viegas,fabricio.asilva,marcus.mendes}@ufv.br

Abstract. Causal reasoning is crucial for advancing AI models beyond correlation-based predictions to infer meaningful cause-and-effect relationships. This paper introduces a novel framework that facilitates causal inference for AI developers with minimal causality expertise. It integrates multiple causal discovery algorithms, standardizes data structures, and employs parallel processing to generate and evaluate causal models efficiently. It automates the selection of appropriate algorithms based on data attributes and prioritizes feature importance to ensure robust causal effect estimation and refutation. The framework effectiveness is demonstrated through a benchmark on a real-world dataset, where it accurately identifies known causal relationships. By providing user-friendly outputs and customizable features, the framework enhances the accessibility and reliability of causal analysis in AI, paving the way for its application in sensitive domains such as healthcare, finance, and justice.

Keywords: Causality · Causal Discovery · Framework · Ethical AI.

1 Introduction

Causation is the concept referring to the cause-and-effect relationship wherein one event (the cause) leads to another event (the effect) and establishes a direct connection or influence between them [17]. Artificial Intelligence (AI) models have demonstrated remarkable capabilities in establishing intricate connections and making accurate predictions from data. However, when confronted with unfamiliar data, these models face challenges, leading to inaccurate causal relationships [16]. This limitation is particularly troubling in ethical AI, where biased associations are highly undesirable, such as the risk of incorrect associations between factors like race and criminal behavior, resulting in unjust outcomes [6].

The complexity of the field, with its requirements of domain knowledge, causal Directed Acyclic Graph (DAG) notation, causal discovery algorithms, estimators, and refuters, can be discouraging. While the adoption of causality analysis has the potential to enhance AI models, the incorporation of these concepts can be overwhelming and challenging due to the steep learning curve and the need to stay current with evolving content. The early stage of development of causal frameworks poses a challenge to their widespread adoption. The field is relatively new, with the most popular and corporate-backed packages released within the last five years [21,24,5]. Additionally, essential performance features like time limits and parallelism still need to be universally implemented, which could be a significant burden for certain self-learning AI models to overcome.

To facilitate the adoption of causal inference, this paper introduces *CausalNest*, a user-friendly framework simplifying causal inference for AI developers with limited causality knowledge. The primary goal of the framework is to build multiple causal models using relevant causal discovery algorithms while considering time constraints and prioritizing potential causal models and their relationships. It can automatically select suitable causal discovery algorithms based on the input data and standardize input and output formats across all processes. Additionally, it produces a graphical output summarizing the pipeline results in a format that is easy for humans to understand.

The article is organized as follows: Section 2 overviews the main causality pipeline: causal discovery and inference. Section 3 presents our proposed framework. Section 4 discusses experimental results, and Section 5 concludes the work.

2 Related Work

This section discusses key components of a causality pipeline that systematically identifies causal relationships from observational data. The main steps of the pipeline are causal discovery to uncover the causal graph and causal inference to estimate effects. To our knowledge, existing literature lacks comprehensive frameworks that can independently address all aspects of a causality problem without the need for manual algorithm selection, model prioritization, and parallel execution, which would typically necessitate a substantial understanding of causality from the user.

2.1 Causal Discovery (CD)

The process of extracting causality knowledge from data involves key steps. It begins with problem formulation and constructing a causal model using a Directed Acyclic Graph (DAG) to represent hypothesized causal pathways among identified variables. Causal Discovery (CD) refers to the process that creates these models from observational data, and various methodologies exist for uncovering causal graph structures.

Causal discovery methods often yield DAGs or undirected Bayesian networks, each presenting its computational complexities. Several factors must be considered when choosing a causal discovery algorithm, including computational complexity, data assumptions, and the ability to extract meaningful causal relationships [17]. The selection criteria involve analyzing the data types, distribution assumptions, and linearity assumptions [23].

A Framework for Practical Causal Discovery from Observational Data

2.2 Causal Inference (CI)

Causal Inference (CI) is a complex task involving estimating causal effects based on a causal graph constructed by domain experts or inferred by advanced causal discovery algorithms. Statistical estimators quantify causal effects between variables in a causal model, and different interpretations of these estimations require domain-specific expertise for accurate understanding [17].

Validating the accuracy of causal effect estimations involves a comprehensive understanding of the data domain and conducting refutation tests. These tests include sensitivity analyses, such as introducing random placebo variables, adding random standard cause variables, or removing subsets from the data to ensure the inferred causal relationships are robust and align with theoretical expectations.

3 *Causal-Nest*

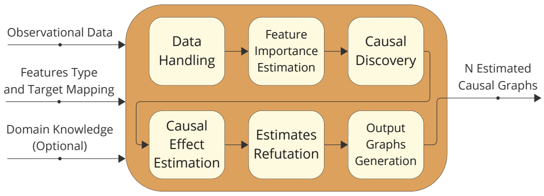


Fig. 1: Summarized pipeline of the proposed framework

This paper proposes a novel framework named *Causal-Nest*[1]. This framework includes a user-friendly interface for easier adoption and automatically manages the many rules and restrictions present in the field of causality. Figure 1 illustrates the key steps in the framework pipeline and their inputs and outputs. All algorithms in the framework were developed in Python 3.9 using packages for graph manipulation, dataset formatting, as well as causal discovery implementation with R backends, and estimation (through high-level packages [9,28,21]). Parallelism was employed in causal discovery, estimation, and refutation using the Pebble package version 5.0.

The framework has a standardized approach to data structures, inputs, and outputs. All graphs are represented as *Networkx* [8] graphs that can be easily translated to DOT, adjacency matrix, and other formats. This standardization ensures a consistent causal process, as the implemented discovery methods in the current causal frameworks operate uniformly while enabling easy customization and adding new methods. The graphical output is in DOT format and can be rendered in SVG, PNG, or other formats.

[1] The code and additional benchmarks are available on GitHub.

3.1 Input and Data Handling

To run the entire pipeline, only a few inputs are necessary: the observational dataset for analysis, the type mapping of variables in the dataset, such as categorical, continuous, and discrete, and the target column representing the observed effect. *Causal-Nest* assumes the input data is processed, cleaned, and formatted. It can handle diverse problem domains and data dimensions. If data is missing, the framework can drop it, apply forward fill or interpolate based on user preference.

Although establishing causality can be challenging, some relationships, such as holidays causing rain, are known to be false upfront. The domain knowledge of forbidden and required edges can be provided as input, formatted in a text file respecting the syntax described in [28].

3.2 Feature Importance Estimation

While correlation does not directly indicate causation, it is improbable for a causal relationship to exist between two variables that are not closely connected. *Causal-Nest* includes user-defined strict time limits for tasks such as effect estimation and refutation. These tasks are ranked in a prioritized list based on their calculated feature importance, which is determined using a Random Forest classifier or regression, and automatically chosen based on the nature of the problem. Establishing this prioritized list helps prevent potentially valuable candidates for causal effects from being overlooked in later stages of the process due to the time limit constraints and their lower position in the sequentially processed list of variables and the time limit constraints.

3.3 Causal Discovery (CD)

Several causal discovery algorithms are available in the literature, each with requirements and restrictions. Choosing the most appropriate one for a particular problem is a complex task. *Causal-Nest* includes twelve CD algorithms and automatically evaluates the ones suitable for the given problem scope, considering the data types of variables, normality distribution (as some models assume Gaussian data), and linearity. Table 1 displays the implemented CD algorithms in *Causal-Nest* with their corresponding backend and type of discovery. The framework also simplifies the process of implementing other discovery methods, both in terms of execution and constraints, to be verified for running in a given dataset. It can be achieved by implementing a straightforward interface that must adhere to two functions: a Boolean function determining if the algorithm is suitable for a given dataset input and another function to output the discovered graph.

The framework validates the data type of each feature in the input dataset to ensure compatibility with the model-accepted data types. If any feature has an unsupported type, the discovery algorithm will be rejected and not applied. A D'Agostino and Pearson-based test is employed for normality checks. This test assumes normality if all features yield a p-value lower than the default threshold of 0.05, applying the null hypothesis that the feature values come from a normal distribution. Lastly, linearity between each variable and the target is assessed using linear regression, expecting the mean of residuals to be lower than the default threshold of 0.05.

CD Algorithm	**Type**	**Backend**
1. Grow-Shrink (GS) [13]	Constraint-based	bnlearn [20]
2. Incremental Association Markov Blanket (IAMB) [25]	Constraint-based	bnlearn [20]
3. Causal Additive Models (CAM) [4]	Constraint-based	CAM R [4]
4. Concave Penalized Coordinate Descent with Reparametrization (CCDr) [3]	Score-based	sparsebn [2]
5. Causal Generative Neural Networks (CGNN) [7]	Score-based	pytorch [1]
6. Greedy Equivalance Search (GES) [15]	Score-based	pcalg [14]
7. Greedy Interventional Equivalance Search (GIES) [15]	Score-based	pcalg [14]
8. Linear Non-Gaussian Acyclic Model (LiNGAM) [22]	Score-based	pcalg [14]
9. Peter-Clark (PC) [11]	Constraint-based	pcalg [14]
10. Structural Agnostic Model (SAM) [10]	Score-based	pytorch [1]
11. Greedy Relaxation of the Sparsest Permutation (GRaSP) [12]	Permutation-based	causallearn [28]
12. BIC Exact Search (BES) [27]	Score-based	causallearn [28]

Table 1: Causal Discovery algorithms implemented by *Causal-Nest*

The framework uses a diverse range of causal discovery algorithms to construct a causal graph from the provided data. This process is parallelized and allows the user to specify the maximum number of workers, which defaults to the number of CPUs in the runtime system. The framework does not prioritize the execution order of algorithms, as the choice depends vastly on the problem domain context. Each worker is responsible for creating the causal graph, generating essential statistics and rankings, and has a user-defined timeout for execution. The automatic algorithm selection, discovery parallelization, timeout execution limits, and user-friendly statistics are fundamental features of the proposed framework, enabling systems that generate real-time data to extract causal knowledge from the data.

Some methods used to derive a causality model from data were not initially designed for causality but for Bayesian networks or other use cases. As a result, the outcome of the discovery algorithms may not always produce a Directed Acyclic Graph (DAG). In such cases, *Causal-Nest* incorporates a custom-defined function to handle these situations. Therefore, every causal discovery graph generated in *Causal-Nest* is a DAG.

The generated statistics in the causal discovery step include the Forbidden Edges Violation Rate (FEVR), the Required Edges Compliance Rate (RECR), and the Knowledge Integrity Score (KIS), which are also contributions of this work. FEVR represents the percentage of forbidden edges present in the generated graph, while RECR indicates the percentage of required edges present in the graph. KIS, a score ranging from 0 to 1, is calculated based on Equation 1, considering FEVR and RECR. A score of 1 indicates that the graph fully adheres to the knowledge restrictions, while a score of 0 indicates that the graph

Algorithm 1: Causal Discovery generated graph priority score

```
Input  : Graph $G$, target node $t$
Output: Ranking score
if $t \notin G.nodes()$ then return 0
$num\_incoming\_edges \leftarrow G.\text{in\_degree}(t)$
$num\_nodes \leftarrow |G.\text{nodes}()|$
$edge\_score \leftarrow \frac{num\_incoming\_edges}{num\_nodes}$
$distances \leftarrow G.\text{get\_nodes\_shortest\_distance\_to(target=}t)$
$avg\_distance \leftarrow \frac{\sum_{d \in distances} d}{|distances|}$
$distance\_score \leftarrow \frac{1}{avg\_distance+1}$
$density\_score \leftarrow G.\text{density}()$
$connectivity\_score \leftarrow 1 \text{ if } G.\text{is\_weakly\_connected}() \text{ else } 0$
$bc \leftarrow G.\text{betweenness\_centrality\_values}()$
$betweenness\_score \leftarrow \sum_{b \in bc} b$
$score \leftarrow 100 \times (edge\_score \times distance\_score \times density\_score \times$
  $connectivity\_score \times betweenness\_score)$
return $score$
```

does not adhere to any imposed restrictions. If no knowledge input is provided, FEVR is defined as 0 and RECR as 1, resulting in a KIS of 1, signifying that no restrictions are violated.

$$\text{KIS} = (1 - \text{FEVR}) \times \text{RECR} \tag{1}$$

Priority Score Another metric proposed in this work and assigned to a causal discovery graph output is the Priority Score, as detailed in Algorithm 1. It systematically organizes graphs generated by multiple causal discovery algorithms based on their potential to contain pivotal causal connections, meaning the higher the value, the more likely it contains the expected causal paths.

Graphs Comparison The generated graphs can be compared by utilizing various metrics to compare them against a ground truth graph, if available, or each other. Within *Causal-Nest* we have implemented three essential metrics for graph comparison: the Area Under the Precision-Recall Curve (AUCPR), Structural Hamming Distance (SHD) [26], and Structural Intervention Distance (SID) [18]. AUC-PRC assesses overall classification accuracy and misclassification when comparing adjacency matrices of graphs. SHD quantifies discrepancies by evaluating missing edges and directionality errors. SID examines causal relationships by computing paths between all pairs of variables to confirm consistency with causal features.

3.4 Causal Effect Estimation

Causal-Nest builds upon the DoWhy package [21] pipeline for estimating the causal effect of a variable. It is accomplished by first identifying the causal expression by assessing the graph structure and then estimating the variable effect. The proposed framework utilizes DoWhy default implementation to extract the identified estimand while disregarding unobserved confounders. Additionally, it employs DoWhy estimation methods, which can be parameterized by the user, defaulting to propensity score stratification. Therefore, the whole variable estimation process takes a causal graph and a variable as input and produces the causal estimate along with the p-value, indicating the statistical significance of the estimated causal effect.

The estimation process must be individually assessed for each variable acting as the treatment, which can lead to increased complexity. For a dataset with N independent variables and K discovered graphs, a total of $N \times K$ estimations will be carried out. In this scenario, the priority score of the discovered graphs (refer to Section 3.3) and the importance of features (refer to Section 3.2) play a crucial role in prioritizing the execution of the estimations. This prioritization is crucial as there is a hard time limit in place, which may prevent certain variables or graphs from being assessed. The estimation execution of a list of discovered graphs follows these steps:

1. Prioritize the graphs and features by sorting them in descending order based on their priority scores.
2. Create a parallelized process queue with a user-configurable maximum number of workers. By default, this maximum is set to the available CPU count in the running system.
3. Each process estimates the features sequentially, following the priority order. It has a user-configurable time limit to estimate the effect of as many variables as possible. If time runs out, partial estimations are allowed, meaning a graph can estimate a fraction of the variables.
4. After completing a process, store the estimation results independently for each model. Then, select the following graph in the queue for estimation.

3.5 Estimates Refutation

Refutation methods are built on top of the DoWhy package [21] and can easily support custom methods if they adhere to a simple interface. These methods take the dataset (and its feature mapping) and an estimation result as input and return an instance with the newly estimated effect and a p-value associated with the significance test of the newly calculated effect and the previous one. Our framework deems a refutation to have passed the refutation test if the p-value exceeds the default threshold of 0.05.

In the refutation phase, hard time limit constraints are applied due to the extensive input. Each variable serving as a treatment needs individual evaluation, and refutation must be handled for each estimation result. Multiple refutation methods are executed for each estimation result, resulting in a significant input size of $N \times K \times Z$, where N is the number of independent variables, K is the number of discovered graphs, and Z is the number of refuters. The proposed framework establishes a global timeout for the total execution time of the refuting pipeline and a model timeout for each estimation result entry, providing flexibility in handling different estimation results and refutation methods.

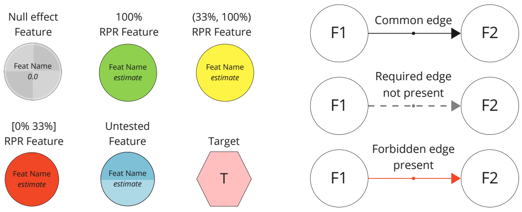
This study uses three methods of refutation: Placebo Treatment Refuter (PTR), Random Common Cause Refuter (RCCR), and Subset Removal Refuter (SRR). Each method tests different aspects of the estimated effect independently. Time constraints may limit the number of refutation tests that can be performed. The refutation execution of a list of estimation results follows these steps:

1. Prioritize the estimation results by arranging them in descending order based on their estimated effect values.
2. Create a parallelized process queue with a user-configurable maximum number of workers. By default, the maximum is set to the available CPU count.
3. Assign the process queue with independent estimation results based on the priority order. Each process tests the estimation result sequentially with the implemented refutation methods.
4. After completing the refutations for an estimated result or reaching the model-timeout threshold, store the refutation results independently for each estimation result. Then, select the next entry in the sorted list if the global timeout has not been reached.

3.6 Output Graphs Generation

The proposed framework provides a user-friendly output for quickly evaluating the generated graphs, including their estimation and refutations, as shown in Figure 2. It generates a custom DOT output graph for each graph discovered, allowing the user to inspect the causal effect estimates, the Refutation Pass Rate (RPR), and the presence of forbidden edges or the absence of required edges. Features include the feature name and the causal estimate below, while the target node is represented as a pink hexagon. Feature nodes follow a color scheme: green for variables that passed all refutation tests, red for those that passed up to 33% of the refutation tests, yellow for values in between, and split blue for variables with no refutation tests executed. A grey node is shown when the estimated effect of the node is 0, indicating that the variable has no estimated causal effect on the target.

If domain knowledge is provided as input to the framework, it may display edges in various styles in addition to the standard filled black arrow, which represents a causal relationship. A filled red arrow denotes an undesired edge, as specified in the domain input. In contrast, a gray dashed edge indicates edges listed as required in the input that were not discovered by the causal discovery algorithm. In the case of a pair of nodes with both a red and a dashed gray edge, this signifies a wrongly directed edge. *Causal-Nest* can correct these edges by modifying the graphs generated during the causal discovery process with the help of an optional flag that is turned off by default.



(a) Nodes notation for output graph

(b) Edges notation for output graph

Fig. 2: Custom nodes and edges style specification for the output graph

4 Experimental Results

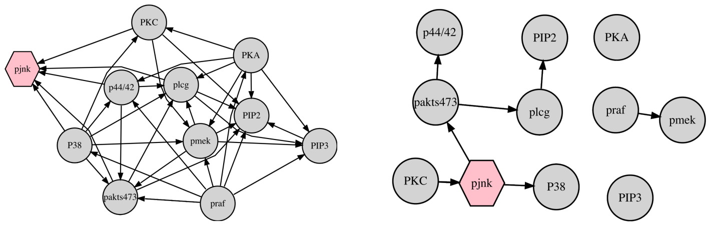
We showcase the execution of the causality pipeline in a real-world dataset that includes a generally accepted ground truth. The experiments were conducted on a 2023 MacBook Pro with an Apple M3 Pro chip and 36 GB of memory. All applied benchmarks utilized identical parameters for parallelization and time limits. Expressly, the maximum number of workers was set to 12. Additionally, a 30-minute timeout was established for each discovery model for causal discovery and estimation. A 30-minute global timeout and a 4-minute timeout per estimation result were applied for refutation.

The dataset created by Sachs *et al.* [19] is a widely used benchmark in causality due to its known ground truth graph, which consists of 19 directed edges. The dataset contains empirical quantitative data with 11 features and 7466 instances.

We treat the ‘pjnk‘ column as the target and map all other columns as continuous features. The dataset does not contain missing data. After the input is given, the feature sorting is applied with calculated associated importance.

The ground truth graph contains two edges pointing to the target node, from PKC and PKA, as Figure 4-a shows. Even though the feature importance ranking is determined by a regressor using the data and does not consider the ground truth, our solution placed these features at the top 4 of the sorted list. Out of the 9 paths to the target node in the ground truth, 7 involve the PKC feature, and PKA has a direct edge to the target, justifying their high ranking. Furthermore, 4 out of the lowest 5 features in the ranking do not have a path to the target in the ground truth.

Out of the 12 discovery methods implemented, *Causal-Nest* identified 8 applicable: PC, GS, CCDr, IAMB, SAM, BES, GRaSP, and CGNN. Due to the dataset size and complexity, SAM and CGNN timed out, while the others yielded DAGs with various calculated priority scores. BES had the highest score at 1.04, followed by IAMB (0.79), GS (0.76), PC (0.63), GRaSP (0.56), and CCDr (0.00). Figure 3 displays the discovered graphs with the highest (a) and lowest (b) priority scores. As expected, the graph generated by BES, containing multiple paths to the target and high interconnectivity, received a higher ranking. Conversely, the final graph in the list, lacking edges pointing towards the target and disconnected, justifies its lower position in the sorting order.



(a) Discovered graph by BES (b) Discovered graph by CCDr

Fig. 3: Discovered graphs sample for Sachs dataset

Table 2 compares the metrics for the discovered graphs. Additionally, Figure 4 shows the dataset ground truth alongside the output generated by the graph with the higher estimated causal effect (generated by PC), depicting the estimated effects of the features and their associated refutations. Notably, algorithms such as CCDr and PC demonstrated competitive performance, with AUCPRC scores of 0.45 and 0.38, respectively, indicating robust predictive power in identifying causal relationships. Achieving a 100% RPR across all algorithms highlights the capability of our solution to effectively generate robust causal models, enhancing the reliability of inferred causal relationships. The distances (SHD and SID) achieved in the benchmarks demonstrate varying levels of structural accuracy across algorithms, reflecting the diverse nature of causal discovery models and indicating the utility of a diverse set of algorithms in causality problems.

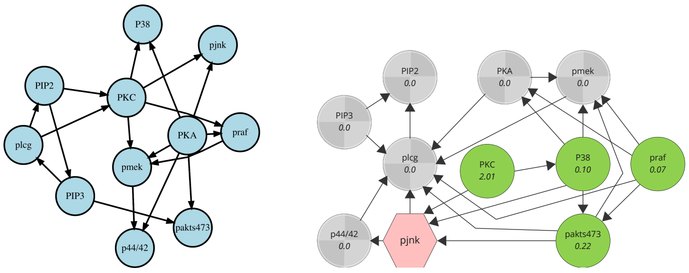
CD Model	**Priority Score**	**AUCPR**	**SHD**	**SID**	**Max Causal Effect Estimate**	**Max Causal Effect p-value**	**Max Causal Effect RPR**
1. GS	0.38	0.42	27	78	1.48	1.94e-05	100%
2. IAMB	0.77	0.25	35	90	1.83	2.57e-26	100%
4. CCDr	0.00	**0.45**	**17**	**72**	1.89	1.13e-18	100%
9. PC	0.63	0.38	28	83	**2.01**	1.50e-80	100%
11. GRaSP	0.18	0.09	42	86	1.87	3.98e-130	100%
12. BES	**1.04**	0.30	42	79	1.37	8.39e-114	100%

Table 2: Stats for the causality pipeline on the Sachs dataset

5 Conclusion

This work proposes a new framework that significantly advances causal inference for AI model developers, particularly those with minimal expertise in causality. By integrating multiple causal discovery algorithms and addressing crucial aspects such as time constraints and prioritization of potential causal models, *Causal-Nest* effectively simplifies the complex process of causal analysis.

Our experimental results demonstrate the capability of the framework to handle real-world problems. The performance of *Causal-Nest* on the Sachs *et*



(a) Sachs ground truth

(b) *Causal-Nest* output for PC in Sachs dataset

Fig. 4: Comparison between Sachs ground truth and a generated output graph

al. [19] dataset underscores its precision in identifying known causal relationships, thereby validating its effectiveness. The framework provides user-friendly outputs, featuring custom DOT graphs and a color-coded scheme, which enhance the interpretability of causal effect estimates and refutation results, making it accessible to users with varying levels of expertise.

In conclusion, *Causal-Nest* bridges a critical gap in AI development, providing a powerful tool for uncovering and validating causal relationships. This advancement is promising to improve decision-making and predictions in sensitive and high-stakes fields such as healthcare, justice, and finance. Future work should continue to refine and expand the capabilities of the framework, ensuring its adaptability to an ever-evolving landscape of AI and causality research.

References

1. Ansel, Jason, Y.e.a.: Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2. p. 929–947. ASPLOS '24, Association for Computing Machinery (2024). `https://doi.org/10.1145/3620665.3640366`
2. Aragam, B., Gu, J., Zhou, Q.: Learning large-scale bayesian networks with the sparsebn package. Journal of Statistical Software **91**(11) (2019). `https://doi.org/10.18637/jss.v091.i11`
3. Aragam, B., Zhou, Q.: Concave penalized estimation of sparse gaussian bayesian networks. The Journal of Machine Learning Research **16**(1), 2273–2328 (2015)
4. Bühlmann, P., Peters, J., Ernest, J.: Cam: Causal additive models, high-dimensional order search and penalized regression (2014)
5. Chen, H., Harinen, T., Lee, J.Y., et al.: Causalml: Python package for causal machine learning (2020). `https://doi.org/10.48550/ARXIV.2002.11631`
6. Ensign, D., Friedler, S.A., Neville, S., et al.: Runaway feedback loops in predictive policing. In: Conference on fairness, accountability and transparency. pp. 160–171. PMLR (2018)
7. Goudet, O., Kalainathan, D., Caillou, P., et al.: Learning functional causal models with generative neural networks. Explainable and interpretable models in computer vision and machine learning pp. 39–80 (2018)

G.F.V de Oliveira, F.A. Silva, M.H.S Mendes.

8. Hagberg, A., Swart, P.J., Schult, D.A.: Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Laboratory (LANL), Los Alamos, NM (United States) (2008)
9. Kalainathan, D., Goudet, O.: Causal discovery toolbox: Uncover causal relationships in python (2019). `https://doi.org/10.48550/ARXIV.1903.02278`
10. Kalainathan, D., Goudet, O., Guyon, I., et al.: Structural agnostic modeling: Adversarial learning of causal graphs. Journal of Machine Learning Research **23**(219), 1–62 (2022)
11. Kalisch, M., Bühlman, P.: Estimating high-dimensional directed acyclic graphs with the pc-algorithm. Journal of Machine Learning Research **8**(3) (2007)
12. Lam, W.Y., Andrews, B., Ramsey, J.: Greedy relaxations of the sparsest permutation algorithm. In: Uncertainty in Artificial Intelligence. pp. 1052–1062. PMLR (2022)
13. Margaritis, D., et al.: Learning Bayesian network model structure from data. Ph.D. thesis, School of Computer Science, Carnegie Mellon University Pittsburgh, PA, USA (2003)
14. Markus Kalisch, Martin Mächler, Diego Colombo, et al.: Causal inference using graphical models with the R package pcalg. Journal of Statistical Software **47**(11), 1–26 (2012). `https://doi.org/10.18637/jss.v047.i11`
15. Nandy, P., Hauser, A., Maathuis, M.H.: High-dimensional consistency in scorebased and hybrid structure learning. The Annals of Statistics **46**(6A), 3151–3183 (2018)
16. Obermeyer, Z., Powers, B., Vogeli, C., et al.: Dissecting racial bias in an algorithm used to manage the health of populations. Science **366**(6464), 447–453 (2019)
17. Pearl, J.: Causality. Cambridge University Press (2009)
18. Peters, J., Bühlmann, P.: Structural intervention distance (sid) for evaluating causal graphs (2014)
19. Sachs, K., Perez, O., Pe'er, D., et al.: Causal protein-signaling networks derived from multiparameter single-cell data. Science **308**(5721), 523–529 (2005). `https://doi.org/10.1126/science.1105809`
20. Scutari, M.: Learning bayesian networks with the bnlearn R package. Journal of Statistical Software **35**(3), 1–22 (2010). `https://doi.org/10.18637/jss.v035.i03`
21. Sharma, A., Kiciman, E.: Dowhy: An end-to-end library for causal inference (2020). `https://doi.org/10.48550/ARXIV.2011.04216`
22. Shimizu, S., Hoyer, P.O., Hyvärinen, A., et al.: A linear non-gaussian acyclic model for causal discovery. Journal of Machine Learning Research **7**(10) (2006)
23. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. MIT Press (2000)
24. Syrgkanis, V., Lewis, G., Oprescu, M., et al.: Causal inference and machine learning in practice with econml and causalml: Industrial use cases at microsoft, tripadvisor, uber. In: Proceedings of the 27th ACM SIGKDD. p. 4072–4073. KDD '21, ACM (2021). `https://doi.org/10.1145/3447548.3470792`
25. Tsamardinos, I., Aliferis, C.F., Statnikov, A.R., et al.: Algorithms for large scale markov blanket discovery. In: FLAIRS. vol. 2, pp. 376–81 (2003)
26. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing bayesian network structure learning algorithm. Machine learning **65**, 31–78 (2006)
27. Yuan, C., Malone, B.: Learning optimal bayesian networks: A shortest path perspective. Journal of Artificial Intelligence Research **48**, 23–65 (2013)
28. Zheng, Y., Huang, B., Chen, W., et al.: Causal-learn: Causal discovery in python (2023)

Causal Discovery: Improving Structure Learning using Knowledge and Synthetic data

José A. Montero[1], Julio Muñoz-Benítez[2], and L. Enrique Sucar[2]

[1] TecNM: Instituto Tecnológico de Acapulco, Calle Cayaco S/N, Puerto Marquez, Acapulco, México.
jose.mv@acapulco.tecnm.mx
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Luis Enrique Erro No.1, Sta. María Tonantzintla, Puebla, México.
{jcmunoz,esucar}@inaoep.mx

Abstract. Causal structure learning is fundamental for understanding complex processes to discover causal relationship between its variables. Traditional algorithms have relied in observational data and assumptions to face this task. However, to estimate reliable causal models is still challenging, in particular with limited data. In this work, we apply two approaches to improve the learning of causal structures using causal discovery algorithms. The first approach, utilizes as prior knowledge the results provided by four variable ordering-based methods. While in the second approach, we apply a novel method to generate synthetic data to expand datasets with small samples. Validation of results is realized by topological and statistical metrics. Preliminary results demonstrate that both approaches are promising.

Keywords: Causal Learning Algorithms · Ordering-based methods · Diffusion data generation.

1 Introduction

Causal Discovery, also known as causal structure learning, is a fundamental field of research in many disciplines related with sciences and engineering. It´s target consists on estimating causal relationships from observational or interventional data. A traditional way to uncover causal relationships is through randomized controlled experiments; this process is the golden standard for learning causality and estimating causal effects, in many real-world applications. However, very often it will be impractical due to its cost, ethical reasons, logistic limitations or even impossible to conduct [1]. Hence, the importance of causal discovery, i.e., the process of revealing causal information through the analysis of purely observational data, which has become increasingly apparent across diverse disciplines [2,3]. For example, in health area, causal discovery has been of great importance to understand the relationships between certain genes and diseases. For this purpose, the researchers have available repositories with biological data. However, is common to observe in these datasets that the number of variables exceed the sample size making it difficult to obtain a reliable causal model.

Current algorithms applied for causal discovery can be broadly classified into following categories [5]: a) Constraint-based, b) Score-based, and c) Functional causal models. Constraint-based and score-based methods have been employed for causal discovery since the 1990s, using conditional independence relationships in data to uncover information about the underlying causal structure. The algorithm representative of the Constraint-based class is the developed by Peter and Clark [6], it assumes causal sufficiency and is generally known as the PC algorithm. In cases without latent confounders, score-based algorithms like the Greedy Equivalence Search (GES) [7] aim to find the causal structure by optimizing a score function and select a graph among the Markov equivalence class. On the other hand, algorithms based on Functional Causal Models, like Direct LiNGAM [8] have exhibited the ability to distinguish between different Directed Acyclic Graphs (DAGs) within the same equivalence class, thanks to additional assumptions on the data distribution beyond conditional independence relations (linear relations among variables and non-Gaussian distribution in the noise signals). All algorithms depend on having *enough* data either to estimate independence relations or to calculate a global score.

Two alternatives in the case of scarce data are (i) incorporating expert knowledge, and (ii) generating synthetic data. The search space of causal discovery algorithms grows exponentially with the increase in the number of variables, the use of expert knowledge reduces this search space and guides the estimation of reliable models [4] [5] Another alternative is to generate synthetic data that provides a similar distribution to the real data, so the amount of data can be increased and in this way improve the performance of the causal discovery techniques. But a challenge is to generate data so that it is not identical to the real data, and at the same time, maintains the same causal relations.

Recently, there has been a growing body of research that justify the use of expert knowledge in causal discovery [9,10]. An alternative is to provide the topological ordering of the variables which reduces the search space, and helps to determine the causal relationships. We leverage this aspect and use this information as domain knowledge in conventional algorithms. Real data generated by randomized controlled experiments does not follow a predictable behavior in their distribution. So, we select some order-based methods with different data distributions, and to use the combination of the intermediate results as domain knowledge in conventional causal discovery algorithms.

It is important to note that these causal discovery methods benefit from using the largest possible amount of observational data. However, this is often not possible because the observed data are limited due to the nature of the phenomenon or activity of interest being analyzed, the high cost of repeating experiments, ethical factors that prevent the reproducibility of the experiment, or the difficulty of recreating the experiment under the same conditions. Due to this, as part of this work, a synthetic data generation model, based on a diffusion approach, was proposed to create datasets with a certain degree of variability by integrating various levels of noise. The objective was to obtain a larger amount of data that would closely resemble the original data.

In this work, we introduce two approaches to improve the learning of causal models using several causal discovery algorithms (PC, GES and DirectLiNGAM).

- We incorporate results provided by causal ordering-based algorithms, which will be used as domain knowledge. Then, we integrate this information as the adjacency matrix to determine the initial state of the causal graphs.
- We developed a tool based on diffusion techniques to generate synthetic data with similar characteristics to the real data.

Preliminary results indicates that both approaches are promising. The use of edges coincidences -to form the adjacency matrix- obtained from the full DAG in topological ordering step, by four order-based algorithms, improve in some cases the model generated by causal discovery algorithms. In the same sense, synthetic datasets generated by the proposed technique obtain comparable results as the real dataset, and when combined with real datasets surpass -in some cases- the estimated causal structure. Although these results are satisfactory, additional experiments with a larger collection of cases are considered as future work.

2 Related Work

There are several reasons that make the task of causal discovery difficult. In general it is not possible to learn all causal relationships from observational data alone; what we can recover is a set of statistically equivalent models known as a *Markov equivalence class*. Additionally, a large data set is required to provide reliable statistical measures for either local or global algorithms.

The work proposed by Heckerman and Geiger [15] is one of the earliest that considers prior knowledge in the structure learning process; it is a soft-constraint approach, where the learning process is guided to reward graphs that are closer to an initial best knowledge graph. One of the first implementations of hard constraints is presented in the score-based K2 algorithm [16], which assumes that the temporal order of the data variables is given, thereby restricting the search space of graphs to be consistent with the temporal ordering. Other relevant studies include the work of [17] on measuring the effect of knowledge about the existence or absence of edges, in addition to node ordering constraints, with application to score-based search methods and the constraint-based PC algorithm.

Causal discovery algorithms benefit by using the largest amount of data available. However, sometimes there is a limited amount of data. An alternative is to generate synthetic data based on the observable data; this is, create new data from the original data. In synthetic data generation, selecting the appropriate model is crucial to ensure that the generated data maintains the desired properties, such as realism, stability, and the inherent structure and relationships of the original data. In this context, various architectures have been employed with Diffusion Models [23], Generative Adversarial Networks (GANs) [21], and Variational Autoencoders (VAEs) [22] being some of the most prominent.

Diffusion models offer a significant advantage in terms of training stability compared to GANs [23,24], which are known for their challenging training process due to the adversarial nature of the method. This competitive process between the generator and the discriminator in GANs can lead to issues such as model collapse, where the generator only produces a limited subset of possible data [24]. On the other hand, although VAEs are more stable than GANs, they often generate less precise data due to the regularization of variability, which can limit the visual quality or accuracy of the generated data [23]. Diffusion models, however, avoid these issues by diffusing noise into the data during training and learning to reverse this process in a controlled manner. A crucial aspect of diffusion models is their ability to capture both the global and local structure of the data, making them especially suitable for applications where preserving relationships and structural coherence is essential. While GANs are effective in capturing complex patterns, they may struggle to maintain the global structure of the data. VAEs, though useful for representing the overall distribution, may sacrifice fine details due to regularization [25].

In this regard, diffusion models stand out for offering a more interpretable process in data generation, making easier to understand how synthetic data is generated from a probabilistic perspective. Thus, diffusion models represent a robust and versatile option for synthetic data generation, overcoming some of the inherent limitations of GANs and VAEs. Additionally, their stability during training, ability to preserve data structure and relationships, and the quality and realism of the generated data make them an ideal approach in situations where these characteristics are critical.

3 Methodology

3.1 Incorporating Variable Order as Prior Knowledge

We used as prior knowledge the ordering of variables from the following algorithms: 1) DiffAN [11]. This algorithm realizes the ordering of variables in observational datasets considering the functional relations to be constrained as nonlinear with additive noise (ANM). DiffAN is a topological ordering algorithm that leverages diffusion techniques for learning a Hessian function to find leaf nodes. 2) SCORE [12]. Recovers causal graphs using the score of the data distribution in non-linear Gaussian models. In order to infer the topological ordering, SCORE estimates the Hessian matrix of the log-likelihood. Then, it finds a leaf node by taking the arg min of the variance over the diagonal elements of the Hessian matrix. 3) CAM [13]. This method was specifically designed for high-dimensional data distributions in non-linear additive models represented by structural equation models. In CAM the topological ordering is inferred by finding the permutation of the graph nodes corresponding to the fully connected graph that maximizes the log-likelihood of the data. 4) NoGAM [14]. NoGAM introduces a topological ordering procedure that does not assume any distribution of the noise terms, the data distribution follows an additive behavior.

DiFFAN, SCORE, CAM and NoGAM algorithms perform causal discovery in a two steps procedure. Given observational i.i.d. data from an Additive Noise

Model without latent confounders, first the method estimates a topological ordering of the causal variables. This partial ordering can be represented as a fully connected graph, where every node has an incoming edge from all its predecessors in the ordering. Figure 1 shows the Full DAG obtained in this step by the algorithm SCORE. The topological ordering of the variables estimated in this step by the SCORE algorithm is: *PKC, Erk, Akt, P38, Jnk, Plcg, Mek, PIP3, Raf, PIP2, PKA*. In the same way and as shown graphically with the SCORE algorithm, the other algorithms estimate the topological ordering of the variables that make up the dataset used.

According with some authors [11,12], the knowledge of the topological ordering is already sufficient for estimating causal effects, so, we are going to take advantage of this aspect and use this information as *a priori* knowledge to determine the initial state in the mentioned causal discovery algorithms. We use the common edges obtained in the first step from the order-based algorithms previously mentioned. In this stage, each algorithm obtains a full DAG, where we can see the hierarchy of the vertices, see Figure 1.

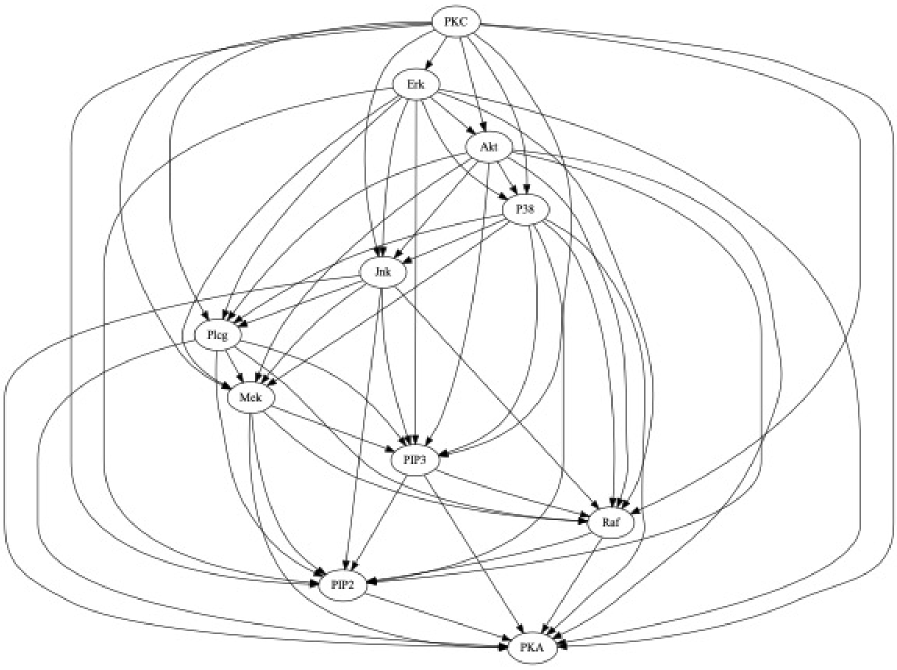


Fig. 1: Full DAG for the Sachs dataset (described in Section 4.1) generated in the ordering phase by the SCORE Algorithm [12].

3.2 Generation of Synthetic Data

In this work, we implemented a diffusion based neural network to generate synthetic data from an original dataset. This approach allowed to generate new samples that maintain inherent relationships in the original data while introducing controlled variations. The diffusion model was built using the TensorFlow library and consists of a deep neural network with multiple dense layers. The architecture includes five hidden layers, designed to effectively capture the complex relationships present in the data. The first three layers contain 128 neurons each and use the *Rectified Linear Unit* (ReLU) as the activation function to capture the complex nonlinear relationships present in the data. The next two layers have 64 neurons each, maintaining the model's capacity to learn finer patterns and specific details. Finally, the output layer has as many neurons as there are variables in the original dataset, allowing the model to generate synthetic data with the same dimensionality, ensuring that the generated data retains the fundamental characteristics of the original dataset. The original data was normalized to asset faster and more stable convergence. During the training stage, the model learns to replicate the distributions of the original data. Once training is complete, the model is able to generate new data that are consistent with the patterns and relationships found in the original data.

To introduce variability Gaussian noise was applied to the data generated by the model. This noise was introduced at different intensity levels (2, 5, 10, 15 and 20%) to simulate varying degrees of uncertainty. These noise levels allowed to vary the similarity of the synthetic data with the original; in the experimental results we evaluate the impact of the noise levels on learning the causal model.

3.3 Implementation

In this work we rely -with light modifications- on the package developed by Zheng et al. [18] called *causal-learn* that provide access to a range of hard knowledge constraints that can guide structure learning. This is a freeware that extends Tetrad [19]. In causal-learn, users can specify knowledge in the form of required and forbidden edges, via the adjacency matrix.

4 Experiments and Results

The results obtained applying the two approaches previously mentioned are presented in this section. First we describe the data sets used. Then we present results generated by the causal discovery algorithms incorporating as prior knowledge the variables' order. Next, we show the results obtained by combining the real and synthetic data.

4.1 Datasets

In this work, we used two datasets that are commonly used to evaluate causal discovery algorithms [11,12,13], SACHS and GAUSSIAN.

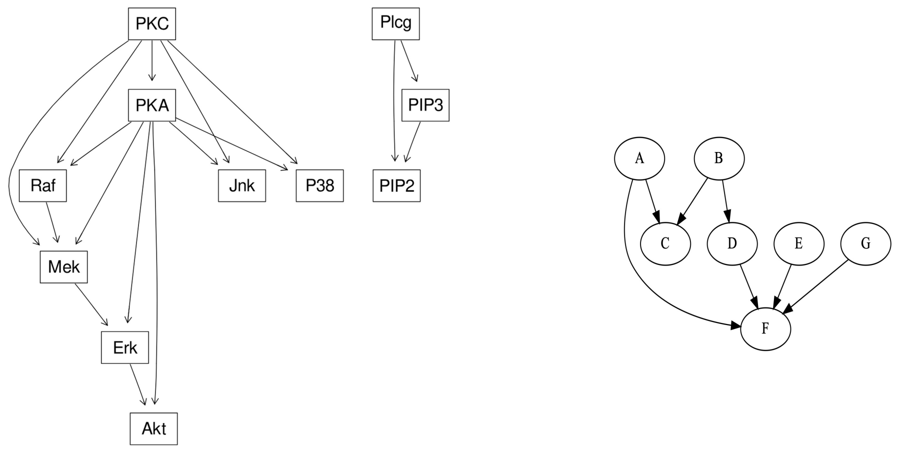


Fig. 2: Benchmark causal graphs that correspond to the Sachs (a) and Gaussian (b) datasets.

SACHS. It is a real dataset that measures the expression levels of multiple phosphorylated proteins and phospholipid components in human cells related with the immune system. It has a small network with 11 nodes and 17 edges (Figure 2(a)). The dataset has both observational and interventional samples. Most causal discovery approaches [11,12,13] use the dataset with 853 observational samples to evaluate their method. In this work, we used the same dataset[3].

GAUSSIAN. The Gaussian dataset is integrated by 5000 samples and 7 variables of synthetic data. This is a small network of 7 nodes and 7 edges (Figure 2(b)), its distribution data has a Gaussian behavior[4].

4.2 Results generated using variable orderings

In this experiment we create the adjacency matrix that represents the knowledge obtained from the results generated in the first step of several order-based algorithms (DiffAN, SCORE, CAM, NoGAM); that means, the full DAG where the hierarchical order of the variables is observed (see Figure **??**). For each algorithm, the order generated is always the same, with exception of DiffAN; in this case we realized five repetitions and chose the variable that appeared most times in each position. We compared the causal graphs generated by the original causal discovery algorithms vs. the models obtained by incorporating the variable order, applied in two datasets: Sachs and Gaussian. The results are presented in Tables 1 and 2.

[3] Can be obtained from: `https://github.com/snarles/causal/blob/master/bnlearn_files/sachs.data.txt`.

[4] This network can be obtained from: `https://www.bnlearn.com/documentation/man/gaussian-test.html`.

Table 1: Metrics obtained for the Sachs dataset without and with prior knowledge. Improved results are boldfaced.

Algorithm/Metric	SHD	Precision	Recall
PC without Expert Knowledge	14	0.5433	0.5433
PC using Expert Knowledge	20	0.3076	0.2766
GES without Expert Knowledge	12	0.6153	0.5433
GES using Expert Knowledge	17	0.4444	0.5333
DirectLinGAM without Expert Knowledge	15	0.5	0.3333
DirectLinGAM using Expert Knowledge	**14**	**0.5433**	**0.5433**

Table 2: Metrics obtained for the Gaussian dataset without and with prior knowledge. Improved results are boldfaced.

Algorithm/Metric	SHD	Precision	Recall
PC without Expert Knowledge	5	0.5833	0.9785
PC using Expert Knowledge	9	0.3333	0.2857
GES without Expert Knowledge	11	0.1666	0.1428
GES using Expert Knowledge	**9**	**0.3333**	**0.2857**
DirectLinGAM without Expert Knowledge	6	0.5555	0.7142
DirectLinGAM using Expert Knowledge	**5**	**0.8777**	**0.2857**

According to these experiments, we observe some improvement in SHD in the case of DirectLinGAM for Sachs, and GES and DirectLinGAM for Gaussian. Figure 3 illustrates the graphs for Sachs estimated by the causal discovery algorithms (PC, GES, Direct-LiNGAM), when prior knowledge is considered.

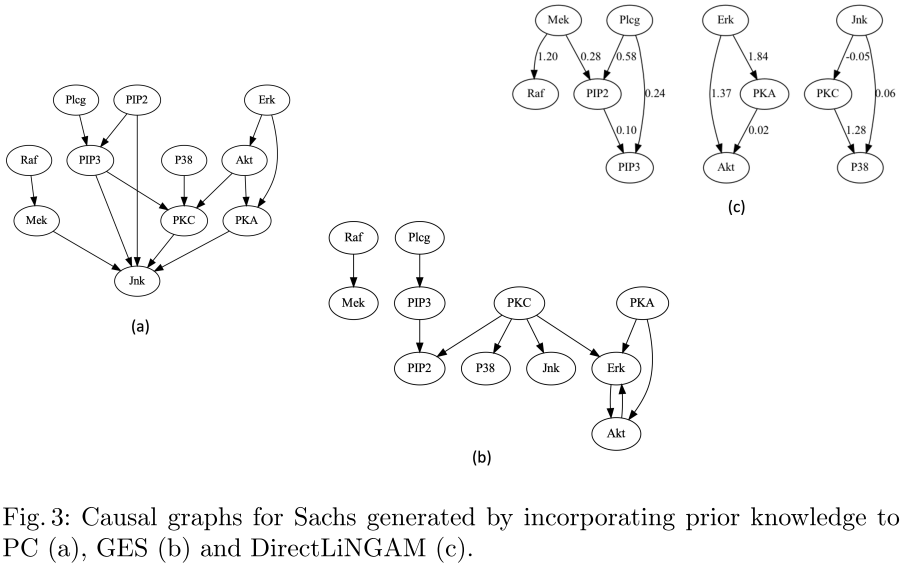


Fig. 3: Causal graphs for Sachs generated by incorporating prior knowledge to PC (a), GES (b) and DirectLiNGAM (c).

4.3 Results obtained using synthetic data

In this section we present the results obtained by generating synthetic data for the Sachs dataset. The model generated 2000 values for each of the 11 variables based on the data of the original SACHS dataset, adding the following noise percentages: 1, 5, 10, 15, 20 and 25. Real and synthetic datasets were utilized individually and mixed taken different proportions of each one. Table 3 depicts the results using the PC and GES algorithms with only synthetic data with a sample size of 853. According this results, we observe that the synthetic dataset equals the values obtained when using the original dataset in three cases (noise of 1%, 5%, and 10% (only the PC algorithm)). Therefore, we can assume that the synthetic data presents a similar behavior to the real data for some noise levels.

Table 3: Metrics obtained applying PC and GES with only synthetic data. Results equal to those obtained with the original dataset are boldface.

Noise percentage / Metrics	SHD	Precision	Recall
Noise 1%			
PC	**14**	**0.5333**	**0.5333**
GES	**12**	**0.6363**	**0.4666**
Noise 5%			
PC	**14**	**0.5333**	**0.5333**
GES	**12**	**0.6666**	**0.4**
Noise 10%			
PC	**14**	**0.5384**	**0.4666**
GES	13	0.5714	0.5333
Noise 15%			
PC	16	04615	0.4
GES	14	0.5384	0.4666
Noise 20%			
PC	16	0.4615	0.4
GES	13	0.5625	0.6

The results of the models estimated by the PC, GES and Direct-LiNGAM algorithms, when using combined data from the Sachs dataset with 853 samples and synthetic data gradually added in different proportions, are shown in Tables 4 (5% noise), 5 (10% noise), and 6 (15% noise). The metrics indicate that the PC algorithm outperform the results by using only real data in four combinations, GES improves in one combination and the Direct-LiNGAM outperform the metrics reached in real dataset in two combinations. According to this results, synthetic data has the potential to improve the performance when combined with real data, depending on the amount of noise and the algorithm. Additional results are presented in Appendix A.

Table 4: Results generated by using the original Sachs dataset and adding gradually synthetic data with 5% noise. The algorithms applied are PC, GES and DirectLiNGAM. Improved values are boldfaced.

Data and Algorithms / Metrics	**SHD**	**Precision**	**Recall**
10% synthetic data added			
GES	**12**	**0.6153**	**0.5333**
PC	**12**	**0.6**	**0.6**
DirectLiNGAM	**14**	**0.5555**	**0.3333**
20% Synthetic data added			
GES	**12**	**0.6153**	**0.5333**
PC	**14**	**0.5333**	**0.5333**
DirectLiNGAM	**14**	**0.5555**	**0.3333**
50% synthetic data added			
GES	15	0.5	0.4666
PC	15	0.5	0.6
DirectLiNGAM	16	0.4285	0.2
70% Synthetic data added			
GES	17	0.4166	0.333
PC	16	0.47	0.5333
DirectLiNGAM	18	0.333	0.2

Table 5: Results generated by using the original Sachs dataset and adding gradually synthetic data with 10% noise. The algorithms applied are PC, GES and DirectLiNGAM. Improved values are boldfaced.

Data and Algorithms / Metrics	**SHD**	**Precision**	**Recall**
10% synthetic data added			
GES	**12**	**0.6153**	**0.5333**
PC	**14**	**0.5294**	**0.6**
DirectLiNGAM	16	0.4444	0.2666
20% Synthetic data added			
GES	**12**	**0.6153**	**0.5333**
PC	**12**	**0.5882**	**0.6666**
DirectLiNGAM	**15**	**0.5**	**0.333**
50% synthetic data added			
GES	**11**	**0.6428**	**0.6**
PC	16	0.4719	0.6666
DirectLiNGAM	17	0.4	0.2666
70% Synthetic data added			
GES	**12**	**0.6153**	**0.5333**
PC	**13**	**0.5555**	**0.6666**
DirectLiNGAM	16	0.4444	0.2666

Table 6: Results generated by using the original Sachs dataset and adding gradually synthetic data with 15% noise. The algorithms applied are PC, GES and DirectLiNGAM. Improved values are boldfaced.

Data and Algorithms / Metrics	**SHD**	**Precision**	**Recall**
20% Synthetic data added			
GES	13	0.5833	0.5333
PC	14	0.5333	0.5333
DirectLiNGAM	17	0.4	0.2666
50% synthetic data added			
GES	13	0.5714	0.5333
PC	**13**	**0.5714**	**0.5333**
DirectLiNGAM	**15**	**0.5**	**0.3333**
70% Synthetic data added			
GES	**12**	**0.6**	**0.6**
PC	15	0.5	0.6
DirectLiNGAM	**15**	**0.5**	**0.3333**

5 Conclusions

The present work proposed two approaches to deal with scarce data in causal discovery. First, we incorporate as prior knowledge the variable order provided by four ordering-based algorithms (DiffAN, SCORE, CAM, NoGAM) into different causal discovery algorithms. The results obtained with the Sachs and Gaussian datasets, demonstrate that in some cases it is possible improve the results of the estimated graph. In the second approach, we generate synthetic data based on a diffusion technique. The experimental results indicate that: i) only synthetic data is capable of generating results comparable with the real data; ii) the combination of real and synthetic data can improve the results with only real data, the gain depends on the algorithm and the amount of noise.

As future work we plan to combine both approaches and apply them to learning the structure of gene regulatory networks.

References

1. David B Resnik. Randomized controlled trials in environmental health research: ethical issues. Journal of environmental health, 70(6):28, 2008.
2. Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. Frontiers in genetics, 10:524, 2019.
3. A. R. Nogueira, J. Gama and C. Ferreira. Causal discovery in machine learning: Theories and applications. Journal of Dynamics and Games, 8(3):203, 2021.
4. Judea Pearl. Causality: models, Reasoning and Inference. Cambridge university Press, 2000. (2nd ed. 2009).
5. Wang Z, Gao X, Yang Y, Tan X and Chen D. Learning Bayesian Networks based on order graphs with ancestral constraints. Knowledge-Based Sys, 211(1):1-12, 2021.
6. Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. Causation, prediction, and search. MIT press, 2000.

7. David Maxwell Chickering. Optimal structure identification with greedy search. Journal of machine learning research, 3(Nov):507–554, 2002.
8. S. Simizu, T. Inazumi, Y. Sogawa, A. Hyvarinem, P. Hoyer and K. Bollen. DirectLiNGAM: A Direct Method for Learning a Linear Non-Gaussian Structural Equation Model. Journal of Machine learning Research, 12, 1225-1248.2011.
9. A. Constantinou, Z. Guo and N. Kitson. The impact of prior knowledge on causal structure learning. Knowledge and Information Systems, 65:3385-3434, 2023, htps://doi.org./10.1007/s10115-023-01858-x
10. M. Munch, P. Buche, H. Angellier-Coussy, C. Manfredotti and P-H Wuillemin. Formalising contextual expert knowledge for causal discovery in linked knowledge graphs about transformation process: application to processing of bio-composites for food packing. Internatiomnal Journal of Metadata, Semantics and Ontologies, 2022, 16(1), pp. 1-15. https://hal.science/hal-04115029.
11. P. Sanchez, X. Liu, A. O´Neil and S. Tsaftaris. Diffusion Models for Causal Discovery via Topological Ordering.ICLR23, pp. 1-20, 2023.
12. P. Buhlmann, J. Peters and Jan Ernest CAM: Causal additive models, highdimensional order search and penalized regression. The Annals of Statistics, 42(6):2526–2556, 2014.
13. F. Montagna, N. Noceti, L. Rosasco, K. Zhang and F. Locatello. Scalable Causal Discovery with Score Matching. 2nd Conference on Causal Learning and Reasoning, Proc. Of Machine Learning Research, Vol 1, pp. 1-20, 2023.
14. F. Montagna, N. Noceti, L. Rosasco, K. Zhang and F. Locatello. Causal Discovery with Score Matching on Additive Models with Arbitrary Noise. 2nd Conference on Causal Learning and Reasoning, Proc. Of Machine Learning Research, Vol 213, pp. 1-24, 2023.
15. Heckerman D, Geiger D. Learning Bayesian networks: a unification for discrete and Gaussian domains. In: Proceedings of the 11th conference on uncertainty in artificial intelligence (UAI95). San Francisco, pp 274–284, 1995.
16. Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Mach Learn 9(4):309–347, 1992.
17. de Campos LM, Castellano JG. Bayesian network earning algorithms using structural restrictions. Int J Approx Reason 45:233–254, 2007.
18. Zheng, Yujia, Huang, Biwei, Wei, Ramsey, Spirtes, Gong, Shimizu and Zhang. Causal-learn: Causal Discovery in Python. Journal of Machine Learning Research, Vol. 25, No. 60, pp. 1-8, 2024.
19. Center for Causal Discovery.Tetrad manual and code. Retrieved in May 2024. `https://www.ccd.pitt.edu/tools/`.
20. K. Sachs. Causal-protein signaling networks derived from multiparameter single cell data. Science 308, 523(2005), DOI: 10.1126/science.1105809.
21. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” 2014
22. D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” arXiv preprint arXiv:1312.6114, 2013
23. J. Ho, A. Jain and P. Abbeel, "Denoising diffusion probabilistic models", Proc. Int. Conf. Neural Inf. Process. Syst., pp. 6840-6851, 2020.
24. F. -A. Croitoru, V. Hondru, R. T. Ionescu and M. Shah, "Diffusion Models in Vision: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 9, pp. 10850-10869, 1 Sept. 2023, doi: 10.1109/TPAMI.2023.3261988.
25. Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” arXiv preprint arXiv:2011.13456, 2020.

A Additional Results with Synthetic Data

The values reported in Tables 7 and 8 were obtained by combining real and synthetic data with different noise proportions, applying the PC and GES algorithms, respectively. The combined data equals the number of 853 samples. It is observed that the PC algorithm equals, but does not exceed, the value generated with only real data. When the GES algorithm is applied to these combined data, it is observed that in two combinations (noise 10% and noise 15%) the value obtained by using only real data is outperformed.

Table 7: Metrics obtained applying the PC algorithm and combining real with synthetic data considering different proportions of noise. Sample size with combined dataset is 853.

Noise percentage / Metrics	SHD	Precision	Recall
Noise 1%			
Real20%-Synthetic80%	**14**	**0.5333**	**0.5333**
Real50%-Synthetic50%	15	0.5	0.4666
Real80%-Synthetic20%	15	0.5	0.4666
Noise 5%			
Real20%-Synthetic80%	17	0.4285	0.4
Real50%-Synthetic50%	**14**	**0.5333**	**0.5333**
Real80%-Synthetic20%	16	0.4615	0.4
Noise 10%			
Real20%-Synthetic80%	18	0.4285	0.6
Real50%-Synthetic50%	**14**	**0.5333**	**0.5333**
Real80%-Synthetic20%	16	0.4666	0.4666
Noise 15%			
Real20%-Synthetic80%	15	0.5	0.5333
Real50%-Synthetic50%	16	0.5333	0.5333
Real80%-Synthetic20%	**14**	**0.5333**	**0.5333**
Noise 20%			
Real20%-Synthetic80%	15	0.5	0.4666
Real50%-Synthetic50%	17	0.45	0.6
Real80%-Synthetic20%	16	0.4666	0.4666
Noise 25%			
Real20%-Synthetic80%	18	0.4285	0.6
Real50%-Synthetic50%	18	0.421	0.5333
Real80%-Synthetic20%	20	0.3809	0.5333

Table 8: Metrics obtained applying the GES algorithm and combining real with synthetic data considering different proportions of noise. In bold appears the combination that reach the same result as real dataset and in cursive bold the combination that improve the use of only real dataset.

Noise percentage / Metrics	**SHD**	**Precision**	**Recall**
Noise 1%			
Real20%-Synthetic80%	14	0.5333	0.5333
Real50%-Synthetic50%	**12**	**0.6153**	**0.4666**
Real80%-Synthetic20%	**12**	**0.6153**	**0.4666**
Noise 5%			
Real20%-Synthetic80%	13	0.6	0.4
Real50%-Synthetic50%	**12**	**0.6153**	**0.5333**
Real80%-Synthetic20%	**12**	**0.6353**	**0.4666**
Noise 10%			
Real20%-Synthetic80%	***11***	***0.625***	***0.666***
Real50%-Synthetic50%	**12**	**0.6153**	**0.5333**
Real80%-Synthetic20%	**12**	**0.6353**	**0.4666**
Noise 15%			
Real20%-Synthetic80%	**12**	**0.6363**	**0.4666**
Real50%-Synthetic50%	***10***	***0.7272***	***0.5333***
Real80%-Synthetic20%	14	0.5333	0.5333
Noise 20%			
Real20%-Synthetic80%	14	0.5333	0.5333
Real50%-Synthetic50%	**12**	**0.6153**	**0.5333**
Real80%-Synthetic20%	**12**	**0.6153**	**0.5333**
Noise 25%			
Real20%-Synthetic80%	16	0.4666	0.4666
Real50%-Synthetic50%	14	0.5384	0.4666
Real80%-Synthetic20%	14	0.5384	0.4666

One-Class Learning for Text Causal Discovery through Hypergraph Neural Networks

Marcos Paulo Silva Gôlo[1] and Ricardo Marcondes Marcacini[1]

Institute of Mathematical and Computer Sciences, University of Sao Paulo
marcosgolo@usp.br and ricardo.marcacini@icmc.usp.br

Abstract. We explore the problem of causal discovery between text pairs. We propose a new method called eCOLGAT (**e**dge **C**lassification through **O**ne-c**L**ass **G**raph **AT**tention autoencoder) that exploits hypergraphs to better learn the representation of edges (causal relations), graph attention networks to perform edge classification in causal graphs, one-class learning to better model the problem and reduce the labeling effort, and interpretability to improve the understanding of the causal discovery learning process. eCOLGAT outperformed other one-class methods and large language models (state-of-the-art for causal discovery), proving to be a promising method for causal discovery in text pairs.

Keywords: Event Causal Discovery · One-Class Classification · Graph Neural Networkds · Text Pair Causal Discovery.

1 Introduction

Understanding an event's causal relations is a challenging task that directly impacts society because this information can be applied in government analyses through decision-makers to reduce harm to society [11]. Although the vast majority of research and applications in causal discovery focus on effect inference tasks, the growing prevalence of textual event reporting in society has introduced new challenges and opportunities. Thus, there is an increasing demand for causal discovery methods specifically tailored to textual data. In particular, we are interested in binary classification tasks that address the causal relationship between two textual statements. For example, does a "*weakening economic environment*" cause "*rising unemployment rates*"? This task is called causal discovery in text pairs in the machine learning literature [8, 13, 12].

Studies have performed textual analysis to discover causality between text pairs by exploiting the Bidirectional Encoder Representations from Transformers (BERT) model [8, 13, 12]. Furthermore, considering text-based models, the state-of-the-art (SOTA) models are Large Language Models (LLMs) [16]. Even obtaining SOTA results, LLMs do not observe the causal relation between sentences, since each pair of texts is analyzed individually. This fact can harm the performance of the models since modeling these relations can allow the models to explore more information. Graphs are an alternative since graphs model this task naturally because each node in the graph is a text, and an edge is created between the nodes when there is a causal relation between the texts [28].

Graph Neural Networks (GNNs) have been widely used to discover cause-effect relations in text pairs modeled by graphs [20, 22, 23]. However, we highlight the limitation of GNNs in learning representations for edges due to biased message passing for node representation learning [10]. Furthermore, GNN works for causal discovery are typically based on binary learning, i.e., during training they learn from labeled instances of causal and non-causal classes. We highlight some gaps of binary learning, such as the need for a significant amount of labeled instances for the algorithm learning step. Furthermore, we highlight the large scope of non-causal relations, which makes labeling challenging [6].

We present a new method for edge classification through one-class learning (OCL) in graphs. The method is called eCOLGAT (**e**dge **C**lassification through **O**ne-c**L**Ass **G**raph **AT**tention autoencoder). eCOLGAT is based on hypergraphs that improve representation learning through GNNs for edges since they transform edges into nodes and thus GNNs can learn better representations [10]. Furthermore, we propose to model the problem of causal discovery in text pairs through OCL since, in OCL, the algorithm trains with only one class (causal relations) and can predict two (causal or non-causal relations). In this way, covering the entire scope of non-causal relations is unnecessary, and the user's labeling effort is reduced since it only labels causal instances [24, 4]. In this sense, we base eCOLGAT on the SOTA loss functions for GNNs and OCL [7, 29]. Finally, we explicitly learn three-dimensional representations to introduce interpretability in eCOLGAT naturally. In summary, our contributions are:

1. We model causal discovery between text pairs through **one-class learning**, providing labeling advantages and more natural modeling for the problem.
2. We model causal discovery between text pairs through **hypergraphs** to better exploit graph neural networks' representation learning for edges.
3. We introduce **interpretable** representation learning on hypergraphs through graph neural networks and one-class learning for causal discovery.

2 Related Work

Hassanzade et al. [8] is one of the pioneering studies in causal discovery between text pairs using language models based on deep neural networks. The authors propose an unsupervised method based on the pre-trained BERT language model called NLM-BERT. First, the authors generated embeddings for a corpus of 17 million causal sentences using Bidirectional Encoder Representations from Transformers (BERT). Second, the method generates embeddings for text pairs to answer whether the first text causes the second. Using a technique based on cosine similarity between the top k similar embeddings, NLM-BERT generates two scores and compares these values with a threshold to decide whether there is causality between the texts. The authors obtained an f1-score of 67%, outperforming four other methods in the Risk Models dataset.

Kayesh et al. [13] proposed fine-tuning the BERT model and its variations to detect causality. The authors fine-tuned with a semi-supervised dataset of 100,000 pairs of causal and non-causal sentences. The authors compared their results with the results of Hassanzade et al. [8]. The developed methods could not outperform NLM-BERT, obtaining an f1-score of 66% on the Risk Models dataset. In the same line of research, Kayesh et al. [12] extended the work of Kayesh et al. [13] by adding another training dataset of 197,000 sentence pairs, three new methods, and three combinations of these methods to detect causality, and used a causality graph through each of the training sets (197,000 and 100,000 sentence pairs). The proposal of Kayesh et al. [12] is based on a knowledge fusion that fuses representations generated in two stages: extraction of causal features (embedding generation through the graph) and extraction of contextual features (embedding generation through attention mechanisms). For the Risk Models dataset, the new model trained on the set of 100 thousand sentence pairs could not outperform NLM-BERT, obtaining 66% of f_1.

The state-of-the-art (SOTA) for causal discovery in textual data is through large language models (LLMs) [16]. LLMs are pre-trained models on a corpus with trillions of words capable of generating text from an input text. In this sense, LLMs can be queried in natural language whether one sentence causes another to work as causal discovery models. In this sense, the PyWhy-LLM library was developed[1]. Other methods that also obtain SOTA results for causal discovery are graph neural networks (GNNs) methods since graphs express the causal relations explicitly, and GNNs obtain SOTA results in graphs [20, 22, 23].

Minghim et al. [20] obtain contextualized embeddings for the words used to build a causal graph. The authors used a gated GNN and recurrent neural network decoder for graph neural network encoding. Finally, a fully connected neural network was used for Event Causality Identification binary classification. The model outperformed baselines, including the BERT model. Sakaji et al. [22] is another study that explores BERT and GNNs. This work explored Graph Attention Networks, pointing out improvements in relation to graph convolutional networks. Finally, the authors used two fully connected neural networks to classify sentences as cause and effect separately. The model outperformed baselines, including the BERT model. Finally, Sasaki et al. [23] perform edge classification in the causal graph through GNNs, in which the edges are the causality relations and the nodes are the sentences. A classifier was used in the final layer of the GNN. The authors do not compare with baselines but obtain 98% of f_1.

Text-based methods (BERT and LLM) do not explore information about the relations between sentences because they do not model the data through graphs. Even using graphs in the strategy, Kayesh et al. [12] do not use graphs as the main part of the method nor GNNs, obtaining inferior results than text-based methods. GNNs explore this relational information and can explore textual information because they need an initial representation for the texts. On the other hand, GNNs focus on node message passage, i.e., in the sentences and not in edges, in which the classification will be performed, thus making causal

[1] https://github.com/py-why/pywhy-llm/tree/main

discovery difficult. Furthermore, we highlight that GNNs are explored as binary supervised methods, i.e., they require labeling of causality and non-causality, making knowledge discovery difficult since labeling what is not causal is costly due to lack of scope. In this sense, in the next section, we present a method based on hypergraphs and one-class learning for causal discovery in text pairs.

3 eCOLGAT: edge Classification through One-cLAss Graph ATtention autoencoder

We propose a novel method called **e**dge **C**lassification through **O**ne-c**L**Ass **G**raph **AT**tention autoencoder (eCOLGAT). Our method for causal detection in text pairs is based on hypergraphs, one-class learning (OCL), and graph neural networks (GNNs). Our method presents several novelties for causal discovery between text pairs. First, eCOLGAT is a pioneering one-class method for causal discovery. Second, using hypergraphs with graph neural networks for causal discovery in text pairs is also novel. Third, exploiting three-dimensional representation learning to provide interpretability for causal discovery in text pairs is another novelty of our method.

Hypergraphs lead better with the edge-representation gap through GNNs. eCOLGAT explores a state-of-the-art one-class loss function to encapsulate causal relations closer to the center within the sphere. Our proposal learns a new three-dimensional latent space to provide interpretable learning, where causal relations are positioned inside a sphere and non-causal relations outside. We learn the new space through a graph attention autoencoder to explore the reconstruction loss as a constraint to the sphere loss function and the attention mechanism to learn better representations for the edges. Finally, our final loss function combines the sphere loss function with the reconstruction loss function. Figure 1 presents an eCOLGAT illustration.

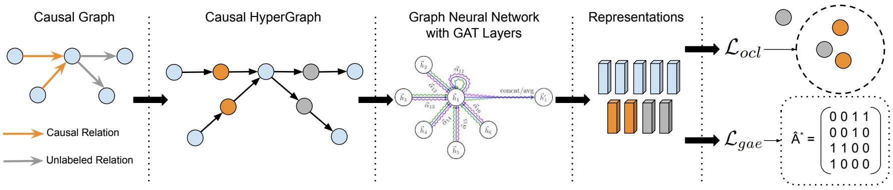


Fig. 1. Our proposed method eCOLGAT. We show all the steps from eCOLGAT: hypergraph generation, representation learning through GAT, one-class sphere loss ($\mathcal{L}_{ocl}$), and GAE loss ($\mathcal{L}_{gae}$).

Causal discovery between text pairs can be defined as a binary classifier with two inputs that output a causal or non-causal label. We define $\{s_1, s_2, s_3, ..., s_m\} \in \mathcal{S}$ as a set of m natural language sentences and $\{causal, non-causal\} \in \mathcal{C}$ as the set of classes. This classifier can be defined as a function $f : \mathcal{S} \rightarrow \mathcal{C}$ that maps two text pairs $(s_i, s_j) \in \mathcal{S}$ to the causal or non-causal label. In this case, $\boldsymbol{S} \in \mathbb{R}^d$ represents the sentence feature space. We define one-class learning for causal discovery on text pairs as the function f^* from a training set with only causal labels $\{((s_1, s_2); causal), ((s_5, s_3); causal), ..., ((s_i, s_j); causal)\}$ that approximates the unknown mapping function f.

In the context of this work, $\boldsymbol{S} \in \mathbb{R}^d$ will be a representation generated by Bidirectional Encoder Representations from Transformers (BERT) for each of our sentences, where d has value 384 [21]. Given the limitations of text-based models, this study models causality pairs through a directed graph. A directed graph is formally defined as $\mathcal{G} = (\mathcal{V}, \boldsymbol{A})$, where each node $v_i \in \mathcal{V}$ and $\boldsymbol{A}$ is the adjacency matrix containing relations between nodes. Thus, $\mathcal{V} \equiv \mathcal{S}$, i.e., the sentences are the nodes, and $\boldsymbol{A}$ contain the causal and non-causal relations [17]. Our method also exploits GNNs, state-of-the-art methods for different tasks, including node classification [9]. Due to the limitation of GNNs in learning edge representation and classification, this study explores the use of hypergraphs to model causality in graphs [10].

We transform each edge into a node to transform our directed graph into a directed hypergraph. Thus, when a node v_i was connected to node v_j, we create a node v_o (which we will call here node-edge) and connect v_i with v_o and v_o with v_j [10]. In this way, GNNs can be explored without the limitation of message passing for edges because the edges are now nodes. We define a hypergraph for causal discovery as $\mathcal{G}^* = (\mathcal{V}^*, \boldsymbol{A}^*)$, where $\mathcal{V}^*$ is the nodes set $\mathcal{V}$ plus the edge set of $\mathcal{G}$ and $\boldsymbol{A}^*$ is the adjacency matrix of the hypergraph with the new generated relations [10]. Another point is the need for GNNs in each node to have an initial representation. In the case of our proposal, part of the set $\mathcal{V}^*$ (the $\mathcal{V}$ set) has the initial representation of BERT, but the nodes-edge does not. Thus, to use GNNs, we add the average of the adjacent nodes-edge as an initial representation of the nodes. In the above example, $\boldsymbol{v_o} = avg(\boldsymbol{v_i}, \boldsymbol{v_j})$.

We exploit GNNs to learn representations in our causality hypergraph. The GNNs consider the structured representation of each node $\boldsymbol{v}_i \in \boldsymbol{V}^*$ and the adjacency matrix $\boldsymbol{A}^*$ as input for the representation learning process. Therefore, $g(\boldsymbol{V}^*, \boldsymbol{A}^*; \mathbf{W})$ represents a GNN with trainable weights $\mathbf{W} = \{\boldsymbol{W}^{(1)}, \cdots, \boldsymbol{W}^{(L)}\}$ in L hidden layers. Formally, for the l-th layer, the GNN propagation can be summarized as follows [26]:

$$\boldsymbol{H}^{(l+1)} = g(\boldsymbol{H}^{(l)}, \boldsymbol{A}^*; \boldsymbol{W}^{(l)}), \tag{1}$$

in which $\boldsymbol{H}^{(l)}$ is the input to the l-th GNN layer, and $\boldsymbol{H}^{l+1}$ is the output of this layer. The representations $\boldsymbol{V}^*$ are the inputs for the first layer, i.e., $\boldsymbol{V}^* \equiv \boldsymbol{H}^{(0)}$. In this sense, $\boldsymbol{H}^{(L)}$ are the learned embeddings for each node. GNNs learn representations $\boldsymbol{H}^{(L)}$ by aggregating information from neighbors.

We chose Graph Attention Networks (GAT) [3] as GNNs, given their improved performance in node classification tasks, including causality discovery [22]. The GAT learns the most important edges through the attention mechanism [3], i.e., the GAT has attention to the main relations in the graph, improving information aggregation. The GAT aggregation step can be defined by Equation 2 [3]:

$$\boldsymbol{h}_{v_i}^{l} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{v_j \in N(v_i)} \alpha_{ij}^{k}\boldsymbol{W}^{k}\boldsymbol{h}_{v_j}^{l-1}\right), \tag{2}$$

in which $\boldsymbol{h}_{v_i}^{l}$ is the aggregation result of the v_i neighbors defined as N_{v_i}, and $\boldsymbol{h}_{v_j}^{l-1}$ is the feature vector of the node v_j at the $l-1^{th}$ layer. $\boldsymbol{W}^k$ are the GAT weights associated with the head k. K is the number of heads in the GAT, and α_{ij}^{k} is the attention computed bu the $k-th$ attention head defined by the Equation 3.

$$\alpha_{ij} = \frac{exp(\boldsymbol{a}^{\intercal} LeakyReLU(\boldsymbol{W}\boldsymbol{h}_{v_i}^{l-1}||\boldsymbol{W}\boldsymbol{h}_{v_j}^{l-1}))}{\sum_{v_u \in N(v_i)} exp(\boldsymbol{a}^{\intercal} LeakyReLU(\boldsymbol{W}\boldsymbol{h}_{v_i}^{l-1}||\boldsymbol{W}\boldsymbol{h}_{v_u}^{l-1}))}, \tag{3}$$

in which, $\boldsymbol{a}$ is the shared attention mechanism, $(\cdot)^{\intercal}$ represents transposition and $||$ is the concatenation operation.

GNNs with sphere loss functions are state-of-the-art for one-class graph neural networks [27, 7]. These methods learn $\boldsymbol{h}_{v_i}^{L}$ encapsulating the nodes of interest. To detect causality through our hypergraph, we explore this strategy [7]. We use the sphere loss function $\mathcal{L}_{ocl}$ defined in Equation 4 [7].

$$\mathcal{L}_{ocl}(\boldsymbol{W}) = \frac{1}{|\mathcal{V}^{\text{in}}|}\sum_{i=1}^{|\mathcal{V}^{\text{in}}|}\begin{cases} o_i + 1, \text{ if } o_i > 0 \\ \exp(o_i), \text{otherwise} \end{cases}, \quad (4) \qquad o_i = \|\boldsymbol{h}_{v_i}^{(L)} - \boldsymbol{c}\|^2 - r^2. \quad (5)$$

in which equation 5 represents the value indicating whether the interest instance v_i is within the hypersphere with radius r and center $\boldsymbol{c}$ and $\mathcal{V}^{\text{in}}$ are the set of interest nodes-edges in our hypergraph (with the causal class).

By using only $\mathcal{L}_{ocl}$, all nodes will converge to the center. In this sense, following the work of [7], we use our GAT layer in a graph autoencoder (GAE) since GAEs have an unsupervised loss function that is a constraint to mitigate the collapse of the sphere. Thus, we combine the sphere loss function with the loss function of GAEs [7], which obtained superior and state-of-the-art results compared to other methods [27]. A GAE uses GNN layers as an encoder and an inner product of the latent representation as a decoder to learn node representations. Equation 6 describes a GAE [14]:

$$GAE = \begin{cases} Encoder : \boldsymbol{H}^{(L)} = g(\mathbf{V}^{*}, \boldsymbol{A}^{*}; \mathbf{W}) \\ Decoder : \hat{\boldsymbol{A}}^{*} = \sigma(\boldsymbol{H}^{(L)} \cdot \boldsymbol{H}^{(L)^{\intercal}}) \end{cases}, \tag{6}$$

in which $\sigma(.)$ is a logistic sigmoid function. The GAE loss function $\mathcal{L}_{gae}$ is defined in Equation 7 (binary cross entropy loss applied in the adjacency matrix). Therefore, our final loss function is defined by: $\mathcal{L} = \mathcal{L}_{ocl} + \mathcal{L}_{gae}$.

$$\mathcal{L}_{gae}(\boldsymbol{W}) = -\frac{1}{|V|}\sum_{i=0}^{|V|}\sum_{j=0}^{|V|}(\boldsymbol{A}_{ij}^{*} \cdot log\hat{\boldsymbol{A}}^{*}{}_{ij} + (1 - \boldsymbol{A}_{ij}^{*}) \cdot log(1 - \hat{\boldsymbol{A}}^{*}{}_{ij})). \tag{7}$$

We show the causal graph, causal hypergraph, the GAT step for representation learned in the hypergraph, the sphere loss, and GAE loss. Even though it is possible to understand the model's decision by observing a sphere and instances inside the sphere (causal nodes-edge) and outside the circle/sphere (non-causal nodes-edge), it is challenging to interpret the learning that generated this decision, since we cannot visualize the representations generated during learning if the dimension is greater than three. On the other hand, with representations in three dimensions, we can observe and interpret the representations generated during learning. In this sense, we bias eCOLGAT learning so that our method learns representations in size three to provide interpretability for representation learning in the scenario of one-class learning for causal discovery in text pairs.

4 Experimental Evaluation

This section presents the experimental evaluation of this article. We present the used dataset, experimental settings, results, and discussion. Our research goal is to demonstrate that our eCOLGAT proposal outperforms other SOTA methods for causal discovery in text pairs. Another goal is to demonstrate that our method learns low-dimensional representations, providing interpretability for the causal discovery scenario. The experimental evaluation codes are publicly available[2].

4.1 Dataset

There are some benchmark datasets for detecting causality between text pairs, such as the three explored by Hassanzadeh et al. [8]. We used the largest of them to perform the empirical evaluation. The other two are very small, as they have only 160 and 59 causal pairs, making them unfeasible to train our one-class learning model. In this sense, we explored the Risk Models dataset [8].

Hassanzadeh et al. [8] explored models built by expert analysts to configure a decision support system [25] as a source of causal knowledge by human experts. The models are graphs in which the nodes are texts represented by descriptions of conditions or events, and the edges show causal relations. The models are based on enterprise risk management, expert knowledge, literature study, and reports. The authors created the cause-effect pairs dataset by transforming each edge in the graph into a pair of texts with the causal label. Finally, the dataset has 368 causal pairs with 223 unique cause/effect sentences. For the non-interest class, the authors randomly chose 368 pairs to be non-causal.

4.2 Experimental Setting

We focus only on unsupervised and one-class methods due to the difficulty of creating a large enough training set with reasonable quality and coverage. We compare the eCOLGAT with the state-of-the-art text-based methods for causal

[2] `https://github.com/GoloMarcos/eCOLGAT`

discovery [16]. We compare our methods with five large language models using the strategy of Pywhy-LLM library. Different LLMs have been proposed in the last three years, and they have differences that generate advantages and disadvantages for each model [30]. We use the 5-fold cross-validation for our experiments. We use four folds of the causal class to train, the remaining fold to test, and one fold of the non-causal class to test. Finally, we use the f_1-macro to compare all models.

We explore in our methodology four families of LLMs open-source: the LLM from meta (LLaMa) [18], from Microsoft (Phi) [19], from Google (Gemma) [5], and from Alibaba Cloud (Qwen) [2]. Each LLM has a number of parameters: Llama 3 (8 and 70 billion of parameters), Phi 3 (14 billion of parameters), Gemma 2 (27 billion of parameters), and Qwen 2 (7 billion of parameters). We also compare eCOLGAT with one-class methods since we have an initial representation for each sentence (BERT embedding) and can generate an initial representation for causal relations (average of causal and effect sentences). In this sense, we explore two one-class methods: One-Class Support Vector Machines (OCSVM) [1] and Isolation Forest (IsoForest) [15]. We use the following parameters for the methods:

- **LLMs**: parameter-free;
- **OCSVM**: kernel = {rbf, poly, sigmoid, linear}, $\nu = \{0.05 * b\}, b \in [1..19]$, and $\gamma = \{$ scale, auto $\}$;
- **IsoForest**: n$^{\text{o}}$ of estimators = $\{1, 2, 5, 10, 50, 100, 200, 500\}$, maximum samples and maximum features = $\{0.1 * b\}, b \in [1..10]$;
- **eCOLGAT**: radius - $\{0.35, 0.45, 0.5\}$, epochs = $\{700, 1000, 1500\}$, heads for GAT = $\{1, 2, 3\}$, learning rate = $\{0.001, 0.0001, 0.0005\}$.

4.3 Results and Discussion

Table 1 presents the results of our study. We present the f_1-macro for all folds. We also present the average f_1 for the seven models explored and eCOLGAT. Higher values are in bold (best models). The second-highest values are underlined (second-best models). eCOLGAT outperforms the other two OCL methods since it obtained the higher f_1. OCSVM obtained the second-best results, followed by Phi 3 and LLaMa 3 (70b). LLaMa 3 (8b) obtains the worst f_1, followed by IsoForest and Qwen 2. Compared to LLM models, we highlight OCL models with state-of-the-art results and outperform these models.

Our method obtained the better f_1-macro in all analyzed folds compared to other methods. In addition, we obtained a 10% gain from the second-best model (OCSVM) and 25% from LLM models (Phi 3). It is worth mentioning that the second-best model also had a significant gain of 6% from LLM models, justifying the use of one-class learning for causal discovery in text pairs. Another point of attention in the results is Isolation Forest, which obtained a performance comparable with the other LLM algorithms. This shows that one-class learning is promising, but the choice of algorithm is important.

Table 1. f_1-macro for each method in the five folds and the average. The best results are in bold, and the second best are underlined.

Models	Folds					Average
	1	2	3	4	5	
LLaMa 3 (8b)	0.338	0.349	0.365	0.374	0.337	0.353
Phi 3 (14b)	0.652	0.612	0.580	0.585	0.657	0.618
Qwen 2 (7b)	0.497	0.467	0.516	0.492	0.524	0.499
Gemma 2 (27b)	0.451	0.521	0.502	0.502	0.556	0.508
LLaMa 3 (70b)	0.511	0.596	0.506	0.568	0.553	0.548
BERT + OCSVM	0.634	0.659	0.659	0.688	0.718	0.672
BERT + IsoForest	0.480	0.492	0.477	0.463	0.539	0.493
eCOLGAT	**0.764**	**0.811**	**0.707**	**0.766**	**0.804**	**0.771**

Typically, the number of LLM parameters influences the performance gain, as shown in other task performances considering LLMs. In the task explored in this study, models with fewer parameters such as Phi 3 obtained better performance than models with more parameters such as LLaMA 3 with 70 billion parameters or Gemma with 27 billion. Therefore, choosing LLMs for causal discovery in text pairs is also not trivial.

Figure 2 presents the confusion matrices for eCOLGAT and all baseline methods. The rows of the confusion matrix represent the true labels, while the columns represent the predicted labels. The cells of the main diagonal are the true positives (TP) and true negatives (TN), and the cells of the secondary diagonal are the false negatives (FN) and false positives (FP), where positive means our causal class and negative means the non-causal class. Thus, the higher the VP and VN values, the higher the accuracy, and the higher the FN and FP values, the higher the errors.

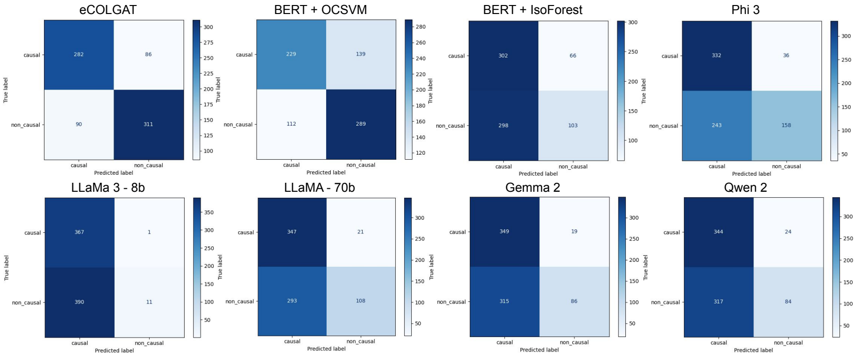


Fig. 2. Confusion Matrix for each method with the sum of the five folds. Higher values are in dark blue, and smaller values are in light blue.

eCOLGAT obtained the highest VN value and the lowest FP value, i.e., it correctly predicted more non-causal relations. LLaMA 3 (8b) obtained the highest VP value and the lowest FN value, i.e., it correctly predicted more causal relations. On the other hand, these correct predictions are due to the model's bias for this class. Note that it obtained the highest FP value and the lowest VN value, i.e., it predicted all instances for the causal class. This behavior was also observed in other models, such as IsoForest, Gemma 2, and Qwen 2. Even though it did not obtain the best VP and FN values, we emphasize that our model balanced its values, resulting in better classification performances.

We present the representations generated by the eCOLGAT learning to demonstrate the interpretability of our method. In this sense, Figure 3 presents the eCOLGAT representations for the node-edges in our hypergraph focusing on the learning process. In the real world, we can show the video of the learning process since we can plot all learning epochs without processing the representation (our learned representations have three dimensions). The epochs are 30, 70, 110, 120, 140, 240, 280, and 360. Purple points represent the causal edges, and yellow points represent the non-causal edges.

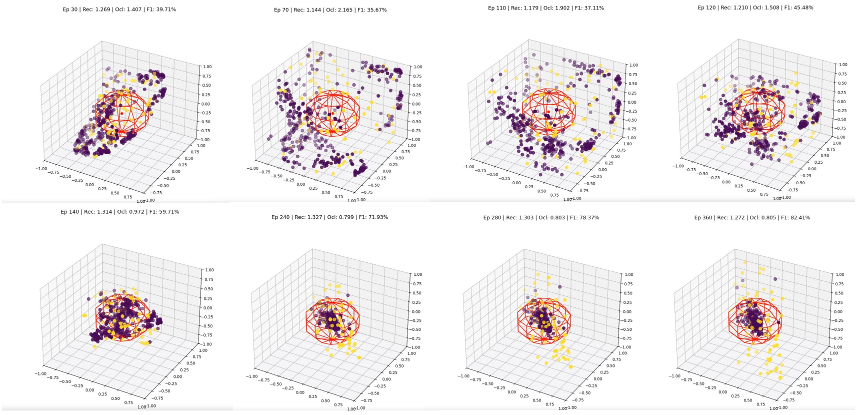


Fig. 3. Interpretability plot considering the three-dimensional last layer learned representations of eCOLGAT in the second fold. The colors indicate the causal (purple) and the non-causal (yellow) classes.

We observe the learning process of our proposal through eCOLGAT interpretability, as shown in Figure 3. In the initial steps, we observe eCOLGAT focusing on graph reconstruction (loss $\mathcal{L}_{gae}$). In the next steps, we observe learning through the two losses since the causal node-edge gets closer to the sphere while non-causal node-edge instances are outside the sphere. In the final steps, we observed that the one-class loss $\mathcal{L}_{ocl}$ encouraged the instances to continue coming to the sphere center while the $\mathcal{L}_{gae}$ maintained the non-causal node-edges outside the sphere.

5 Conclusion and Future Work

We propose eCOLGAT, a one-class GNN for causal discovery through edge classification. eCOLGAT explores a hypergraph to improve the edge representation learning and explores GAT as the GNN layer. Our eCOLGAT is also based on the SOTA sphere loss function for OCL and the reconstruction loss function. eCOLGAT explores three-dimensional representations during classification, providing interpretability for learning causal discorevy on text pairs.

Our experiments show that LLMs still have limitations in classifying causal relations, making it important to consider alternative representations. The proposed method incorporates both textual features from pre-trained models and relations between different cause-effect pairs, outperforming other LLM and OCL methods. In future work, we intend to explore a heterogeneous version of eCOLGAT on more causal text pair datasets, as well as incorporate classic effect inference tasks beyond textual data by mapping their attributes through natural language descriptions and modeling them in graphs.

Acknowledgments: This work was supported by FAPESP (grant numbers 2023/10100-4 and 2019/07665-4), CAPES (grant number 88887.671481/2022-00), CNPQ (grant number 316507/2023-7), Center for Artificial Intelligence (C4AI-USP), IBM Corporation and LatAm Google Ph.D. Fellowship.

References

1. Alam, S., Sonbhadra, S.K., Agarwal, S., Nagabhushan, P.: One-class support vector classifiers: A survey. Knowledge-Based Systems **196**, 105754 (2020)
2. Alibaba: Qwen technical report (2023), `https://arxiv.org/abs/2309.16609`
3. Brody, S., Alon, U., Yahav, E.: How attentive are graph attention networks? In: International Conference on Learning Representations (2022)
4. Emmert-Streib, F., Dehmer, M.: Taxonomy of machine learning paradigms: A data-centric perspective. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **12**(5), e1470 (2022)
5. Gemma: Gemma 2: Improving open language models at a practical size (2024), `https://arxiv.org/abs/2408.00118`
6. Gôlo, M.P.S., Marcacini, R.M.: Text representation through multimodal variational autoencoder for one-class learning. In: Anais do XXXVI Concurso de Teses e Dissertações. pp. 148–157. SBC (2023)
7. Gôlo, M.P.S., Junior, J.G.B.M., Silva, D.F., Marcacini, R.M.: Olga: One-class graph autoencoder. arXiv (2024)
8. Hassanzadeh, O., Bhattacharjya, D., Feblowitz, M., Srinivas, K., Perrone, M., Sohrabi, S., Katz, M.: Answering binary causal questions through large-scale text mining an evaluation using cause-effect pairs from human experts. In: IJCAI (2019)
9. Jin, M., Koh, H.Y., Wen, Q., Zambon, D., Alippi, C., Webb, G.I., King, I., Pan, S.: A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. Transactions on Pattern Analysis and Machine Intelligence (2024)
10. Jo, J., Baek, J., Lee, S., Kim, D., Kang, M., Hwang, S.J.: Edge representation learning with hypergraphs. In: NeurIPS. pp. 7534–7546. NeurIPS foundation (2021)

11. Kayesh, H., Islam, M.S., Wang, J.: Event causality detection in tweets by context word extension and neural networks. In: Int. Conf. on parallel and distributed computing, applications and technologies. pp. 352–357. IEEE (2019)
12. Kayesh, H., Islam, M.S., Wang, J.: Answering binary causal questions using roleoriented concept embedding. Transactions on Artificial Intelligence (2022)
13. Kayesh, H., Islam, M.S., Wang, J., Anirban, S., Kayes, A., Watters, P.: Answering binary causal questions: A transfer learning based approach. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–9. IEEE (2020)
14. Kipf, T.N., Welling, M.: Variational graph auto-encoders. stat **1050**, 21 (2016)
15. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422. IEEE (2008)
16. Liu, X., Xu, P., Wu, J., Yuan, J., Yang, Y., Zhou, Y., Liu, F., Guan, T., Wang, H., Yu, T., et al.: Large language models and causal inference in collaboration: A comprehensive survey. arXiv preprint arXiv:2403.09606 (2024)
17. Ma, J., Wan, M., Yang, L., Li, J., Hecht, B., Teevan, J.: Learning causal effects on hypergraphs. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1202–1212 (2022)
18. Meta: The llama 3 herd of models (2024), `https://arxiv.org/abs/2407.21783`
19. Microsoft: Phi-3 technical report: A highly capable language model locally on your phone (2024), `https://arxiv.org/abs/2404.14219`
20. Minghim, R., Milos, E., Provia, K., et al.: Ggnn@ causal news corpus 2022: Gated graph neural networks for causal event classification from social-political news articles. In: Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE). pp. 85–90 (2022)
21. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bertnetworks. In: Proceedings Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics (2019)
22. Sakaji, H., Izumi, K.: Financial causality extraction based on universal dependencies and clue expressions. New Generation Computing **41**(4), 839–857 (2023)
23. Sasaki, H., Fujii, M., Sakaji, H., Masuyama, S.: Enhancing risk analysis with gnn: Edge classification in risk causality from securities reports. International Journal of Information Management Data Insights **4**(1), 100217 (2024)
24. Seliya, N., Abdollah Zadeh, A., Khoshgoftaar, T.M.: A literature review on oneclass classification and its potential applications in big data. Journal of Big Data **8**, 1–31 (2021)
25. Sohrabi, S., Katz, M., Hassanzadeh, O., Udrea, O., Feblowitz, M.D., Riabov, A.: Ibm scenario planning advisor: Plan recognition as ai planning in practice. Ai Communications **32**(1), 1–13 (2019)
26. Tang, J., Liao, R.: Graph neural networks for node classification. Graph Neural Networks: Foundations, Frontiers, and Applications pp. 41–61 (2022)
27. Wang, X., Jin, B., Du, Y., Cui, P., Tan, Y., Yang, Y.: One-class graph neural networks for anomaly detection in attributed networks. Neural computing and applications **33**, 12073–12085 (2021)
28. Zanga, A., Ozkirimli, E., Stella, F.: A survey on causal discovery: theory and practice. International Journal of Approximate Reasoning **151**, 101–129 (2022)
29. Zhang, F., Fan, H., Wang, R., Li, Z., Liang, T.: Deep dual support vector data description for anomaly detection on attributed networks. International Journal of Intelligent Systems **37**(2), 1509–1528 (2022)
30. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv (2023)

Integrating Causal Inference into Dynamic Incentive Design

Sebastián Bejos[1], Eduardo F. Morales[1], Luis Enrique Sucar[1], and Enrique Munoz de Cote[2]

[1] Computer Science Department, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México. {sebastian.bejos,emorales,esucar}@inaoep.mx
[2] Secondmind, Cambridge CB2 1LA, United Kingdom, enrique@people-ai.com

Abstract. The design of incentives that modifies the behavior of self-interested agents to optimize the performance of a Multi-Agent System (MAS) remains a significant challenge. This is mainly due to the fact that after modifying the agents' rewards by incentives, the resulting system outcomes and changes to the agents' joint behavior are generally difficult to predict. Multi-armed bandit approaches in online learning represent interesting solutions to incentive design via exploration–exploitation strategies. However, the design of incentives for MAS that exploit causal feedback to make inferences about the performance of incentives remains uncharted territory, and the incorporation of causal reasoning in this context is an open problem. This paper introduces a way for integrating causal inference to solve dynamic incentive design problems in MAS, using Dynamic Causal Bayesian Optimization (DCBO). We use a generalization of an important representative class within MAS, called Minority Games, to show how dynamic incentive design can be addressed as a causal sequential decision problem incorporating causal reasoning by using DCBO.

Keywords: Multi-agent Systems · Causal Dynamic Incentive Design · Dynamic Causal Bayesian Optimization · Causal Sequential Decision Process · Multi-asset Minority Games

1 Introduction

In Dynamic Incentive Design (DID) problems [12], a central institution modifies the behavior of self-interested agents in a Multi-Agent System (MAS) to optimize the overall performance by introducing an incentive function to modify their individual payoffs. For instance, the central institution may want to drive the system performance to a more desirable behavior that, e.g., maximizes revenue or the social welfare. As an example, consider a ridesharing market MAS like Uber, in which it is common for users in neighborhoods with low user density to waive the platform, given that an habitual strategy among drivers is not to commit to trips in these type of areas but to remain biddable with trips that do not take them to far away from high user density zones. This also prevents the market from growing in these areas due to poor service, resulting in sub-optimal results in this MAS. For this issue, incentives can be designed, such as subsidizing fuel costs or giving a monetary reward for motivate services in this type of zones, to avoid this selfish strategies of the drivers. DID problems can be formulated as a bi-level optimization problem or equivalently as a reverse Stackelberg game [14]; in which an incentive designer agent (the principal agent), sequentially proposes an incentive function that affects the behavior of follower agents, and adapts sequentially this incentive function based on the followers' responses. DID has been a subject of interest to economics and control theory for long time [9]. In recent years there has been an increasing interest in studying DID in machine learning through the lens of online learning [8, 6, 11]. Other recent efforts have adopted the agent-based simulation paradigm and have taken state-of-the-art agent learning methods, such as Multi-Agent Meta Reinforcement Learning [15, 13, 7]. Nevertheless, to the best of our knowledge, there is no work that integrates causal reasoning to address DID problems. The use of causal inference to make predictions about incentive performance and solve for optimal incentives is the main motivation for this research. Causal inference would allow to take full advantage of data samples of observations on past incentive functions to infer the effectiveness of new alternative interventions. We deal with the integration of causal inference into DID problems by incorporating the Dynamic Causal Bayesian Optimization (DCBO) framework to handle the sequential decision process encountered by the principal agent in this scenario.

We define the main aspects of the principal-agents problem [12], the El Farol Bar problem [3, 5], and the DCBO [1] in Section 2 below. After in section 3 we show how to integrate DCBO to solve DID problems by setting the principal-agents problem as a dynamic probabilistic causal model. Considering the general causal structure of the principal-agent problem, in Section 4 we illustrate how the structural equations can be derived from a bi-level optimization formulation of a principal-agent problem, adopting the Multi-Asset Minority Games as an instance for such derivation. Conclusions and future work are given in Section 5.

2 Preliminaries

2.1 The Principal-Agents Problem

We restrict our attention to the principal–agents problem of DID, in which there is only one principal agent and a set $\boldsymbol{N}$ of followers agents, with $|\boldsymbol{N}| = n$, in a MAS, where the underlying model of the environment dynamics can be described as a differential or difference equation in a continuous or discrete time MAS, respectively[1][12]. Let $J_p(v^t, \boldsymbol{u}^t, s^t)$ be the utility function of the principal, with $J_p : \boldsymbol{V} \times \boldsymbol{U}_1 \times \ldots \times \boldsymbol{U}_n \times \boldsymbol{S} \to \mathbb{R}$, and let $\{J_{a_i}(v^t, \boldsymbol{u}^t, s^t) \mid i \in [n]\}$ be the set of utility functions for the followers agents, with $J_{a_i} : \boldsymbol{V} \times \boldsymbol{U}_1 \times \ldots \times \boldsymbol{U}_n \times \boldsymbol{S} \to \mathbb{R}$, where $\boldsymbol{V}$ is the action space of the principal, $\boldsymbol{U}_i$ is the action space for the follower

[1] In this work we focus in discrete time multi-agent systems

agent a_i, v^t and $\boldsymbol{u}^t$ are the decisions of the principal and the followers agents, respectively, at time t. $\boldsymbol{S}$ is the state space of the multi-agent system, s^t is the state at time t, and the system state dynamics is given by $s^{t+1} = d(v^t, \boldsymbol{u}^t, s^t)$. Considering a T horizon, there is a game between the principal and the followers agents, where there is an specific order of play. For rounds $t = 1, \ldots, T$, the interaction protocol in the game is as follows:

1. The principal p decides and announces an incentive function $\gamma^t : \boldsymbol{U}_1 \times \ldots \times \boldsymbol{U}_n \rightarrow \boldsymbol{V}$ with full, an estimation or no knowledge at all (as applicable) of the utility functions of the followers agents $\{J_{a_i} \mid i \in [n]\}$.
2. Then, with knowledge of this incentive function, each follower agent $a_i \in \boldsymbol{N}$ selects an action $u_i^{t^*}$ that maximizes their utility. So, at the end of this step, the decisions vector $\boldsymbol{u}^{t^*} = (u_1^{t^*}, \ldots, u_i^{t^*}, \ldots, u_n^{t^*})$ is public to the principal agent p, where $u_i^{t^*} \in \arg\max J_{a_i}(\gamma^t(\boldsymbol{u}^t), \boldsymbol{u}^t, s_t)$ for each $i \in [n]$.

The goal of the game is to find $\{(v^{t^*}, \boldsymbol{u}^{t^*})\}_t \in \arg\max \sum_{t\in[T]} J_p(v^t, \boldsymbol{u}^t, s^t)$, i.e., maximize the principal's cumulative utility, by selecting an incentive function γ^t in a set of admissible incentives functions $\boldsymbol{\Gamma} = \{\gamma : \boldsymbol{U}_1 \times \ldots \times \boldsymbol{U}_n \rightarrow \boldsymbol{V}\}$, at each time $t \in [T]$. In other words, at each round t the principal selects an incentive function $\gamma^t \in \boldsymbol{\Gamma}$ such that the followers agents chooses an action that leads to the maximization of the principal's utility, which usually is equivalent to the system utility or what is best for the multi-agent system, e.g., maximize the social welfare, i.e., the sum of expected gains of all agents in the long run.

2.2 El Farol Bar Problem and Minority Games

In the *El Farol Bar* problem a set $\boldsymbol{N}$ of n agents have to decide independently on each time t whether to go to the bar, $u_i^t = 1$, or not go, $u_i^t = 0$, i.e., the action space for each agent a_i is $\boldsymbol{U}_i = \{0, 1\}$, with $i \in [n]$. The Farol Bar has a capacity limit, $L < n$, and the bar is enjoyable only if it is not overcrowded, i.e., only if the attendance $A^t = \sum_{i\in[n]} u_i^t$ does not exceed L. In order to make their decisions, agents aim to predict whether the bar will be crowded or not on any given time t based on the past attendances. It is assumed that agents base their predictions on the attendances of a finite number $m \in \mathbb{N}$ of past times, and the information available to agents at time t is encoded in the string $\lambda^t = \big(\zeta[(L - A^{t-1}], \ldots, \zeta[(L - A^{t-m}]\big) \in \{0, 1\}^m$, where $\zeta(\cdot)$ is the Heaviside function, i.e., $\zeta[(L-A^t] = 1$ if the bar was enjoyable ($A^t < L$) while $\zeta[(L-A^t] = 0$ if the bar was overcrowded ($A^t \geq L$) at time t.

Given information λ^t, each agent a_i have a set $\boldsymbol{\Lambda}_i = \{\Lambda_{i1}, \ldots, \Lambda_{iK_i}\}$ of functions Λ_{ik} with $k \in [K_i]$ for some $K_i \in \mathbb{N}$ called strategies or predictors, that maps information strings $\{0, 1\}^m$ into the binary actions set $\{0, 1\}$ of go or do not go, i.e., every strategy $\Lambda_{ik} \in \boldsymbol{\Lambda}_i$ is a function such that $\Lambda_{ik} : \{0, 1\}^m \rightarrow \{0, 1\}$.

Ranking the strategies of each agent in the El Farol Bar problem can be approached through different learning processes, here we focus on the cumulative utility based ranking [5]. These processes help agents evaluate and adapt their strategies based on their performance as the game progresses. Let $u_{\Lambda_{ik}}^{\lambda^t} \in \{0, 1\}$

denote the prediction of strategy $\Lambda_{ik} \in \boldsymbol{\Lambda}_i$ of agent a_i under the information λ^t at time t, and let $\Omega^t_{\Lambda_{ik}}$ be the cumulative utility of agent a_i using strategy Λ_{ik} up to time t. At every time t, agent a_i selects a strategy with the highest cumulative utility ${\Lambda^t}^*_{ik} \in \arg\max_{\Lambda_{ik}} \Omega^t_{\Lambda_{ik}}$, and acts accordingly, i.e., $u^t_i = u^{\lambda^t}_{\Lambda^*_{ik}}$.

In order to decide which strategy to adopt on every time t, agents keep track of their performance via the cumulative utility that is updated according to the rule: $\Omega^{t+1}_{\Lambda^*_{ik}} = \Omega^t_{\Lambda^*_{ik}} + (1 - 2u^{\lambda^t}_{\Lambda^*_{ik}})[A^t - L]$, with the rationale that strategies suggesting not to go ($u^{\lambda^t}_{\Lambda^*_{ik}} = 0$) are rewarded when the attendance is higher than L and punished when it is lower than L (and vice versa when $u^{\lambda^t}_{\Lambda^*_{ik}} = 1$).

For modeling purposes, Minority Games serve as a class of simple models which are able to produce some of macroscopic features being observed in the real financial markets. The basic Minority Game corresponds roughly to the case where $L = \lfloor \frac{n}{2} \rfloor$ of the El Farol Bar problem, but here the agents actions are either to buy, $u^t_i = 1$, or sell, $u^t_i = 1$, an specific stock, to model speculative trading in financial markets [3, 2, 4].

2.3 Dynamic Causal Bayesian Optimization (DCBO)

A graphical causal model consist of a four tuple $\langle \boldsymbol{W}, \boldsymbol{Z}, P(\boldsymbol{Z}), \boldsymbol{F} \rangle$ and a directed acyclic graph $\mathcal{G}$, where $\boldsymbol{W}$ is the set of observed endogenous variables, $\boldsymbol{Z}$, is a set of exogenous variables expressing a random disturbance distributed according to $P(\boldsymbol{Z})$, and $\boldsymbol{F} = \{f_1, \ldots, f_{|\boldsymbol{W}|}\}$ is a set of functions known as structural equations, such that $W_i = f_i(\boldsymbol{pa}(W_i), Z_i)$, for each $W_i \in \boldsymbol{W}$, with $\boldsymbol{pa}(W_i)$ denoting the parents of W_i [1]. The graph $\mathcal{G}$ encodes the causal relationship between the variables in $\boldsymbol{W}$. Within $\boldsymbol{W}$ we distinguish three different types of variables: non-manipulative variables $\boldsymbol{C}$, treatment variables $\boldsymbol{X}$ that can be set to specific values, i.e., intervene them, and output variable Y that represent the outcome of interest. In order to reason about interventions that are implemented in a sequential manner, i.e., at each time t we decide which intervention to perform in the system. Let $\mathcal{M}^t$ be a dynamic graphical causal model defined as $\mathcal{M}^t = \langle \mathcal{G}^{1:t}, \boldsymbol{W}^{1:t}, \boldsymbol{Z}^{1:t}, P(\boldsymbol{Z}^{1:t}), \boldsymbol{F}^{1:t} \rangle$, where $1:t$ denotes the union of the corresponding graphs, variables or functions up to time t, $\boldsymbol{W}^{1:t} = \boldsymbol{X}^{1:t} \cup \boldsymbol{C}^{1:t} \cup \boldsymbol{Y}^{1:t}$. The goal of DCBO is to find a sequence of interventions, optimizing a target variable, at each time t, in a graphical causal model $\mathcal{M}^t$. Given $\mathcal{M}^t$, at every time step t, we wish to optimize Y_t by intervening on a subset of the manipulative variables $\boldsymbol{X}_t$. The optimal intervention variables $\boldsymbol{X}^*_{s,t}$ and intervention levels $\boldsymbol{x}^*_{s,t}$ are given by:

$$\boldsymbol{X}^*_{s,t}, \boldsymbol{x}^*_{s,t} = \underset{\boldsymbol{X}_{s,t} \in \mathcal{P}(\boldsymbol{X}_t), \boldsymbol{x}_{s,t} \in D(\boldsymbol{X}_{s,t})}{\arg\max} E[Y_t \mid do(\boldsymbol{X}_{s,t} = \boldsymbol{x}_{s,t}), \mathbb{1}_{t>1} \cdot \boldsymbol{I}_{1:t-1}]. \quad (1)$$

where $\mathcal{P}(\boldsymbol{X}_t)$ is the power set of $\boldsymbol{X}^*_{s,t}$, $D(\boldsymbol{X}_{s,t})$ represents the interventional domain of $\boldsymbol{X}_{s,t}$, $\boldsymbol{I}_{1:t-1} = \bigcup_{i=1}^{t-1} do(\boldsymbol{X}^*_{s,i} = \boldsymbol{x}^*_{s,i})$ denotes previous interventions, and $\mathbb{1}_{t>1}$ is the indicator function. The expected value is over the interventional distribution $P(Y_t \mid do(\boldsymbol{X}^*_{s,t} = \boldsymbol{x}^*_{s,t}))$, given $\mathbb{1}_{t>1} \cdot \boldsymbol{I}_{1:t-1}$. DCBO make the assumptions of invariance of causal structure, i.e., $\mathcal{G}^t = \mathcal{G}^{t+1}$ for all $t \in [T]$, and absence of unobserved confounders in $\mathcal{G}_{1:T}$ [1].

3 Causal Dynamic Incentive Design

In this section we show how causal inference can be integrated in the principal-agent problem described in Section 2. This is done by first proposing the general causal structure $\mathcal{G}^t_{pap}$ for a dynamic graphical causal model representing the principal agent problem $\mathcal{M}^t_{pap}$. The dynamics between the principal and the follower agents are modeled in $\mathcal{G}^t_{pap}$ in the following way. The set of observed endogenous variables is given by $\boldsymbol{W^{1:T}} = \{\boldsymbol{V}^{1:T}, \boldsymbol{U}^{1:T}_i, \boldsymbol{J}^{1:T}_{a_i}, \boldsymbol{J}^{1:T}_P\}$, for all $i \in [\boldsymbol{N}]$, where $\boldsymbol{V}^{1:T}$ is the variable representing the principal actions, $\boldsymbol{U}^{1:T}_i$ represents the agent i actions, $\boldsymbol{J}^{1:T}_{a_i}$ represents the agents i utility function, for all $i \in [\boldsymbol{N}]$, and $\boldsymbol{J}^{1:T}_P$ represents the utility function of the principal, all up to time T. The Figure 1 shows the directed graph $\mathcal{G}^{t:t+1}_{pap}$, where the direct causal relation are show for the variables in $\boldsymbol{W^{t:t+1}}$, i.e., for times or rounds t and $t+1$ of the principal-agents protocol (see Section2), and from which all the direct causal relation for $\boldsymbol{W^{1:T}}$ can be inferred. We are using the following color convention for the types of variables that appear in $\mathcal{G}^{t:t+1}_{pap}$ shown in Figure 1 (as well as for the next figures): blue for the variables that are not manipulable, i.e., $\boldsymbol{U}^{1:T}_i \cup \boldsymbol{J}^{1:T}_{a_i} = \boldsymbol{C}^{1:T}$, green for those that are manipulable, i.e., $\boldsymbol{V}^{1:T} = \boldsymbol{X}^{1:T}$ and orange for the target variables, i.e., $\boldsymbol{J}^{1:T}_p = \boldsymbol{Y}^{1:T}$, at each time t.

.

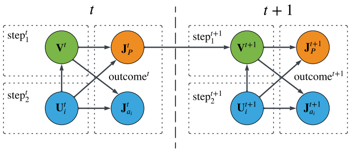


Fig. 1: $\mathcal{G}^{t:t+1}_{pap}$.

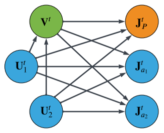


Fig. 2: $\mathcal{G}^t_{pap}$ for $n = 2$.

We are doing an important simplification on the graphical representation in Figure 1 as the real dynamic graphical causal structure most contemplate one variable for each of the agents' action space and one variable for each of the utility functions of each of the agents. Figure 2 shows what the structure $\mathcal{G}^t_{pap}$ would actually look like for the case of $n = 2$ on a given time t. In Figure 1, we are using the variables $\boldsymbol{U}^t_i$ and $\boldsymbol{J}^t_{a_i}$ to represent all the variables for each each of the agents' action space $\boldsymbol{U}^t_1, \dots \boldsymbol{U}^t_n$ and for each of the utility functions $\boldsymbol{J}^t_{a_1}, \dots \boldsymbol{J}^t_{a_n}$ of each of the agents, respectively, to keep the drawings understandable and not overly edged.

Figure 3 (a) shows the simplification pattern for variables $\boldsymbol{U}^t_i$ and and its causal relationship with $\boldsymbol{V}^t$. On Figure 3 (b) it is shown what this pattern represents, where the causal structure from variables $\boldsymbol{U}^t_1, \dots \boldsymbol{U}^t_n$ is given by an empty graph, i.e., no edges or causal relations between $\boldsymbol{U}_1^t, \ldots \boldsymbol{U}_n^t$. Figure 3 (c) shows an instance of the other scenario where the causal structure between variables $\boldsymbol{U}_i^t$ is not an empty graph, it is shown a sub-DAG for the action variables $\boldsymbol{U}_i^t$ of four agents in this case. In some application contexts maybe important to perform a causal discovery method to find the underlying causal structure of variables $\boldsymbol{U}_i^t$. We leave this kind of scenario for later work.

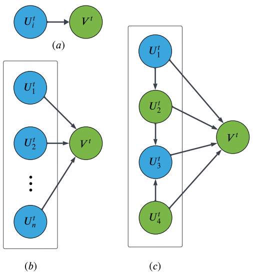


Fig. 3: The pattern $\boldsymbol{U}_i^t \rightarrow \boldsymbol{V}$.

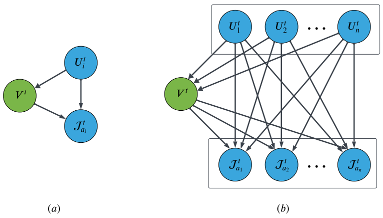


Fig. 4: The directed triangle pattern.

Figure 4 (a) depicts another graphical simplification in the $\mathcal{G}_{pap}^t$ shown in Figure 1, we call it the directed triangle pattern, which represent the direct causal relation of $\boldsymbol{U}_i^t \rightarrow \boldsymbol{V}^t$, $\boldsymbol{V}^t \rightarrow \boldsymbol{J}_{a_i}^t$, and $\boldsymbol{U}_i^t \rightarrow \boldsymbol{J}_{a_i}^t$. In Figure 4 (b) it is shown the unfold directed triangle pattern, i.e., the real structure that it represents. Something similar occurs for the directed triangle pattern representing the direct causal relations of $\boldsymbol{U}_i^t \rightarrow \boldsymbol{V}^t$, $\boldsymbol{V}^t \rightarrow \boldsymbol{J}_p^t$, and $\boldsymbol{U}_i^t \rightarrow \boldsymbol{J}_p^t$. Another aspect that is shown in the graph of Figure 3 (c) is that in some application contexts maybe important to consider interventions on a subset of agents' action space variables. As shown in Figure 3, some U_i^t variables are colored green meaning they are in the set of manipulable variable $\boldsymbol{X}^t$. Manipulating certain U_i^t variables can be helpful for exploring the effects of specific actions taken by a group of agents on the causal model.

We focus in the case of no edges between the $U_1^t, \ldots, U_n^t$ variables and with none of these variables in the set of manipulable variables $\boldsymbol{X}^t$. Given this developed causal structure $\mathcal{G}_{pap}^t$, it is possible to completely specify the DCBO method (See Section 2) to solve a general principal-agent problem by defining the action space of the principal $\boldsymbol{V}^t$, the action space of the agents $\boldsymbol{U}_i^t$, for all $i \in [n]$, and the structural equations $\boldsymbol{F}^t = \{f_{\boldsymbol{V}^t}^t, f_{\boldsymbol{J}_p^t}^t, f_{\boldsymbol{J}_{a_i}^t}^t\}$, for all $i \in [n]$, which have the following general arrangement due to the causal structure $\mathcal{G}_{pap}^t$: $f_{\boldsymbol{J}_{a_i}^t}^t = f^{\boldsymbol{J}_{a_i}^t}(v^t, \boldsymbol{u}_i^t, \epsilon_{\boldsymbol{J}_a}^t)$, $f_{\boldsymbol{J}_p}^t = f^{\boldsymbol{J}_p^t}(v^t, \boldsymbol{u}_i^t, \epsilon_{\boldsymbol{J}_p}^t)$, and

$$f_{\boldsymbol{V}^t}^t = \begin{cases} f^{\boldsymbol{V}^t}(\boldsymbol{u}_i^t, \epsilon_{\boldsymbol{V}}^t), & \text{if } t = 1, \\ f^{\boldsymbol{V}^t}(\boldsymbol{u}_i^t, \boldsymbol{J}_p^{t-1}, \epsilon_{\boldsymbol{V}}^t), & \text{if } t > 1, \end{cases}$$

where $\epsilon_{\boldsymbol{J}_a}^t, \epsilon_{\boldsymbol{J}_p}^t, \epsilon_{\boldsymbol{V}}^t \in \boldsymbol{Z}^t$ are the exogenous variables expressing a random disturbance distributed according to $P(\boldsymbol{Z}^t)$.

The detailed definition of these structural equations for the causal dynamic graphical model $\mathcal{M}_{pap}^t$ will depend on the application context. In the application settings, it is a common practice to formulate a principal-agent problem as a bi-level optimization problem [12]. Under this type of formulation of a principal-agent problem, the utility of the principal and the agents are presented as the objective functions of the upper level and lower level problem, respectively [14]. In Section 4, we present an interesting instance of the principal agent problem for which we give its bi-level optimization formulation and show how from the objective functions we can obtain directly the structural equations for the $\mathcal{M}_{pap}^t$. It is necessary to mention that at each time t the interaction protocol between the principal and the followers (Seen in Section 2) is respected. That is, the principal first on step 1 decides (or chooses) an incentive function γ from a set of possible incentive function $\boldsymbol{\Gamma} = \{\gamma : \boldsymbol{U}_1 \times \ldots \times \boldsymbol{U}_n \to \boldsymbol{V}\}$. Then, on step 2, with the knowledge of the principal's selected incentive function γ, as this is incorporated in the follower's utility function $\boldsymbol{J}_{a_i}(v^t, \boldsymbol{u}_i^t) = \boldsymbol{J}_{a_i}(\gamma(\boldsymbol{u}_i^t), \boldsymbol{u}_i^t)$, the follower agents decide $\boldsymbol{u}_i^t \in \boldsymbol{U}_i^t$. After the followers agent's decisions, on step 2, the outcome for time t is computed as the variables $\boldsymbol{J}_{a_i}^t$ and $\boldsymbol{J}_p^t$ for every time $t \in [T]$. This interaction protocol for each time t is shown in Figure 1 by partitioning the vertex set of $\mathcal{G}_{pap}^t$ with the sets $step_1^t = \{\boldsymbol{V}^t\}$, $step_2^t = \{\boldsymbol{U}_i^t\}$, and $outcome^t = \{\boldsymbol{J}_{a_i}^t, \boldsymbol{J}_p^t\}$. In the light of all that and the given specification of the general dynamic graphical causal model $\mathcal{M}_{pap}^t$ the agent-principal problem can be solve by finding a sequence of interventions on variables $\boldsymbol{V}^{1:T}$ optimizing the target variables $\boldsymbol{J}_p^{1:T}$, i.e., by computing:

$$v^{t*} = \gamma^{t*}(\boldsymbol{u}_i^t) = \underset{\gamma^t \in \Gamma}{\arg\max}\, \mathbb{E}\big[\boldsymbol{J}_p^t \mid do(\boldsymbol{V}_t = \gamma^t(\boldsymbol{u}_i^t) = v^t), \mathbb{1}_{t>1} \cdot \boldsymbol{I}_{1:t-1}\big], \qquad (2)$$

with the important remark that intervene the variable $\boldsymbol{V}^t$ is to assign an incentive function $\gamma \in \Gamma$ to variable $\boldsymbol{V}^t$, representing the choose of γ by the principal on step 1, which after step 2 gets a value $v^t \in \boldsymbol{V^t}$ by the causal relation $\boldsymbol{U}_i^t \to \boldsymbol{V}^t$. Moreover, the causal relation $\boldsymbol{J}_p^{t-1} \to \boldsymbol{V}^t$ on $\mathcal{G}_{pap}^t$ stablish the utilization of previous results for $\boldsymbol{J}_p^t$ to inform the exploration for the optimal incentive function selection, using the result of previous interventions $\boldsymbol{I}_{1:t-1}$. We call Causal Dynamic Incentive Design (CDID) to the method described in this section for solving the principal-agent problem.

4 CDID on the Multi-Asset Minority Game

We explore a generalization of the El Farol Bar problem and the Minority Game, where agents must decide not only whether to go to or not to go to the bar or to buy or to sell a stock but also which bar to attend or which stock to buy or sell from a collection of bars or stocks, where each asset has its own capacity. We call this generalization a Multi-Asset Minority Game. Let M be the number of assets and let each asset has a capacity limit L_j with $L_j < n$ for $j \in [M]$. Let $u_{ij}^t \in \{0, 1\}$ be the decision variable indicating whether agent i decides in to the asset j at time t. Let $\mu_j^t = \big(\zeta[(L_j - A_j^{t-1}], \dots, \zeta[(L_j - A_j^{t-m}]\big) \in \{0,1\}^m$ be the string encoding the information available to agents at time t assuming finite memory of m past times references over the asset $j \in [M]$, with $A_j^t = \sum_{i \in [N]} u_{ij}^t$. The information available to agents at time t is encoded in the following string:

$$\mu^t = \Big(\bigoplus_{j \in [M]} \mu_j^t \Big) \in \{0,1\}^{m|M|},$$

where $\bigoplus_{j \in [M]} \mu_j^t$ denotes de concatenation of the M strings $\mu_j^t \in \{0,1\}^m$.

Let $\boldsymbol{\Lambda_i} = \{\Lambda_{i1}, \Lambda_{i2}, \dots, \Lambda_{iK_i}\}$ be the strategies set of agent a_i, $u_{\Lambda_{ik}}^{\mu_j^t} \in \{0,1\}$ denote the prediction of strategy Λ_{ik} with $k \in [K_i]$ for some $K_i \in \mathbb{N}$ of agent i for asset $j \in [M]$, under the information μ_j^t at time t. Let $U_{\Lambda_{ik}}^j(t)$ the the cumulative utility of agent i using strategy Λ_{ik} up to time t for asset $j \in M$. Every time t, agent i selects the strategy Λ_{ik}^* with the highest cumulative utility for each asset $j \in M$, and acts accordingly, i.e., $u_{ij}^t = u_{\Lambda_{ik}^*}^{\mu_j^t}$. Let Λ_{ik}^* be such strategy in $\boldsymbol{\Lambda}_i$, i.e.,

$$\Lambda_{ik}^* = \underset{\Lambda_{ik} \in \boldsymbol{\Lambda}_i}{\arg\max}\, \Omega_{\Lambda_{ik}}^j(t),$$

In order to decide which strategy to adopt for each asset $j \in M$ on every time t, agents keep track of their performance for each asset $j \in M$ via the cumulative utility $\Omega_{\Lambda_{ik}}^j(t)$ that is updated according to the following rule:

$$\Omega_{\Lambda_{ik}^*}^j(t+1) = \Omega_{\Lambda_{ik}^*}^j(t) + (1 - 2u_{\Lambda_{ik}^*}^{\mu_j^t})[A_j^t - L_j],$$

4.1 Bi-level Optimization Formulation of the Multi-Asset Minority Game with Incentives Design

We now extend the Multi-Asset Minority Game (MAMG) to include Dynamic Incentive Design (DID), presenting a formulation of this Multi-Asset Minority Game as a principal-agents problem. For this, we present the Multi-Asset Minority Game with Dynamic Incentives as a bi-level optimization problem below, specifying the principal's (upper level) and agents' (lower level) objective functions, each with their own constraints. The objective function of the principal is given as follows:

$$\text{(Principal)} \quad \max_{\gamma_j^t(\boldsymbol{u}_{ij}) \in \boldsymbol{\Gamma}} \quad \mathbb{E}\Bigg[\sum_{t \in [\boldsymbol{T}]} \sum_{j \in [\boldsymbol{M}]} R_j\big(A_j^t\big) - C_j\big(A_j^t, \gamma_j^t(\boldsymbol{u}_{ij})\big)\Bigg] \tag{3}$$

$$\text{s. a. } \gamma_j^t(\boldsymbol{u}_{ij}) = \begin{cases} \alpha\left(\frac{L_j - A_j^t}{L_j}\right) \leq \omega_j^t & \text{if } A_j^t \leq L_j, \text{ where } \omega_j^t \text{ is the budget at } t. \\ \alpha\left(\frac{A_j^t - L_j}{L_j}\right) & \text{if } A_j^t > L_j, \text{ for some } \alpha > 0 \in \mathbb{R}. \end{cases}$$

where $R_j\left(A_j^t\right)$ is the revenue function assuming a revenue per unit of attendance (e.g., trading volume) for asset j, and $C_j\left(A_j^t, \gamma_j^t(\boldsymbol{u}_{ij})\right)$ is the cost function, presuming a cost per unit of attendance, and a cost per unit of incentive. Note that we define the incentives in a way that when $A_j^t \leq L_j$ the principal offers a subsidy which is given based on how far the attendance is from the limit, on the other hand, when $A_j^t > L_j$ the principal imposes taxes on the agents according to the excess of attendance. The agents' objective function is shown below:

$$\text{(Agents)} \quad (\boldsymbol{u}_{ij}^*) \in \underset{\boldsymbol{u}_{ij} \in \boldsymbol{U}_{ij}}{\arg\max} \left\{ \mathbb{E}\left[\sum_{t \in [T]} \sum_{j \in [M]} \Big((1 - 2u_{ij}^t)[A_j^t - L_j] \Big) + \gamma_j^t(\boldsymbol{u}_{ij}) \right] \right\}_{i \in [n]}$$

$$\text{s. a.} \sum_{j \in [\boldsymbol{M}]} u_{ij}^t \leq |M|, \quad \forall i \in [n],\ \forall j \in [M],\ \forall t \in [T]. \tag{4}$$

It is important to note that the above are not two independent optimization problems, one for the primary agent's objective function and constraints and the other for the followers' objective functions and constraints. Rather, it is a single bi-level optimization problem, where an optimal solution for the lower level problem, i.e., the one corresponding to the follower agents, is only a feasible solution for the upper level optimization problem, the one corresponding to the principal agent.

4.2 CDID for the Multi-asset Minority Game with Incentives Design.

From the analysis given in Section 3, we can use CDID to solve the Multi-asset Minority Game with incentives design by first defining the action spaces of the principal and follower agent, i.e., variables $\boldsymbol{V}^t$, and $\boldsymbol{U}_i^t$, for all $i \in [\boldsymbol{N}]$, which for this principal-agent MAS we can state as:

$$\boldsymbol{V}^t = \left\{ \boldsymbol{v}^t = (v_1^t, \ldots, v_M^t) \in \mathbb{R}^M \mid v_j^t = \gamma_j^t(\boldsymbol{u}_{ij}^t),\ \text{for } j \in [M], \gamma_j^t \in \mathit{\Gamma} \right\}$$

$$\mathcal{U}_i^t = \left\{ \boldsymbol{u}_i^t = (u_{i1}^t, \ldots, u_{iM}^t) \in \{0,1\}^M \mid u_{ij}^t = u_{\mathit{\Lambda}_{ik}^*}^{\mu^t} \in \{0,1\},,\ \text{for } j \in [M] \right\}$$

Subsequently, given the general structure for the dynamic causal model representing a principal-agent problem $\mathcal{M}_{pap}^t$ as given in Section 3, it simply remains to state the structural equations $\boldsymbol{F}^t = \{f_{\boldsymbol{V}^t}^t, f_{\boldsymbol{J}_P^t}^t, f_{\boldsymbol{J}_{a_i}^t}^t\}$, for all $i \in [n]$. Using the above bi-level optimization formulation of the principal-agent problem for the Multi-asset Minority Game with incentives design, we can set up the structural equations $\boldsymbol{F}^t$ for its dynamic causal model $\mathcal{M}_{pap}^t$ as follows:

$$f^t_{\boldsymbol{J}_p} = f^{\boldsymbol{J}^t_p}(v^t, \boldsymbol{u}^t_i, \epsilon^t_{\boldsymbol{J}_p}) = \mathbb{E}\left[\sum_{j\in[M]} R_j\big(A^t_j\big) - C_j\big(A^t_j, \gamma^t_j(\boldsymbol{u}_{ij})\big) + \epsilon^t_{\boldsymbol{J}_p}\right] \tag{5}$$

$$f^t_{\boldsymbol{J}^t_{a_i}} = f^{\boldsymbol{J}^t_{a_i}}(v^t, \boldsymbol{u}^t_i, \epsilon^t_{\boldsymbol{J}_a}) = \mathbb{E}\left[\sum_{i\in[M]} \Big((1-2u^t_{ij})[A^t_j - L_j]\Big) + \gamma^t_j(\boldsymbol{u}_{ij}) + \epsilon^t_{\boldsymbol{J}_a}\right] \tag{6}$$

$$f^t_{\boldsymbol{V}^t} = \begin{cases} f^{\boldsymbol{V}^t}(\boldsymbol{u}^t_i, \epsilon^t_{\boldsymbol{V}}), & \text{if } t = 1, \\ f^{\boldsymbol{V}^t}(\boldsymbol{u}^t_i, \boldsymbol{J}^{t-1}_p, \epsilon^t_{\boldsymbol{V}}), & \text{if } t > 1, \end{cases} = \begin{cases} \big((0,\ldots,0) + \boldsymbol{\epsilon}^t_{\boldsymbol{J}_p}\big) \in \mathbb{R}^M \text{ if } t = 1, \\ \boldsymbol{v}^{t*} = (v^{t*}_j) \in \mathbb{R}^M, \text{ if } t > 1, \text{ where} \end{cases} \tag{7}$$

$$v^{t*}_j = \gamma^{t*}_j(\boldsymbol{u}^t_{ij}) = \underset{\gamma^t_j \in \boldsymbol{\Gamma}}{\arg\max}\, \mathbb{E}\big[\boldsymbol{J}^t_p \mid do(\boldsymbol{V}^t = \gamma^t_j(\boldsymbol{u}^t_{ij}) = v^t_j), \mathbb{1}_{t>1} \cdot \boldsymbol{I}_{1:t-1}\big],$$

which is exactly Equation 2 from Section 3, but for each asset $j \in [M]$, where $\epsilon^t_{\boldsymbol{J}_a}, \epsilon^t_{\boldsymbol{J}_p}, \epsilon^t_{\boldsymbol{V}} \in \boldsymbol{Z}^t$ for which we assume $P(\epsilon^t_{\boldsymbol{J}_a}) = P(\epsilon^t_{\boldsymbol{J}_p}) = P(\epsilon^t_{\boldsymbol{V}}) \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$, with $\mu = 0$ and $\sigma = 1$. Observe that Equations 5 and 6, which correspond to the structural equations of the utility variables $\boldsymbol{J}^t_p$ and $\boldsymbol{J}^t_{a_i}$, for all $i \in [n]$, are practically the objective functions of the bi-level optimization problem for the principal agent (Equation 3) and the follower agents (Equation 4), respectively. Finally, an important design aspect for using the CDID framework and in particular to fully define the set $\boldsymbol{V}^t$ in the MAMG with DID is to define the set of admissible incentive functions $\boldsymbol{\Gamma}$. Many variants of the $\boldsymbol{\Gamma}$ set can be proposed, yet we exhibit two instances of incentive functions families that independently each of them could be established as $\boldsymbol{\Gamma}$, but also the union of both families of incentive functions. For example, to distribute clients optimally across assets, the principal can implement a dynamic pricing scheme by stablish a cost

$$DPS_j(A^t_j) = bp_0 + \beta \frac{A^t_j}{L_j},$$

for access or buying an asset $j \in [M]$ as a function of attendance A^t_j, where bp_0 is the base price and β is a scaling factor. Likewise, agents could be penalized for choosing overcrowded assets by defining a congestion penalty incentive function

$$CPIF(A^t_j) = \beta\big[max\big(0, (A^t_j - L_j)\big)\big],$$

that reduces individual utility based on attendance. Where β is a scaling factor. Each agent choosing an asset j receives a penalty if the asset is over capacity, incentivizing them to opt for less crowded assets.

5 Conclusions

The generation of experimental data in the dynamic incentive design on a MAS is expensive. In this context, such experimental data translates into testing different incentives and measuring their effects on the multi-agent system in question. The field of causal inference features a rich set of tools to evaluate the performance of untested incentives and solve for optimal incentives, thereby allowing to make the most of limited data samples, i.e., experience with past incentive functions. That is, using observations on the MAS and data from past incentive functions effect on the system, causal inference can infer the effectiveness of new alternative interventions by evaluating post-intervention distributions and rate different incentive functions cheaper. The main contribution in this paper is the presentation of CDID method in which causal inference can be incorporated to deal with incentive design in MAS. For this purpose, we leverage of the DCBO method and characterize a generic dynamic causal probabilistic model $\mathcal{M}_{pap}$ representing a principal-agent problem in general, and show how the structural equations of this dynamic causal probabilistic model can be derived from a bi-level optimization formulation of a principal-agents problem in MAS. An interesting result is that in the case of homogeneous follower agents, only three structural equations of the dynamic causal model are needed for the application of DCBO. For the more general case, i.e., contemplating heterogeneous follower agents MAS with g different groups of agents, it would require the formulation of $2g + 1$ structural equations of the dynamic causal model.

5.1 Future Work.

We are working in the validation this proposal using data from Agent Based Modeling simulations [10] of Multi-asset Minorty Games and other MASs, like traffic assignment systems, and compare with solutions based on Bayesian Optimization for DID [11], hoping to reach faster convergence in the search for optimal incentives functions. In this research, we focus on a single principal with several follower agents version of the DID, but another goodness about our proposal is that it can be straightforward extended to contemplate more than one principal agent. Moreover, it can also be extended to consider more than two levels of hierarchy among agents, e.g., that there were principal agents at a third level who incentivize second level principals. Additionally, the proposed framework for DID may be appropriate for considering information asymmetries, such as adverse selection. Which can be characterized as the follower agent's utilities being dependent on some parameter $\theta \in \Theta$ representing the agent's type, so the utility functions J_{a_i} for $i \in [n]$ can be expressed as $J_{a_i}(v, \boldsymbol{u}; \theta)$, and θ is unknown a priori to the principal. We intend to investigate these variants that can be easily accommodated within our framework by incorporating new variables into the probabilistic causal model in future research.

F. Author et al.

References

1. Aglietti, V., Dhir, N., González, J., Damoulas, T.: Dynamic causal bayesian optimization. Advances in Neural Information Processing Systems **34**, 10549–10560 (2021)
2. Challet, D., Marsili, M., Zhang, Y.C.: Minority games: interacting agents in financial markets. OUP Oxford (2004)
3. Challet, D., Zhang, Y.C.: Emergence of cooperation and organization in an evolutionary game. Physica A: Statistical Mechanics and its Applications **246**(3-4), 407–418 (1997)
4. Coolen, A.C.: The mathematical theory of minority games: statistical mechanics of interacting agents. OUP Oxford (2005)
5. De Martino, A., Marsili, M.: Statistical mechanics of socio-economic systems with heterogeneous agents. Journal of Physics. A, Mathematical and General **39**(43), R465–R540 (2006)
6. Fiez, T., Sekar, S., Zheng, L., Ratliff, L.J.: Combinatorial bandits for incentivizing agents with dynamic preferences. arXiv preprint arXiv:1807.02297 (2018)
7. Guresti, B., Vanlioglu, A., Ure, N.K.: Iq-flow: Mechanism design for inducing cooperative behavior to self-interested agents in sequential social dilemmas (2023)
8. Ho, C.J., Slivkins, A., Vaughan, J.W.: Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. In: Proceedings of the fifteenth ACM conference on Economics and computation. pp. 359–376 (2014)
9. Laffont, J.J., Martimort, D.: The Theory of Incentives: The Principal-Agent Model. Princeton University Press, Princeton, NJ, USA (2001)
10. Manzo, G.: Agent-based models and causal inference. John Wiley & Sons (2022)
11. Mguni, D., Jennings, J., Sison, E., Valcarcel Macua, S., Ceppi, S., Munoz de Cote, E.: Coordinating the crowd: Inducing desirable equilibria in non-cooperative systems. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. p. 386–394. AAMAS '19, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2019)
12. Ratliff, L.J., Dong, R., Sekar, S., Fiez, T.: A perspective on incentive design: Challenges and opportunities. Annual Review of Control, Robotics, and Autonomous Systems **2**(1), 305–338 (2019)
13. Yang, J., Wang, E., Trivedi, R., Zhao, T., Zha, H.: Adaptive incentive design with multi-agent meta-gradient reinforcement learning. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems. p. 1436–1445. AAMAS '22, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2022)
14. Zemkohoo, A., Dempe, S.: Bilevel optimization advances and next challenges (2020)
15. Zheng, S., Trott, A., Srinivasa, S., Parkes, D.C., Socher, R.: The ai economist: Taxation policy design via two-level deep multiagent reinforcement learning. Science advances **8**(18) (2022)

Semantic enrichment of causal graphs for strategic foresight

Jože M. Rožanec[1[0000−0002−3665−639X]] and Gaël Gendron[2,3[0000−0002−2457−934X]]

[1] Jožef Stefan Institute, Jamova ulica 39, Ljubljana, Slovenia
joze.rozanec@ijs.si
[2] NAOInstitute, University of Auckland, New Zealand
gael.gendron@auckland.ac.nz

Abstract. Strategic foresight has been identified as a key tool to enhance policymaking and guide decision-making in private and public organizations. While the potential of artificial intelligence has been recognized in this domain, it has only been applied to specific tasks, and no AI-first approach has been developed yet. Among the multiple strategic foresight methodologies, one of the most frequently used ones is scenario planning. Nevertheless, creating such scenarios requires specific domain knowledge, particularly about causal relationships, to understand how forces of change may affect potential future outcomes. In this research, we describe some early results we obtained from semantic enrichment performed on causal graphs extracted from media news. The experiments were performed using ChatGPT 4o on 50 media events that correspond to oil prices in the first quarter of 2023. The results show that when performing semantic linking, different results are obtained if the extracted causal variables or the causal relationships are considered. While there is complete agreement regarding the assigned wiki concepts in 9% of the cases, such agreement falls below 33% in most of the cases. Furthermore, nearly 23% of the proposed wiki concepts do not correspond to real ones.

Keywords: Causal discovery · NLP · Graphs · Semantic enrichment · Strategic foresight.

1 Introduction and related work

Strategic Foresight aims to provide a structured approach to gathering information regarding plausible futures to prepare for change adequately. It frequently leverages experts' knowledge regarding trends and emerging issues to understand better how decisions and policies may influence the future and guide strategic planning and policy-making [4]. The ability to lead to better future outcomes has promoted an increasing adoption in the private and public sectors [12].

While artificial intelligence is increasingly being used across different domains, it has only been adopted to aid in specific strategic foresight tasks [11], and no end-to-end tool exists to (semi-)automate the complex process performed by domain experts. Among the most frequently used strategic foresight methods, we find scenario planning [1], which aims to foresee relevant scenarios based on trends and factors of influence to understand better how actions can influence the future [14]. Building such scenarios requires identifying the driving forces of change and possible outcomes while acknowledging the accompanying uncertainties. Knowledge about such causal relationships and context is currently owned by experts, who then estimate plausible outcomes. Artificial intelligence could automate this process using LLMs and causal inference frameworks to extract and test causal relationships and infer future scenarios. In particular, we envision using LLMs to extract candidate causal relationships from media news, assess whether such relationships effectively mean a causal relationship, and merge them into a causal graph [2]. The resulting causal graph could be analyzed, and subgraphs of interest could be identified to build scenarios about plausible futures for human examination.

LLMs have shown promising results but face challenges in causal inference from text due to the inherent complexity of natural language, which is often unstructured, high-dimensional, and semantically ambiguous [9]. LLMs have been used for causal discovery, effect estimation, and tasks like counterfactual reasoning. Research on causal discovery focuses on pairwise causal direction [8], though issues arise with LLMs repeating embedded knowledge [15] or inferring causal relations from entity order [7]. For full causal graph discovery, LLMs have been mostly applied on datasets, where they have even outperformed baselines [10]. Among the few studies focusing on extracting causal links from the text, we can mention [3] and [5]. Jin et al. [6] proposed an alternative approach using a chain-of-thought prompting strategy to extract a causal graph and context (e.g., conditional or interventional probabilities) to perform correct causal inferences and answer causality questions.

Gendron et al. [2] have recently described an approach leveraging LLMs for causal relationship extraction from media news. Their approach shows promise in extracting observed and hidden causal variables and their causal relationships. Nevertheless, the experiments were executed only on a handful of cases and the causal variables and relationships they extract are encoded as strings. While encoding such knowledge as strings can sometimes provide rich expressiveness, we consider that they could be further enriched associating them with a semantic concept. This would allow for a normalized understanding of the causal variables and enable further interoperation with ontologies and knowledge bases. This work, therefore, explores on how such enrichment can be performed, and provides a quality assessment of the causal graphs described above.

2 Experiments and results

This research is part of a wider research effort on how artificial intelligence can be applied to enhance strategic foresight [13]. The manuscript aims to describe experiments and results we obtained when pursuing two research goals (a) semantically enrich the causal variables identified in media news by following the prompts described by Gendron et al. [2] and (b) assess how reliable is the semantic linking obtained through LLM prompting.

Methodology The experiments were performed considering 50 media news events related to oil prices which were reported in the first quarter of 2023 and extracted from EventRegistry. A total of 169 causal graphs from the events were obtained by applying the prompt described in Gendron et al. [2]. Their observed edges and nodes were further processed to semantically enrich them. In particular, two enrichments were performed: (i) considering the strings describing the already extracted causal variables and (ii) considering the edge description that explains how two causal variables are related to each other. The outcomes were then compared to understand to what extent did (i) and (ii) lead to the same understanding. To that end, we measured the Jaccard similarity of the extracted wiki concept URLs for causal variable pairs associated through a causal relationship. Finally, we measured how many of the wiki concept URLs issued by the LLM corresponded to real ones. The work was performed using the OpenAI ChatGPT 4o model.

Results The results we obtained from the semantic enrichment show that in most cases the wiki concepts retrieved from the causal variables do not match the ones described when obtained from the causal relationships. In fact, in 54% of the cases, the Jaccard similarity was zero, in 36% of the cases the Jaccard similarity was 0.33, and in just 9% of the cases we could observe a perfect match. We randomly picked some cases to understand the quality of the extracted wiki concepts and understand whether it would be better to consider the extracted causal variables or rely on the causal relationships. One such example was *»Influence of Russian supply cuts on oil prices«*. The causal graph prompt correctly identified that *»Russian supply cuts«* influence *»Oil prices«*. When executing the additional prompts, the causal variables were mapped to *Economy of Russia: Natural resources and energy exports* and *Oil price* and the causal relationship to *Oil supply* and *Oil prices*, which is not tied to a particular country and more accurately describes the key resource being considered in the market dynamics. Finally, when assessing how many unique wiki links were valid, we found that 204 unique links corresponded to real wiki concepts, while 59 entries were misleading.

Conclusion and Future work Our research regarding causal extraction and counterfactuals has not been tested on an extensive dataset. Further research is required to understand (i) how causal graph extraction prompts can be enhanced to yield more accurate results, (ii) how to enhance the semantic enrichment to avoid LLM hallucination scenarios. Future work will address this gap by (a) refining the causal graph extraction, (b) enrich the semantic linking considering domain-specific ontologies, and (c) testing the proposed approach on a dataset spanning multiple years of media news.

Acknowledgments. This research was developed as part of the Graph-Massivizer project funded under the Horizon Europe research and innovation program of the

European Union under grant agreement 101093202 and supported by the Slovenian Research Agency.

Disclosure of Interests. The authors declare that they have no competing interests.

References

1. Ebadi, A., Auger, A., Gauthier, Y.: Detecting emerging technologies and their evolution using deep learning and weak signal analysis. Journal of Informetrics **16**(4), 101344 (2022)
2. Gendron, G., Rožanec, J.M., Witbrock, M., Dobbie, G.: Counterfactual causal inference in natural language with large language models. arXiv preprint arXiv:2410.06392 (2024)
3. Gopalakrishnan, S., Garbayo, L., Zadrozny, W.: Causality extraction from medical text using large language models (llms). arXiv preprint arXiv:2407.10020 (2024)
4. Greenblott, J.M., O'Farrell, T., Olson, R., Burchard, B.: Strategic foresight in the federal government: a survey of methods, resources, and institutional arrangements. World futures review **11**(3), 245–266 (2019)
5. Hobbhahn, M., Lieberum, T., Seiler, D.: Investigating causal understanding in llms. In: NeurIPS ML Safety Workshop (2022)
6. Jin, Z., Chen, Y., Leeb, F., Gresele, L., Kamal, O., Zhiheng, L., Blin, K., Adauto, F.G., Kleiman-Weiner, M., Sachan, M., et al.: Cladder: Assessing causal reasoning in language models. In: Thirty-seventh conference on neural information processing systems (2023)
7. Joshi, N., Saparov, A., Wang, Y., He, H.: Llms are prone to fallacies in causal inference. arXiv preprint arXiv:2406.12158 (2024)
8. Kıcıman, E., Ness, R., Sharma, A., Tan, C.: Causal reasoning and large language models: Opening a new frontier for causality. arXiv preprint arXiv:2305.00050 (2023)
9. Ma, J.: Causal inference with large language model: A survey. arXiv preprint arXiv:2409.09822 (2024)
10. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
11. Pratt, L., Bisson, C., Warin, T.: Bringing advanced technology to strategic decision-making: The decision intelligence/data science (di/ds) integration framework. Futures **152**, 103217 (2023)
12. Rosa, A.B., Gudowsky, N., Repo, P.: Sensemaking and lens-shaping: Identifying citizen contributions to foresight through comparative topic modelling. Futures **129**, 102733 (2021)
13. Rozanec, J., Šircelj, B., Cochez, M., Leban, G.: Back to the future: predicting causal relationships influencing oil prices. In: The Second Tiny Papers Track at ICLR 2024
14. Wilkinson, A.: Strategic foresight primer. European Political Strategy Centre (2017)
15. Zečević, M., Willig, M., Dhami, D.S., Kersting, K.: Causal parrots: Large language models may talk causality but are not causal. arXiv preprint arXiv:2308.13067 (2023)

Causal Discovery of Non-Stationary Dynamic Causal Bayesian Networks

Mario De Los Santos[1], Felipe Orihuela-Espina[2], and Luis Enrique Sucar[1]

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México
[2] University of Birmingham, Birmingham, United Kingdom
andres.santos@inaoep.mx,f.orihuela-espina@bham.ac.uk,esucar@inaoep.mx

Abstract. Non-stationary time series present challenges in causal discovery due to their evolving statistical properties. Existing probabilistic causal frameworks often assume static structures or parameters, limiting their effectiveness in dynamic environments. This research introduces a non-stationary dynamic causal Bayesian network (nsDCBN) model designed to iteratively detect and adapt to significant structural and parametric shifts in time series data. The proposed method identifies and captures these shifts using a probabilistic graphical approach. Preliminary results on synthetic fNIRS data demonstrate the algorithm's ability to detect changes in causal structures despite noise. These findings suggest potential applications in brain connectivity analysis during neurorehabilitation, where evolving brain connectivity is crucial.

Keywords: Causal Discovery · Non-stationarity · Causal Models

1 Introduction

Stationarity is a complex concept, as many processes exhibit stationary and non-stationary behaviors depending on the sampling context. This creates challenges for causal discovery, where models often assume a fixed causal structure. These assumptions are inadequate in scenarios like neurorehabilitation, where brain connectivity evolves due to neuroplasticity.

Causality has been defined in various ways, with contributions from Granger, Pearl, Lamport, Kosko, and others. While Pearl's framework dominates causal Bayesian networks, applying traditional algorithms to non-stationary processes is insufficient for streaming data, an underexplored area.

To address this, we propose a non-stationary Dynamic Causal Bayesian Network (nsDCBN) that adapts its structure in response to changes. Using Functional Near-Infrared Spectroscopy (fNIRS) as a test case, we enhance signal quality through preprocessing and detect structural shifts. We demonstrate our model's potential through synthetic fNIRS data with applications in neurorehabilitation. This paper presents our approach and preliminary results.

2 Methods

2.1 Synthetic Data

Synthetic data enables evaluation and validation of the proposed model before applying it to real-world data. We have reimplemented the bilinear model for fNIRS [1] and extended it to simulate structural and parametric changes, incorporating realistic noise components (e.g., motion artifacts, physiological, and instrumental noise) [2] to mimic neural, hemodynamic, and optical responses in fNIRS including semisynthetic data where noise from experimental data -rather than synthetic- is added.

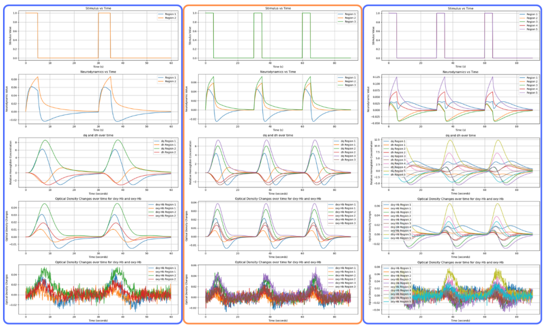


Fig. 1. Example of synthetic data generation based on a causal model.

The model simulates neurodynamics, hemodynamics, and optics, allowing for benchmarking causal discovery algorithms against known ground truths [6]. This provides a solid basis for testing noise removal and detecting structural changes in non-stationary environments. In particular, the extended bilinear model allows for the generation of non-stationary processes.

2.2 Non-Stationary Detection

We propose analyzing four statistical moments—mean, variance, skewness, and kurtosis—to identify stationary to non-stationary state transitions through statistical tests (e.g., Kwiatkowski-Phillips-Schmidt-Shin). The algorithm operates iteratively, comparing changes across overlapping and non-overlapping time windows to detect emerging trends and trigger the causal discovery process. This approach iteratively adapts to changes in structure and parameters.

2.3 Causal Discovery Framework

Once a change in the statistical moments is detected, the algorithm proceeds to uncover causal relationships in the data. We exploit existing algorithms, such as CD-NOD [3], PC [5], or LiNGAM [4], to iteratively refine the causal graph, depending on the specific characteristics of the synthetic and semisynthetic datasets. This iterative approach allows for adapting the causal structure within each time slice, improving the representation of the evolving system.

2.4 Non-Stationary Dynamic Causal Bayesian Network (nsDCBN) Learning

The final stage of the methodology focuses on learning the non-stationary Dynamic Causal Bayesian Network (nsDCBN). If this is our proposal, we must explain it in detail here. Once a causal structure is identified, it is used as prior knowledge in subsequent iterations, allowing the model to adapt to the new stochastic state and capture shifts in causal relationships over time. This approach requires less data to update the model.

2.5 Experiments

Validation occurs in two stages: first, using synthetic datasets to ensure alignment with known ground truths, and second, applying the model to observational fNIRS data to assess its effectiveness in capturing brain connectivity changes during neurorehabilitation. Concurrence and statistical tests will confirm the model's accuracy.

Performance will be assessed using structural hamming distance (SHD), which quantifies the differences between the discovered causal structure and the known ground truth. This ensures that the framework captures underlying causal dynamics in non-stationary environments.

3 Preliminary Results

We generated two synthetic data sets with differing structures, each simulating three regions. Each dataset spans 80 seconds, sampled at 10.84 Hz. To evaluate noise robustness, we simulated synthetic noise with a 10% noise-to-signal ratio, incorporating sources such as heart rate variability, breathing patterns, and external perturbations. This controlled dataset allows for precise validation of the causal discovery models by providing known ground truths.

3.1 Non-Stationarity Detection:

The detection algorithm uses the first four statistical moments (mean, variance, skewness, kurtosis) across overlapping and independent time windows and the KPSS test to identify shifts from stationary to non-stationary states.

Initial validation on a synthetic sine signal with controlled amplitude changes hinted at the algorithm's effectiveness (Figure 2A), with reliable identification of changes. Further testing on synthetic fNIRS data confirmed accurate detection of structural changes in real-time, with an average time lag of ±10 seconds

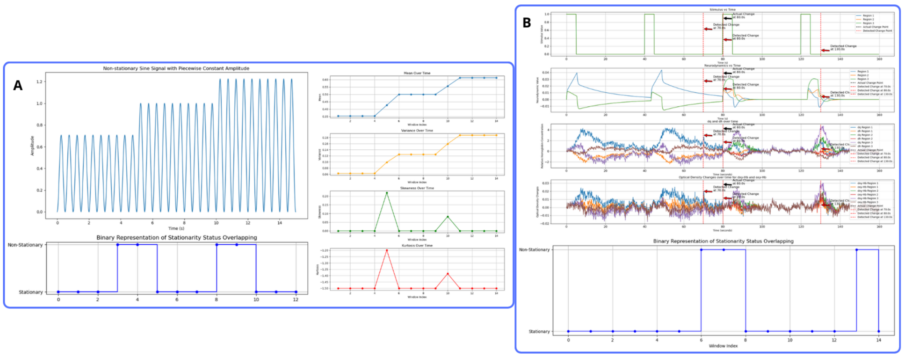


Fig. 2. Example of non-stationarity detection in (A) sine signal and (b) synthetic fNIRS data.

in a sequence of 160 seconds (Figure 2B). Although a few false positives were observed, likely attributable to elevated noise levels in the signal, the overall false-positive rate remained low, indicating the algorithm's capabilities even under noisy conditions.

4 Conclusions

We presented a first approximation to causal discovery in non-stationary streaming data, combining non-stationarity tests with local applications of existing causal algorithms. Preliminary results on synthetic fNIRS data demonstrate reliable structural change detection with minimal time lag and low false positives.

Ongoing work will strengthen the algorithm with additional statistical tests and noise reduction methods. Future efforts will focus on refining the nsDCBN as changes occur, validating the model with synthetic and real fNIRS data to recover brain connectivity. We also aim to explore real-time applications to improve scalability and accuracy in dynamic systems.

References

1. Tak, S., et al.: Dynamic causal modeling for fNIRS. Neuroimage **111**, 338–349 (2015)
2. Elwell, C. E., et al.: Oscillations in cerebral haemodynamics. In: Oxygen Transport to Tissue XXI, pp. 57–65. Springer, Boston, MA (1999)
3. Huang, Biwei, et al. "Causal discovery from heterogeneous/non-stationary data." Journal of Machine Learning Research 21.89 (2020): 1-53.
4. Shimizu, Shohei. "LiNGAM: Non-Gaussian methods for estimating causal structures." Behaviormetrika 41.1 (2014): 65-98.
5. Spirtes, P., Glymour, C., Scheines, R. (1993). "Causality, Prediction, and Search". Springer-Verlag.
6. De Los Santos-Hernández, M., Orihuela-Espina, F., Sucar, L. E. (2024). A framework for synthetic fNIRS data generation. Poster presented at the fNIRS Conference 2024.

Section 3

Applications

Preventing Collisions in Self-driving Cars using Probabilistic Logic Counterfactual Reasoning

Verónica Rodríguez[1][0000−0002−5976−9338], Héctor Avilés[2][0000−0001−5310−3474],
Rubén Machucho[2][0000−0002−5731−6677], Alberto Reyes[3][0000−0002−8509−6974],
Marco Negrete[4][0000−0002−5468−2807], Gabriel Ramírez[5][0000−0001−5226−5615],
Alberto Petrilli[6][0000−0002−2766−7599], Ingridh Gracia[2][0009−0004−7101−6656],
Gloria De-La-Garza[2][0009−0004−2446−6435], Karelly Rivera[2][0000−0002−4749−0663],
and Rafael Kiesel[7][0000−0002−8866−3452]

[1] Technological University of the Mixteca, 69004 Oaxaca, Mexico
veromix@mixteco.utm.mx
[2] Polytechnic University of Victoria, 87138 Tamaulipas, Mexico
{havilesa,rmachuchoc,2330319,2130071,1930435}@upv.edu.mx
[3] National Institute of Electricity and Clean Energy, 62490 Cuernavaca, Mexico
areyes@ineel.mx
[4] National Autonomous University of Mexico, 04510 Mexico City, Mexico
marco.negrete@ingenieria.unam.edu
[5] Center for Research and Advanced Studies of the National Polytechnic Institute,
Tamaulipas Campus, 87138 Tamaulipas, Mexico
grtorres@cinvestav.mx
[6] Tohoku University, 980-8577 Miyagi, Japan
a.petrilli@srd.mech.tohoku.ac.jp
[7] Vienna University of Technology, 1040 Vienna, Austria
rafael.kiesel@tuwien.ac.at

Abstract. We propose counterfactual reasoning through probabilistic logic twin networks (*PLTNs*) to prevent collisions in self-driving cars. The basis of a PLTNs is a causal Bayesian network (*cBN*) partially learned from simulated self-driving car data and synthetic data. The cBN include the lane of the self-driving car, the presence of up to 4 surrounding vehicles, an indicator for potential collisions, and 6 driving actions. Counterfactual queries through the PLTNs intervene with alternative actions to identify which minimizes the probability of a collision. For evaluation, three cBNs are learned with 1%, 50%, and 100% of a training dataset. For querying, 120 state-action examples labeled as leading to a crash are selected randomly. Each one is associated with six possible interventions. The probability of a collision is then queried from PLTNs, provided that a potential collision has been warned, and the current state and action are known. Results show that all intervened actions minimizing the probability of a crash does not lead to a car crash, suggesting the effectiveness of this approach in developing collision prevention schemes for self-driving cars. To the best of our knowledge, this is the first application of PLTNs for counterfactual reasoning in autonomous vehicles.

Keywords: Counterfactual reasoning · probabilistic logic twin networks · Autonomous vehicles.

1 Introduction

Collision prevention is an important concern in self-driving cars. Although diverse approaches exist to achieve this goal, many rely on adjusting the current trajectory or behavior of the self-driving car to a new, safer course of action [19–21]. In this context, we propose probabilistic counterfactual queries to pose questions about prospective, hypothetical situations like:

> "*What is the probability that a situation of potential collision could prevail, given that the current action and the state of the system are known, if a different action were chosen?*"

Answers to that kind of "*What-if*" questions can help self-driving cars evaluate the convenience of alternative driving actions to perform safer maneuvers under hazardous situations [16]. Additionally, the probabilistic causal models associated with causal questions can be framed in probabilistic logic programming [15], a paradigm well-suited for modeling causal relationships due to both its clear and flexible rule-based representation, and the availability of sophisticated inference procedures to efficiently solve probabilistic queries on complex probability distributions [6, 8].

Therefore, in this paper, we present our ongoing work on counterfactual reasoning using probabilistic logic twin networks (*PLTNs*) as a counterfactual model to identify alternative actions that may prevent collisions in self-driving cars. PLNTs are twin networks [2] encoded as probabilistic logic programs. Inference is carried out through Counterfactuals [11], an efficient and effective tool recently developed to solve probabilistic counterfactual queries in PLTNs. Causal Bayesian networks (*cBN*) are proposed as the basis of the PLTNs. Three cBNs are constructed, for the most part, from a large dataset of examples taken from multiple runs of simulations of our self-driving car under autonomous and human control, and from synthetic data. The state variables included in the cBNs are the lane in which the self-driving car is traveling and six "occupancy" variables that indicate the presence of surrounding vehicles. On each state, the self-driving car can choose one of six possible actions. When a potential collision is warned, the goal is to calculate the probability of the collision given the current state and action, by intervening in the causal model with alternative actions, one at a time. It is assumed that the counterfactual action minimizing the probability of the collision is the best alternative for the self-driving car to avoid or mitigate the severity of the impact.

To evaluate our approach, three cBNs were constructed using 1%, 50%, and 100% of a training dataset containing over 1,900,000 examples of state-action pairs. For testing, 120 examples were randomly selected from a pool of 288 state-action pairs labeled as potential crashes. Each of these examples was subsequently associated with six alternative actions implemented in our simulated self-driving car, forming a group of six examples that encompass the state, the observed action, and possible interventions. Examples within each group are independently queried to a PLTN to compute its corresponding probability value of a crash.

A qualitative analysis of the results shows that, in all 120 cases the counterfactual actions with the lowest probability of potential collision within each group does not lead to a crash in any of the PLTNs. This finding suggests the viability of the proposed approach for developing a collision avoidance module in self-driving cars.

2 Related work

Recent research into autonomous vehicles (*AVs*) and their safety has made significant strides using counterfactual reasoning, what-if analysis, and advanced simulation techniques to address collision risks and enhance traffic safety. Studies have employed counterfactual simulations to evaluate the performance of AVs in preventing collisions and improving safety. For example, research on the Waymo Driver [17] demonstrated its effectiveness in avoiding fatal collisions by simulating crash scenarios and showing that it could prevent or mitigate a substantial percentage of crashes. Similarly, the impact of Advanced Driver Assistance Systems (*ADAS*) has been analyzed using counterfactual reasoning to assess their safety benefits and drawbacks in real-world scenarios [3]. This approach allows for a nuanced understanding of how different safety technologies might alter crash outcomes and driver behavior. Another relevant study focused on predicting crash configurations and the impact of Autonomous Emergency Braking (*AEB*) systems [12]. This research utilized counterfactual simulations to identify specific crash scenarios that AEB systems could not address, thus highlighting areas for improvement. Additionally, counterfactual reasoning has been applied to estimate the importance of objects in autonomous driving environments [10], enhancing the system's ability to prioritize critical objects and reduce collision risks. The effectiveness of pre-crash safety technologies, such as AEB and Electronic Stability Control (*ESC*), has also been evaluated in the context of reducing severe injuries in crashes [14]. This study underscores the role of these technologies in mitigating collisions and emphasizes the need for continued advancements to address remaining safety concerns. Furthermore, research on the design and evaluation of Automated Driving Systems (*ADS*) has utilized injury risk modeling to understand the potential outcomes of various crash scenarios [13]. By developing comprehensive injury risk surfaces, this research provides valuable insights into how different ADS designs could impact overall safety. The importance of considering driver behavior models in counterfactual simulations has been highlighted as well in [5]. Different models can significantly influence the effectiveness estimates of safety systems, underscoring the need for accurate simulations to evaluate safety technologies properly.

3 Methodology

3.1 Testbed and datasets

Our development framework involves a self-driving car simulated in race-like environments using the Webots simulator. An example of the self-driving car and

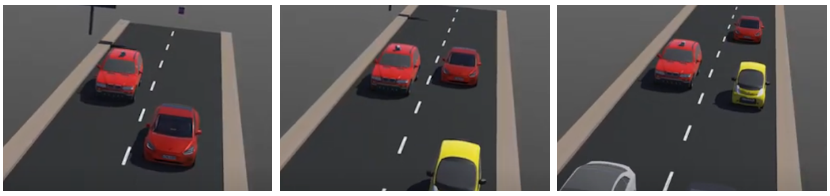


Fig. 1. Race-like environment considered in this study for our self-driving car (in bright red).

its environment is shown in Fig. 1. The car can travel on a two-lane road with straight segments and curves and up to 10 obstacle vehicles distributed over the road and either static or in motion. However, in this work, we are considering straight roads only. It is assumed that the vehicles traveling on the left lane moves faster than those on the right lane, while the self-driving car is the fastest (the maximum tested speed is near 50 km/h).

The architecture of the self-driving vehicle includes modules for *perception*, *driving control* and *decision making*. Perception utilizes an accelerometer for collision detection, 3D laser readings for detecting other vehicles, and a RGB camera to detect lane borders. The control module is responsible for lane tracking, and speed and steering estimation, following standard control laws. Decision making selects driving behaviors or actions based on a state-action policies estimated by solving probabilistic logic factored Markov decision processes (*PL-MDPs*) [1,4]. These policies primarily promote traveling on the right lane while using the left lane only for overtaking. For decision making, the vehicle behind the self-driving car in the same lane is not explicitly considered. State variables are `curr_lane` which identifies the current lane of the self-driving car, and occupancy variables called `free_E`, `free_NE`, `free_NW`, `free_SE`, `free_SW` and `free_W` which indicate whether there is a vehicle or not in the location indicated in the name of each variable, relative to the self-driving car. Figure 2 depicts the locations represented by the occupancy variables on each lane. In this work, we incorporate a new variable called `latent_collision` used to warn a potential collision in the trajectory of the self-driving car. All these variables are binary. Finally, a multi-valued variable called `action` identifies one of four driving maneuvers for the self-driving car: **change_to_left** and **change_to_right** used for overtaking on the left and returning to the right lane, respectively, **cruise** to reach a steady (maximum) speed, **keep** to maintain a safe, steady distance to a vehicle ahead in the same lane, and **swerve_right** and **swerve_left** which perform a controlled veer to the side of the lane while reducing speed. We consider the two latter actions also as safe, evasive maneuvers in case of unexpected situations.

We have recorded several runs of the self-driving car system under both autonomous control and human command via a human-friendly interface developed on purpose[8]. These datasets include 1,238,869 driving decisions through

[8] These datasets are available at: https://www.kaggle.com/autonomousvehicle/

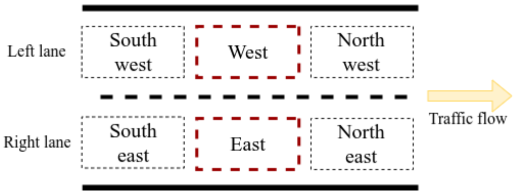


Fig. 2. Predefined locations for other vehicles around the self-driving car (dashed red lines indicate the space the self-driving car occupy on each lane). When the vehicle is on the left (resp. right) lane, only the locations Northwest, Northeast, East and Southeast (resp. Northeast, Northwest, West and Southwest) are meaningful.

more that 300 km. The source code of our self-driving vehicle and some videos of the self-driving system running are available at: https://github.com/hector-aviles/AIRJ2024/.

The approach used in this work requires labeling each state-action pair in the dataset as either leading to a collision or not. In the proposed driving environment, we consider that a crash may occur principally in two situations: (a) when the self-driving car applies the action **cruise** and there is a vehicle ahead in the same lane (rear-end collision), or (b) when there is a car either next to or ahead in the lane the self-driving car merges into (sideswipe and rear-end crash, respectively). Following the previous criteria, from the complete space of $2^7 \times 6 = 768$ state-action pairs, we carefully labeled by hand 288 pairs as leading to a potential collision. Some tate-action pairs that involve **change_to_left**, **change_to_right** and **cruise** emerged as the "unsafe" driving combinations. With this new list as reference, we found that the automated and human control databases contain only a small number of potential crash examples (39,695 pairs, all of which are repetitions of 132 unique pairs). Unfortunately, as of the time of writing, sampling potential crashes by human control has not been completed due to its tedious and time-consuming nature. Instead, the 288 state-action pairs were replicated 2,500 times to generate a synthetic dataset of 720,000 potential crashes. The number of repetitions was selected arbitrarily, with the aim of approximating a balance between collision and non-collision examples. During the process, continuous variables such as the speed and steering of the self-driving car were random sampled from their known data distributions, although they are not considered in the present work. This new dataset of crashes complements the driving records originally obtained from the autonomous and human control of the self-driving car. Each example in the integrated dataset was labeled using `latent_collision` as a potential collision or not. Table 1 summarizes the number of actions labeled as potential collisions or non-potential collisions in the complemented dataset. Table 2 shows the number of unique state-action pairs on each subset.

Table 1. Number of actions in state-action pairs labeled as potential collisions and non-potential collisions in the integrated dataset.

Actions	# of examples of non-potential collisions	# of examples of potential collisions	**Total:**
change_to_left	40,308	289,809	330,117
change_to_right	39,814	294,459	334,273
cruise	617,930	175,429	793,359
keep	496,558	0	496,558
swerve_left	2,883	0	2,883
swerve_right	1,681	0	1,681
Total:	1,199,174	759,695	1,958,869

Table 2. Number of unique state-action pairs labeled as non-potential collisions and potential collisions ("safe" and "unsafe", respectively).

Actions	# of unique "safe" state-action pairs	# of unique "unsafe" state-action pairs	**Total:**
change_to_left	14	112	126
change_to_right	15	112	127
cruise	53	64	117
keep	67	0	67
swerve_left	36	0	36
swerve_right	29	0	29
Total:	214	288	502

3.2 Learning of the causal Bayesian networks

Structural and parameter learning of cBNs were instrumented with the **bnlearn** package in R package [18]. Its directed acyclic graph is shown in Fig. 3. Structural learning was carried out by means of hill-climbing greedy search with Bayesian information criterion. Initially, we devised a preliminary graph that we wished to test, so undesired links between variables were forbidden (for instance, those between occupancy variables, or from `action` to `curr_lane`). In contrast, only the relation that goes from `action` to `latent_collision` was explicitly requested to be included, while letting hill-climbing to decide about the rest of the connections. We believe the resulting structure resembles a causal graph with confounders, with `action` being the treatment variable, `latent_collision` being the output variable and the current lane and occupancy variables jointly playing the role of a single, multi-valued confounder. The parameters of the cBN were fitted by the *maximum likelihood estimation* criterion.

For training, we considered the integrated dataset containing 1,958,869 examples. Training data is organized into `state-action-latent_collision` triplets. From our perspective, data leakage (that may occur when non-disjoint datasets

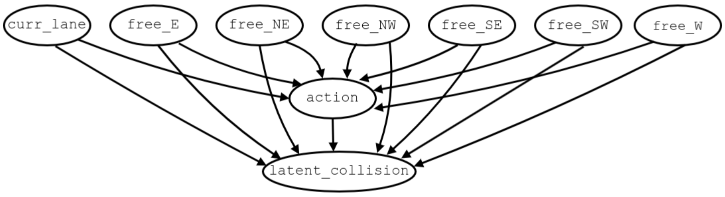


Fig. 3. Causal Bayesian network (*cBN*) proposed in this work.

Table 3. Number of unique state-action pairs as a function of the action used for training and testing each cBN.

	Training dataset			Test
Action	1%	50%	100%	dataset
change_to_left	122	126	126	49
change_to_right	118	126	127	46
cruise	101	114	117	25
keep	31	62	67	0
swerve_left	7	32	36	0
swerve_right	4	29	29	0

are used for training and testing) does not represent a major concern in this setup. This is because the core of the evaluation focuses on comparing conditional probabilities under interventions for a counterfactual model, rather than studying the generalization capabilities of PLTNs to previously unknown data. Despite this, we consider constructing cBNs using training sets of varying sizes for comparison purposes. To achieve this, we randomly selected 1%, 50%, and 100% of the training dataset for structural and parameter learning of the cBNs. Table 3 summarizes the number of unique state-action pairs used for training and testing for each driving action and cBN.

3.3 Counterfactual queries

Counterfactual querying requires a cBN to be provided as a parameter to the Counterfactuals library, formatted as a probabilistic logic program in ProbLog syntax [9]. To achieve this, we developed a purpose-built R script to syntactically translate the cBN learned with **bnlearn** into a ProbLog program. Although this conversion results in a large format size in comparison to the well-known compactness of traditional conditional probability tables, we have found that the probabilistic logic representation of a cBN as a set of probabilistic facts and rules was convenient during development for inspection, debugging and communication. In the Counterfactuals library, two copies of the ProbLog program are required to represent both the real and the counterfactual worlds. The two ProbLog programs have the same rules as the original program and share probabilistic facts representing not observable external factors. To obtain the twin networks, each rule of the original ProbLog program is duplicated, labeling the variables on each copy with the superscript **e** for the real, and **i** for the counterfactual world, respectively. As an example, consider the following Counterfactuals representation of a twin network inspired in the model proposed in this work:

```
% Probabilistic facts
0.5349662::u1.
0.7046031::u2.
0.4785868::u3.
0.1143778::u4.

% Rules for the real world
free_NE^e :- u1.
free_W^e :- u2.
action(keep)^e :- free_NE^e, free_W^e, u3.
latent_collision^e :- action(keep)^e, free_NE^e, free_W^e, u4.

% Rules for the counterfactual world
free_NE^i :- u1.
free_W^i :- u2.
action(keep)^i :- free_NE^i, free_W^i,u3.
latent_collision^i :- action(keep)^i, free_NE^i, free_W^i, u4.
```

A twin ProbLog program of this type constitutes a counterfactual model, which is used to solve probabilistic counterfactual queries. A counterfactual query is evaluated by first absorbing the evidence in the real ProbLog rules, then performing the intervention in the counterfactual ProbLog rules, and finally, using the internal inference engine of Counterfactuals [7] to assess the conditional probability of a hypothetical event given the observed values of certain variables as evidence. This scheme allows us to perform queries about the probability value of any variable of the cBN being true, given the evidence available, along with interventions on the truth values of one or more variables. In our case, the query is focused on the variable `latent_collision`. The evidence includes the current lane, the occupancy variables and the action it is performed. The counterfactual variable is `action`. Thus, we are interested in calculating probability values of the type[9]:

$$
\begin{aligned}
&P\big(\texttt{latent\_collision}^i = \mathbf{T} \big| \texttt{latent\_collision}^e = \mathbf{T}, \texttt{curr\_lane}^e = \textbf{right}, \\
&\quad \texttt{free\_E}^e = \mathbf{T}, \texttt{free\_NE}^e = \mathbf{F}, \texttt{free\_NW}^e = \mathbf{F}, \texttt{free\_SE}^e = \mathbf{T}, \\
&\qquad \texttt{free\_SW}^e = \mathbf{T}, \texttt{free\_W}^e = \mathbf{F}, \texttt{action}^e = \textbf{cruise}, \\
&\qquad\qquad do(\texttt{action}^i = \textbf{keep})\big)
\end{aligned}
\tag{1}
$$

[9] The truth value **T** means the truth-value true, and **F** means false (in the case of the occupancy variables, **T** indicates that a location is empty and **F** that it is occupied by another vehicle).

Table 4. Number of groups with a unique minimum probability value and the number of ties for first place (ranging from 2 to 5) across all groups and counterfactual models.

# of ties (1^{st} place)	# of cases (1%)	# of cases (50%)	# of cases (100%)	**Total:**
No ties	43	23	20	86
2	32	32	33	97
3	28	38	42	108
4	15	26	24	65
5	2	1	1	4
Total:	120	120	120	360

4 Evaluation and results

The evaluation of a counterfactual model involves a random selection of 120 unique `state-action-latent_collision` triplets from the list of identified potential collisions (that is, in which `latent_collision` = **T**). These triplets are further expanded by adding an intervention over the observed action with each of the six actions that the self-driving car can perform (including the action originally observed). Each unique triplet is thus transformed into a group of six `state-action-latent_collision-intervention` quartets, all sharing the same observed `state-action-latent_collision` triplet, but incorporating a different intervention on the action. From each quartet, a query similar to that in Equation 1 is derived to be solved by the counterfactual models. In total, each counterfactual model solves 720 queries. The objective is to identify an action (among the six intervention alternatives) that minimizes the probability of `latent_collision`. It is assumed that the counterfactual actions with the lowest probability of `latent_collision` represents the best decision at hand to prevent or reduce the severity of an accident under the observed circumstances.

Table 4 presents the number of ties for the first place observed across the 120 groups for the three counterfactual models when ranking the quartets within each group from lowest to highest, according to their probability values. We consider there can be from 2 to 5 ties on each group (the action currently observed has a probability value of 1 and there is at least one safe action on each state). For tie-breaking (that is, selecting an appropriate intervention among those sharing the lowest probability value), we implemented a straightforward, standard approach based on random selection. In accordance with our initial labeling of state-action pairs as potential or non-potential collisions, results show that in all three counterfactual models and test examples, the intervened action associated to the lowest probability value for `latent_collision` may indeed prevent a crash. Table 5 resume the number of actions selected as the best alternative for each counterfactual model, following the previous scheme. The average time for solving each counterfactual query on the three models is 5.74 seconds (SD=0.15). All tests were performed on a standard Core i7 laptop computer.

Table 5. Number of actions selected as the optimal intervention in the three counterfactual models.

Action	# of best interventions (1%)	# of best interventions (50%)	# of best interventions (100%)	**Total:**
change_to_left	1	1	0	2
change_to_right	0	4	2	6
cruise	23	25	22	70
keep	59	51	47	157
swerve_left	22	24	30	76
swerve_right	15	15	19	49
Total:	120	120	120	360

Table 6. State-action pairs with 5 tied actions. The first two pairs, from top to bottom, correspond to the counterfactual model trained with 1% of the data, and the last one is the same for the models trained with 50% and 100% of the data (all actions other than the observed are safe).

State variables							Observed
`curr_lane`	`free_E`	`free_NE`	`free_NW`	`free_SE`	`free_SW`	`free_W`	action
right	**F**	**T**	**T**	**F**	**T**	**T**	*CTR*
right	**F**	**T**	**T**	**F**	**F**	**T**	*CTR*
left	**T**	**T**	**T**	**F**	**F**	**T**	*CTL*

4.1 Discussion

The initial results described above are encouraging. A qualitative analysis revealed that 100% of the best driving decisions obtained through counterfactual reasoning prevent car collisions. When ties occur between alternative interventions, choosing an appropriate action for the current driving scenario becomes particularly important. In our case, this is especially relevant due to the significant number of ties identified on closer inspection of the results. A thorough analysis of the ranking of `state-action-latent_collision-intervention` quartet with respect to their probability values within each group demonstrated that, in all groups, all interventions with the lowest probability value are also safe actions (note that there may be more than one safe action for some states). For example, Table 6 show the 3 unique cases in which there are 5 equivalent interventions. Yet factors beyond safety that include efficiency on energy consumption, better use of available space, better adherence to traffic rules, cooperation with other vehicles and human preferences can also be taken into account to improve tie-breaking.

5 Conclusions and future work

We have presented a counterfactual reasoning approach to prevent collisions in a self-driving car using probabilistic logic twin networks. Twin networks are derived from causal Bayesian networks designed through expert knowledge, data from simulations and synthetic data. Three causal Bayesian networks were constructed and tested using training sets of varying sizes. Counterfactual reasoning allowed us to decrease the probability of a car crash by intervening with alternative maneuvers, yielding promising results. Although we have not yet fully exploited the descriptive and inferential capabilities of the Counterfactuals library, we recognize its suitability for development, debugging and analysis. This probabilistic logic approach does not only enhance clarity of the causal model, but it help us consider its deductive capabilities in explainability tasks in which it is useful to track the inference path.

As a future work, we plan to perform an exhaustive testing with more collision cases, make more extensive use of the pobabilistic logic approach to include new information for tie-breaking, and integrate counterfactuals into the decision-making module of our self-driving vehicle. Additionally, we will incorporate new information to the causal model, such as discrete versions of the distance from the self-driving car to other vehicles, as well as speed and steering of all the vehicles on the road. Furthermore, we will explore Counterfactuals to achieve counterfactual explanations about relevant variables and rules in decision-making, with the primary aim of enhancing transparency and trustworthiness in autonomous vehicle systems.

Acknowledgments. This work was partially founded by UNAM-DGAPA under grant IT102424 and AI Consortium - CIMAT-CONAHCYT.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Avilés, H., Negrete, M., Reyes, A., Machucho, R., Rivera, K., de-la Garza, G., Petrilli, A.: Autonomous Behavior Selection For Self-driving Cars Using Probabilistic Logic Factored Markov Decision Processes. Applied Artificial Intelligence **38**(1), 2304942 (2024)
2. Balke, A., Pearl, J.: Probabilistic evaluation of counterfactual queries, p. 237–254. Association for Computing Machinery, New York, NY, USA, 1 edn. (2022)
3. Bärgman, J., Victor, T.: Holistic assessment of driver assistance systems: how can systems be assessed with respect to how they impact glance behaviour and collision avoidance? IET Intelligent Transport Systems **14**(9), 1058–1067 (Aug 2019)
4. Bueno, T.P., Mauá, D.D., De Barros, L.N., Cozman, F.G.: Markov decision processes specified by probabilistic logic programming: representation and solution. In: 2016 5th Brazilian Conference on Intelligent Systems (BRACIS). pp. 337–342. IEEE (2016)

5. Bärgman, J., Boda, C.N., Dozza, M.: Counterfactual simulations applied to shrp2 crashes: The effect of driver behavior models on safety benefit estimations of intelligent safety systems. Accident Analysis & Prevention **102**, 165–180 (2017)
6. Chavira, M., Darwiche, A.: On probabilistic inference by weighted model counting. Artificial Intelligence **172**(6-7), 772–799 (2008)
7. Eiter, T., Hecher, M., Kiesel, R.: aspmc: An algebraic answer set counter. In: ICLP Workshops (2021), https://ceur-ws.org/Vol-2970/plppaper1.pdf
8. Eiter, T., Hecher, M., Kiesel, R.: aspmc: New frontiers of algebraic answer set counting. Artificial Intelligence **330**, 104109 (2024)
9. Fierens, D., Van den Broeck, G., Renkens, J., Shterionov, D., Gutmann, B., Thon, I., Janssens, G., De Raedt, L.: Inference and Learning in Probabilistic Logic Programs using Weighted Boolean Formulas. Theory and Practice of Logic Programming **15**(3), 358–401 (2015)
10. Gupta, P., Biswas, A., Admoni, H., Held, D.: Object Importance Estimation using Counterfactual Reasoning for Intelligent Driving. IEEE Robotics and Automation Letters **9**(4), 3648–3655 (2024)
11. Kiesel, R., Rückschlo, K., Weitkämper, F.: "What if?" in Probabilistic Logic Programming. Theory and Practice of Logic Programming **23**(4), 884–899 (2023)
12. Leledakis, A., Lindman, M., Östh, J., Wågström, L., Davidsson, J., Jakobsson, L.: A method for predicting crash configurations using counterfactual simulations and real-world data. Accident Analysis & Prevention **150**, 105932 (2021)
13. McMurry, T.L., Cormier, J.M., Daniel, T., Scanlon, J.M., Crandall, J.R.: An omnidirectional model of injury risk in planar crashes with application for autonomous vehicles. Traffic Injury Prevention **22**(sup1), S122–S127 (2021), pMID: 34402345
14. Puthan, P., Lubbe, N., Davidsson, J.: Characterizing future crashes on indian roads using counterfactual simulations of pre-crash vehicle safety technologies. IATSS Research **46**(4), 479–491 (2022)
15. Riguzzi, F.: Foundations of probabilistic logic programming. River Publishers, New York (May 2023)
16. Ruiz-Tagle, A., Lopez-Droguett, E., Groth, K.M.: A novel probabilistic approach to counterfactual reasoning in system safety. Reliability Engineering & System Safety **228**, 108785 (2022)
17. Scanlon, J.M., Kusano, K.D., Daniel, T., Alderson, C., Ogle, A., Victor, T.: Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. Accident Analysis & Prevention **163**, 106454 (2021)
18. Scutari, M.: Learning Bayesian Networks with the bnlearn R Package. Journal of Statistical Software **35**(3), 1–22 (2010)
19. Sharma, P., Rana, C.: Artificial intelligence based object detection and traffic prediction by autonomous vehicles – a review. Expert Systems with Applications p. 124664 (2024)
20. Tolba, M.A., Kamal, H.A.: SDC-Net++: End-to-End Crash Detection and Action Control for Self-Driving Car Deep-IoT-Based System. Sensors **24**(12), 3805 (2024)
21. Zhang, N., Tian, R., Fu, G.: Design and application of automatic driving emergency collision avoidance control algorithm based on artificial intelligence technology. Measurement: Sensors p. 101248 (2024)

Visual robot navigation incorporating causal models in deep reinforcement learning

Nilda G. Xolo-Tlapanco[1,2], Eduardo F. Morales[1], L. Enrique Sucar[1], and J. Ernesto Gomez-Balderas[2]

[1] Instituto Nacional de Astrofisica, Optica y Electrónica, Santa María Tonazintla, Puebla, México (nilda.xolo,emorales,esucar)@inaoep.mx
[2] GIPSA-Lab, Univ. Grenoble Alpes, CNRS, Grenoble INP, 38000 Grenoble, France gomezbaj@univ-grenoble-alpes.fr

Abstract. Deep reinforcement learning can be used for visual navigation in mobile robots but needs substantial computational resources and long training time. To reduce exploration time and improve adaptation to novel situations or environments, we created an algorithm to learn a Causal Bayesian Network of the task and use it in a Deep Reinforcement Learning algorithm (DQN). Experimentally, we show its effectiveness in simulated environments for a visual robot navigation task.

Keywords: Causal Reinforcement Learning · Deep Reinforcement Learning · Micro Air Vehicles.

1 Introduction

The use of Unmanned Aerial Vehicles (UAV) or drones, has recently increased, in recent years, due to the growth interest in using them for entertainment, but also for military, agriculture ([22]), delivery services ([9]), and rescue applications ([18]), among others. Their principal disadvantage is the need for a human pilot to control it, so multiple research works have used different artificial intelligence techniques to develop autonomous pilots ([12][8] [11]). One of the most popular techniques is Reinforcement Learning (RL) which has been used for collision avoidance and seeking an objective/goal in different environments ([4] [25] [14]). RL algorithms require a large amount of data and long times for training. A promising alternative is to use causal discovery to construct a causal model of the environment which can help accelerate the training phase.

Causal reinforcement learning integrates causal inference into RL, with the aim of utilizing the underlying causal relationships in the environment. Incorporating knowledge from a causal model and their construction into RL algorithms, e.g., [7] [28] [19], can greatly speed up the learning process by reducing the need for extensive exploration, but their use in robotics task has not been yet proved. In this research, we show how it can be used to autonomously control navigation tasks in a drone in a simulated environment and build two algorithms: one that learns a Causal Bayesian Network (CBN) of the task and environment during training of the DQN algorithm; and another one (with two variants) that uses a CBN to guide the selection of actions in the DQN algorithm.

2 Fundamentals

Reinforcement learning studies sequential decision problems. Mathematically, we can formalize these problems as Markov Decision Processes (MDPs). An MDP can be formally described by a 4-truple (S,A,P,R) where:

- S is a state space in which the process of evolution takes place.
- A is a set of all possible actions that control the state dynamics.
- $P_a(s, s') = Pr(S_{t+1} = s'|S_t = s, A_t = a)$ denotes the probability of transition (at time t) from state s to state s' under action a.
- $R_a(s, s')$ provides the immediate reward after transition from state s to s' with action a.

MDPs allow us to model the state evolution dynamics of a stochastic system when this system is controlled by an agent choosing and applying the actions a_t at every time step t. The procedure of choosing such actions is called action policy or strategy and is written as π. A policy specifies which action to take at a particular state. Solving a Markov decision problem implies searching for a policy that optimizes a performance (or optimality) criterion. One way to solve MDPs, particularly when the state and action spaces are too large for tabular methods, is the use of Deep Reinforcement Learning (DRL).

DRL combines the perception capabilities of deep learning with the decision-making capabilities of reinforcement learning by using deep neural networks, this can handle more complex scenarios with less engineered feature extraction. One of the most commonly used DRL algorithms is Deep Q-Learning (DQL), which updates the Q-values using the Bellman equation:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

Where:

- $Q(s, a)$ is the state-action value function, representing the expected utility of taking action a in state s.
- α the learning rate $(0 < \alpha \leq 1)$ determines to what extent newly acquired information overrides old information.
- r is the reward received after taking action a.
- $\max_{a'} Q(s', a')$ is the maximum predicted reward obtainable from the next state s', over all possible actions a'.
- γ is the discount factor $(0 \leq \gamma < 1)$. It represents the difference in importance between future rewards and immediate rewards.

Traditional Q-Learning stores Q-values in a Q-table, but this is infeasible for large, continuous state spaces. DQL solves this by using a neural network to approximate Q-values, inputting states and outputting actions but it needs extensive interactions to gather enough data for training. This data can be used for action selection or causal discovery.

2.1 Causal Discovery

Causal Modeling attempts to solve questions about possible causes by providing explanations of phenomena (effects) as the result of previous phenomena (causes). Several writers [17][15] specify three conditions that must be met in order to infer the existence of a causal relationship between two variables, X and Y: (1) a covariation between X and Y, (2) a time-ordered asymmetry, and (3) the elimination of other possible causes. Graphs are used to represent dependencies, where connected variables are dependent, and independent relations are implicit. Causal Bayesian Networks (CBNs) offer a framework for causal inference and prediction and represent stronger assumptions than Bayesian networks; CBNs assume all relationships correspond to actual causal connections [26].

The combination of RL and causal modeling is a relatively new field called Causal Reinforcement Learning (CRL). CRL is a suite of algorithms incorporating additional assumptions or prior causal knowledge into RL to analyze and understand the causal mechanisms underlying actions and their consequences. This enables agents to make more informed and effective decisions for more effective model learning, policy evaluation, or policy optimization [5]. It is generally divided into two categories, the first category is based on the prior causal information, where such methods typically assume the causal structure about the environment is given *a priori* by experts; while in the second category the causal information has to be learned [27]. Our work belongs to the second category, we developed an algorithm to learn the causal information from the interactions of the RL agent with the environment to later use it in policy construction.

3 Related Work

3.1 Deep reinforcement learning

DRL has been widely used to develop autonomous drone pilots to avoid obstacles and achieve their goals. Darwish et al. [4] introduced a model-free DRL method using a depth-RGB camera, outperforming DQN in intercepting specified targets. Shin et al. [25] demonstrated a drone navigating 3D obstacles with a dual input of an RGB camera and depth map for precise path finding. Cetin et al. [2] proposed DRL for autonomous navigation in suburban environments using depth images. Kersand et al. [14] used depth images and heading to train DQL algorithms. A challenge in training DRL agents is the time and data required. Causal models can be used for interpretability, task transfer, and faster training.

3.2 Causal Modeling

Causal Modeling (CM) needs rigorous data collection and expert knowledge. In machine learning, it has been used mainly to generate explanations. Shi et al. [24] introduce a model for interpreting causal relationships in a temporal-spatial context, capturing the causal connections between consecutive observations and decisions made by an RL agent. In [13], the authors use structural causal models to make human insight explicit in the causal relations used during the development of AI systems. Cetin et al. [3] use causal information to explain why the agent performs an specific action. It is based on the generation of a saliency map to identify the critical regions in an input image that influences the predictions made by the DQN agent. Diehl et al. [6] learn a CBN from simulation data to learn a cause-effect model of the environment, generating causal explanations.

3.3 Causal Reinforcement Learning

As we mentioned before, CRL is divided into two categories. The first category is based on prior causal information. Feliciano et al. [7] demonstrate better performance even with partial and spurious relationships by integrating a CM into Q-Learning in the light switch control tasks. Zhu et al. [28] use DQL in the Emotional Pendulum and Windy Pendulum tasks. Gonzalez et al.[10] introduced a decision-making approach for agents operating in environments characterized by uncertainty and underlying causal dynamics. This approach enabled the agent to continually update beliefs about the causal environment based on interactive outcomes. In their experimental setup, it was presupposed that the agent had prior knowledge of the causal framework governing the environment.

In the second category, the causal information has to be learned. Méndez et al. [19] learn a Causal Dynamic Bayesian Network for each of the agent actions and uses those models to improve the action selection process in the Coffee task. The same authors [20] developed a framework for simultaneously learning and using causal models to speed up policy learning in online MDPs, evaluated in the Coffee, Taxi and Taxi Atari tasks. MOCODA [21] applies a learned factored dynamics model to an augmented distribution of states and actions to generate counterfactual transitions for RL, and is used to train an offline RL agent to solve a robotics manipulation task.

It is clear that causal knowledge can help to accelerate reinforcement learning; however, there are very few approaches that learn causal models from data generated in RL and it has not been used for autonomus drones navigation.

4 Methodology

Data recollected by the agent during RL is used to learn a Causal Bayesian Network, which in turn is used in the action selection process of RL. We select Deep Q-Learning (DQN) as the algorithm to implement the visual navigation task in the drone. DQN uses two strategies to facilitate a more stable learning process: (i) An experience replay buffer, which stores the experiences of the agent and from which random samples are selected for the gradient descent process, (ii) the use of two networks to reduce the variance of the gradients, one fixed and used as a reference, and other one which is updated. The algorithm is described in Algorithm 1, the modifications on DQN are lines 5-7, where the Causal Bayesian Network is learned, and lines 8 and 14, where the Causal Bayesian Network is used.

Algorithm 1 Deep Q-Learning Algorithm

```
Initialize replay memory $D$ to capacity $N$, action-value function $Q$ with random weights $\theta$ and target action-value function $\hat{Q}$ with weights $\theta^- = \theta$
for episode $= 1, M$ do
    Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$
    for $t = 1, T$ do
        if step % k == 0 then
            Learn the Causal Bayesian Network
        end if
        Set the probability of each action to reach a negative and positive reward with the evidence of the state $t$
        rand = random number
        if rand < explorationRate then
            action $a_t$ = random action
        else
            action $a_t$ = MaxIndex(qValues)
            Use of the Causal Bayesian Network (Algorithm 3)
        end if
    end for
end for
```

4.1 Learning of the Causal Bayesian Network

The structure of the CBN is learned while training the agent in DRL using the Hill Climb search algorithm and the BIC score. Once we have the structure of the directed graph, we need to learn the parameters. For this, we calculate the Markov blanket of the reward node and delete synchronous links in the graph. To perform the inference with the observable state, we use the variable elimination algorithm [16]. The learning of the CBN is described in Algorithm 2, where k is a predefined number of steps to update the Causal Bayesian Network, and *numActions* is the number of actions in the actions set.

4.2 Use of the CBN in DQN

If the CBN model is already known, it can be used from the onset of the learning process. However, when the CBN needs to be learned, the RL algorithm must first accumulate sufficient data, this is necessary to ensure that the data collected is adequate for learning an accurate approximation of the true CBN. Algorithm 3 describes how to use the CBN within DQN. At each step, it takes the state elements present in the Markov blanket of the reward node as *filtered evidence*, which is used in the inference of reaching each state value. We divide this inference into two arrays, for the probability of transition to a positive reward (*ProbAct*) and for the probability to transition to a negative reward (*ProbActNeg*). Lines 10-13 describes the normal behavior of DQN, after that, depending of the probability of consulting the CNB *ProbCausalModel* (lines 14-26), the algorithm decides which action to take. If the action selected has a high probability (*ThresholdProb*) to reach a negative reward, the action is eliminated as

Algorithm 2 Learning the Causal Bayesian Network

```
At each step in the QDN algorithm save the state representation at time $t$, $t+1$,
    reward, and accumulated reward in the variable *data*
if step % k == 0 then
    for i = 0 to numActions do
        Check for the previous model
        if there is a previous model then
            Start with a pre-defined structure for the CBN
        else
            Start with a random initial structure for the CBN
        end if
        model[i]= HillClimbSearch(BIC score, *data*)
        Delete synchronous links, calculate the Markov blanket for the node Reward
            and create/update CPTs
        inference[i] = VariableElimination(model[i])
    end for
end if
```

an option and another one is randomly selected (lines 16-20). If the selected action has a high probability to obtain a positive reward, the action is executed (lines 22-24). We define a variant of the previously described method where the decision to consult the CBN is performed before the behavior of the standard DQN.

5 Experimental results

For the implementation of DQN for visual navigation in the drone, we followed the implementation of Anas et al. [1] and adapted it to our task.

State space: The agent's state spaces consist of: (i) An image with dimensions 84x84x3 pixels used to evaluate the performance of the baseline algorithm without causal models. (ii) An image with dimensions 84x84x3 pixels + 9 values for the learning and use of the CBN.

Action space: the agent can take eight discrete actions namely: *forward* and *backward* moves (move the drone in the y axis), *turn left* and *turn right* (change the orientation of the drone in the x axis), *right* and *left* (move the drone in the x axis), and *ascend* and *descend* (move the drone in the z axis).

Target: a goal image.

Reward: calculated from the following formula.

$$\text{reward} = \begin{cases} 100 - \text{Dist. to goal} - |\text{angle to goal}| & \text{if the target is recognized} \\ 1000 & \text{if the goal is reached} \\ -1000 & \text{if the drone crashes} \\ -9 & \text{otherwise} \end{cases}$$

Termination conditions: 1) The agent reaches the goal within 0.2 meters. 2) The agent collides with an obstacle. 3) The number of steps exceeds 200.

Algorithm 3 Use of the Causal Bayesian Network for DQN

```
At each step initialize Evidence of the state $t$
for i = 0 to numActions do
    filter evidence of the Markov blanket of the node reward
    result = inference[i] for variable reward with filtered evidence
    ProbActNeg[i]=Probability of the action to obtain a negative reward
    ProbAct[i]=Probability of the action to obtain a positive reward
end for
initialize ProbCausalModel and ThresholdProb
rand = random number
if rand < explorationRate then
    action $a_t$ = random action
else
    action $a_t$ = MaxIndex(qValues)
    rand = random number
    if rand < ProbCausalModel then
        if ProbActNeg[action $a_t$] > ThresholdProb then
            ActionNegative = action
            while action $a_t$ == ActionNegative do
                action $a_t$ = random action
            end while
        else
            if ProbAct[action $a_t$] < ThresholdProb then
                action $a_t$ = MaxIndex(ProbAct)
            end if
        end if
    end if
end if
```

Network Architecture: The Prediction and Target Networks share the same structure.

- Input Layer: 84x84x3 RGB image.
- Convolutional Layers (ReLU activated):
    - Layer 1: 32 filters, 8x8 kernel, stride 4.
    - Layer 2: 64 filters, 4x4 kernel, stride 2.
    - Layer 3: 64 filters, 3x3 kernel, stride 1.
- Flatten Layer: Converts output to 1D.
- Dense Layer: 512 units for processing features.
- Output Layer: 8 nodes for Q-values of actions.
- Loss Function: Mean Squared Error (MSE).
- Optimizer: RMSprop, learning rate 0.00025, discount 0.9, epsilon 0.00000001.

To represent the information needed for the construction of the CBN we used 9 values: (i) Distance of five defined sections of the image (center, top left, top right, bottom left, and bottom right). In each section we select a number specific of random points (at least 20 points), taking the smallest distance as the overall distance of the drone to the objects in this section. (ii) A boolean value if the goal is in the field of view. (iii) Distance and angle to the goal, and (iv) Altitude of the drone. To obtain these values we need three sensors: an RGB camera to process the image to detect the goal, a Depth Camera to obtain the distances, and a barometer or similar to obtain the altitude of the drone.

The goal is a sign of "Heliport". Usually, the goal is defined with a set of coordinates, but we try not to depend on the agent seeing the goal all the time, and, in real cases, access to the real coordinates requires the help of GPS or other sensors that are more difficult to access. To enable the drone to recognize the goal, we trained a custom YOLO v8 model [23] for 50 epochs and obtained a precision of 0.943 and a recall of 0.902 for goal detection, using a dataset of 1070 images from simulations and real-world data. The YOLO model returns a list of detected objects, each with a bounding box coordinates and a score, representing the confidence in the detected object. We set a threshold of 0.7 for the confidence score. Once the model detects the object and the bounding box coordinates are obtained, the distance is calculated in the same way as the sections. We also establish an angle between the center of the image and the center of the goal.

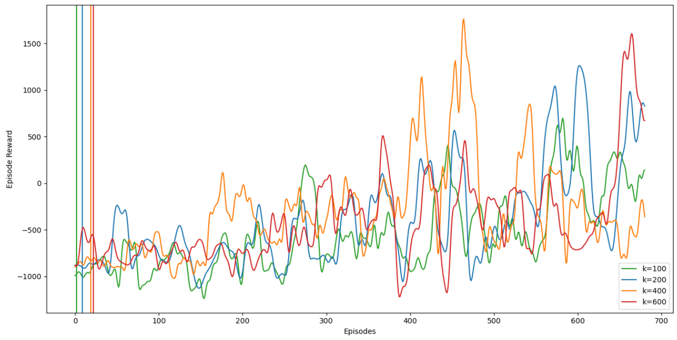
For experimental purposes we construct a CBN of the task to use in the reinforcement learning algorithm, taking into account the 9 values obtained from the RGB and depth image. To simplify and reduce the observation space we discretized all the values, taking two values for the distance: close and far; a boolean value for the goal in sight; three values for the angle: center, far left and far right, and three values for the altitude: good, close and far ground.

5.1 Results

We begin our experiments by using the state representation for learning and using the CBN to determine the optimal value for k. The parameter k dictates the point at which the CBN learning algorithm starts its learning process. We evaluated its effectiveness across various k values: 100, 200, 400, and 600. Their performance is illustrated in Figure 1, the vertical axis represents the reward obtained in each episode after applying a moving average of size 20 to smooth out short-term fluctuations and highlight longer-term trends, rewards range from below -1000 up to approximately 1500. The horizontal axis tracks the number of episodes, ranging from 0 to 700. Before the first k steps the algorithm behaves as DQN, after k steps the algorithm can consult the current CBN model.

Lower k values seem to have more stability compared to higher k values, which exhibits more dramatic ups and downs. This indicates that lower k values provide a more consistent learning experience or adaptation, whereas higher k values, while capable of achieving higher peaks, may introduce volatility. For this reason, we select the value of 200 for k to compare with the other algorithms.

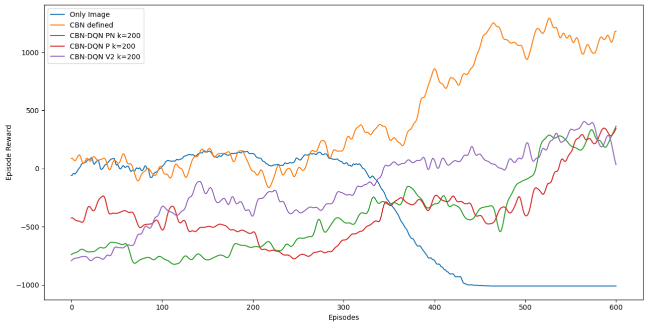
To evaluate the performance of our algorithm, we compared the DQN algorithm as baseline (only image) with the proposed algorithm that adds the discrete state representation, using: (i) the defined Causal Bayesian Network (ii) learning the CBN (CBN-DQN P) (iii) a variant of the algorithm considering the probability of receiving a positive and negative reward from the CBN to decide



,

Fig. 1. Episode reward obtained after 700 episodes in the learning and use of the CBN in the DQN algorithm with k equals to 100, 200, 400, and 600. The vertical lines indicate from which episode the CBN is learned and subsequently used and updated. We utilize a moving average of size 20.

which actions to take versus considering only the probability of positive rewards for the first version (CBN-DQN PN) and (iv) learning the CBN in the second version of the algorithm (CBN-DQN V2), all with k=200



,

Fig. 2. Episode reward obtained after 700 episodes with base DQN, DQN with the CBN defined, CBN-DQN PN, CBN-DQN P and CBN-DQN V2 with k=200, with a moving average of 50.

Their performance is illustrated in Figure 2, the vertical axis represents the reward obtained in each episode after applying a moving average of size 50. The best behavior is when the Causal Bayes Network is manually defined in advance because it is used from the beginning of the learning process and the

CBN is more accurate. Using only the image as state representation has not adequate information to reach the goal and the network learns to "do nothing". In comparison, the CBN-DQN V2 is the closest to the behavior of the algorithm with a predefined causal model, compared to use only the action with a higher probability of carrying a positive reward (CBN-DQN P) and using both (CBN-DQN PN). This could be because the first version of the algorithm benefits more from performing random actions at the first episodes depending of the exploration rate, so it might require more training episodes for better behavior compared to the version two of the algorithm where the probability of consulting the CBN is dominated by the exploration rate.

We also analyze the behavior of the CBN learning method. We can observe the evolution of the CBN for CBN-DQN V2, for the action forward in Figure 3. It begins with 4 connections between time t and time $t+1$ but at the end, it has 7 connections. It is interesting to note the dependence between the reward and the angle for the same $t+1$, which depends of seeing the goal and its angle goal in the previous time. This behavior of the learned CBNs helps us to note dependencies that we did not take into account in the manual definition of the network. Also, the lack of direct connections to the reward node can indicate that there was not sufficient information involving seeing the goal and consequently the angle and distance to the goal.

From the experiments we can conclude the following:

- A Causal Bayesian Network (given or learned) can be used to improve the performance of an RL agent in robotic tasks with a partially continuous space state.
- A Causal Bayesian Network can be learned from data obtained during the reinforcement learning process but more episodes are needed to obtain a good model.

6 Conclusions

In this work, we developed two algorithms: one that learns a Causal Bayesian Network of the task and environment during the training of the DQN algorithm

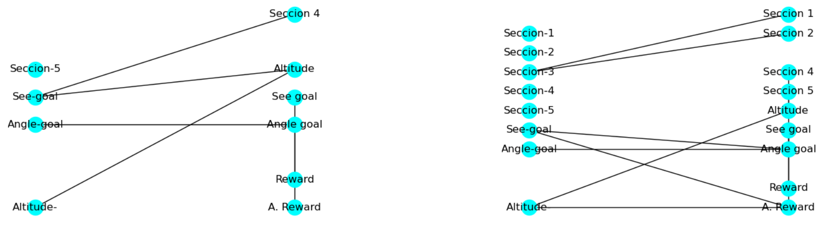


Fig. 3. Comparison of the first and last Causal Bayesian Network for action *forward*.

and another one (with two variants) that uses a CBN to guide the selection of the actions in the DQN algorithm. We tested the algorithms in the task of autonomous navigation of a drone in a simulated environment. The results show that the use of causal knowledge can accelerate learning a policy; and that it is possible to learn partial CBNs, while learning a policy, but more episodes are required to reach a stable and accurate CBN.

As future work, we need to train our algorithms for more episodes to prove the stability of the algorithm and confirm if the learned CBN reaches the accuracy of the model with the network previously defined. We also need to perform more tests on different environments to assess the generality of the proposed approach. We would also like to include an exploration strategy, in case the objective is out of sight. Finally, we believe that the learned CBNs can be transferred to other, although similar, domains where the causal relationships are still valid.

Acknowledgments The authors would like to acknowledge the funding support for this work from CDP-BOOT and CONAHCYT.

References

1. Anas, H., Ong, W.H., Malik, O.A.: Comparison of deep q-learning, q-learning and sarsa reinforced learning for robot local navigation. In: Kim, J., Englot, B., Park, H.W., Choi, H.L., Myung, H., Kim, J., Kim, J.H. (eds.) Robot Intelligence Technology and Applications 6. pp. 443–454. Springer, Cham (2022)
2. Çetin, E., Barrado, C., Muñoz, G., Macias, M., Pastor, E.: Drone navigation and avoidance of obstacles through deep reinforcement learning. In: 2019 IEEE/AIAA 38th Digital Avionics Systems Conf. (DASC). pp. 1–7 (2019)
3. Çetin, E., Barrado, C., Pastor, E.: Explainability of deep reinforcement learning method with drones. In: 2023 IEEE/AIAA 42nd Digital Avionics Systems Conf. (DASC). pp. 1–9 (2023)
4. Darwish, A.A., Nakhmani, A.: Drone navigation and target interception using deep reinforcement learning: A cascade reward approach. IEEE Journal of Indoor and Seamless Positioning and Navigation **1**, 130–140 (2023)
5. Deng, Z., Jiang, J., Long, G., Zhang, C.: Causal reinforcement learning: A survey (2023), https://arxiv.org/abs/2307.01452
6. Diehl, M., Ramirez-Amaro, K.: Why did i fail? a causal-based method to find explanations for robot failures. IEEE Robotics and Automation Letters **7**(4), 8925–8932 (2022)
7. Feliciano-Avelino, I., Méndez-Molina, A., Morales, E.F., Sucar, L.E.: Causal based action selection policy for reinforcement learning. In: Batyrshin, I., Gelbukh, A., Sidorov, G. (eds.) Advances in Computational Intelligence. pp. 213–227. Springer, Cham (2021)
8. García, M., Caballero, R., González, F., Viguria, A., Ollero, A.: Autonomous drone with ability to track and capture an aerial target. In: 2020 Int. Conf. on Unmanned Aircraft Systems (ICUAS). pp. 32–40 (2020)
9. Gatteschi, V., Lamberti, F., Paravati, G., Sanna, A., Demartini, C., Lisanti, A., Venezia, G.: New frontiers of delivery services using drones: A prototype system exploiting a quadcopter for autonomous drug shipments. In: 2015 IEEE 39th Annual Computer Software and Applications Conf. vol. 2, pp. 920–927 (2015)

10. Gonzalez-Soto, M., Sucar, L.E., Escalante, H.J.: Playing against nature: causal discovery for decision making under uncertainty (2018), https://arxiv.org/abs/1807.01268
11. Hanover, D., Loquercio, A., Bauersfeld, L., Romero, A., Penicka, R., Song, Y., Cioffi, G., Kaufmann, E., Scaramuzza, D.: Autonomous drone racing: A survey. IEEE Transactions on Robotics **40**, 3044–3067 (2024)
12. Hasan, K.M., Abdullah-Al-Nahid, Alim, M.A., Maniruzzaman, M., Atiqur Rahaman, G.M., Ahsan, M.S., Newaz, S.S.: Design and development of an aircraft type multi-functional autonomous drone. In: 2020 IEEE Region 10 Symposium (TENSYMP). pp. 734–737 (2020)
13. Heyn, H.M., Knauss, E.: Structural causal models as boundary objects in ai system development. In: 2022 IEEE/ACM 1st Int. Conf. on AI Engineering – Software Engineering for AI (CAIN). pp. 43–45 (2022)
14. Kersandt, K., Muñoz, G., Barrado, C.: Self-training by reinforcement learning for full-autonomous drones of the future. In: 2018 IEEE/AIAA 37th Digital Avionics Systems Conf. (DASC). pp. 1–10 (2018)
15. Klecka, W.: Discriminant Analysis. SAGE Publications, Inc. (1980). https://doi.org/10.4135/9781412983938
16. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. The MIT Press (2009)
17. Lewis-Beck, M.: Applied Regression. SAGE Publications, Inc. (1980)
18. Mario Silvagni, Andrea Tonoli, E.Z., Chiaberge, M.: Multipurpose uav for search and rescue operations in mountain avalanche events. Geomatics, Natural Hazards and Risk **8**(1), 18–33 (2017)
19. Méndez-Molina, A., F. Morales, E., Sucar, L.E.: Causal discovery and reinforcement learning: A synergistic integration. In: Salmerón, A., Rumí, R. (eds.) Proceedings of The 11th Int. Conf. on Probabilistic Graphical Models. Proceedings of Machine Learning Research, vol. 186, pp. 421–432. PMLR (05–07 Oct 2022)
20. Méndez-Molina, A., Morales, E.F., Sucar, L.E.: Carl: A synergistic framework for causal reinforcement learning. IEEE Access **11**, 126462–126481 (2023)
21. Pitis, S., Creager, E., Mandlekar, A., Garg, A.: Mocoda: Model-based counterfactual data augmentation (2022), https://arxiv.org/abs/2210.11287
22. Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T., Moscholios, I.: A compilation of uav applications for precision agriculture. Computer Networks **172**, 107148 (2020)
23. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 779–788 (2016)
24. Shi, W., Huang, G., Song, S., Wu, C.: Temporal-spatial causal interpretations for vision-based reinforcement learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(12), 10222–10235 (2022)
25. Shin, S.Y., Kang, Y.W., Kim, Y.G.: Automatic drone navigation in realistic 3d landscapes using deep reinforcement learning. In: 2019 6th Int. Conf. on Control, Decision and Information Technologies (CoDIT). pp. 1072–1077 (2019)
26. Sucar, L.E.: Probabilistic graphical models. Advances in computer vision and pattern recognition, Springer Nature, Cham, Switzerland, 2 edn. (Dec 2020)
27. Zeng, Y., Cai, R., Sun, F., Huang, L., Hao, Z.: A survey on causal reinforcement learning (2023), https://arxiv.org/abs/2302.05209
28. Zhu, W., Yu, C., Zhang, Q.: Causal deep reinforcement learning using observational data. In: Proceedings of the Thirty-Second Int. Joint Conf. on AI. p. 4711–4719. IJCAI-2023, Int. Joint Conf. on AI Organization (Aug 2023)

Combining Literature with Causal Discovery in Environmental-Conflict

Maarten Vonk[1][2]

[1] Natural Computing, Leiden Institute of Advanced Computer Science, Einsteinweg 55, Leiden 2333CC
[2] The Hague Centre for Strategic Studies, Organization, Lange Voorhout 1, The Hague, 2514EA, Netherlands, https://hcss.nl/

Abstract. As causal discovery algorithms have not matured up to the point where they can consistently identify complete causal structures from observational data, human expertise remains crucial in distinguishing cause from effect. In scientific practice, this human expertise is manifested in literature studies. This paper presents a causal discovery application of the environmental-conflict, conflict dynamic in Iraq where an extensive literature review is paired up with a causal discovery algorithm to uncover a causal mechanism. Specifically, the literature review identifies relevant variables for the causal dynamic and offers a hypothesized causal structure, which is then used as a benchmark to evaluate the causal structures derived from the causal discovery algorithms. Our proof of concept shows that such an approach can add specificity and refine our understanding of a causal mechanism.

Keywords: Causal Discovery · Human Expertise · Conflict Studies

1 Introduction

Environmental-conflict is a form of tension or violence triggered by competition over natural resources or environmental pressures, such as scarcity or degradation. Although environmental-conflict literature has advanced in examining the causal relationships between climatological factors and the onset of armed conflict, these studies often focus on specific connections rather than a comprehensive mechanism of naturally induced conflict [11, 24, 15]. The unsuitability of randomized controlled trials for the study of armed conflict further hampers the extraction of causal relations, leaving a gap in the literature for methodologies that can infer causality from non-experimental observations.

One way to address this gap is causal discovery. While causal discovery algorithms have gained traction over the years, research has shown that causal discovery algorithms can be unstable [14] or that only limited parts of the causal graph can be discovered from pure observations [19]. To this purpose, domain knowledge can be used to refine the performance of causal discovery algorithms and has been incorporated via tiered background knowledge [2] user interactions [16] or the penalization of the search process [9].

This extended abstract extracts a causal mechanism from environmental-conflict data in Iraq by complementing the power of a causal discovery method with hypothesized causal linkages from a literature review. First, a hypothesized causal mechanism is derived from a literature review, which informs the selection of variables for causal discovery. A causal discovery algorithm is then applied to the environmental-conflict data, and the resulting causal structure is benchmarked against the established linkages from the literature.

While the hypothesized causal linkages from the literature are discussed in Section 2, the data, causal discovery algorithm, and results are described in Section 3. Finally, the conclusions and limitations are discussed in Section 4.

2 Literature Review

This section formulates hypotheses on causal linkages contributing to the emergence of environmental conflict in Iraq, grounded in a comprehensive literature review that systematically examines regional environmental stressors and conflict dynamics identified across academic sources.

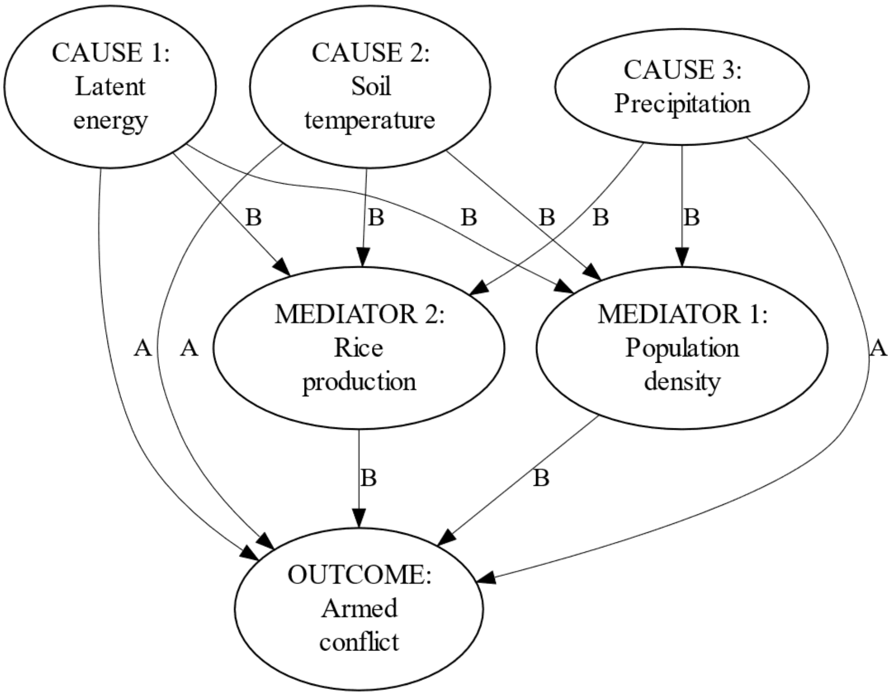

CAUSE 1: Latent energy
CAUSE 2: Soil temperature
CAUSE 3: Precipitation
MEDIATOR 2: Rice production
MEDIATOR 1: Population density
OUTCOME: Armed conflict
A
B

Fig. 1. The hypothesized causal structure: it distinguishes between direct and indirect linkages (i.e., paths A and B, respectively).

Long-term weather patterns have been directly linked to armed conflict [12, 11, 25, 8, 4]. Environmental changes can disrupt livelihoods, prompting fragmented communities to take action in an effort to mitigate the adverse effects. Such changes include variations in precipitation, evaporation, temperature, and the differential absorption or release of accumulated heat by physical environments, all of which may contribute to the emergence of conflict [11, 24].

On the other hand environmental variables have also been argued to cause armed conflict indirectly [3, 24, 15]. One possibility is that the effect of environmental changes on armed conflict are mediated by the scarcity of vital resources [24, 10]. This materializes into the scarcity of crops due to disrupted weather patterns, of which rice production has been shown to mediate causal effects of temperature on the emergence of actual violence [5].

Many studies also tie the emergence or existence of armed conflict in relation to environmental changes to population sizes [27, 20, 22]. Specifically, resource scarcity affects denser populations more significantly than less dense ones [3, 1], as denser populations might be less able to mitigate tensions effectively.

With all of the above hypotheses, it is possible to compose the entire hypothetical causal structure of linkages between environment and conflict, as shown in Figure 1. This hypothesized causal structure is also in line with research of other scholars [24].

3 Experimentation and Results

The units of analysis are all the 294 Iraqi municipalities. The literature study was leading in the choice of extracted variables. First, the number of civilian fatalities, conflict events, and conflict fatalities were sourced from Armed Conflict Location and Event Data Program [21] (ACLED). In addition, the climatological-related variables such as precipitation, soil temperature, and latent energy were retrieved from IMERG quality index [13], ERA5-Land dataset [18] and NASA MODIS dataset [23] respectively. Finally, we extracted rice production data from MapSPAM [26] and population density data from the International Earth Science Information Network at Columbia University [7]). Since the time horizon was from January 1, 2020, to January 1, 2022. we aggregated the values of explanatory variables, so much so each aggregation emphasized extreme values of the observed explanatory variables.

The causal discovery algorithm that is being applied to the data is GES [6]. Although the study would benefit from a more exhaustive use of causal discovery algorithms, GES has been chosen as it has been deemed suitable in the case of simulation studies with small sample sizes [17].

As the literature review pointed out, there are multiple subselections of variables that are hypothesized to play a role in the environmental-conflict dynamic. While no causal linkages or partial causal structure is imposed in the causal discovery phase, a total of 136 extracted graphs are benchmarked against the hypothesized graph of Figure 1. The graph that shares the most directed edges with the hypothesized graph is shown in Figure 2.

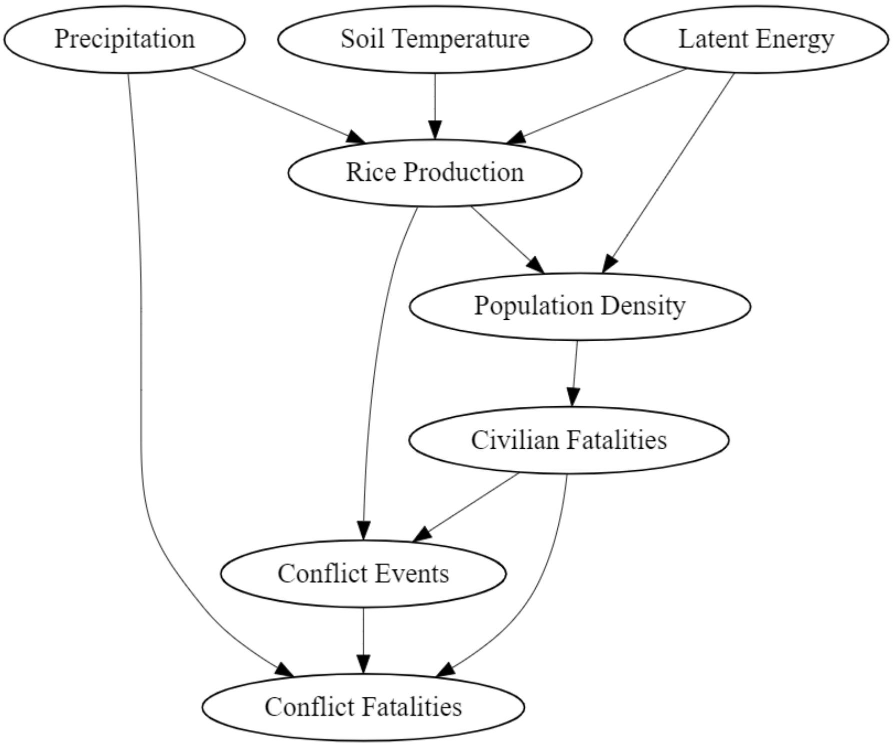


Fig. 2. GES-retrieved causal structure that is most aligned with the literature review

The literature review only hypothesized the possible existence of direct or indirect effects, which is why the causal discovery process reveals more specifically which directed arrows exist between the variables. The empirically retrieved causal structure reveals that precipitation is the only environmental variable with a direct effect on conflict, while other effects are mediated by rice production or population density.

4 Conclusion and Limitations

In this work, we have provided a proof of concept of how the strengths of both thorough literature review as well as causal discovery methods can be merged to achieve a causal mechanism in the environmental-conflict context of Iraq.

Since the results of causal discovery are filtered based on their alignment with existing findings, the method can only corroborate established conclusions and is unable to identify causal structures that are not already supported by the literature. Nonetheless, as the literature consensus can be ambiguous, pairing them up with causal discovery algorithms can contribute significantly to refining our understanding of complex causal structures.

Finally, a limitation of our current approach to environmental-conflict modeling is that the causal discovery process assumes that the data is independent and identically distributed (i.i.d.). However, it is plausible that climatological variables of one municipality may influence the climate-conflict dynamics of another municipality, leading to violations of the i.d.d. assumption. This spatial confounding calls for causal discovery methods that can account for such violations [28].

References

1. Acemoglu, D., Fergusson, L., Johnson, S.: Population and conflict. The Review of Economic Studies **87**(4), 1565–1604 (2020)
2. Andrews, B., Spirtes, P., Cooper, G.F.: On the completeness of causal discovery in the presence of latent confounding with tiered background knowledge. In: Chiappa, S., Calandra, R. (eds.) Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 108, pp. 4002–4011. PMLR (26–28 Aug 2020), https://proceedings.mlr.press/v108/andrews20a.html
3. Ash, K., Obradovich, N.: Climatic stress, internal migration, and syrian civil war onset. Journal of Conflict Resolution **64**(1), 3–31 (2020)
4. Bai, Y., Kung, J.K.s.: Climate shocks and sino-nomadic conflict. Review of Economics and Statistics **93**(3), 970–981 (2011)
5. Caruso, R., Petrarca, I., Ricciuti, R.: Climate change, rice crops, and violence: Evidence from indonesia. Journal of Peace Research **53**(1), 66–83 (2016)
6. Chickering, D.M.: Optimal structure identification with greedy search. Journal of machine learning research **3**(Nov), 507–554 (2002)
7. CIESIN, S.: Gridded population of the world, version 4 (gpwv4): population density. Center for International Earth Science Information Network-CIESIN-Columbia University. NASA Socioeconomic Data and Applications Center (SEDAC). https://doi. org/10.7927/H4DZ068D (2015)
8. Ghimire, R., Ferreira, S.: Floods and armed conflict. Environment and Development Economics **21**(1), 23–52 (2016)
9. Hasan, U., Gani, M.O.: Kcrl: A prior knowledge based causal discovery framework with reinforcement learning. In: Lipton, Z., Ranganath, R., Sendak, M., Sjoding, M., Yeung, S. (eds.) Proceedings of the 7th Machine Learning for Healthcare Conference. Proceedings of Machine Learning Research, vol. 182, pp. 691–714. PMLR (05–06 Aug 2022), https://proceedings.mlr.press/v182/hasan22a.html
10. Homer-Dixon, T.F.: Environment, scarcity, and violence. Princeton University Press (2010)
11. Hsiang, S.M., Burke, M., Miguel, E.: Quantifying the influence of climate on human conflict. Science **341**(6151), 1235367 (2013)
12. Hsiang, S.M., Meng, K.C., Cane, M.A.: Civil conflicts are associated with the global climate. Nature **476**(7361), 438–441 (2011)
13. Huffman, G.: Imerg v06 quality index. NASA, Available for download from: https://gpm. nasa. gov/sites/default/files/2020-02/IMERGV06_QI_0. pdf (2019)
14. Kitson, N.K., Constantinou, A.C.: The impact of variable ordering on bayesian network structure learning. Data Mining and Knowledge Discovery pp. 1–25 (2024)

15. Koubi, V.: Climate change and conflict. Annual Review of Political Science **22**(1), 343–360 (2019)
16. Mäkelä, J., Melkas, L., Mammarella, I., Nieminen, T., Chandramouli, S., Savvides, R., Puolamäki, K.: Technical note: Incorporating expert domain knowledge into causal structure discovery workflows. Biogeosciences **19**(8), 2095–2099 (2022). https://doi.org/10.5194/bg-19-2095-2022, https://bg.copernicus.org/articles/19/2095/2022/
17. Malinsky, D., Danks, D.: Causal discovery algorithms: A practical guide. Philosophy Compass **13**(1), e12470 (2018)
18. Muñoz-Sabater, J., Dutra, E., Agustí-Panareda, A., Albergel, C., Arduini, G., Balsamo, G., Boussetta, S., Choulga, M., Harrigan, S., Hersbach, H., et al.: Era5-land: A state-of-the-art global reanalysis dataset for land applications. Earth system science data **13**(9), 4349–4383 (2021)
19. Peters, J., Janzing, D., Schölkopf, B.: Elements of causal inference: foundations and learning algorithms. The MIT Press (2017)
20. Raleigh, C., Hegre, H.: Population size, concentration, and civil war. a geographically disaggregated analysis. Political geography **28**(4), 224–238 (2009)
21. Raleigh, C., Linke, R., Hegre, H., Karlsen, J.: Introducing acled: An armed conflict location and event dataset. Journal of peace research **47**(5), 651–660 (2010)
22. Reuveny, R.: Climate change-induced migration and violent conflict. Political geography **26**(6), 656–673 (2007)
23. Running, S., Mu, Q.: Mod16a2 modis/terra evapotranspiration 8-day l4 global 500m sin grid. NASA LP DAAC (2015)
24. Sakaguchi, K., Varughese, A., Auld, G.: Climate wars? a systematic review of empirical analyses on the links between climate change and violent conflict. International Studies Review **19**(4), 622–645 (2017)
25. Slettebak, R.T.: Don't blame the weather! climate-related natural disasters and civil conflict. Journal of Peace Research **49**(1), 163–176 (2012)
26. Statistics, G.S.D.C.P.: Data for 2010 version 2.0. IFPRI: Washington, DC, USA (2019)
27. Tir, J., Diehl, P.F.: Demographic pressure and interstate conflict: linking population growth and density to militarized disputes and wars, 1930-89. Journal of Peace Research **35**(3), 319–339 (1998)
28. Urdangarin, A., Goicoa, T., Ugarte, M.D.: Evaluating recent methods to overcome spatial confounding. Revista Matemática Complutense **36**(2), 333–360 (2023)

Discovering the causal structure of students' entry into higher education in Chile from data

Paulo Quinsacara$^{1[0000-1111-2222-3333]}$, Billy Peralta$^{1[0000-0002-5457-2157]}$, and Pablo Schwarzenberg$^{1[0000-0002-7753-353X]}$

Facultad de Ingeniería, Universidad Andres Bello, Santiago, Chile
p.quinsacarajofr@uandresbello.edu,{b.peralta,pablo.schwarzenberg}@unab.cl

Abstract. Higher education is a critical driver of social mobility and economic development globally. Access to universities and the factors influencing academic performance are essential for shaping the future workforce and ensuring equitable opportunities. In Chile, the higher education system plays a pivotal role in providing pathways for students from diverse socioeconomic backgrounds. This study focuses on the discovery of causal relationships influencing student admission to higher education in Chile, using data from the Higher Education Access Test of 2021. The research aims to identify the factors affecting academic performance and access to university by applying causal inference algorithms, specifically PC, GES, and LINGAM. These algorithms help uncover directed acyclic graphs (DAGs) from observational data, revealing the underlying causal structure among variables like socioeconomic background and academic scores. The results highlight the potential causal relationships between these factors, providing critical insights for educational policy-making. Key findings demonstrate the value of such causal models in understanding the dynamics that affect educational outcomes. Future work should explore the application of additional sensitivity analyses and broader datasets to further validate and refine these causal models.

Keywords: Higher education · Causal learning · Causal structure.

1 Introduction

Higher education is a cornerstone of societal progress, fostering innovation, critical thinking, and economic development. Around the world, universities serve as hubs for research, knowledge creation, and the cultivation of skills necessary for addressing complex global challenges. Access to higher education is linked to improved employment prospects, social mobility, and the reduction of income inequality. In Chile, higher education is similarly vital, providing pathways for individuals to advance professionally and economically. The country's education system has undergone significant reforms to promote equity and broaden access, yet disparities remain, particularly for students from lower socioeconomic backgrounds. Addressing these inequalities is crucial for ensuring that all Chilean students have the opportunity to succeed in a rapidly evolving global economy.

Within the context of the Higher Education Access Test (PAES), which is the instrument that allows access to higher education, that is, the universities attached to the access system, it is highly desirable to understand the key factors that can explain access to higher education. The Higher Education Access Test (PAES) of Chile, which replaced the old University Selection Test (PSU), in its most recent versions offers academic and socioeconomic data that can potentially be used to analyze the influence of multiple factors on academic performance and access to higher education. In the context of computer sciences, establishing the identification that is materialized in causal relationships is a fundamental challenge in artificial intelligence, especially if the focus is on the educational area, where decisions based on correct analysis can have a significant impact on policies and on the development of students as well as on their academic and professional performance.

The main objective of this study is to identify the key causal relationships between various educational variables specified in the PAES 2021 dataset. This will contribute to a better understanding of the factors influencing university admissions and, subsequently, academic performance. The available dataset includes a range of variables, such as scores across different areas of knowledge, socioeconomic background, and demographic characteristics of the students. The findings are expected to offer new insights into the interactions between these educational variables and provide a solid foundation for making informed decisions in educational policy or guidelines that support continuous improvement.

The literature on causal inference in educational data has had a sustained growth in recent years, which is why there are studies that have used machine learning and statistics methods to identify patterns and relationships in academic performance data. However, the specific application of causal structure discovery algorithms in the context of education is relatively new. These methods allow not only identifying correlations, but also establishing causal directions, which makes them particularly useful to understand the complex dynamics in education [9].

2 Literature review

In recent years, causal algorithms have been used in various educational studies to identify key factors that influence academic performance, educational equity, and the efficiency of educational policies. Below we show some relevant works found in our bibliographic review

Peralta et al. [11] propose an approach using Bayesian networks to identify, analyze, and weigh causal relationships influencing student outcomes. By leveraging real-world data from the Universidad Católica de Temuco, their model offers a more comprehensive understanding of the underlying factors contributing to dropout and graduation. The findings reveal key variables and relationships that align with expert opinions, suggesting the superior ability of our model to represent the causal dynamics of these events. Ultimately, this research paves the way for the development of more effective retention policies and timely degree completion strategies.

Chen et al. [2] developed a model, CIRF-MLP, which combines causal inference methods with machine learning to predict students' post-graduation outcomes based on their academic data. The study also introduced a visual analytics tool, CausalCareerVis, that helps analyze the causal relationships between academic performance and career choices. The model showed high accuracy and interpretability, providing insights that can assist students with career planning and guide university administrators in curriculum development.

Rodriguez et al. [12] investigate the relationship between the educational use of information and communication technology (ICT), students' technological skills, and academic performance among university students in Barinas, Venezuela. A sample of 410 students from various universities was surveyed to assess their ICT usage, proficiency, and academic results. The analysis employed structural equation modeling to explore causal links. The findings revealed that while ICT use significantly impacts students' technological skills, its direct effect on academic performance was minimal.

Misiunas et al. [8] employ Bayesian Networks (BNs) to uncover causal relationships within a dataset comprising demographic, academic, and financial characteristics of university students. Factor analysis on mixed data is utilized to identify groups of dependent variables and validate BN structure. The BN, learned through bootstrapping, reveals two primary structures: one focused on academic performance and the other on financial, housing, and demographic factors. Prediction accuracy using pre-college evidence is low (55%), while on-going college evidence significantly enhances accuracy (75%).

Silva et al. [15] investigate the causal effects of educational interventions on student performance. By combining educational data mining techniques with traditional theory-driven models, they address the limitations of previous EDM studies that lack causal reasoning. Using large-scale assessment data from Brazil, the authors identify key unobserved confounders through causal graph analysis and incorporate them into a two-way logistic regression fixed effects model. The findings highlight the significance of socioeconomic factors, faculty education policies, and the role of Brazilian states in shaping student performance.

On the other hand Gerard et al. [5] focuses on identifying causal relationships among six criteria that define the synthetic indicator of the quality of education systems (ISQ) in OECD countries. Utilizing directed acyclic graphs (DAGs) and causal Bayesian networks, the research aims to measure and interpret the causal relations between these criteria and the ISQ 2018 score. The study emphasizes the importance of factors such as equity, parental engagement, and efficiency in determining educational effectiveness.

In the reviewed works we did not find direct applications in data from university entrance exams in Chile regarding the discovery of causal structures. Given the availability of the data, we propose for the first time to apply various causal discovery algorithms to find such structures in the local context.

3 Causal models

Causal models are a fundamental tool in understanding the relationships between variables, allowing researchers to infer cause-and-effect connections rather than mere correlations. These models help identify the underlying mechanisms that drive changes in one variable due to another, often supporting decision-making and policy development.

This model considers a graph $G = (V, E)$ consists of a set of nodes V and edges E. In artificial intelligence learning algorithms, nodes represent variables in the input dataset, so the terms "node" and "variable" are often used interchangeably [7]. Two nodes X and Y are adjacent if they are connected by an edge $(X - Y)$, and the nodes adjacent to a node X in graph G are denoted as $\mathrm{adj}(X, G)$. An edge is directed if it has an orientation towards one of the nodes. If the edge is directed from X to Y (XßY or Y ¸X), we say that Y is a descendant of X, or that X is the parent of Y. A Directed Acyclic Graph (DAG) is defined as one where all edges are directed, and no directed cycles exist. A DAG has a unique topological order if there is a directed path that includes all nodes.

To determine if the nodes X and Y are d-separated or d-connected, three rules are considered [10]:

1. Unconditional Separation: Nodes X and Y are d-connected if there is a free path between them, meaning a path that can be traversed by following the direction of the links. Otherwise, the nodes are d-separated.

2. Blocking by Conditioning: When variables Z are measured in a system, the conditional dynamics of other variables change. In a DAG, this dynamic is expressed by the dependence (d-connected) of variables when conditioned on Z.

3. Conditioning on Colliders: If a collider is part of the conditioning set Z or has a descendant in Z, it does not block the path traced by the collider.

In the causal interpretation, the DAG G represents a causal structure where directed edges indicate direct causes between nodes. Understanding causal structures, given by the graph, is crucial for identifying the underlying factors that influence student admission to higher education. However, discovering these causal relationships is inherently challenging.

4 Proposed methodology

This work is based on data from the 2021 Higher Education Access Test (PAES), provided by the Chilean Ministry of Education (MINEDUC). The dataset includes detailed information on academic performance in different areas, along with sociodemographic variables. A data cleaning and normalization approach was used to ensure the quality and relevance of subsequent analyses. Missing values were removed, categorical variables were coded, and continuous variables were normalized, ensuring that the assumptions of the causal algorithms were satisfied.

4.1 Variable selection

For the construction of the causal graphs, variables were selected that are considered to have a significant impact on academic performance and access to higher education [1]. These include, but are not limited to, scores in Mathematics, Language and Communication, family economic situation, type of educational establishment (public/private), and geographic location.

4.2 Causal structure discovery algorithms

PC Algorithm The PC algorithm (*Peter-Clark*) was the first to be applied for discovering causal relationships among the selected variables [6]. Their steps are given by:

1. **Initialization**: A completely connected graph is started.
2. **Independence Tests**: Conditional independence tests are conducted using Fisher's Z Test.
3. **Arc Removal**: Arcs that do not meet the established significance threshold were removed.
4. **Arc Orientation**: The remaining arcs are oriented according to the constraints imposed by the independence tests and the structure of the graph.

4.3 GES Algorithm

The GES (*Greedy Equivalence Search*) algorithm was used to explore different possible causal structures. GES is a heuristic search algorithm that employs a forward and backward search approach to identify the most probable causal structure [3]. The implemented steps were:

1. **Forward Search**: Arcs are added to the initial graph, evaluating the *Bayesian Information Criterion* (BIC) at each iteration.
2. **Backward Search**: Subsequently, arcs that do not improve the BIC score were removed.
3. **Model Selection**: The model with the highest BIC score is selected as the definitive causal graph.

4.4 LINGAM Algorithm

The LINGAM (*Linear Non-Gaussian Acyclic Model*) algorithm were designed to address the potential non-normality of relationships between variables. This method is particularly useful in contexts where the relationships between variables cannot be adequately modeled under Gaussian assumptions [14]. The implementation follows these steps:

1. **Initial Modeling**: A structural equation model is constructed assuming linear relationships.
2. **Decomposition**: The variables are decomposed to isolate the non-Gaussian relationships.
3. **Arc Orientation**: Arcs in the final graph are oriented based on the decomposed structure.

Table 1. Variables of dataset

COLUMN	Type
CODIGO-ENS	Nominal
LOCAL-EDUCACIONAL	Nominal
UNIDAD-EDUCATIVA	Nominal
DEPENDENCIA	Nominal
CODIGO-REGION-EGRESO	Nominal
CODIGO-PROVINCIA-EGRESO	Nominal
CODIGO-COMUNA-EGRESO	Nominal
ANYO-DE-EGRESO	Ordinal
PTJE-NEM	Ordinal
PORC-SUP-NOTAS	Ordinal
PTJE-RANKING	Ordinal
PROM-CM-ACTUAL	Ordinal
CODIGO-COMUNA-DOMICILIO	Nominal
SEXO	Nominal
TIENE-TRABAJO-REM	Nominal
HORARIO-TRABAJO	Nominal
CUANTOS-TRABAJAN-GRUPO-FAM	Ordinal
PERSONAS-ESTUDIAN-SUP	Ordinal
ACTIVIDAD-JEFE-FAMILIA	Nominal
INGRESO-PERCAPITA-GRUPO-FA	Ordinal
EDUCACION-MADRE	Nominal
COMPLETO-EDUCACION-MADRE	Nominal
EDUCACION-PADRE	Nominal
COMPLETO-EDUCACION-PADRE	Nominal
ESTUDIO-INSTITUCION-SUPERIOR	Nominal

4.5 Model Evaluation and Validation

The evaluation of the generated causal graphs was conducted using a cross-validation approach. The data was divided into training and test subsets. The training set was used to construct the causal graphs, while the test set was employed to assess their predictive capability and the coherence of the identified causal relationships. Additionally, sensitivity analyses were performed to ensure the model's robustness against variations in the data and algorithm parameters [10], [16].

5 Experiments

The final dataset consists of a total of 25 variables considering 270 412 records. Previously, we delete all the records with incomplete information. The variables of the dataset are shown in Table 1. The causal algorithms were implemented using library DoWhy [13].

5.1 Application of algorithms for structure causal discovery

For the application of the PC algorithm, we consider one random percent of the records. The reason were the computing resources available as this algorithm were extremely slow. Stillmore, this processing required seven days in our equipment. In Figure 1, we can see the outcome of the execution of the PC algorithm. In a future version, we will consider all available data for this algorithm.

For the application of the GES algorithm, all the records were processed, yielding the following result. In Figure 2, we can see the outcome of the GES algorithm execution. For the application of the LINGAM algorithm, all the records were selected, yielding the given in Figure 3.

6 Discussion

The PC algorithm produces graphs that are connected as well as some that are not, and its computation time is greater, while also providing very little visual analysis ease for decision-makers. In the GES algorithm, while all nodes are related and none are excluded, the issue is that it is not easy for users to understand.

When executing the three causal algorithms, LINGAM's main function is to take a DAG and data corresponding to the dataset as input and return an evaluation result. When executing the function ‘result = falsify-graph(g-true, data, plot-histogram=True)‘, it is found, as expected, that the true DAG is not rejected. It is clear what it does: when we provide a specific DAG, it tests for violations of the Local Markov Conditions (LMC) by performing conditional independence (CI) tests. For each node in the graph, it tests "falsify-graph" [4].

Next, we randomly permute the nodes of the given DAG and retest for LMC violations. This can be done for a fixed number of permutations or for the total number of nodes in the given DAG. The probability that a random node permutation (the null) has as few or fewer violations as the given DAG (the test statistic) can then be used as a measure to validate the given DAG (the p-value reported in the upper right corner of the graph) [4].

Given the above, an oracle test is executed for each permutation with the given DAG, meaning that if the given DAG were the true DAG, how many LMC violations would be expected for some permutation? It should be noted

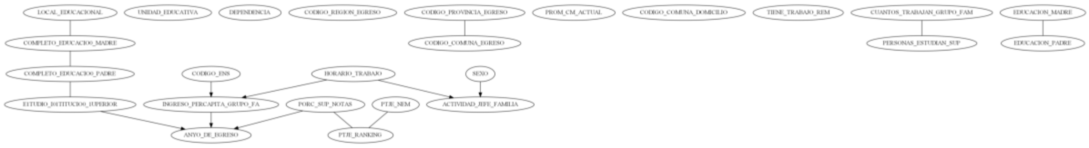


Fig. 1. Result of PC algorithm. Please see details in PDF file.

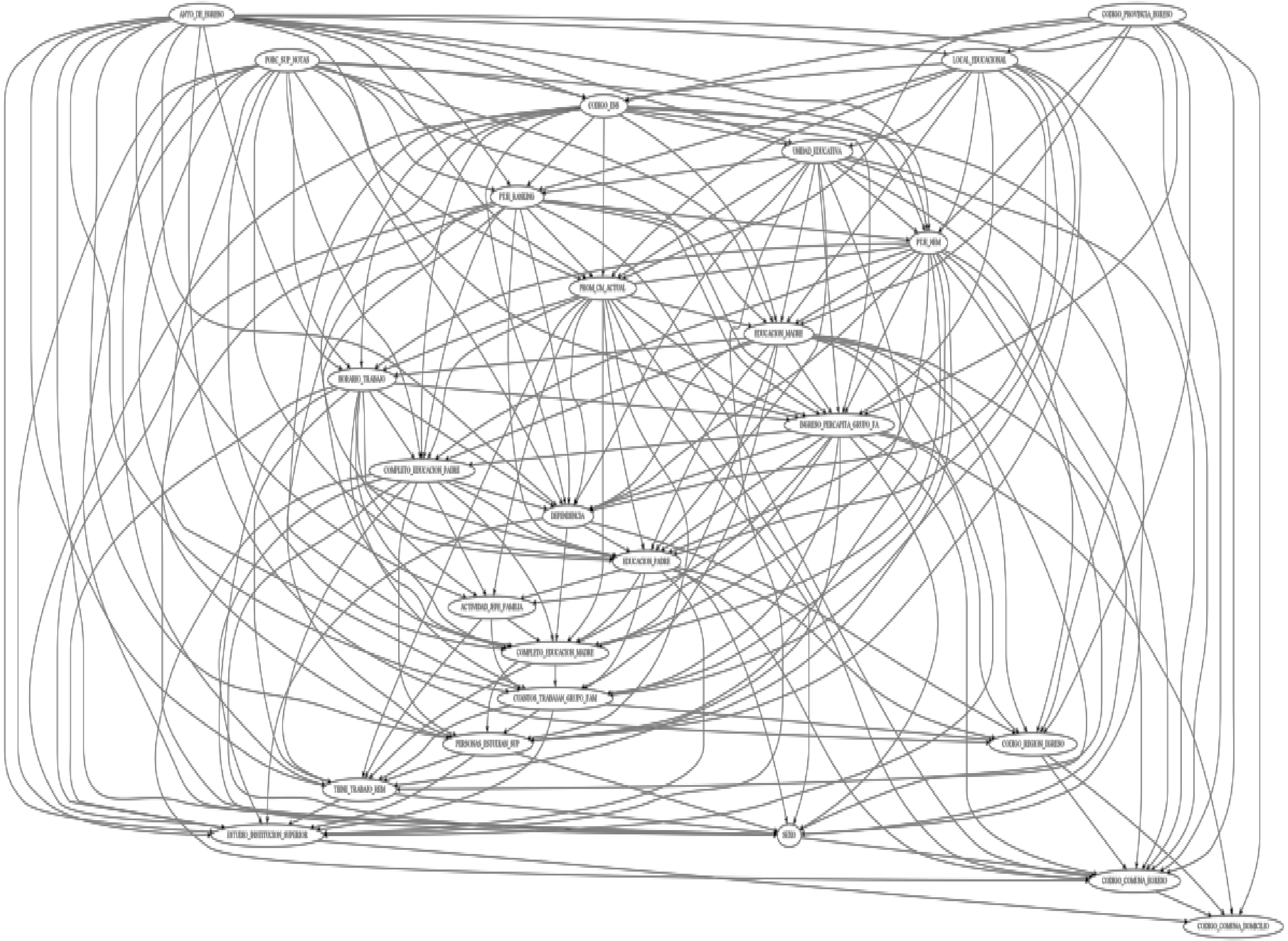


Fig. 2. Causal graph generated by GES algorithm.

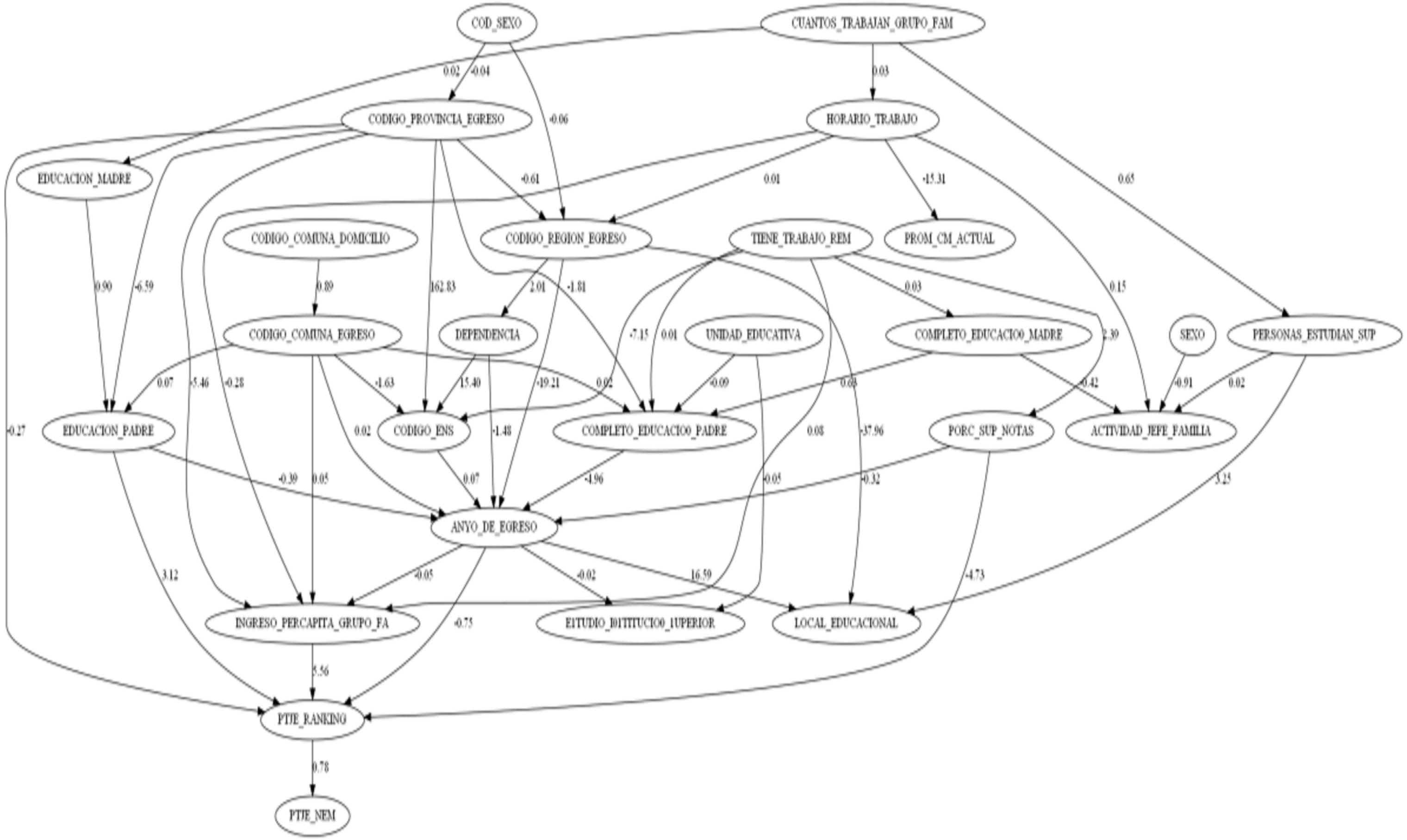


Fig. 3. Causal graph generated by LINGAM algorithm

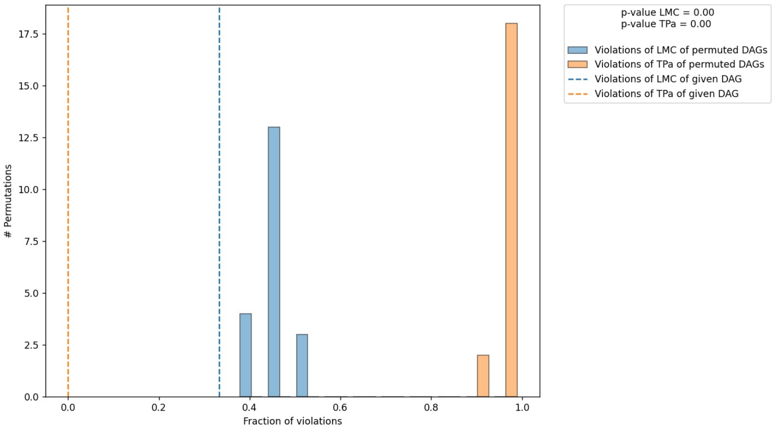


Fig. 4. Result of the Falsification Test in LINGAM Algorithm

that asking about the number of permutations that violate zero LMC is identical to asking how many DAGs lie in the same Markov equivalence class (MEC) as the given DAG. The method uses the number of permuted DAGs that lie in the same MEC as the given DAG (with zero tPA violations) as a measure of how informative the given DAG is. Fig. 4. shows the results of test.

Given the previous graph, the histogram of LMC violations of the permuted DAGs (blue) and d-separation violations (oracle, orange) of the permuted DAGs is shown. The dashed orange and blue lines indicate the number of LMC violations (blue) / d-separation violations (orange) of the given DAG. As expected for the true DAG, both histograms overlap significantly (except for statistical errors in the CI tests) [4].

From the application, the following result is obtained: The given DAG is informative because 0/20 of the permutations lie in the Markov equivalence class of the given DAG (p-value: 0.00). The given DAG violates 77/231 LMC and is better than 100.0% of the permuted DAGs (p-value: 0.00). According to the provided significance level (0.05) and since the DAG is informative, we do not reject the DAG.

Causal discovery methods require appropriate assumptions for accuracy guarantees, and thus there will be variation among the results returned by different methods in practice. However, these methods can be usefully combined with domain knowledge to construct the final causal graph. The causal effect estimate of INGRESO-PERCAPITA-GRUPO-FA on PTJE-NEM is positive and corresponds to 4.817 which appears reasonable. Note that PTJE-NEM is the more relevant variable given that it indicates the points that a student obtains in the test, therefore, we choose as output variable. We choose INGRESO-PERCAPITA-GRUPO-FA as treatment variable because is the more relevante variable as it indicates the income of the student's family.

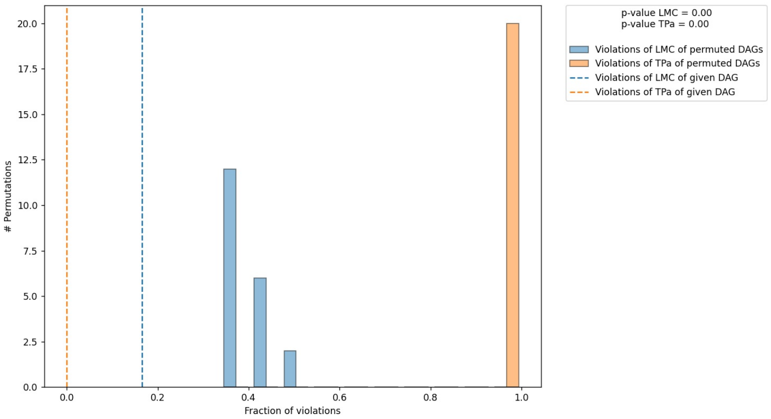


Fig. 5. Result of the Falsification Test in GES Algorithm

For the GES algorithm, the result was similar to that of LINGAM, yielding the result of Fig. 5. The given DAG is informative because 0/20 of the permutations are in Markov. The given DAG violates 20/121 LMC and is better than 100 percent of the permuted DAGs (p-value: 0.00). According to the significance level provided (0.05) and because the DAG is informative, we conclude that the test do not reject the DAG. Complementary, in the estimation of the causal effect of INGRESO-PERCAPITA-GRUPO-FA on the PTJE-NEM is 4.018 which also appears reasonable.

In terms of usability, the presented graphs may pose challenges for decision-makers. We believe that, in particular, the graph given by LINGAM appears more interpretable; however, as future work, we plan to validate it functionally. Regarding decision-making, our next steps involve simulating interventions based on the causal graph to observe the effect and thus aid in decision-making, for example, by helping students in specific conditions. Finally, to mitigate the impact of hidden variables, we plan to incorporate more variables in the future, such as weather or the presence of policies.

7 Conclusion

In conclusion, the evaluation of the PC, GES, and LINGAM algorithms highlights the strengths and weaknesses of each in causal discovery. With more success, LINGAM effectively tests for Local Markov Condition violations and provides a clear validation framework through permutations. The analysis reveals that the given DAG is informative, as it consistently outperforms the permutations in identifying true causal relationships. The discovery of causal structure appears promisory to help to identify the causal structures inside the educational data in Chile.

Acknowledgments. P. Quinsacara thanks the support of the program of PhD in Advanced Systems Engineering, Universidad Andres Bello. B. Peralta appreciate the support of the National Center for Artificial Intelligence CENIA FB210017, Basal ANID..

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Card, D.: Estimating the return to schooling: Progress on some persistent econometric problems. Econometrica **69**(5), 1127–1160 (2001)
2. Chen, Y., Wei, W., Wang, L., Dong, Y., Liang, C.J.: Where do they go next? causal inference-based prediction and visual analysis of graduates' first destination. Journal of Visualization pp. 1–24 (2024)
3. Chickering, D.M.: Optimal structure identification with greedy search. Journal of Machine Learning Research **3**, 507–554 (2002)
4. Eulig, E., Mastakouri, A.A., Blöbaum, P., Hardt, M., Janzing, D.: Toward falsifying causal graphs using a permutation-based test. Machine Learning (2023)
5. Gerard, F., Hugonnier, B., Varin, S.: Bayesian networks and causal discovery: what lessons for the synthetic indicator of the quality of education systems in oecd countries. International Journal of Research in Social Sciences **13**(1), 73–81 (2023)
6. Kalisch, M., Bühlmann, P.: Estimating high-dimensional directed acyclic graphs with the pc-algorithm. Journal of Machine Learning Research **8**, 613–636 (2007)
7. Lipsky, A.M., Greenland, S.: Causal directed acyclic graphs. JAMA **327**(11), 1083–1084 (2022)
8. Misiunas, N., Raspopovic, M., Chandra, K., Oztekin, A.: Sensitivity of predictors in educational data: A bayesian network model. In: 2015 INFORMS Workshop on Data Mining and Analytics. The Institute for Operations Research and the Management Sciences (2015)
9. Murnane, R.J., Willett, J.B.: Methods matter: Improving causal inference in educational and social science research. Oxford University Press (2010)
10. Pearl, J.: Causality. Cambridge university press (2009)
11. Peralta, B., Salazar, J., Levano, M., Nicolis, O.: A causal modelling for desertion and graduation prediction using bayesian networks: a chilean case. In: 2021 IEEE International Conference on Automation/XXIV Congress of the Chilean Association of Automatic Control (ICA-ACCA). pp. 1–7. IEEE (2021)
12. Rodríguez, C.G.: Educative uses of the ict, technological skills and academic performance of the venezuelan university students (barineses): a causal perspective. International Journal of Education and Development using ICT **2**(4), 28–43 (2006)
13. Sharma, A., Kiciman, E.: Dowhy: An end-to-end library for causal inference. arXiv preprint arXiv:2011.04216 (2020)
14. Shimizu, S., Hoyer, P.O., Hyvärinen, A., Kerminen, A.: A linear non-gaussian acyclic model for causal discovery. Journal of Machine Learning Research **7**, 2003–2030 (2006)
15. Silva Filho, R.L.C., Brito, K., Adeodato, P.J.L.: Leveraging causal reasoning in educational data mining: An analysis of brazilian secondary education. Applied Sciences **13**(8), 5198 (2023)
16. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. MIT Press (2000)

Exploring Connectivity in Parkinson's Disease Using Graphical Models and fNIRS

Samuel Montero-Hernandez[1] and Edgar Guevara[2]

[1] University of Birmingham, School of Computer Science s.montero@bham.ac.uk
[2] CONAHCYT-Universidad Autónoma de San Luis Potosí, Mexico

Abstract. Bilateral motor functional connectivity during a motor task was investigated in patients with Parkinson's disease (PD) and healthy controls using Graphical Models and continuous-wave functional Near-Infrared Spectroscopy (fNIRS). Here, we integrate probabilistic graphical models (PGMs) with fNIRS data to compare functional connectivity between PD patients and healthy controls. Results revealed significant differences in connectivity, particularly during left-hand motor tasks, highlighting disrupted communication in PD motor networks. Using structure Hamming distance (SHD) metrics, we quantified these differences, demonstrating the potential of PGMs to capture the altered brain network architecture in PD. Our results present GM as a promising tool for functional connectivity analysis in the fNIRS field.

Keywords: Functional connectivity · Parkinson's disease · fNIRS.

1 Introduction

1.1 Parkinson's disease and functional neuroimaing

Parkinson's disease (PD) is a progressive neurodegenerative disorder characterized by motor symptoms such as bradykinesia, tremor, rigidity, and postural instability, as well as non-motor symptoms like cognitive decline, depression, and sleep disturbances. In Latin America, the prevalence of PD is increasing due to the aging population and improved access to diagnostic tools. Countries such as Brazil, Argentina, and Mexico report the highest prevalence rates, estimated at 40–50 cases per 100,000 individuals [7]. However, healthcare disparities, late diagnosis, and limited access to specialized care continue to present significant challenges in the management of PD in this region [11].

Functional neuroimaging studies have demonstrated alterations in resting-state networks, including disrupted connectivity within the motor and cognitive networks in PD patients [3]. These neuroimaging findings have contributed to identifying potential biomarkers for early diagnosis and monitoring of disease progression. Functional brain connectivity studies have become crucial in unraveling the neural network disruptions associated with Parkinson's disease. Functional near-infrared spectroscopy (fNIRS) has recently emerged as an alternative tool to explore brain connectivity, particularly in cortical regions. The characteristics of fNIRS, such as portability, non-invasiveness, and tolerance to motion, make this modality suitable for interrogating brain activity in the PD population. Initial studies using fNIRS in PD patients have shown altered cortical connectivity during motor tasks, providing a complementary approach to fMRI in understanding the neural mechanisms of PD [8].

1.2 Probabilistic and Causal Graphical Models in Brain Connectivity

Probabilistic graphical models (PGMs), such as Bayesian networks, enable the modeling of probabilistic dependencies between brain regions, providing insights into functional connectivity by estimating conditional independencies. These models are especially powerful in brain connectivity studies as they can disentangle direct from indirect interactions, allowing researchers to infer not just correlations but potential causal pathways in neural networks [5, 1, 10]. The application of fNIRS to study functional connectivity in Parkinson's disease is still relatively new. Few studies have employed PGMs to explore the altered connectivity patterns in PD patients during motor tasks, revealing impaired communication between the pre-motor and supplementary motor areas [14]. However, comprehensive exploration of causal pathways using causal graphical models in fNIRS studies of PD remains limited. Still, the potential to map causal disruptions in these networks could lead to early detection markers and better treatment strategies [8, 9].

2 Methods

2.1 Experimental protocol

The study was approved by the Institutional Review Board and took place at the Central Hospital "Dr. Ignacio Morones Prieto" in San Luis Potosí, Mexico, from October 2021 to October 2022. The study included twenty patients with PD (mean age 69.9, SD = 10.1, 60% male) and 20 healthy subjects matched for age and sex, without any movement disorders (mean age 65.6 years, SD = 9.9, 45% male). PD patients were assessed using the United Kingdom PD Society Brain Bank criteria. Disease severity was determined using the Hoehn and Yahr scale and the MDS-UPDRS. The brain activity was measured using the portable fNIRS system Brite MKII (Artinis Medical Systems BV, the Netherlands)(760 and 850 nm at 25 Hz). The setup included 20 channels (10 per hemisphere) placed across bilateral motor regions of the brain to measure changes in hemoglobin. Following a previously used paradigm [2], participants performed a 10-second finger-tapping task followed by a 20-24 s recovery period targeting the primary motor cortex (M1) activation. The recording included 20 blocks (10 per hand) lasting approximately 11 minutes (approximately 4000 samples after resampling at 6Hz). Data is publicly available online (https://doi.org/10.5281/zenodo.7966829).

2.2 Graphical Models

The PC algorithm, developed by Spirtes and Glymour, learns the structure of a causal graph from observational data. It assumes conditional independence and uses statistical tests to identify these independencies. The algorithm consists of three phases: identifying the graph's skeleton, orienting edges, and resolving any remaining ambiguities in edge directionality. It is widely used due to its scalability and ability to handle large datasets, making it suitable for applications like brain connectivity modeling [13].

2.3 Assessment metrics

Here, we opted for the structural Hamming distances to evaluate the intra-group structural differences [12, 4]. We computed the matrix SHD in which each entry (i, j) is obtained as $SHD_{i,j}(G_i, G_j)$, where G_i and G_j are the models discovered by the PC algorithm for the subjects i and j $(i \neq j)$. Considering the patterns (equivalence classes) obtained, we computed four SHD matrices from the combination of the groups (control and PD) and stimulation type (left finger-tapping and right finger-tapping). Finally, we determined the statistical intra-group differences by using a T-test with an alpha value of 0.05.

3 Results

The time series from HbO were used as input data to the PC algorithm. We used the stable (order independent) PC version with $\alpha = 0.05$ and Fisher's Z as the conditional independence test [?]. Then, the PC algorithm generated a pattern (equivalence class) P_i for each subject i. Next, the $SHD_{C,R}, SHD_{PD,R}, SHD_{C,L}$, and $SHD_{PD,L}$ were calculated for the control and PD groups and left and right finger tapping task. Examples of a structure $P_i, SHD_{C,L}$, and $SHD_{PD,L}$ are shown in Fig. 1 a and b.

From the comparison of the structural Hamming distance (SHD) matrices, we observed a smaller variance within the LFT control group. In Figure 1 c, a box plot set depicts the SHD results distribution from subjects within the same group. In this sense, a higher intra-group variable was observed in the PD group. The analysis indicated that the control group exhibited statistically significant differences compared to the PD group, specifically to LFT (t-value = -8.29, dof = 378, p-value < 0.001).

4 Conclusion

In this work, we have used graphical models to compare the functional connectivity differences between control and Parkinson's disease patients using fNIRS. We employed a finger-tapping task to stimulate hemodynamic activity and processed the time series data using a standard pipeline for input into the PC algorithm. By comparing the structures of each pair of subjects, we quantified

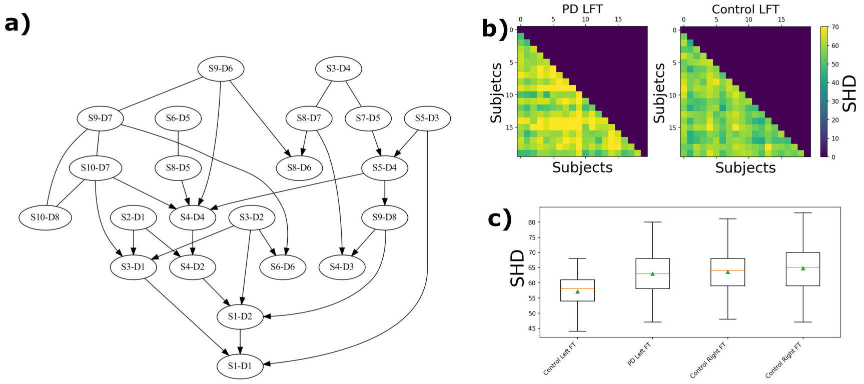


Fig. 1. a. Example of the structure of a graphical model G_i from Control 02. b Examples of $SHD_{C,L}$, and $SHD_{PD,L}$. c. Box plots of the $SHD_{C,R}, SHD_{PD,R}, SHD_{C,L}$, and $SHD_{PD,L}$ groups. The mean value is represented with a green triangle.

the difference utilizing the structure Hamming distance (SHD). Our results revealed distinct differences in connectivity between the control and PD groups, particularly during the left finger tapping task. This can be caused by the inter-subject variability, typically observed in the individual structure approach [6]. Our approach highlights the potential of graphical models in identifying connectivity variations in fNIRS data between healthy and non-healthy populations. Our findings indicate that employing more specialized methodologies could elucidate even more nuanced distinctions among groups and commonalities within individuals.

References

1. Friston, K.J., et al.: Dynamic causal modeling. NeuroImage **19**(4), 1273–1302 (Aug 2003)
2. Guevara, E., Rivas-Ruvalcaba, F.J., Kolosovas-Machuca, E.S., Ramírez-Elías, M., de León Zapata, R.D., Ramirez-GarciaLuna, J.L., Rodríguez-Leyva, I.: Parkinson's disease patients show delayed hemodynamic changes in primary motor cortex in fine motor tasks and decreased resting-state interhemispheric functional connectivity: a functional near-infrared spectroscopy study. Neurophotonics **11**(2), 025004 (2024)
3. Hacker, C.D., Perlmutter, J.S.: Resting state functional connectivity of the striatum in parkinson's disease. Brain **135**(12), 3699–3711 (2012)
4. Heinze-Deml, C., Maathuis, M.H., Meinshausen, N.: Causal structure learning. Annual Review of Statistics and Its Application **5**, 371–391 (2018)
5. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge, MA, USA (2009)
6. Li, J., Wang, Z.J., Palmer, S.J., McKeown, M.J.: Dynamic bayesian network modeling of fmri: A comparison of group-analysis methods. NeuroImage **41**(2), 398–407 (2008)

7. Marras, C., Beck, J.C., Bower, J.H., et al.: Prevalence of parkinson's disease across north america. NPJ Parkinson's Disease **4**(1), 21 (2018)
8. Muthalib, M.M., Besson, P., Rothwell, J., Perrey, S.: Effects of anodal high-definition transcranial direct current stimulation on bilateral sensorimotor cortex activation during sequential finger movements: An fnirs study. In: Advances in Experimental Medicine and Biology, vol. 789, pp. 89–95. Springer (2013)
9. Naseer, N., Hong, K.S.: fnirs-based brain-computer interfaces: A review. Frontiers in Human Neuroscience **9**, 3–15 (Jan 2015)
10. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press, Cambridge, U.K., 2nd edn. (2009)
11. Perez-Lloret, S., Rascol, O.: The prevalence and incidence of parkinson's disease in latin america. Movement Disorders **33**(Suppl 1), S1–S5 (2018)
12. Scutari, D., Choi, J.D.: Bayesian network structure learning with application to brain functional networks. Frontiers in Neuroscience **14**(103), 1–15 (2020)
13. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. MIT Press, Cambridge, MA, USA, 2nd edn. (2001)
14. Wu, T., Wang, L., Hallett, M., Chen, Y., Li, K., Chan, P.: Effective connectivity of brain networks during self-initiated movement in parkinson's disease. NeuroImage **55**(1), 204–215 (2011)