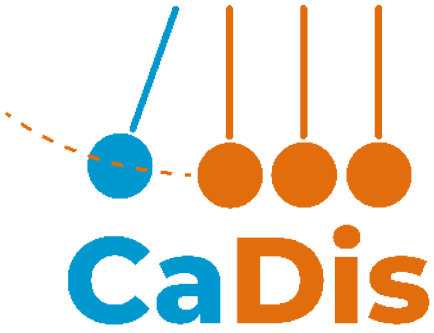
Workshop on Causal Discovery
CaDis 2023



ACADEMIA MEXICANA DE COMPUTACIÓN, A. C.

Proceedings of the 1st Workshop on Causal Discovery CaDis 2023
Editors: Luis Enrique Sucar Succar, Julio César Muñoz Benítez

In collaboration with Academia Mexicana de Computación.

First Edition 2023.
Academia Mexicana de Computación, A. C.

ISBN:

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

Printed in Mexico.

Proceedings of the 1st Workshop on Causal Discovery
CaDis 2023

Workshop Chairs:

Luis Enrique Sucar, INAOE
Julio César Muñoz-Benitez, INAOE

Program Committee:

Armando Aguayo, Universidad de Deusto
Nicandro Cruz, Universidad Veracruzana
Adnan Darwiche, University of California, Los Angeles
Hugo Jair Escalante, INAOE
Mauricio González, University of Vienna
Eduardo Morales, INAOE
Julio César Muñoz-Benitez, INAOE
Luis Enrique Sucar, INAOE

Foreword

This volume contains the proceedings of the 1st Workshop on Causal Discovery (CaDis 2023). The workshop was held at the National Institute of Astrophysics, Optics and Electronics (INAOE) in Tonatzintla, Puebla, Mexico, June 19–21, 2023.

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 2023 program included invited talks by Prof. Adnan Darwiche (professor and former chairman of the computer science department at UCLA) and Dr. Rubén Sánchez-Romero (Rutgers-Newark Center for Molecular and Behavioral Neuroscience). Video recordings of these talks are available at the workshop website: https://cadisworkshop.com.mx/.

After a review by at least three members of the program committee, eight papers were accepted for publication and from these, six are included in these proceedings. In an analogous way as the workshop, this proceeding are divided in three parts: (i) Invited talks abstracts, (ii) Fundamentals and Algorithms for Causal Dis-

cover, and (iii) Applications.

We hope that this workshop will help to increase the interest of the Mexican and Latin American computing community in causal reasoning and discovery, and we plan to held a second workshop in 2024.

Luis Enrique Sucar and Julio César Muñoz-Benitez
Workshop Chairs

Contents

Section 1

Invited Talks

Optimizing Causal Objective Functions

Author: Porf. Adnan Darwiche

Abstract: A causal objective function scores objects, called units, based on how likely they are to exhibit a certain mode of causal behavior. We discuss the syntax and semantics of causal objective functions (i.e., causal loss functions) and present an exact algorithm for optimizing a broad class of such functions. We also discuss results that bound the complexity of the algorithm and identify the complexity class of this optimization problem. Optimizing causal objective functions is quite related to the "unit selection" problem introduced by Li & Pearl, with two key distinctions: (1) we treat a broad class of causal objective functions that include the "benefit function" used by Li & Pearl as a special case and (2) we take an algorithmic direction that assumes a fully specified causal model—to compute point values of causal objective functions—instead of focusing on computing bounds on the causal objective function using observational and experimental data.

Unveiling brain network mechanisms supporting cognitive activation with activity flow models and causal functional connectivity

Author: Dr. Rubén Sánchez-Romero

Abstract: Activity flow models estimate task-evoked brain activity moving across connections to explain network based task functionality. While these models accurately predict brain activation, they face limitations due to causal interpretation issues in standard functional connectivity pairwise measures, for example, confounding from common causes or causal chains. To address this, we show that connectivity measures that leverage conditional independence information and are grounded in causal principles can provide accurate predictions of task-evoked activation and facilitate causal mechanistic interpretations of activity flow models. We compare the performance of correlation, multiple regression, combinedFC and the PC algorithm in simulations and empirical fMRI data across a large battery of cognitive tasks. Finally, applying PC-based causal activity flow models to the dorso-lateral prefrontal cortex during a working memory task, we uncover distributed causal network mechanisms supporting well documented working memory effects. These results have the potential to inform future interventions aimed to reduce the impact of working memory deficits from cognitive decline.

Section 2

Fundamentals and Algorithms for Causal Discovery

Data Imputation with Adversarial Neural Networks for Causal Discovery from Subsampled Time Series

Julio Muñoz-Benítez[1] and L. Enrique Sucar[1]

Instituto Nacional de Astrofísica Óptica y Electrónica, Coordinación de Ciencias Computacionales, Puebla, Mexico {jcmunoz,esucar}@inaoep.mx

Abstract. A relevant and challenging problem is causal discovery from time series data. This helps to understand dynamics events present in real world scenarios. However, causal interactions may occur at a time scale faster than the measurement frequency, resulting in a subsampled time series. This can lead to significant errors during causal discovery. We propose an approach based on imputing the missing data using adversarial neural networks to try to recover the true causal structure. The trained model is fed with the sub-sampled time series in order to generate data that behaves similarly to the original time series, so that the original causal structure can be recovered. The completed data series is then fed to a causal discovery algorithm. Experimental results on several synthetic dynamic models show that the imputed data time series is close to the original one, and that the causal structure derived from this data resembles the correct causal structure. The proposed method is applied to real data from a weather monitoring site using information of nearby sites, recovering a causal structure based on imputed data close to the original structure when subsampling is present.

Keywords: Causal Discovery · Time Series · Sub Sampling

1 Introduction

Inferring causal relations from time series have served as the basis for causal discovery in various fields of science such as climate systems, ecological networks, effective connectivity in the brain, and finance [5, 8, 14, 17]. Data collected can provide precise measurements at regular points of time [1]. One of the main advantages of using observational data from time series is that the temporal order of the information can simplify causal analysis [9, 14]. That is, the causal driver can be identified as the variable that occurred first, as the future can't affect the past [1, 16, 15]. However, the study of causal relations in time series is still a challenging issue, which is partly due to the complexity and dynamism of real world systems and, in many cases, the time series data may contain erroneous measurements, inconsistent data, or even missing data.

One of the main challenges of causal discovery from time series is that causal interactions may occur on a time scale faster than the frequency of measurement [8, 9], this phenomena is known as *subsampling*. This can lead to a loss of

valuable information to determine the true causal relationships between events. Subsampling could lead to significant errors in the obtained causal structure, as shown in previous work [2]. Although causal discovery in subsampled time series is relatively under explored [3], it is a challenge that must be addressed in order to avoid learning incorrect causal relations from observational data when studying dynamic events. Previous work that considers the subsampling problem has focused on obtaining an *equivalence class* of causal structures consistent with the subsampled data measurements [2, 8, 15]. However, they can not determine the *true* causal structure in the equivalence class.

We present a novel approach to solve this problem, how to obtain a unique causal structure given undersampled data. Our approach is based on imputing the missing data using generative adversarial neural networks (GANs). We assume that the rate of subsampling is known, and we estimate the missing data between the data samples provided. We train a GAN to estimate the missing samples based on several time series, and then, given new data and the subsampling rate, we estimate the missing samples. Once the time series is completed, we obtain the causal structure using a method for causal discovery from time series [13] and it is verified whether the resulting causal structure is consistent with the possible causal structures derived from the original structure [8].

We have evaluated the proposed approach in different scenarios of increasing complexity. The results show that the imputed data is close to the original data, and that the discovered causal structure is also very close the correct one that generated the data. We propose the use of the adjacency matrix as a way to compare the causal structures. Thus, the main contributions of this paper are: (a) a deep learning model based on a GAN architecture for data imputation in time series affected by subsampling; (b) a method based on the use of the adjacency matrix that provides a numerical score to compare causal models; and (c) an experimental evaluation of the proposed approach.

2 Background

2.1 Causal Graphical Models for Time Series

In order to model dynamical systems one may use graphical models such as *Directed Acyclic Graphs (DAGs)*, which consist of a series of nodes connected through edges or links directed from parent nodes to child nodes [11]. The nodes in the DAG represent the variables and the links indicate the causal relationships between these variables. In particular, in the case of dynamic systems, this representation is known as a *dynamic causal Bayesian network* [12]. In this work, it is assumed that there is causal sufficiency, that is, that there are no hidden variables that affect the observed variables [2, 14]. In addition, it is also assumed that the causal relationships are invariant over time [2, 15]. An example of the representation of a causal structure from a time series can be seen in Figure 1a.

2.2 Subsampling in Time Series

One of the main challenges of using data from time series is that causal interactions may occur on a time scale faster than the frequency of measurement [8, 9]. This can lead to a loss of valuable information to determine the true causal relationships between events. An example of this can be seen in Figure 1, where the original causal structure of the time series is shown (Fig. 1a); and the causal structure of the same process under subsampling, obtained by making observations every two time steps (Fig. 1b). If it is assumed that the structure of Fig. 1b is correct, valuable information about the true causal relationships between the variables is lost. This may lead to believe that variable Z can be intervened to control Y, but the true influence of Z on Y is mediated by X. Thus, an intervention in X would be more effective. Similarly, if the structure of Fig. 1b is used, the predictions of the behavior of the variables can be completely different from those obtained if the true causal structure of the time series is used [8].

3 Related Work

Most causal discovery methods are designed to analyze identically distributed and independent data (IID). Causal discovery from time series requires a different approach [2, 9, 11, 15]. Next we present a summary of related work, including the causal discovery algorithm we use in this work; the application of deep learning for causal discovery; and causal discovery for the case of subsampling.

3.1 PCMCI Algorithm

The PCMI algorithm is focused on causal discovery in time series [14], solving some limitations of the PC algorithm [17]; in particular the processing time in data sets with high dimensionality, and eliminating irrelevant variables that could lead to the appearance of inconsistent causal relationships. The PCMI algorithm aims to solve these problems through the selection of conditions to

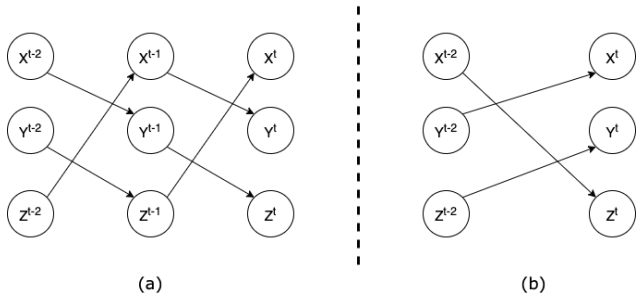


Fig. 1. Causal structures for a time series with variables$\{X,\ Y,\ Z\}$. (a) Original structure. (b) Structure obtained from subsampled data (every two time steps).

eliminate irrelevant variables and a test of conditional independence for the discovery of causal relationships between variables [13].

This is achieved through the implementation of two stages. The first being the condition selection in order to identify the most relevant conditions for all the variables in the time series; that is, only those variables with the largest associations are selected rather than selecting all possible combinations. Subsequently, momentary conditional independence (MCI) is used as an estimator for causal strength, based on auto correlation, and as an identifier of false positives by means of a conditional independence test. In this way, the causal relationships with the highest probability are established, estimating the causal strength as well as the type of correlation between them. However, the PCMCI algorithm, as most causal discovery algorithms from time series, assumes that the data is sampled at the appropriate time scale; so if it is presented with subsampled data it will produce, in general, an incorrect causal structure.

3.2 Deep Learning in Causal Discovery

One of the novel approaches for causal discovery is the use of deep learning techniques, such as learning the causal structure from observational data taking advantage of continuous optimization [18]. The use of neural networks allows data to be analyzed to infer causal relationships [4, 6]; likewise, the use of adversarial neural networks has been a promising approach for generating missing samples. In [19], a deep learning framework is used to impute data on an incomplete observational data set. Synthetic data generated by this approach helps the causal discovery of existing relationships with the objective of generating the causal graph. However, these data is invariant over time, this means that the observational data set is not part of a time series. The work proposed in [7] reflects the versatility of the use of neural networks for causal discovery of observational data in time series, obtaining good results in the inference of causal relationships, including their direction and intensity, although it is assumed that the time series is complete and is not affected by subsampling.

3.3 Causal Discovery in Subsampled Time Series

Danks te al. [2] developed an algorithm that allows learning a set of causal structures even if the level of subsampling is unknown. This is performed through a graphical representation of the causal structure of the time series which is known or inferred. Subsequently, all the possible causal structures are obtained, comparing them with the initial causal structure, which may be affected by some degree of sub-sampling. In this way, if the new structures are consistent with the original structure they are considered as a possible causal structures, obtaining an *equivalence class* of causal structures.

The use of this approach presents computational challenges that limits its use to *small* models. Furthermore, since the causal structure is obtained through the time series data, statistical errors may occur that imply that some structures are not consistent with the original causal structure or that structures that are

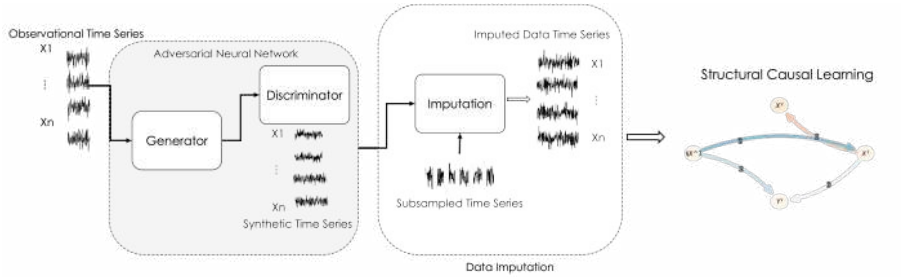


Fig. 2. A block diagram of the proposed method. The gray block at the left represents the training stage and the white block at the right the imputation phase. The generator model generates samples until the discriminator model accepts the synthetic time series as valid. This synthetic time series complements the missing values so that the output is a time series with imputed data that resembles the original time series. The completed data is fed to a causal structure learning algorithm to obtain the causal structure.

actually consistent are not taken into account. [8] extends the previous approach by proposing a constraint satisfaction procedure which is computationally more efficient, and can also recover from conflicts due to statistical errors. Recent work [15] extends this approach to obtain an equivalence class of causal structures with multiple measurement timescales. In this way, it is possible to indicate how many structures are part of the subset of possible causal structures given an initial one. This allows to quantify the resolution, or gain, of the size of the equivalence class and to evaluate whether or not a causal structure belongs to the subset of possible causal structures.

The previous developments can find the set of possible causal structures that are consistent with the under-sampled data given a known or even unknown subsampling rate, but can not select among these the *correct* one. The present work proposes an approach to impute the missing data due to subsampling, in order obtain to a single causal structure *close* to the correct one.

4 Data Imputation for Causal Discovery of Time Series

The proposed approach aims to minimize subsampling in time-series, by imputing data generated in an artificial way, to obtain the original causal structure. A conceptual diagram of the proposed model is shown in Fig. 2.

4.1 Representation and Assumptions

In this work, the time series data is composed of a set of variables $V^t = \{X_1^t, X_2^t, X_3^t, ...\}$ that may take discrete or continuous values in discrete points of time. This means that the time series can be represented as a dynamic Bayesian network. The following assumptions are considered: (i) Time invariant; that is,

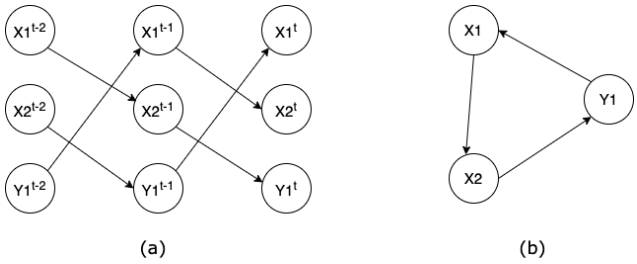


Fig. 3. (a) A time series causal structure where the causal links are repeated over time. (b) The same causal structure showed as a rolled graph.

the causal links between variables are repeated through time. (ii) Causal sufficiency; that is, V^{t-1} includes all common causes of V^t and there are not causal links of the form $X_i^t \rightarrow X_j^t$ [8]. Figure 3 shows an example of the structure of a time series and a simplified representation (*rolled graph*) for the same structure.

4.2 Data Imputation

Imputation methods aim to make use of the available information and estimate missing data to obtain a complete data set. For data generation and imputation a generative adversarial neural network (GAN) [20] was used. This type of model learns regular patterns from the input data in such a way that the model can generate output data that may have a similar behavior, such that the generated data may be considered as part of the original data set. In this sense, we may capture the original distribution by making the distribution of the outputs (synthetic data) approximate the original data distribution. This is achieved by two models: the *generator* that is trained to generate data based on the original data set, and the *discriminator* that aims to classify the received data as real or fake. These two models work together until the discriminator model accepts the generated data as if these data belong to the original data set.

4.3 Generative Adversarial Model Architecture

Figure 4 shows the architecture of the generator and discriminator. The Generator receives the input data and outputs a synthetic sample *G(z)*. The Discriminator takes either a training sample x or a synthetic sample *G(z)* as input. The output is a scalar indicating the probability that x or *G(z)* follows the original data behavior. The generator performs 1-D, or 1 stride, fractional convolutions, often called as deconvolutions, using rectified linear units (ReLU). The discriminator is an inverse of the generator. The features of the time series are extracted using 1-D kernel layers based on convolutions with stride 1 which outputs a scalar. For imputation we use the generator based loss as the loss function; we employ back-propagation to find the closest latent value of input data and then use the samples generated by the generator to impute the missing values.

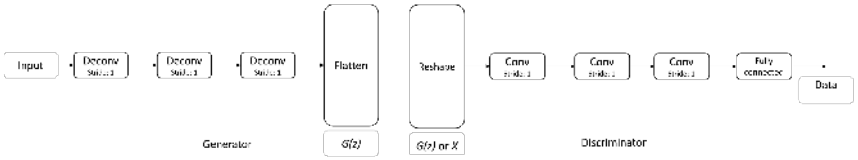


Fig. 4. Architecture of the generator and the discriminator, which contain three fractional convolution layers (Deconv) and three strided convolutional layers.

Training The discriminator was trained to minimize the classification loss, and the generator was trained to maximize the discriminator's missclassification rate. Considering the data input $\mathbf{X}$, the main objective is to generate data that approaches to the data distribution, $P(\mathbf{X})$. Thus, the imputed data can be obtained by replacing missing entries with the generated corresponding values according to the learned distribution.

Testing In the test phase, the input to the ANN is a time series data affected by the subsampling. Assuming a subsampling rate of two, the input consist of an incomplete time series, $\mathbf{X_1}, \mathbf{X_3}, \mathbf{X_5}, ...\mathbf{X_{N-1}}$; and the output is the completed time series generated by the ANN: $\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}, \mathbf{X_4}, ...\mathbf{X_{N-1}}, \mathbf{X_N}$. The data that complements the subsampled time series are data that has the highest probability for the missing values of the time series. That is, the input values are taken into account to predict the value of data that were not observed.

4.4 Causal Structure Learning

Once the data generated complements the time series, the PCMCI algorithm [14] is used to reconstruct its causal structure [1]. This algorithm was used due to its good performance in the causal discovery of time series, which have not been affected by subsampling. Likewise, the causal links between the variables are reconstructed specifying both causal strength of the relationships between them and the time step in which these relationships arise. In this way, this algorithm serves to analyze and compare the original causal structure and the structure resulting from the imputed data. Thus, we can verify if the data generated by the ANN maintains the causal relationships.

4.5 Causal Structure Verification

As a way to verify the causal structure of the time series learned from the imputed data, the method proposed by [8] is used. This approach takes into account a subsampled time series and assumes that the subsampling rate is known. A causal

[1] The Python module Tigramite [14], which implements the PCMCI algorithm, was used.

structure of the subsampled time series $\mathcal{H}$ is obtained, and used to generate all possible causal structures $\mathcal{G}$ that are consistent with the original causal structure. In this way, it can be verified whether the causal structure, V, obtained with the imputed data is found as a possible graph in $\mathcal{G}$ consistent with $\mathcal{H}$. If $V \notin G$, we select the *most similar* causal structure in $\mathcal{G}$.

4.6 Adjacency Matrix

We use the adjacency matrix as a way to evaluate the difference between the causal structures, such that each link in the original causal structure is represented as *1* if it is present or as *0* if it is absent. This is represented for each time step until we reach the maximum lag in which causal links may appear[2] [10]. Based on the adjacency matrix, the *Mean Average Error* (MAE) is used to evaluate the difference between two adjacency matrices:

$$MAE_{(Y,\hat{Y})} = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \hat{y}_i| \tag{1}$$

Where n is the number of elements in the adjacency matrix, y_i is a link in the original causal structure and $\hat{y}_i$ is a link of the estimated causal structure. If there are causal links at different time lags, an adjacency matrix and MAE are obtained for each lag, and the total MAE is the sum of the MAEs for each time lag. Using this metric we can compare causal structures, in particular the structure obtained with the imputed data vs. the original structure, and the one obtained from the subsampled data. Additionally, this metric is important for the verification stage described in the previous section, as it allows us to select the causal model that is closest to the predicted one.

5 Experimental Results

5.1 Experimental Setup

To perform an analysis of the imputation of data on the time series and compare the resulting causal structure, artificially generated time series were used. In this way, the resulting causal structure of the time series is known before hand to be compared with the structure obtained from the subsampled time series, and the resulting structure of the time series with the imputed data.

To generate the data, the structure and parameters of the time series are specified, and the tool developed in [9] was used to generate N data points. Then a fraction of this data points is deleted to simulate subsampling ($N/2$ for a subsampling rate of two), and we apply the proposed method to this data. These data points are generated based on linear models affected by some degree of noise. The time series generated are based on a structural causal model that assumes that the child nodes in a causal graph have a functional dependence

[2] See Table 1 in the experimental results for an example of an adjacency matrix.

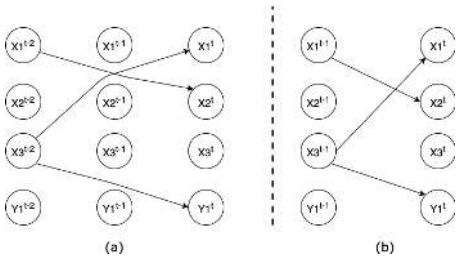


Fig. 5. Causal structure of the time series used in the first experiment: (a) The original causal structure shows a link from X_1 to X_2, a link from X_3 to X_1 and X_3 to Y_1 at two time steps. (b) The causal structure of the subsampled time series shows a link from X_1 to X_2 at one time step, as well for the links from X_3 to X_1 and to Y_1. This demonstrates how subsampling affects the causal relationships between variables.

on their parents. That is, given a set of variables $\{X_1, X_2, ..., X_n\}$ each variable X_i can be represented in terms of a function F_i and its parents $Pa(X_i)$, as $X_i = F_i(Pa(X_i), N_i)$ where F_i are linear models and N_i are noise terms with a given distribution (Guassian, Student's t, Laplace, Uniform).

We performed a series of experiments. First, a simple example with four variables in which the training and testing time series have the same probability distribution. Next, more complex scenarios with four variables and causal links at different time steps, in which the training and test time series have different distributions. In each experiment we compare: (i) the reconstruction of the time series, comparing the original data with the imputed data; (ii) the causal structures learned form the subsampled and imputed data vs. the correct structure.

5.2 Experiment 1: Subsampled Time Series with the Same Distributions

For an initial experiment we consider a simple time series with four variables, X_1, X_2, X_3, and Y_1. A time series composed of 1,000 observations was generated for each one of the variables. This same time series was affected by a subsampling rate of two, that is, the observed data comprise values of the variables every two time steps. Figure 5a depicts the causal structure of the original time series where there is a causal link from *X1* to *X2* every two time steps. However, when analyzing the causal structure of the time series affected by subsampling, Figure 5b, the resulting causal link, although it is specified in a correct way from *X1* to *X2*, is represented in an single time step. Similarly for the links from *X3* to *X1* and *X3* to *Y1*. These represent errors in the causal structure.

The time series was completed with imputed data for each one of the observed variables: X_1, X_2, X_3 and Y_1. A comparison is depicted in Figure 6, where it can be seen that the generated data is very close to real data, presenting a similar behavior over time. The mean absolute error, which is the measure of the difference between both sets of values, allows to quantify the precision of the generated values compared to the original values, resulting in a value of 0.0209

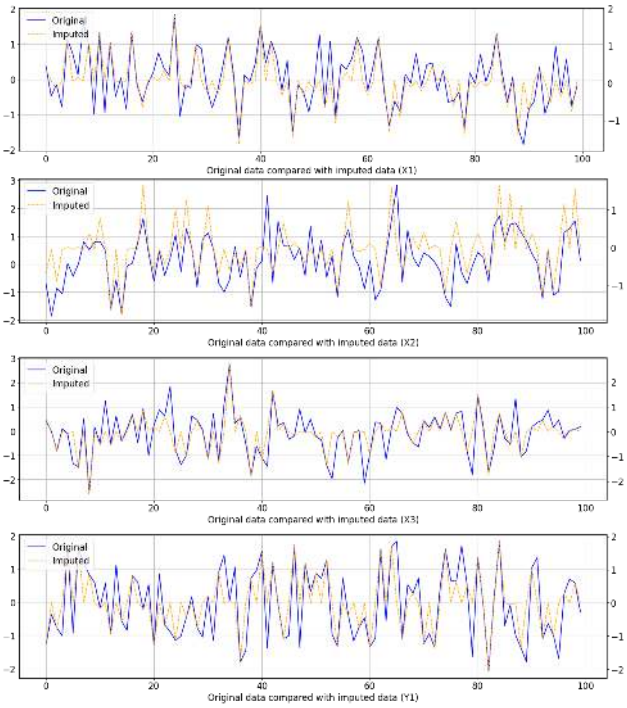


Fig. 6. Graphs of the original (blue) and generated (orange) data for variables (from top to bottom) X_1, X_2, X_3 and Y_1. The generated data for each one of the variables resembles the behavior of of the original values. (Best seen in color.)

(approx. 2%). In this way, the resulting time series from the imputed data may be considered *very* similar to the original time series.

A comparison of the causal structures of the three scenarios was made, that is, the causal structure of the time-series obtained with the imputed data compared with the original causal structure and the one obtained from the subsampled data. This comparison is shown in Figure 7. In Fig. 7c it can be seen that the causal structure obtained from the subsampled data has a causal link from X_1 to X_2 at the incorrect time step compared to the original causal structure (Fig. 7a); while the structure from the imputed data, Fig. 7b, has this link at an correct time step. However, the appearance of a causal link from X_1 to Y_1 is appreciated in 7c, although this link has a minimal causal strength. The links from X_3 to Y_1 and from X_3 to X_1 appear at the correct time step in the imputed data (7b). The causal structure of the imputed data time series is very close to the original causal structure even though it is obtain from a subsampled time series.

As previously mentioned, one way to compare the causal structures is through their adjacency matrix. Table 1 represents the causal interactions of the times series of the imputed data at the second time step. The adjacency matrix rep-

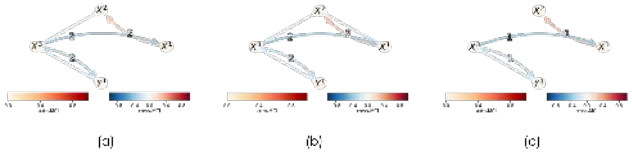


Fig. 7. Experiment 1: causal structure for (a) the original time series, (b) the time series with imputed data and (c) the subsampled time series. Each graph represents the causal structure via a compact representation (*rolled graph*), indicating the time delay (number associated to the link) and strength (color code) of each causal link. (Best seen in color.)

resents the appearance, denoted with *1*, or the absence, denoted with *0*, of the causal links for each of the variables. When evaluating the MAE (time lag 2) for the causal structure of the subsampled time series the resulting value is 0.1875, while the error value for the imputed time series is 0. This reflects how the imputed time series maintained the causal interactions of the variables even after the original time series was affected by subsampling.

5.3 Experiments 2–5: Subsampled Time Series with Different Distributions and Time Steps

Next we present four more challenging scenarios, using the same number of variables but changing the maximum lag in which causal links may appear and considering different noise distributions for the training and test time series. Subsequently, these time series were affected by subsampling. For each of the time series the proposed approach was used, where imputed data complemented the time series, and their causal structure was obtained. This causal structure was compared versus the original causal structure of the time series and the causal structure of the time series affected by subsampling for each scenario. Figure 8 shows the comparison for the four different scenarios: of the original causal structure (a); the causal structure of the time series with imputed data (b); and the causal structure of the time series affected by subsampling (c).

The resulting causal structures where compared using their respective adjacency matrix. As mentioned before, this was used as a way to obtain the error to measure the differences between the causal structure of the original time series

Table 1. Adjacency matrix of the imputed time series at the second time step, Experiment 1.

	X1	X2	X3	Y1
X1	0	0	1	0
X2	1	0	0	0
X3	0	0	0	0
Y1	0	0	1	0

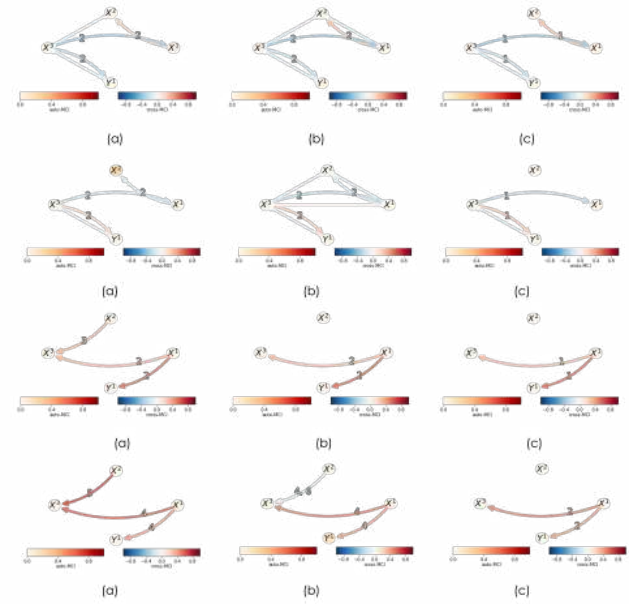


Fig. 8. Causal structure of the times series for experiments 2-5: (a) original causal structures, (b) causal structures of imputed data, and (c) causal structures of subsampled time series.

and the time series with the imputed data. Table 2 summarizes the four experiments. In the first two experiments, the causal structure of the imputed data is the same as the original structure while the one of the subsampled time series presents some errors. However, as we increased the maximum lag the errors in the causal structure of the imputed data started to increase, this is, there were differences with the original causal structure. Although this error increased, the differences of the subsampled causal structure were significantly higher. This highlights the impact of subsampling in the discovery of causal relations in time series. Although the causal structure on scenarios 4 and 5 were not completely recovered, the causal structure of the imputed data is one of the possible graphs in the set $\mathcal{G}$ of consistent causal structures according to the subsampling [8].

Table 2. Summary of the characteristics and results for Experiments 2–5.

	Number of Variables	Maximum Lag	MAE Subsampled Time Series	MAE Imputed Time Series
Scenario 2	4	2	0.1875	0
Scenario 3	4	2	0.156	0
Scenario 4	4	3	0.1041	0.0208
Scenario 5	4	5	0.052	0.0312

5.4 Discussion

The following conclusions can be reached from these experiments: (i) The imputed data is in general very close to the original data. (ii) The causal structure obtained from the imputed data tends to maintain the *strong* causal links in the original model with the correct time scale. In contrast, the structure derived from the subsampled data tends to have causal links at an incorrect time scale. (iii) Some *weak* causal links may be deleted or added in the structure derived from the imputed data. (iv) The monitoring stations scenario shows a possible practical case for the application of the method.

6 Conclusions and Future Work

We have proposed a way to minimize how subsampling affects causal discovery of time series by using a GAN to estimate the missing data. In this way, the imputed time series presents a similar behavior to the original one, so causal discovery algorithms can produce a causal structure closer to the true one. Experimental results with synthetic and real data, considering a known subsampling rate, show promising results.

Future work includes applying the proposed approach to other scenarios, such as neuroimaging where some modalities have a sampling frequency that is known to be lower that the causal mechanisms in the brain. An important assumption is that the subsampling rate is known. A possible solution is to train the GAN model for different subsampling rates (within certain range), and at the testing stage choose the most probable and consistent causal structure.

References

1. Danks, D.: Causal search, causal modeling, and the folk. A companion to experimental philosophy pp. 463–471 (2016)
2. Danks, D., Plis, S.: Learning causal structure from undersampled time series. JMLR: Workshop and Conference Proceedings (2014)
3. Gain, A., Shpitser, I.: Structure learning under missing data. In: International Conference on Probabilistic Graphical Models. pp. 121–132. PMLR (2018)
4. Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., Sebag, M.: Learning functional causal models with generative neural networks. In: Explainable and interpretable models in computer vision and machine learning, pp. 39–80. Springer (2018)
5. Granger, C.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica **37**(3), 424–438 (1969)
6. Grover, A., Zweig, A., Ermon, S.: Graphite: Iterative generative modeling of graphs. In: Inter. Confer. on Machine Learning. pp. 2434–2444. PMLR (2019)
7. Huang, Y., Fu, Z., Franzke, C.L.: Detecting causality from time series in a machine learning framework. Chaos: An Interdisciplinary Journal of Nonlinear Science **30**(6), 063116 (2020)

8. Hyttinen, A., Plis, S., Järvisalo, M., Eberhardt, F., Danks, D.: A constraint optimization approach to causal discovery from subsampled time series data. International Journal of Approximate Reasoning **90**, 208–225 (2017)
9. Lawrence, A., Kaiser, M., Sampaio, R., Sipos, M.: Data generating process to evaluate causal discovery techniques for time series data. Causal Discovery & Causality-Inspired Machine Learning Workshop at NeurIPS (2020)
10. Lütkepohl, H.: Introduction to multiple time series analysis. Springer Science & Business Media (2013)
11. Malinsky, D., Danks, D.: Causal discovery algorithms: A practical guide. Philosophy Compass **13**(1), e12470 (2018)
12. Murphy, K.P.: Dynamic bayesian networks: representation, inference and learning. University of California, Berkeley (2002)
13. Runge, J.: Causal network reconstruction from time series: From theoretical assumptions to practical estimation. Chaos: An Interdisciplinary Journal of Nonlinear Science **28**(7), 075310 (2018)
14. Runge, J., Nowack, P., Kretschmer, M., Flaxman, S., Sejdinovic, D.: Detecting and quantifying causal associations in large nonlinear time series datasets. Science Advances **5**(11) (2019)
15. Solovyeva, K., Danks, D., Abavisani, M., Plis, S.: Causal learning through deliberate undersampling. In: 2nd Conference on Causal Learning and Reasoning (2023)
16. Spirtes, P.: Introduction to causal inference. JMLR **11**(5) (2010)
17. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search, vol. 81 (01 2001). https://doi.org/10.1007/978-1-4612-2748-9
18. Vowels, M.J., Camgoz, N.C., Bowden, R.: D'ya like dags? a survey on structure learning and causal discovery. arXiv preprint arXiv:2103.02582 (2021)
19. Wang, Y., Menkovski, V., Wang, H., Du, X., Pechenizkiy, M.: Causal discovery from incomplete data: a deep learning approach. arXiv:2001.05343 (2020)
20. Yoon, J., Jarrett, D., Van der Schaar, M.: Time-series generative adversarial networks (2019)

Causality Aware Reinforcement Learning in Online Markov Decision Process Settings

Arquímides Méndez-Molina[1[0000−0002−2441−5265]], Eduardo F. Morales[1[0000−0002−7618−8762]], and L. Enrique Sucar[1[0000−0002−3685−5567]]

Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro # 1, Tonantzintla, Puebla, 72840 México
arquimides.mendez@gmail.com, emorales@inaoep.mx, esucar@inaoep.mx

Abstract. Causal Reinforcement Learning (CRL) is an emerging field in which previous work has shown how causality can contribute to mitigate some of the main limitations of Reinforcement Learning (RL), ranging from data-inefficiency, lack of interpretability, and long learning times. However, how to use reinforcement learning to support causal discovery (CD) has so far been less explored. In this article, we introduce CARL, a Causality-Aware Reinforcement Learning framework for simultaneously learning and using causal models to speed-up the police learning in online Markov decision process (MDP) settings. Our method alternates between: (i) (RL for CD), where it promotes the selection of actions to obtain better causal models in fewer episodes than traditional methods of obtaining data in RL, (ii) (RL using CD), where the learned models are used to select actions that speed up the learning of the optimal policy by reducing the number of interactions with the environment, and (iii) (CD), where the system is used to learn causal models. Experiments in the Taxi scenario show that our method achieves better results in policy learning than traditional model-free and model-based algorithms while it is also able to learn the underlying causal models.

Keywords: Causal Reinforcement Learning · Reinforcement Learning · Causal Discovery · Markov Decision Process

1 Introduction

Reinforcement Learning (RL) has emerged as a powerful framework for creating autonomous agents that can learn through interactions with their environment. The ultimate goal for these agents is to determine the optimal policy for each state, i.e., the best action to take at each point in time. This is accomplished by exploring the environment and learning from the rewards associated with the state. RL algorithms have achieved impressive results in several areas: video games [33], robotics [1], and medical care [9].

Causal Discovery (CD) aims for uncovering the causal relationships that exist between a set of variables [27]. Although numerous algorithms have been developed to learn causal relations, this remains a challenging task, especially in

real world scenarios. A limitation of causal discovery is the need for interventions on model variables to ensure a unique model. Nonetheless, once the causal model is established, it enables intelligent systems to predict the effects of interventions, improving planning and enabling counterfactual predictions.

Reinforcement Learning (RL) and Causal Discovery (CD) have traditionally been treated as separate areas, this trend has recently shifted, leading to the emergence of a new area named Causal Reinforcement Learning (CRL). The works in CRL can be divided into two main groups depending if the causal knowledge is given *a priori* or if it has to be learned. The latter is a more challenging task, especially in online Markov Decision Process (MDP) settings where the agent does not know anything about the environment in advance. However, two advantages to consider are the data's temporal order and its interventional nature. In this paper, it is shown how we can provide an intelligent agent with the ability to simultaneously learn and use better causal models to speed-up the learning time in online MPD settings. By "better" models we mean that the structure of the causal models will be closer to the actual one, compared to the traditional RL data collection process. By "speed-up" we mean that a nearly-optimal policy can be obtained in fewer episodes than traditional RL methods.

The main contributions of this work are:

1. A new combination algorithm for an integration between causal discovery and reinforcement learning in MDP settings.
2. A causality-aware action selection algorithm (RL for CD).
3. A transfer learning capability of the learned causal models to more complex tasks.

2 Preliminaries

2.1 Reinforcement Learning

Reinforcement learning is an interactive learning paradigm where an agent learns the optimal actions to take in a given situation, with the objective of maximizing its total reward over the long term. A fundamental assumption of RL is that the environment has the Markov property (future states only depend on the current state). An RL task that satisfies the Markov property is known as a Markov decision process (MDP).

Markov decision processes: A Markov Decision Process (MDP) is represented by the tuple $M = \langle S, A, T, R \rangle$, where S is the set of states, $A(s) \in A$ is a set of possible actions on each state $s \in S$, T is the transition function $T : S \times A \times S \to [0, 1]$, and R is the reward function $R : S \times A \times S \to \mathbb{R}$. A transition from state s to state s' caused by taking action $a \in A(s)$ occurs with probability $P(s'|a, s)$ and receives a reward $R(s, a, s')$. A policy $\pi : S \to A$ for M specifies which action $a \in A(s)$ to execute when an agent is in state $s \in S$, i.e., $\pi(s) = a$. To find a solution for a given MDP is to identify a policy that maximizes the long-term expected cumulative sum of rewards. The action-value function for policy π, denoted by $Q^\pi(a, s)$, is defined [32] as :

$$Q^{\pi}(s,a) = E_{\pi}\left(\sum_{k=0}^{\infty}\gamma^{k}R_{t+k+1}|S_t = s, A_t = a\right). \quad (1)$$

It represents the expected return starting form s, taking the action a and thereafter following policy π. γ is a value between 0 and 1 that indicates how much weight should be given to future rewards when calculating the overall expected reward for taking an action in a particular state.

We can divide the algorithms to solve an MDP and obtain the optimal policy into:

Model-free RL: In model-free RL, the optimal policy is estimated without relying on or estimating the dynamics of the environment, instead, the algorithm directly estimates the value function or the policy from the agent's interaction with the environment.

Model-based RL: In contrast, model-based RL uses the transition and reward functions to estimate the optimal policy.

2.2 Causality

Let X and Y be two random variables representing the cause and effect, respectively. According to Pearl's causal inference framework [27], causality is defined as the relationship between X and Y such that:

- X is a necessary cause of Y: the occurrence of X is necessary for Y to occur.
- X is a sufficient cause of Y: the occurrence of X alone is enough to bring about Y.
- There exists no other variable Z, such that Z is a common cause of both X and Y, and there is no direct causal path from Z to Y that bypasses X.

A causal model can represent causal relations between a set of variables. There are several alternative ways to represent a causal model, in this work we represent causal models as directed acyclic graphs (DAGs). In the DAG representation, each node is a variable, and each directed edge represents a direct causal relationship between the two variables it connects. In the context of MDPs we represent causality through two-slice causal dynamic Bayesian networks.

Two-slice causal dynamic Bayesian networks: A two-slice causal dynamic Bayesian network (CDBN) is a probabilistic graphical model that represents the causal relationships between random variables over two consecutive time steps. Let $\mathbf{X}^{(t)} = X_1^{(t)}, X_2^{(t)}, \ldots, X_n^{(t)}$ denote the set of random variables at time t, and let $\mathbf{X}^{(t+1)} = X_1^{(t+1)}, X_2^{(t+1)}, \ldots, X_n^{(t+1)}$ denote the set of random variables at time $t+1$. A two-slice CDBN is represented by the graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \mathbf{V}^{(t)} \cup \mathbf{V}^{(t+1)}$ is the set of nodes and $\mathbf{E}$ is the set of directed edges between the nodes.

The joint probability distribution over $\mathbf{X}^{(t)}$ and $\mathbf{X}^{(t+1)}$ is factorized according to the two-slice DAG as follows:

$$P(\mathbf{X}^{(t)}, \mathbf{X}^{(t+1)}) = \prod_{i=1}^{n} P(X_i^{(t+1)} | X_i^{(t)}, \mathrm{Pa}(X_i^{(t+1)})) P(X_i^{(t)} | \mathrm{Pa}(X_i^{(t)}))$$

where $\mathrm{Pa}(X_i^{(t+1)})$ denotes the set of parent nodes of $X_i^{(t+1)}$ in $\mathcal{G}$, and $\mathrm{Pa}(X_i^{(t)})$ denotes the set of parent nodes of $X_i^{(t)}$ in the slice of $\mathcal{G}$ at time t.

3 Related Work

In recent years, research work on the relationship between RL and CD have emerged, including the first survey [36] on the area named Causal Reinforcement Learning (CRL). The existing works can be divided into two groups depending on whether the causal information is given or learned.

CRL given causal models as side information: These works assume that the causal information is known or given *a priori* in explicit or implicit way from experts. The causal information is used for different purposes: To deal with latent confounders in different settings like Multi-Armed Bandit (MAB) [3,8,30,35,14,20], MDP [15,12,37], and off-police evaluation (OPE) [4], to mitigate heterogeneity and data scarcity [17], or to derive causal explanations about the behavior of model-free RL agents [21]. More closely related with our work, in [7] and [22] it is shown how it is possible to speed-up police learning in goal-conditioned MDP settings via causal knowledge. In [19] the authors introduce Causal Markov Decision Processes. Rather than proposing to have one causal graph for each action like we do, they suggest having two causal graphs (the reward and the transition graphs) for every state, which structure and part of the parameters are given to the learner. The authors suggest as a promising idea to develop a causal algorithm that can learn the causal information and the optimal policy simultaneously and achieve lower regret than standard non-causal RL algorithms.

CRL with Unknown Causal Information: The task here becomes more challenging because the method first needs to learn the causal information and then use it for a given task. To discover the causal structure, different techniques are proposed, such as constraint-based and score-based algorithms, interventions, and deep neural networks. In the Bandits settings, in [18] it is proposed the first causal bandit algorithm with better regret guarantees than standard multi-arm bandit (MAB) methods without knowing the causal structure. In Hierarchical Reinforcement Learning, a framework named CDHRL that leverages the advantages from causality is presented in [28]. Another area more closely related to our work is transfer learning. In [26] it is presented a method for causal induction using visual observations for goal directed tasks which achieves generalization abilities on different tasks in novel environments. Unlike our proposal, the proposed method makes the strong assumption that it can access the ground-truth causal relationships while training the causal model. Schema networks [13] is another example of how learning causal relationships and using them to plan can

result in better transfer than model-free policies. The networks have the ability to disentangle multiple causes of events and reason in reverse through causes to accomplish objectives. Although in our work we also explore the possibility of transferring the learned models between similar tasks, our main goal is to learn and use the models simultaneously with the policy for the task to be solved. To the best of our knowledge the only work that attacks that problem so far is [23], setting the foundations for the combination strategy between reinforcement learning and causal discovery in MPD, that we improve and extend in the present work.

4 Causality-Aware Reinforcement Learning in MDP Settings

We present the Causality-Aware Reinforcement Learning (CARL) framework, which simultaneously learns and uses causal models for induction of causality and task policies in online MDP settings, see Figure 1. Guided by the combination algorithm, alternately the agent performs actions in the environment with the purpose to obtain quality data for causal discovery (RL for CD) or it performs causal discovery (CD) or it uses the learned causal models to speed-up the RL process (RL using CD).

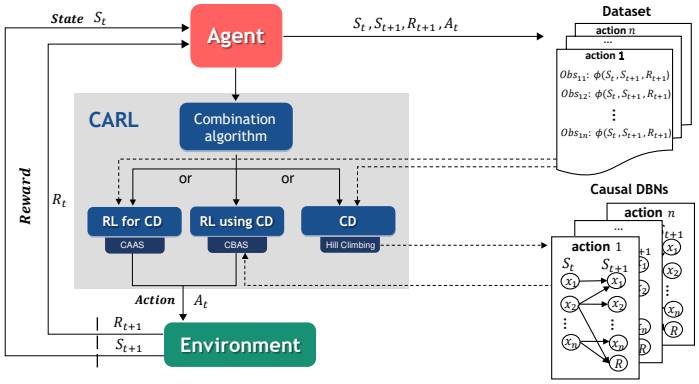


Fig. 1: Overview of the proposed framework. The interaction between the agent and the environment is controlled by the combination algorithm. In the RL for CD stages, the agent uses the causality-aware action selection (CAAS) algorithm to select those actions that give more useful information for causal discovery. In the RL using CD stages, the agent uses the causal-based action selection (CBAS) algorithm to filter among the possible actions taking into account the expected immediate reward. In CD stages the agent employs a score-based hill-climbing algorithm over the collected dataset to discover the causal DBNs.

4.1 Assumptions and limitations

There are some important assumptions in our method: (i) The action space must be discrete, and (ii) the learned causal model corresponding to each action $\mathcal{G}_a$ are causal graphs where the Markov, minimality and faithfulness conditions (described in [2]) are assumed to be satisfied. Finally, we propose to use a mapping $\phi : \mathcal{S} \rightarrow \mathcal{X}$ between the original space of the state variables $s_i \in S$ and the causal model variables $x_i \in X$. Specifically we propose that the causal model variables should be relational variables as used in relational reinforcement learning [6]. The mapping from S to X may be trivial, when all state variables in the task description are relational or it can involve for instance deep neural networks when states are represented by images. This mapping is not a mandatory condition for our method to work, however we must make sure that if we use the original state variables they comply constraint (ii).

4.2 Combination algorithm

In Algorithm 1 the synergistic combination between RL and CD is detailed. Our agent, which is trying to learn the optimal policy for the given task but also to discover the underlying causal structure, interacts with the environment according to the combination strategy $C = [stg_1, \ldots, stg_N]$ where each element $stg_i \in \{\text{RL for CD, CD, RL using CD}\}$. That strategy tells the agent what to execute for the following T episodes [1]. Each subset of stages in the strategy can be executed one or repeated several times (indicated by the symbol $*$). We propose to use the following strategy, $C_1 = [$ RL for CD, CD, RL using CD, CD$]^*$. In the case of ($stg_i =$ CD) the agent performs causal discovery using the interventional data collected for each action up to the current moment. A causal model is learned for each of the agent's actions relating the relational state variables at time t and relational state variables and reward at time $t + 1$. In the other cases, the agent acts similarly to a classical temporal difference reinforcement learning algorithm with the difference that, in the exploration stages, instead of taking random actions with a probability of ϵ, it takes the corresponding action according to the stage indicated in C. If $stg_i =$ RL for CD the action is selected with focus on further causal discovery. That means to select on each state the action that gives more information about the causal structure. If the $stg_i =$ RL using CD the action is selected with focus on police learning. That means to select the action that has a higher probability to give positive reward or an action to avoid negative reward, and in this way speed-up the convergence of the value function. The agent acts on the environment and it updates the value function using temporal difference and collects interventional data (s_t, s_{t+1}, r_{t+1}) that is converted to the relational representation and added to the corresponding action dataset. To secure the convergence of the algorithm to the optimal policy, there is a probability of $1 - \epsilon$ that our agent ignores the action indicated by

[1] In the case of $stg_i =$ CD the T parameter is ignored because that process do not required any interaction with the environment.

Algorithms 2 and 3 and selects the action which maximizes the expected reward in the next state. The process stops when the maximum number of episodes is reached.

Algorithm 1: Simultaneous RL + CD

input : The combination strategy $C = [stg_1, \ldots, stg_N]$ where $stg_i \in \{\text{RL for CD, CD, RL using CD}\}$, the set of actions A, the exploration factor start and minimum values ϵ: $\epsilon_{\max}$ and $\epsilon_{\min}$, the learning rate $\alpha \in (0, 1]$, the discount factor $\gamma \in (0, 1]$, the number of episodes of a given stage T, the maximum number of steps per episode H, the min frequency value f used to select select and action for causal discovery, the confidence threshold value th used to select an action suggested by a causal model, the maximum number of episodes E

output: A value function Q, a set of causal models G

Initialize $Q(s, a)$ in zeros
$\mathcal{D} \leftarrow \emptyset$ ▷ The empty dataset to collect (s_t, s_{t+1}, r_{t+1}) observations for each action $a \in A$)
$\mathcal{G} \leftarrow \emptyset$ ▷ The empty set of causal models, one for each action $a \in A$)
$episode \leftarrow 0$
while $episode < E$ **do**
  for $i \leftarrow 0$ **to** $|C|$ **do**
    $stg \leftarrow C[i]$; **if** *stg = (CD)* **then**
      foreach $a \in A$ **do**
        G [a] ← causal_discovery(D [a]) ▷ See section 4.4
      end
    end
    else
      for $stg\_epi \leftarrow 0$ **to** T **do**
        Randomly set initial state s_t
        $\epsilon = \epsilon - \frac{episode}{E} \times (\epsilon_{\max} - \epsilon_{\min})$
        for $t \leftarrow 0$ **to** H **do**
          if *stg = (RL for CD)* **then**
            $a_t \leftarrow$ Algorithm 2(s_t, A, ϵ, f, $\mathcal{D}$)
          end
          if *stg = (RL using CD)* **then**
            $a_t \leftarrow$ Algorithm 3($\mathcal{G}$, A, s_t, ϵ, th)
          end
          Choose action a_t and observe s_{t+1}, r_{t+1}
          $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$
          $x_t \leftarrow \phi(s_t)$ ▷ Convert variables in s_t to relational space
          $x_{t+1} \leftarrow \phi(s_{t+1})$ ▷ Convert variables in s_{t+1} to relational space
          $x_r \leftarrow \phi(r_{t+1})$ ▷ Convert the reward variable r_{t+1} to relational space
          add the variables (x_t, x_{t+1}, x_r) into D $[a_t]$
          $s_t \leftarrow s_{t+1}$
          if s_t *is terminal* **then**
            break
          end
        end
        $episode \leftarrow episode + 1$
      end
    end
  end
end
return Q, G

4.3 Reinforcement Learning for Causal Discovery (RL for CD)

According to [28] causality is discovered within adjacent steps. The effects of a given action in the next state and the reward are completely determined by the current state of the world, so we learn a "two-slice" causal DBN for each one of the agent's actions. On each causal DBN we have one set of variables representing the state of the world prior to the action (X_t), and the same set of relational variables representing the relational state of the world after the action plus the reward variable (X_{t+1}, r), and directed arcs representing causal relations between the variables from slice t to slice $t+1$.

In this paper, we introduce a new stage that we have called (RL for CD) in which the agent is concerned with trying to select actions that favor causal discovery rather than learning the optimal policy. Although we cannot directly intervene in the state variables, we can intervene in the action variable. This combined with our proposal to learn a causal model per action results in a simple but effective causality-aware action selection algorithm, that is used in every episode during the (RL for CD) stages, so that we can learn causal models with fewer interactions than if we were to use traditional RL data collection methods. The idea is presented in Algorithm 2. Basically, the agent keeps a record of the number of times it has performed each action in each relational state. Then, if it is time to explore, instead of directly performing a random action like a traditional RL methods does, the agent first tries to select the less explored actions in the given relational state. In this way, the datasets of observations (one for each action) remain balanced and we will have a better chance that the models will be correct when it is time to perform causal discovery. When all actions in a given state are executed at least (f) times we start to select random actions again. This limit ensures that we explore different options sufficiently before returning to random exploration. Because the (RL for CD) stage is performed several times during training and we do not want to neglect the learning of the optimal policy, there is always a probability of $(1-\epsilon)$ that the agent will perform the optimal action according to the value function Q.

4.4 Causal Discovery (CD)

Each time the combination algorithm tells the agent that causal discovery (CD) is to be done, we will have $|A|$ data sets (one for each action) with $n \times 2 + 1$ variables (n for relational state variables at time t, n for relational state variables at time $t+1$ and one for the reward at time $t+1$). The number of observations varies depending on the number of times the agent has executed the action thus far. To identify the causal models, we can take advantage of several constraints: (i) No time $(t+1)$ variable can cause a variable at time (t) and (ii) variables at the same time point cannot cause one another. Given the structural constraints of the causal models, the causal discovery process becomes easier than traditional causal discovery because learning the skeleton of G is equivalent to learning its full structure [25]. However, to guarantee that the learned skeleton fully corresponds to the ground truth causal model we need a generative mechanism that

Algorithm 2: Causality-aware action selection (CAAS)

Input : A state s sensed by the agent, the value function Q, the exploration factor ϵ, the min frequency value f, the dataset of interventional data $\mathcal{D}$
Output: An action a.

Choose a random number $r \in [0, 1]$
if $r > \epsilon$ **then**
  $index \leftarrow$ random choice from $i \in [1, |Q(s)|] \mid Q_s[i] = \texttt{max}(Q(s))$ ▷ Exploit, takes the best action according the value function Q
  return $A[index]$
end
else
  ▷ Explore, takes the less selected action in the corresponding relational state so far
  $\mathbf{x} \leftarrow \phi(s)$ ▷ Convert s to relational state $\mathbf{x} = [x_1, \ldots, x_N]$ where $x_i \in \mathcal{X}$
  $Z_x \leftarrow$ vector calculated using $\mathcal{D}$, indicating the number of times action a_i has been done in the relational state x for actions performed less than f times
  if $|Z_x| > 0$ **then**
    $index \leftarrow$ random choice from $i \in [1, |Z_x|] \mid Z_x[i] = \min(Z_x)$
  end
  else
    $index \leftarrow$ random choice from $|A|$
  end
end
return $A[index]$

allows the agent to set each relational state variables $x_i \in X_t$ at time t to all its possible values before taking an action and sampling the transition and reward. In our setting, we do not have such mechanism but instead our agent sequentially interacts with the environment according to the corresponding stage of the combination algorithm visiting different states along the way. For that reason we can not guarantee that the learned models are complete. However, it has been shown in [23] and [7] that partially correct causal models are enough to speed up policy learning. In our experiments we use the score-based structure learning algorithm Hill Climbing (HC) implementation from the BNlearn package [29]. Initially, the discovered models may not be perfect, but with more data, they improve over time.

4.5 Reinforcement Learning using Causal Models (RL using CD)

Once the agent has learned about the implicit causality in the environment, it uses that knowledge to speed-up policy learning for the given task. In the remaining T episodes, the agent is going to use the set of learned causal models for action selection. It has been shown in previous works [22] that a causal model relating state, action and reward variables can be used to accelerate the police learning process in MDPs by guiding the action selection process, even if those models are partially correct [7]. In this work, we extend the causal selection algorithm presented in [23] to take into account, not only the actions to obtain an immediate positive reward (+), but also to filter those actions that can lead to an undesirable large negative reward (−).

In Algorithm 3 we can see the pseudo code of the action selection strategy. As in Algorithm 2, there is a probability of $(1-\epsilon)$ that the agent selects the best

Algorithm 3: Causal-based action selection (CBAS)

Input : A state s sense by the agent, a set of actions A, the value function Q, the set of causal DBN models G, one for each action, the exploration factor ϵ, the confidence threshold value th used to select an action suggested by a causal model

Output: An action a.

Choose a random number $r \in [0, 1]$
if $r > \epsilon$ **then**
  $index \leftarrow$ random choice from $i \in [1, |Q(s)|] \mid Q_s[i] = \texttt{max}(Q(s))$ ▷ Exploit, takes the best action according the value function Q
  return $A[index]$
end
else
  $possible\_actions \leftarrow A$ ▷ Initially, all actions are possible
  $\mathbf{x} \leftarrow \phi(s)$ ▷ Convert s to relational state $\mathbf{x} = [x_1, \ldots, x_N]$ where $x_i \in \mathcal{X}$
  foreach $a \in A$ **do**
    $p \leftarrow P(reward|x, G[a])$
    if $p[+] > th$ **then**
      return a
    end
    if $p[-] > th$ **then**
      $possible\_actions \leftarrow possible\_actions \setminus a$
    end
  end
end
return *random choice from possible_ actions*

action for policy learning according the value function, otherwise with a probability of ϵ, for each of the possible actions a, the agent calculates the probability distribution p for the reward variable given a representation of the state x and the causal model learned for that action $C[a]$. Inference is performed by taking into account only the x_i variables that are parents of r in the corresponding causal model. If there is a probability greater than a threshold value (th) of obtaining immediate positive reward ($p[+]$), action a is selected, otherwise, if there is a probability greater than th of obtaining high negative reward ($p[-]$), action a is discarded from the set of possible actions in the current state. The threshold value th can be seen as the level of confidence the agent has in the causal models it uses. In our experiments we used a fixed value of $th = 0.7$. In summary, our action selection algorithm acts like a filter in the action space compared to other epsilon-greedy strategies that select a random action in a given state. By filtering the bad actions or taking the good ones according to the learned model so far, it is expected that our agent will reach promising states in fewer episodes and learn the optimal policy faster.

5 Experimental Results

To test our method, we use the OpenAI Gym [2] implementation of the Taxi task proposed by [5]. Figure 2 illustrates the problem. In this 5×5 grid world, at each episode there is a passenger randomly placed at one of the four possible

[2] `https://gymnasium.farama.org/environments/toy_text/taxi/`

locations which wants to be transported to one of the other three locations. A taxi must pick up the passenger and drop him off at the destination. In the original formulation, there are 500 possible states using four variables: 5 options for the taxi row (tr), 5 options for the taxi column (tc), 5 passenger's locations (pl) (the predefined four places plus one for being inside the cab), and 4 passenger's destinations (pd). There are six actions: four actions that move the taxi one square on the desired direction (North, South, East or West), a Pick-up action and a Drop-off action. There is a reward of $+20$ for successful drop-off of the passenger at the destination, and penalties of -10 each time the taxi crashes with a wall or -1 in other cases.

In our method, every state is converted to a relational representation using a mapping function $\phi : \mathcal{S} = \{tr,\ tc,\ pl,\ pd\} \rightarrow \mathcal{X} = \{l,\ wp,\ nw\}$. The conversion to relational variables is as follows: (i) taxi location $l \in 0, 1, 2$ indicating when the taxi is on the road (0), on the origin of the passenger (1) or at the destination of the passenger (2), (ii) taxi with passenger $wp \in 0, 1$ indicating when the passenger is in the taxi (1) or not (0), and (iii) nearest wall $nw \in 0, 1, 2, ..., 14$ indicating the positions of the walls at adjacent squares relative to the taxi position. To obtain a number between 0 and 14 we use the decimal number corresponding to the binary representation of a four bits vector (S, N, E, W) indicating the positions of the walls. For example, in Figure 2 the nearest adjacent wall relative to the taxi position is only at North, so the associated bit vector is $(0, 1, 0, 0)$ and the corresponding decimal number is 4. Note that the vector $(1, 1, 1, 1) = 15$ indicating that the taxi is surrounded by walls is not possible. The full relational state corresponding to Figure 2 is $x = (l = 0, wp = 0, nw = 4)$. With this mapping, we reduce the original space from 500 to 90 and also include information that may be relevant to the agent such as the position of the nearest walls.

Fig. 2: Sketch of the Taxi task v3.0 environment. The grid world size is 5 x 5. There are four possible locations for the passenger pick-up and drop-off marked using colored squares.

We perform a set of experiments [3] to measure the advantages of our proposed framework against model-free (Q-Learning [34]) and model-based (Dyna-Q [31]) algorithms. Specifically we want to answer the following questions:

- Can our RL for CD stage contribute to learning better causal models in fewer episodes than using a traditional RL exploration scheme?
- Can our combination algorithm speed-up the learning time (in terms of episodes) and also obtain the underlying causal models?

5.1 Causal Discovery using the new RL for CD stage

The first set of experiments was designed to measure the effectiveness of the novel action selection strategy to improve causal discovery (see section 4.3) using the CARL framework. With this in mind, we compared an agent performing exploration using the traditional epsilon-greedy exploration scheme (RL agent) against an agent who is concerned with taking actions that favor the discovery of the causal models using our proposed RL for CD stage in the taxi task. The RL agent uses the combination strategy C_2 = [**RL**, CD, RL using CD, CD]* proposed in [23]. In this strategy the agent learns and uses causal models, but in the RL stage it does not select actions for causal discovery purposes, it uses epsilon-greedy instead. The RL for CD agent, on the other hand, uses our proposed combination strategy C_1 = [**RL for CD**, CD, RL using CD, CD]* where the (RL) stage is replaced by the (RL for CD) stage and the actions are selected based on Algorithm 2.

In CARL, an important parameter is T. T indicates the duration in episodes of each of the stages, which directly influences the amount of data collected by the agent. The larger T is, the more data will be available at the time of causal discovery. This is why we tested with different values of $T \in 10, 20, 50, 100$ to measure the effect of using RL for CD stage vs RL independently of T. In all experiments, the exploration rate (ϵ) is decreased from 1.0 to 0.1 uniformly over 1000 episodes. Our hypothesis was that the agent using the RL for CD stage will discover *better* models in fewer episodes than the traditional RL agent independent of T. For each T value, we run 10 trials[4] for each agent and we report the structural hamming distance (SHD) among the discovered causal model and the ground truth (the lower the value is better). As we have a causal model for each action, the reported SHD in a given episode is the sum of the SHD among all discovered models for each agent. For experimental purposes, the accuracy of the causal discovery process is determined by comparing the discovered graph against the ground truth, using the structural Hamming distance. This distance represents the minimum number of edge changes required (insertions, deletions, and modifications) to transform one model into another. A lower Hamming distance indicates greater accuracy.

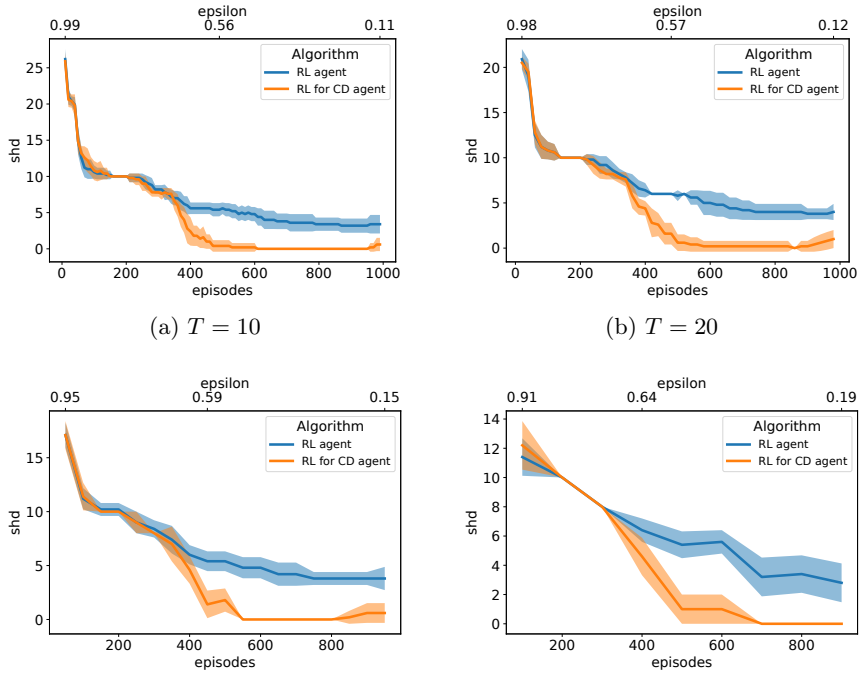
[3] The full code and instructions can be consulted in the following Github repository (`https://github.com/arquimides/causal_rl`)

[4] On each trial both agents share the set of random generated initial states. The set is re-generated on each trial.

In Figure 3 we can see the results of these experiments with different duration for each stage, T. In all cases that the agent with the RL for CD stage manages to discover better models in fewer episodes than the RL agent. In all the scenarios, in the first episodes both agents obtain similar results as the exploration value is high; however, from that point onwards, our agent starts to obtain better results, even managing to discover the complete models ($SHD = 0$) in approximately 600 episodes or less. For smaller values of T (subplot (a)) it can be seen that at the time of the first discovery (10 episodes) the structural hamming distance is larger ($SHD = 25$) compared to the $T = 100$ scenario (subplot (d)) where the structural hamming distance is 14. That is because for larger T we have more data and probably more useful information at the moment of causal discovery.

5.2 Causality-aware RL against model-free and model-based RL

Once the advantages of our new (RL for CD) stage for causal discovery had been determined, the next step was to evaluate the CARL framework against a model-free (Q-Learning [34]) and model-based (Dyna-Q [31]) algorithms. Q-learning does not require any prior knowledge or model of the environment,



(a) $T = 10$

(b) $T = 20$

(c) $T = 50$

(d) $T = 100$

Fig. 3: RL vs RL for CD using the CARL framework for different T.

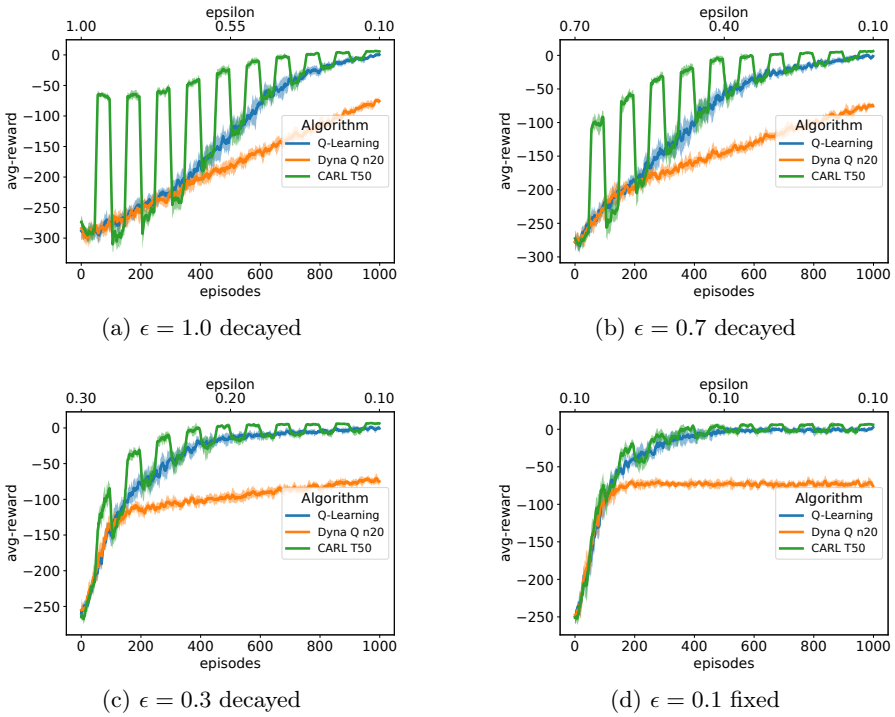
and it directly estimates the Q-values through the real experiences gathered by the agent. On the other hand, Dyna-Q is a model-based reinforcement learning algorithm that extends Q-learning by incorporating an estimated model of the environment that it uses to plan ahead (n) steps in imagination and update the Q-values accordingly, while also using real experiences to improve the model's accuracy. Our hypothesis was that CARL, using the combination strategy, $C_1 =$ [RL for CD, CD, RL using CD, CD]* is able to learn the optimal policy faster than both algorithms but it is also able to discover the underlying causal models in the process.

The task to be solved is the taxi problem in the basic scenario (Fig 2). To make a fair comparison, we tested with different values of the exploration rate (epsilon) from largest to smaller, since this parameter has a significant influence on the performance of the RL algorithms. High values of exploration favor the quality of the learned models to the detriment of policy learning and vice versa. A widely applied strategy [10,11,24] is to eventually decrease the exploration factor as a function of the number of episodes. This decrease is necessary to ensure that eventually the learning algorithm will converge. We test with ϵ starting at different initial values $(1.0, 0.7, 0.3, 0.1)$ and uniformly decreasing the value over 1000 episodes. In the scenario of $\epsilon = 0.1$ that value is small enough so we do not decrease the value among the episodes. This time we use $T = 50$ but we performed experiments for different values of T where we could appreciate that regardless of this value our algorithm behaves similar. We run 10 trials of each experiment for each algorithm and report the average reward and the standard deviation (graphically represented by the width of the shaded region around the corresponding curve) among the episodes. On each episode, the agents starts at a random state (the three agents share the initial state on all episodes) and stops when it reaches the goal state (the passenger is dropped-off at the destination position) or when a maximum number of steps (100) is reached. We use a learning rate of $\alpha = 1.0$ and a discount factor of $\gamma = 0.95$.

In Figures 4 and 5 we show the experimental results. The subplots from (a) to (d) in Figure 4 show the average reward for each algorithm at different starting epsilon values while the subplots from (a) to (d) in Figure 5 show the corresponding causal discovery effectiveness, measured as the structural hamming distance (SHD) against the ground truth causal models for the taxi task. Remember that our agent has no access to such ground truth, however we use it to evaluate the causal discovery. The first thing to notice is the fact that both our algorithm and the model-free RL algorithm perform much better than the model-based algorithm in all scenarios. While Dyna-Q is often more efficient than Q-Learning when it comes to trajectory planning problems, it is also recognized for its low search efficiency, slow convergence speed, and inability to converge in complex dynamic environments. These issues arise due to the sparse reward function and the vast search space involved [16]. Although the taxi problem was not considered *a priori* as such a complex scenario, it is true that the positive reward is quite sparse, since it is only obtained when the agent succeeds in re-

leasing the passenger at the destination. We think that this factor was the one that negatively influenced the Dyna-Q algorithm.

With respect to task learning (measured as the average reward among trials at each episode) we see interesting results. In the episodes where it is performing exploration to improve the model (RL for CD) the reward is similar or slightly below Q-Learning, however, CARL obtains much better rewards than Q-Learning in the episodes where it is using the causal model (Rl using CD). The difference is proportional to the exploration factor. Our method benefits notably when the exploration is high, while when the exploration is low it behaves quite similar to the model-free agent. This makes sense since both the (RL for CD) and (RL using CD) stages are performed only on exploration time (large epsilon). On the other hand, we can see how something similar happens with respect to the learning of causal models. In the case of much exploration Figure 5(subplot (a)) our agent manages to completely discover the models (SHD = 0) in about 600 episodes, while for the case of little exploration (subplot (d)) the minimum value of SHD = 6, indicating that the discovered causal models are still incomplete.



(a) $\epsilon = 1.0$ decayed

(b) $\epsilon = 0.7$ decayed

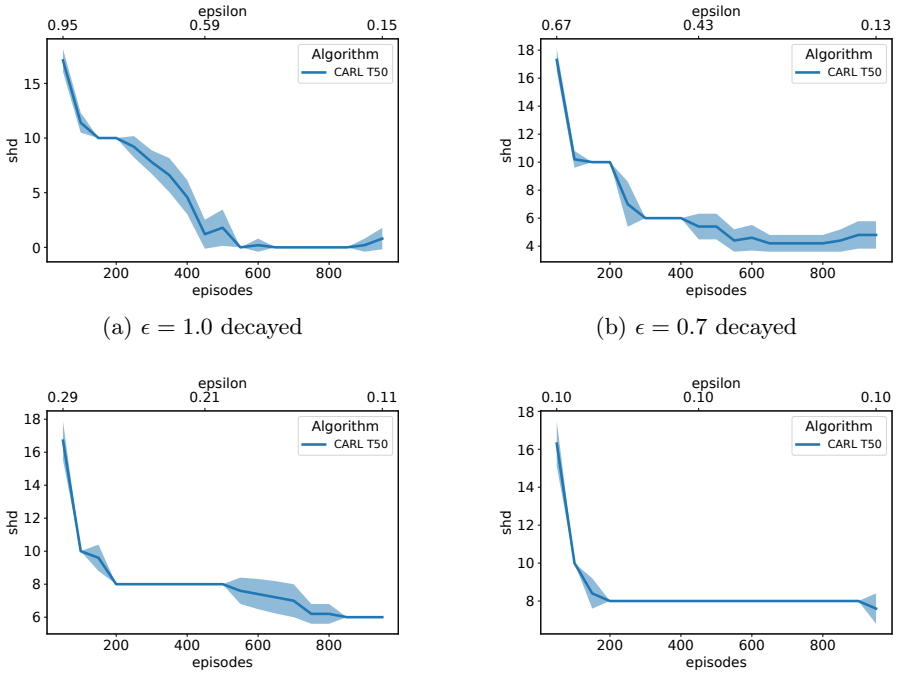
(c) $\epsilon = 0.3$ decayed

(d) $\epsilon = 0.1$ fixed

Fig. 4: CARL vs Q-Learning and Dyna-Q with n=20 at different starting exploration levels.

Discover once, use forever. An interesting observation in the above experiments is that our algorithm seems to obtain better results when it starts using the discovered causal models, even if these are incomplete. In the next set of experiments we want to test what would happen if we only perform (RL for CD) stage once and subsequently learn and use the discovered models, this time without worrying about further improving the models. The corresponding combination strategy will be C_3 = [RL for CD, (CD, RL using CD)*] and all the remaining parameters are equal to the previous experiment.

In Figure 6 we depict the results of using this strategy. As can be seen, regardless of the exploration factor and even though the models used are not complete when they are first used, our method achieves significant momentum when starting to use the discovered models, which it maintains throughout the episodes, being able to learn the optimal policy in far fewer episodes than Q-Learning and Dyna-Q. The disadvantage of this strategy is that the causal models may not be fully discovered as we can see in Figure 7. Even in the high exploration scenario (subplot a) the best SHD value obtained is 8. This is because the (RL using CD) stage significantly impairs the action exploration promoted by the (RL for CD) stage.



(a) $\epsilon = 1.0$ decayed

(b) $\epsilon = 0.7$ decayed

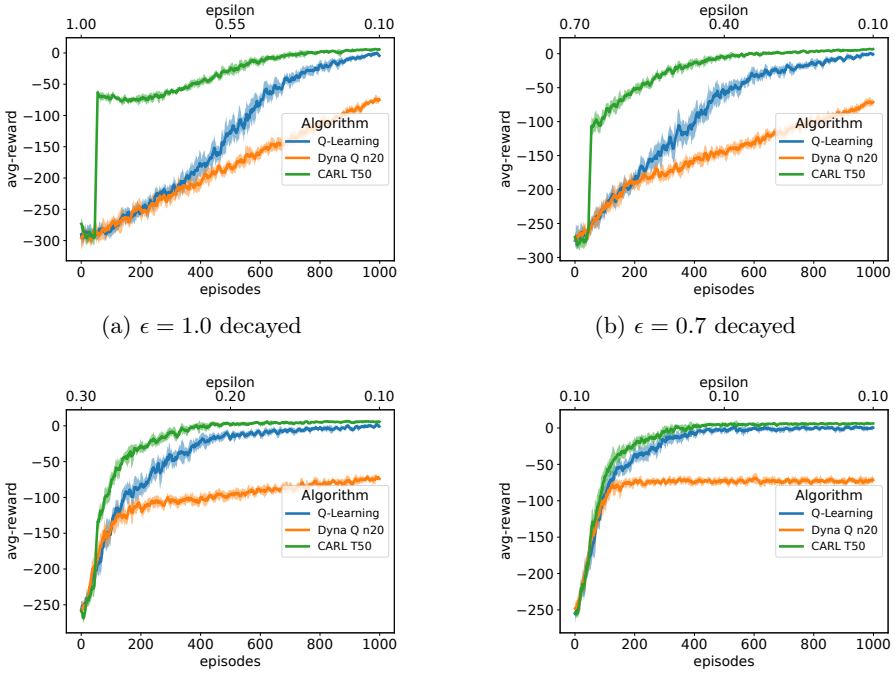
(c) $\epsilon = 0.3$ decayed

(d) $\epsilon = 0.1$ fixed

Fig. 5: Causal Discovery results of CARL at different exploration levels.

5.3 Discussion

Based on the results, our method demonstrates improved performance when initialized with a high exploration rate; it can learn the optimal policy for the given task in far fewer episodes than traditional model-free and model-based agents while also acquiring the causal models. This difference is reduced as the level of exploration decreases, since the agent tends to repeat the action it considers the best in a state given the value function, which decreases the variability in the data that is necessary to learn the causal model. This well-known dilemma between exploration and exploitation affects the performance of our method, however, even in the worst case ($\epsilon = 0.1$) our method behaves slightly better than the model-free agent, and it is also able to discover part of the causal models. On the other hand, the strategy of discovering the model with the RL for CD data only once and then using the learned model until the end, proved to be a very good alternative for scenarios in which policy learning is preferred and we are not so concerned about learning the full causal models.



(a) $\epsilon = 1.0$ decayed

(b) $\epsilon = 0.7$ decayed

(c) $\epsilon = 0.3$ decayed

(d) $\epsilon = 0.1$ fixed

Fig. 6: CARL vs Q-Learning and Dyna-Q at different exploration levels. In this experiment we discover the causal models once and then we use them for the rest of the episodes.

6 Conclusions

In this work, we presented a combination algorithm in which an agent alternately explores the environment to gather information about the underlying causal model (RL for CD), learns a two-slice Dynamic Bayesian Causal Model for each action (CD), and uses those models to improve action selection while learning the policy for the given task (RL using CD). We tested the method in various experimental settings, including modifying the complexity of the environment, the rate of exploration, the number of episodes to alternate between stages, and the sequence of stages. Based on our experimental results, we arrived to the following conclusions:

(i) The RL for CD step allows our agent to learn better models in fewer episodes than if we only used the collected data from epsilon-greedy RL interactions. (ii) A score-based algorithm can be used to learn the graphical structure and parameters of Dynamic Bayesian networks to discover the underlying causal models in RL settings with satisfactory results. (iii) Our proposed method

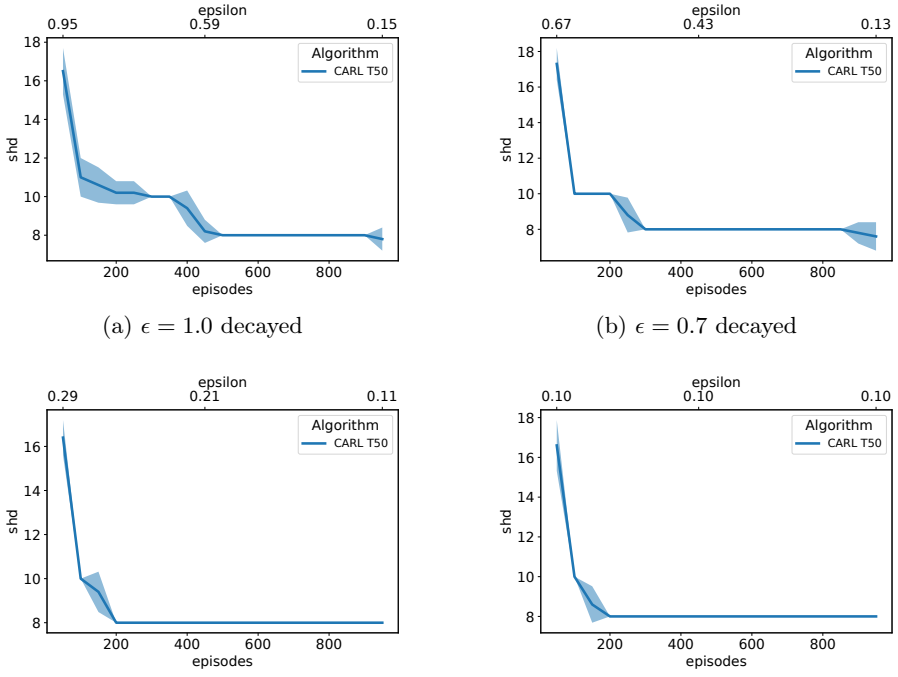


Fig. 7: Causal Discovery results of CARL at different exploration levels. In this experiment the stage of (RL for CD) is performed only once and then (CD and RL using CD) is repeated for the rest of the episodes.

achieves similar or faster time in policy learning than both model-free and model-based RL methods, while also discovering the underlying causal models in the process. (iv) The trade-off between exploration and exploitation also affects our method. High exploration promotes causal discovery, but hinders policy learning.

Future work includes evaluating the transfer capability in similar tasks, adapting our method for continuous action and state spaces, and performing tests on more complex scenarios.

Acknowledgements

This work was partially supported by CONACYT, Project A1-S-43346 and scholarship 754972 (first author).

References

1. Akkaya, I., Andrychowicz, et al.: Solving rubik's cube with a robot hand. arXiv preprint arXiv:1910.07113 (2019)
2. Arntzenius, F.: Reichenbach's common cause principle (Aug 2010), `https://plato.stanford.edu/entries/physics-Rpcc/`
3. Bareinboim, E., Forney, A., Pearl, J.: Bandits with unobserved confounders: A causal approach. In: Advances in Neural Information Processing Systems. pp. 1342–1350 (2015)
4. Bennett, A., Kallus, N., Li, L., Mousavi, A.: Off-policy evaluation in infinite-horizon reinforcement learning with latent confounders. In: Banerjee, A., Fukumizu, K. (eds.) The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 130, pp. 1999–2007. PMLR (2021), `http://proceedings.mlr.press/v130/bennett21a.html`
5. Dietterich, T.G.: Hierarchical reinforcement learning with the maxq value function decomposition. J. Artif. Int. Res. **13**(1), 227–303 (Nov 2000), `http://dl.acm.org/citation.cfm?id=1622262.1622268`
6. Džeroski, S., De Raedt, L., Driessens, K.: Relational reinforcement learning. Machine learning **43**, 7–52 (2001)
7. Feliciano-Avelino, I., Méndez-Molina, A., Morales, E.F., Sucar, L.E.: Causal based action selection policy for reinforcement learning. In: Mexican International Conference on Artificial Intelligence. pp. 213–227. Springer (2021)
8. Forney, A., Pearl, J., Bareinboim, E.: Counterfactual data-fusion for online reinforcement learners. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 1156–1164. PMLR (2017), `http://proceedings.mlr.press/v70/forney17a.html`
9. Gottesman, O., Johansson, et al.: Evaluating reinforcement learning algorithms in observational health settings. arXiv preprint arXiv:1805.12298 (2018)
10. van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Schuurmans, D., Wellman, M.P. (eds.) Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. pp. 2094–2100. AAAI Press (2016), `http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12389`

11. Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M.G., Silver, D.: Rainbow: Combining improvements in deep reinforcement learning. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 3215–3222. AAAI Press (2018), https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17204
12. Kallus, N., Zhou, A.: Confounding-robust policy evaluation in infinite-horizon reinforcement learning. Advances in Neural Information Processing Systems **33**, 22293–22304 (2020)
13. Kansky, K., Silver, T., et al.: Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. In: Proc of the 34th Int Conf on ML. pp. 1809–1818 (2017)
14. Lattimore, F., Lattimore, T., Reid, M.D.: Causal bandits: Learning good interventions via causal inference. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 1181–1189 (2016), https://proceedings.neurips.cc/paper/2016/hash/b4288d9c0ec0a1841b3b3728321e7088-Abstract.html
15. Li, J., Luo, Y., Zhang, X.: Causal reinforcement learning: An instrumental variable approach. CoRR **abs/2103.04021** (2021), https://arxiv.org/abs/2103.04021
16. Liu, Y., Yan, S., Zhao, Y., Song, C., Li, F.: Improved dyna-q: A reinforcement learning method focused via heuristic graph for agv path planning in dynamic environments. Drones **6**(11), 365 (2022)
17. Lu, C., Huang, B., Wang, K., Hernández-Lobato, J.M., Zhang, K., Schölkopf, B.: Sample-efficient reinforcement learning via counterfactual-based data augmentation. CoRR **abs/2012.09092** (2020), https://arxiv.org/abs/2012.09092
18. Lu, Y., Meisami, A., Tewari, A.: Causal bandits with unknown graph structure. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual. pp. 24817–24828 (2021), https://proceedings.neurips.cc/paper/2021/hash/d010396ca8abf6ead8cacc2c2f2f26c7-Abstract.html
19. Lu, Y., Meisami, A., Tewari, A.: Causal markov decision processes: Learning good interventions efficiently. CoRR **abs/2102.07663** (2021), https://arxiv.org/abs/2102.07663
20. Lu, Y., Meisami, A., Tewari, A., Yan, W.: Regret analysis of bandit problems with causal background knowledge. In: Adams, R.P., Gogate, V. (eds.) Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI 2020, virtual online, August 3-6, 2020. Proceedings of Machine Learning Research, vol. 124, pp. 141–150. AUAI Press (2020), http://proceedings.mlr.press/v124/lu20a.html
21. Madumal, P., Miller, T., Sonenberg, L., Vetere, F.: Explainable reinforcement learning through a causal lens. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. pp. 2493–2500. AAAI Press (2020), https://ojs.aaai.org/index.php/AAAI/article/view/5631

22. Méndez-Molina, A., Avelino, I.F., Morales, E.F., Sucar, L.E.: Causal based q-learning. Research in Computing Science **149**, 95–104 (2020)
23. Méndez-Molina, A., Morales, E.F., Sucar, L.E.: Causal discovery and reinforcement learning: A synergistic integration. In: Salmerón, A., Rumí, R. (eds.) International Conference on Probabilistic Graphical Models, PGM 2022, 5-7 October 2022, Almería, Spain. Proceedings of Machine Learning Research, vol. 186, pp. 421–432. PMLR (2022), https://proceedings.mlr.press/v186/mendez-molina22a.html
24. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M.A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nat. **518**(7540), 529–533 (2015). https://doi.org/10.1038/nature14236, https://doi.org/10.1038/nature14236
25. Mutti, M., Santi, R.D., Rossi, E., Calderón, J.F., Bronstein, M.M., Restelli, M.: Provably efficient causal model-based reinforcement learning for systematic generalization. CoRR **abs/2202.06545** (2022), https://arxiv.org/abs/2202.06545
26. Nair, S., Zhu, Y., Savarese, S., Fei-Fei, L.: Causal induction from visual observations for goal directed tasks. arXiv preprint arXiv:1910.01751 (2019)
27. Pearl, J.: Causality. Cambridge university press (2009)
28. Peng, S., Hu, X., Zhang, R., Tang, K., Guo, J., Yi, Q., Chen, R., Zhang, X., Du, Z., Li, L., Guo, Q., Chen, Y.: Causality-driven hierarchical structure discovery for reinforcement learning. CoRR **abs/2210.06964** (2022). https://doi.org/10.48550/arXiv.2210.06964, https://doi.org/10.48550/arXiv.2210.06964
29. 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
30. Sen, R., Shanmugam, K., Dimakis, A.G., Shakkottai, S.: Identifying best interventions through online importance sampling. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 3057–3066. PMLR (2017), http://proceedings.mlr.press/v70/sen17a.html
31. Sutton, R.S.: Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In: Porter, B.W., Mooney, R.J. (eds.) Machine Learning, Proceedings of the Seventh International Conference on Machine Learning, Austin, Texas, USA, June 21-23, 1990. pp. 216–224. Morgan Kaufmann (1990). https://doi.org/10.1016/b978-1-55860-141-3.50030-4, https://doi.org/10.1016/b978-1-55860-141-3.50030-4
32. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
33. Vinyals, O., Babuschkin, et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature pp. 1–5 (2019)
34. Watkins, C.J., Dayan, P.: Q-learning. Machine learning **8**(3-4), 279–292 (1992)
35. Yabe, A., Hatano, D., Sumita, H., Ito, S., Kakimura, N., Fukunaga, T., Kawarabayashi, K.: Causal bandits with propagating inference. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 5508–5516. PMLR (2018), http://proceedings.mlr.press/v80/yabe18a.html

36. Zeng, Y., Cai, R., Sun, F., Huang, L., Hao, Z.: A survey on causal reinforcement learning. CoRR **abs/2302.05209** (2023). https://doi.org/10.48550/arXiv.2302.05209, https://doi.org/10.48550/arXiv.2302.05209
37. Zhang, J., Bareinboim, E.: Markov decision processes with unobserved confounders: A causal approach. Tech. rep., Technical report, Technical Report R-23, Purdue AI Lab (2016)

Should Causal AI Rule over Deep Learning?*

Alberto D. Horner[1]

Universidad Nacional Autónoma de México, C.U. Coyoacán 04510, México
dohalberto@gmail.com

Abstract. This is an argumentative paper. First, it surveys the reasons to prefer causal methods over neural networks methods. Second, it addresses Cartwright's objection against the Causal Markov Condition, which has been raised as a reason to reject causal inference methods as general methods. The present paper proposes that, among other properties, the Causal Markov Condition enables causal models to be a better option for ruling artificial intelligence complex systems. To conclude, causal models are argued to be a better way to model human cognition than current (artificial) neural approaches. It is asserted that Causal AI should rule over Deep Learning on inferential and cognitive grounds.

Keywords: Causal AI · Deep Learning · AI Project.

1 Reasons to prefer Causal AI

Should Causal AI rule over Deep Learning? The answer, as argued here, is 'yes'. The question does not ask which one, if any, is a perfect method. AI progress and applications will not stop soon. Scientists need to decide, therefore, in which direction will they promote progress, and which methods will they apply. It is not reasonable to demand perfect, indisputable methods. Instead, scientists are expected to choose and improve the best available ones. Sometimes a new method is discovered and it should be evaluated in comparison with the other existing methods.

In the present paper, 'Causal AI' is understood as AI based on Causal Bayesian Networks (CBNs) and specifically based on, but not limiting to all the causal inference methods related with Judea Pearl's work. Causal models, Bayesian Networks (BNs), their assumptions, and the algorithms to infer them are here understood as defined in his book *Causality* [1]. A Causal AI system is expected to convey causal explanations of its behavior, and to operate with information about the precise relations between variables.

On the other hand, claims about Deep Learning techniques are intended to apply to any densely connected neural network. For the sake of specificity, let us

* The present paper presents preliminary conclusions that are part of a wider research and reasoning. The full research and argumentation will be presented soon as a thesis for a Master's degree on Philosophy of Science at UNAM. I would like to thank Dr. Francisco Hernández Quiroz, who is my thesis advisor and two anonymous reviewers whose gentle feedback improved this paper.

say that claims about Deep Learning refer to the methods described in Chollet's book *Deep Learning with Python* [8].

Deep Learning and Causal AI do not exclude each other. Some tasks may be better performed by a deep learning algorithm. Others, indeed, can be solved by a cooperation of both methods. Recently we have seen considerable progress towards fusing both approaches; for example, it has been proved that Testing Bayesian Networks (TBNs) (BNs extended with testing units) are universal approximators [3, 4]. Notwithstanding, when cooperation is at hand, or when scientific investigation is at stake, a predominant one needs to be chosen.

There are some well-known advantages of causal methods versus deep learning ones (I mean, well-known within the causal AI community). The strongest advantage is that causal models grasp changes in the probability distribution over the variables, that is to say, they deal with a family of 'n' probability distributions; while deep learning networks just consider one distribution.[1] If we want to use the same deep learning network in a different context with a different probability distribution, we have to retrain the network.

Other well-known advantages are interpretability and simplicity. Both amount to better understanding of phenomena and the ability to translate knowledge into policies. ChatGPT will always have an answer (perhaps a wrong or a discouraging one), and it can not tell you why it produced such an answer [7]. In contrast, causal models explicitly explain the outcomes. Also, they require less computing power.

The list of advantages is large: the feasibility of encoding previous (expert) knowledge, modeling situations in which the agent itself modifies the probability distribution, measuring causal effects with a precise number, grasping invariant qualitative knowledge, and so on [1].

The neural approach has its own advantages too. First of all, they are universal approximators (CBNs are not [3, 4]). Second, expert knowledge is not strictly required for them to work. Third –and perhaps the best advantage: they have found no competitors in the problems they solve. For instance, although it is possible to criticize Large Language Models, up to now there is no causal-powered language model available.

In what follows, (first) an answer to the main objection against causal inference is provided, and (second) an argument in favor of causal AI preference is presented on cognitive grounds.

2 The Causal Markov Condition could be an advantage

There are three main assumptions in causal inference methods: minimality, the Causal Markov Condition, and stability. For instance, in the IC algorithm [1]. It would be practically unreasonable to criticize the first one. So, the main two objections against causal inference are those against the other two assumptions. I focus on the Causal Markov Condition. Before getting to the objections, I will

[1] The reader can compare the definition of causal model [1] with Vapnik's empirical risk minimization approach [5].

pose what is obtained as a consequence of the Markov Condition, contrasting it with other non-causal approaches.

Vladimir Vapnik [5],[2] an advocate of the non-causal statistical learning,[3] distinguishes between the classical approach and the new approach in statistics. The classical approach, explains Vapnik, looks for a small set of strong features and uses simple functions (vg. linear functions) to explain phenomena, while the new techniques use a large number of weak features (vg. data mining) and looks for 'smart' functions to approximate the unknown dependency. Neural networks belong more properly to the 'new technique', and causal models seem to be closer to the classical approach.

Perhaps the new techniques gain a better adjustment to the data, but they lose comprehension. From a comprehensiveness point of view, the contrast between both approaches rises a big difference: for neural networks, with all their weak features and 'smart' functions, there is no feasible procedure to select a few neural paths that will render the others negligible. Anyone agreeing that comprehension entails the understanding of a limited and (under some criteria) sufficient set of relations between the variables, should also agree that there is no 'comprehension' in neural networks.

In behalf of the comprehension of the phenomena we want to understand with these methods, we have causal models. In fact, within causal models it is feasible to select a few causal paths that will render the others insignificant. And the possibility of such a procedure relies on the Causal Markov Condition.

Now, what Nancy Cartwright objects to the Markov condition is that there are many contexts where it does not hold [12].

She presents five scenarios where structural dependencies (i.e. dependencies that occur when conditioning in all relevant factors) are not due to causal relations and, therefore, the Causal Markov Condition fails in those systems. Those scenarios include separately (i) common causes, (ii) causes cooperating to produce one effect, (iii) mixed populations, (iv) changes in the same direction in time, and (v) by-products. She then concludes that the connection between structural dependencies and causal relations is not tight. In other words, that it will not work all the time.

Cartwright's objection calls attention on the fact that not all circumstances are susceptible of causal treatment. We must be careful in evaluating the characteristics of the population we are facing. Her concerns are about the adequacy of these models with real cases, not about the consistency of the model. Fortunately, for a variety of macroscopic phenomena (including those studied by epidemiology, medical diagnosis, economics and climate sciences), the Causal Markov Condition is an adequate assumption [2]. For many of those phenomena

[2] Vladimir Vapnik has been recently working on what he calls *The Complete Statistical Learning Theory* [6]. The reason why here his previous theory is taken as a reference is that most of the current practices of machine learning are grounded on the previous theory [8–11].

[3] He does not reject causality explicitly. Here I call his approach non-causal just because it does not address causal questions.

it is reasonable to assume the presence of non-observed variables (this assumption allows us to restore Causal Markov Condition in non-Markovian models) [1].

Note that predominant statistical learning approaches in current practices are not committed with the causal assumptions (causal parameters and assumptions are distinct from statistical ones). Specifically, they do not assume anything about the existence or non-existence of non-observed variables.

Non-observed variables agnosticism could be rendered as a scientific attitude, but it could also be inadequate for some phenomena. In defence of causal models, it must be said that, though they require strong assumptions, they address adequately the considerable many processes which traditional Deep Learning can not. Namely, those in which decisions alter probability distributions.

It turns out that decisions are related with control tasks. Hence, in any complex system, it will be desirable to subordinate the neural algorithms to the causal ones. This idea closely follows Minsky's *Society of Mind* model [13]. Minsky claimed that, though neural networks can model some tasks of general intelligent systems, they will never be enough to stand as a complete general intelligent system. Just wonder if it is possible to design an AI system with a CBN as its main structure, one which operates not based on raw data inputs, but on results obtained by different neural networks that process different kinds of data.

3 On cognitive grounds

Neuroscience and Artificial Intelligence mutually enrich themselves. All the computational neural-networks framework was motivated by the idea that we could solve pattern recognition problems studying how human brain actually solves them. Conversely, when a cognitive problem solver algorithm is found, scientists often wonder if the brain could be implementing the same algorithm.

As far as AI does not overstep human intelligence (it does in some specific tasks, but not in general yet), human cognition stands as the natural benchmark of intelligence development.

Leading scientists as Geoffrey Hinton [14] and Francois Chollet [8] assert that the most widespread artificial neural architectures are not a good model for the brain. For example, it is highly implausible that the human brain could implement backpropagation. Even though some very refined proposals as the NGRAD hypothesis have been presented, the physiological compatibility with those algorithms remains implausible [15].[4]

[4] The NGRAD hypothesis, 'neural gradient representation by activity differences' presents a family of backprop-like algorithms that approximate backprop results by different feedback processes and which are less physiologically implausible. The physiological incompatibility is due to the backward pass. Human neurons seem to perform a backward pass through the same path as the forward pass, but it is not symmetrical. Neurons modify their states as the feedback information is transmitted by them.

Hinton has been working in order to design neural algorithms that could actually model human brain activity. He currently proposes the Forward-Forward (FF) algorithm as a plausible model [14]. Two main aspects of the FF algorithm make it worth attention. (i) It does not perform a backward pass, that is to say, it neither needs to have complete access to its whole forward pass information, nor is required to compute the derivatives. (ii) It is trained in two different stages, with real and with negative data, and they may correspond to awake and sleep phases. The FF algorithm is expected to tell whether the data provided is real or negative data.

Hinton is right in that the brain is not computing backpropagation, yet he tries to use an equally superficial model. Superficial in the sense of lacking comprehension. Besides telling apart real and negative data, FF can be trained by supervised learning introducing the label as part of the input: it can fit a distribution. Nevertheless, it cannot answer why, not to say computing results that would have occurred if things were different.

Despite the label 'neural', it is healthy to question if real neurons, and overall our human cognitive system, operate within a neural network framework as it is understood by machine learning scientists. The answer is that they probably do not operate within that framework.

Recent results have challenged the prevalent hypothesis of prospective associative learning. They show that dopamine release activity is rather related with retrospective learning [16]. Instead of predicting effects (subsequent events) from causes (previous events), humans start from effects and retrospectively look for causes.

This evidence supports the claim that counterfactuals are an intrinsic feature of human cognition. Furthermore, it suggests that this kind of reasoning is anchored in low level synaptic information transmission. Since deep learning networks are unable to perform counterfactual reasoning, they utterly fail as models of human cognition.

4 Concluding remarks

The main well-known advantages of Causal AI over Deep Learning were presented in the first section. In the second section, it was argued that Cartwright's objection against Causal Markov Condition is not a defeating one, inasmuch as it does not apply to the problems which causal models are meant to solve. As a consequence of this defence, it was stated that causal models are especially adequate for control tasks and, furthermore, they should rule over deep learning methods in complex systems.

Minsky's *Society of Mind* was very influential in the present argument. So were the recent results of H. Jeong *et al.*. In the third section, based on those results it was argued that retrospective causal reasoning models are better on modeling human cognition than the most recent proposals of the artificial neural networks approach.

It would be helpful to keep in mind that originally both methods (Deep Learning and Causal AI) were designed for different purposes. The first neural network, i.e. the Perceptron, was intended to solve a perception problem [17]. The Pattern recognition problem has conceptually guided the landmarks (Backprop, CNNs, LLMs, etc.) in neural networks [5], while the landmarks in causal inference (solving Simpson Paradox, d-separation, IC algorithm, etc.) have been guided by a different kind of problem: the problem of precisely determining relations between variables and the consequences of those relations. They do not pursue the same scientific goal. It is not the same to grasp a pattern than to understand the data generating process.

Since both methods obey distinct goals, we can not just combine them. While recognizing that both are extremely useful, it is needed to chose a main goal in order to put them to collaborate. Recall that despite it is possible to fit a distribution without making assumptions about non-observed variables, it is not possible to infer a data generating process without estimating the actual distribution. This could sound as a basic statement for anyone acquainted with CBNs: first you estimate the probability distribution, then you infer relations between variables. It has a non obvious consequence, though. Causal AI looks for a deeper and more complete knowledge. In this sense, it is deeper than deep learning, and deserves a preeminent place in our research and scientific activities.

The whole argument of the present paper points to answer 'yes, Causal AI should rule over Deep Learning, both as a scientific tool for understanding macroscopic phenomena, and as a path towards more intelligent AI systems.'

References

1. Pearl, J.: *Causality: Models, Reasoning and Inference. 2nd edn.* Cambridge University Press, New York (2009)
2. Pearl, J. Mackenzie, D.: *The Book of Why.* Basic Books, New York (2018)
3. Choi, A., Darwiche, A.: On the Relative Expressiveness of Bayesian and Neural Networks. PGM. *Proceedings of Machine Learning Research* 72, 157–168 (2018)
4. Choi, A., Wang, R., Darwiche, A.: On the Relative Expressiveness of Bayesian and Neural Networks. *International Journal of Approximate Reasoning (IJAR)* 113 303–323 (2019)
5. Vapnik, V. N.: *The Nature of Statistical Learning Theory. 2nd edn.* Springer Science+Business Media, New York (2000)
6. Vapnik, V., Izmailov, R.: Complete Statistical Learning Theory (Learning Using Statistical Invariants). *Proceedings of Machine Learning Research* 128, 1–37 (2020)
7. Monteiro, C., Silva, A.: Do Large Language Models Understand Chemistry? A Conversation with ChatGPT. *Journal of Chemical Information and Modeling* **63**(6), 1649–1655 (2023)
8. Chollet, F.: *Deep Learning with Python. 2nd edn.* Manning Publications, New York (2021)
9. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning.* MIT Press, Massachusetts (2016)
10. Lee, W.: *Python Machine Learning.* Wiley, Indiana (2019)
11. Alpaydin, E.: *Machine Learning.* MIT Press, Massachusetts (2021)

12. Cartwright, N.: *Hunting Causes and Using Them*. Cambridge University Press, New York (2007)
13. Minsky, M., Papert, S.: *Perceptrons (Expanded Edition)*. MIT Press, Massachusetts (1988)
14. Hinton, G.: The Forward-Forward Algorithm: Some Preliminary Investigations. *Google Brain*. (2022). https://www.cs.toronto.edu/∽hinton/absps/FFXfinal.pdf. Last accessed 15 May 2023
15. Lillicrap T., Santoro, A., Marris, L., Akerman, C., Hinton, G.: Backpropagation and the Brain. *Nature Reviews Neuroscience* 21 335–346 (2020)
16. Jeong, H. *et al.*: Mesolimbic Dopamine Release Conveys Causal Associations. *Science* 378, eabq6740 (2022). https://doi.org/10.1126/science.abq6740
17. Rosenblatt, F.: The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review* 65:6 386–408 (1958)

Section 3

Applications

Causal discovery of Mexican COVID-19 data

Verónica Rodríguez-López[1][0000−0002−5976−9338] and
L. Enrique Sucar[2][0000−0002−3685−5567]

[1] Computer Science Institute, Universidad Tecnológica de la Mixteca, Oaxaca, México
veromix@mixteco.utm.mx
[2] Computer Science Dept., Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México esucar@inaoep.mx

1 Introduction

In Mexico, the COVID-19 disease caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) was the leading reason for death in 2020 and 2021 [5]. By April 2022, Mexico had been affected by four pandemic periods caused for different variants of SARS-CoV-2: the first wave was from March 29th, 2020 to September 26th, 2020, the second wave, from September 27th, 2020 to April 17th, 2021, the third wave, from May 23rd, 2021 to November 6th, 2021, and the fourth wave, from December 19th, 2021 to March 19th, 2022 [3]. During these four pandemic periods, the Mexican population was affected differently depending on factors such as gender, age, habits, and the presence of comorbidities [3,5].

Causal discovery methods aim to recover graphical models encoding the causal relations between the factors of phenomena from non-experimental data. Score-based methods use score functions to discover the causal graphical model that is consistent with the likelihood of data [1]. Improvements in these methods have made it possible to discover models from high-dimensionality data with a high number of variables and cases [6], some of them even assuming insufficient data [7]. In this work, we apply score-based causal discovery methods to identify the factors that mainly impact the severity of COVID-19 among the Mexican population.

This work presents the causal analysis performed with Mexican COVID-19 data collected during the second period of the pandemic. We aim to analyze the impact of age, gender, habits, and some comorbidities on the COVID-19 severity among Mexico City and Yucatan populations. Our preliminary results reveal some findings that are consistent with those reported in some investigations.

2 Methods

We worked on the causal analysis with the Mexican COVID-19 data that provides organized and standardized information on the epidemiological and demographic evolution of the COVID-19 pandemic in Mexico (available for research purposes at http://covid-19.iimas.unam.mx). We only considered data collected for the second period of the pandemic (from September 27th, 2020 to April 17th, 2021) of the SARS-CoV-2 confirmed cases for Mexico City and Yucatan. We included in our study 600521 cases of

Mexico City and 19666 of Yucatan, including information on gender, comorbidities, conditions, hospitalized status, and final survival status of the COVID-19 patients.

In our study, we search for the causal associations between comorbidities, age, and gender with the risk of hospitalization and death in patients of the second pandemic wave and the age group of 41−60 of the populations. We discovered the causal graphical models of Mexico City and Yucatan by applying the FGES method [6]. Taking into account that FGES requires sufficient data to find a reliable causal graphical model and Yucatan datasets have small sample sizes, we also discovered the models of Yucatan by applying KTL-WeFGES [7], using the Mexico City datasets as source data. These causal graphical models help to confirm the relations found for Yucatan with FGES.

3 Results and Discussion

In Figures 1 and 2, we present the graphical models of Mexico City and Yucatan for the second wave and the age group of 41 − 60, respectively. The causal graphical models of the second wave for Mexico City and Yucatan include relations that have been reported for some studies [2,4]. They indicate that directly impacted death age, gender, COPD, immunosuppression, heart disease, hypertension, and chronic kidney disease. They also reveal that directly impacted hospitalized some comorbidities such as obesity, hypertension, and chronic kidney disease. They also suggest some causal paths between comorbidities with hospitalization and death. These possible causal paths seem to confirm that patients with three or more comorbidities had a higher risk of death [2].

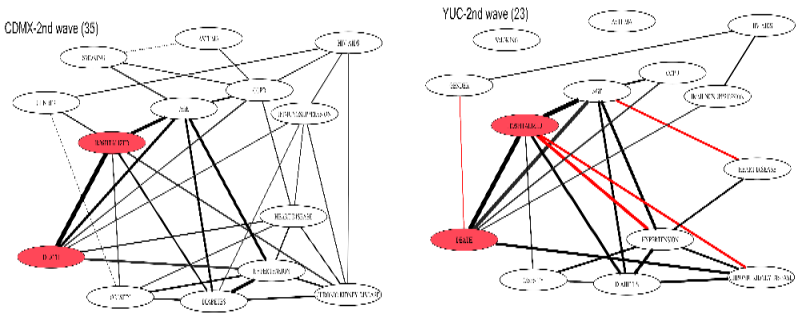


Fig. 1. Causal graphical models of Mexico City and Yucatan for the second wave. The number in parentheses in each model indicates the number of edges. The thickness of the lines indicates the strength of the correlation between the variables, dotted lines, the relations with a correlation factor of less than 0.2, and red, those relations of Yucatan that are not present in the causal graphical model of Mexico City. (Best seen in color.)

For its part, the causal graphical models for the 41 − 60 age group reveal the more frequent comorbidities among Mexico City and Yucatan populations. They indicate how the comorbidities among Mexico City and Yucatan populations impacted differently to the severity of COVID-19.

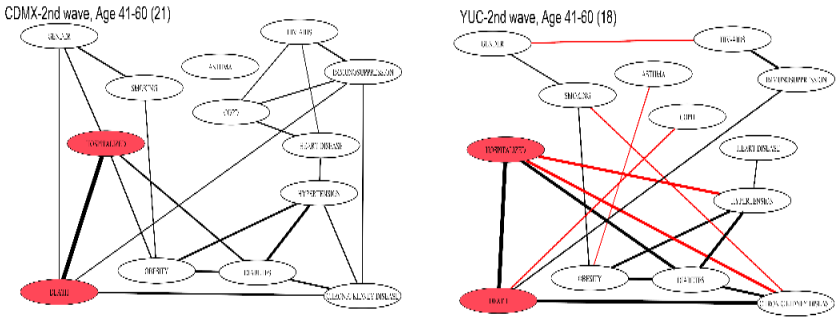


Fig. 2. Causal graphical models of Mexico City and Yucatan for 41 – 60 age group of the second wave. The number in parentheses in each model indicates the number of edges. The thickness of the lines indicates the strength of the correlation between the variables, dotted lines, the relations with a correlation factor of less than 0.2, and red, those relations of Yucatan that are not present in the causal graphical model of Mexico City. (Best seen in color.)

In summary, the graphical models discovered by our analysis reveal some main factors that probably be the causes of the severity of COVID-19. We believe that our preliminary results help increase understanding of how COVID-19 impacted the Mexican population. These models reveal possible causal paths between comorbidities besides confirming their relationship with the severity of COVID-19. Furthermore, they expose relations between comorbidities and other factors such as age and smoking.

Further research is needed to analyze the causal graphical models of the other periods of the pandemic and age groups. It will be important that future work investigate the orientation of the causal relations and their variations across the pandemic periods and age groups. Future work should also include validating and interpreting the causal graphical models and their variations made by epidemiologists.

Acknowledgements We thank Antonio Loza and Rosa María Gutiérrez-Ríos for their assistance and for providing us with the Mexican COVID-19 data. We also thank epidemiologists María Eugenia Jiménez Corona and Rosa María Wong for the review of the preliminary causal graphical models. This work was supported in part by CONACYT project A1-S-43346.

References

1. Chickering, D.M.: Optimal structure identification with greedy search. Journal of Machine Learning Research **3**, 507–554 (2003). https://doi.org/10.1162/153244303321897717
2. Kammar-García, A., Vidal-Mayo, J.d.J., Vera-Zertuche, J.M., Lazcano-Hernández, M., Vera-López, O., Segura-Badilla, O., Aguilar-Alonso, P., Navarro-Cruz, A.R.: Impact of comorbidities in mexican SARS-CoV-2-positive patients: a retrospective analysis in a national cohort. Revista de investigación clínica **72**(3), 151–158 (2020). https://doi.org/10.24875/ric.20000207

3. Loza, A., Wong-Chew, R. M.and Jiménez-Corona, M.E., Zárate, S., López, S.and Ciria, R., Palomares, D., García-López, R., Iša, P., Taboada, B., Rosales, M., B.C., A., H.E., Mojica, N.S., Rivera-Gutierrez, X., Muñoz Medina, J.E., Salas-Lais, A.G., Sanchez-Flores, A., Vazquez-Perez, J.A., Arias, C.F., Gutiérrez-Ríos, R.M.: Two-year follow-up of the COVID-19 pandemic in Mexico. Frontiers in public health **10**(1050673), 1–14 (2023). https://doi.org/10.3389/fpubh.2022.1050673
4. Márquez-González, H., Méndez-Galván, J.F., Reyes-López, A., Klünder-Klünder, M., Jiménez-Juárez, R., Garduño Espinosa, J., Solórzano-Santos, F.: Coronavirus disease-2019 survival in mexico: A cohort study on the interaction of the associated factors. Frontiers in Public Health **9** (2021), https://www.frontiersin.org/article/10.3389/fpubh.2021.660114
5. Palacio-Mejía, L., Hernández-Ávila, J., Hernández-Ávila, M., Dyer-Leal, D., Barranco, A., Quezada-Sánchez, A., Alvarez-Aceves, M., Cortés-Alcalá, R., Fernández-Wheatley, J., Ordoñez Hernández, I., Vielma-Orozco, E., Muradás-Troitiño, M., Muro-Orozco, O., Navarro-Luévano, E., Rodriguez-González, K., Gabastou, J., López-Ridaura, R., H., L.G.: Leading causes of excess mortality in mexico during the covid-19 pandemic 2020-2021: A death certificates study in a middle-income country. Lancet Regional Health Americas **13**(100303), 1–15 (2022). https://doi.org/doi: 10.1016/j.lana.2022.100303
6. Ramsey, J.D.: Scaling up greedy equivalence search for continuous variables. CoRR **abs/1507.07749** (2015), http://arxiv.org/abs/1507.07749
7. Rodríguez-López, V., Sucar, E.: Knowledge transfer for causal discovery. International Journal of Approximate Reasoning **143**, 1–25 (2022). https://doi.org/10.1016/j.ijar.2021.12.010

Reinforcement learning through relational representation and causal modeling

Armando Martínez Ruiz, Eduardo F. Morales, and L. Enrique Sucar[1]

Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro # 1, Tonantzintla, Puebla, México C.P. 72840

Abstract. In reinforcement learning, an agent learns to behave, through trial and error, in a dynamic environment. This way of learning requires a considerable amount of data and time. Relational representation and causal models allow to abstract the state space, and therefore they can help the learning process to be faster, and also, abstract representations are easier to transfer to to other similar domains that share these abstractions; so it is proposed to investigate whether the agent learns faster using these methods, as well as to verify that this abstraction is transferable to other tasks.

Keywords: Reinforcement Learning · Relational Representation · Causal Models · Transfer Learning.

1 Introduction

Learning is one of the main areas of artificial intelligence and, in general, it tries to build programs that improve their performance automatically with experience. In this sense, machine learning studies and computationally models learning processes in their various manifestations.

Reinforcement learning is a paradigm of machine learning that studies how an agent maximizes a future reward it receives from the environment through its interactions with the environment[1]. At each iteration, the agent receives a signal from its current state s and selects an action a. The action possibly changes the state and the agent receives a reward signal r. The goal is to find a policy, which is a function that maps states to actions, that maximizes the total expected reward.

One of the problems with reinforcement learning is that the interactions of an agent and the environment in which it finds itself require a considerable amount of data and time for the agent to learn, so the learning process in this approach is time consuming. In addition, reinforcement learning suffers from its poor ability to generalize learned knowledge to new but related problems.

Based on the above, the use of relational representation and causal models can allow us to abstract certain characteristics of the environment, so that the agent's learning can be accelerated.[2]

2 Methodology

The proposed approach is applied to ATARI 2600 game environments. The choice of this type of environments is due to the fact that important features can be generalized using relationships between the agent and its environment.

A first step is the incorporation of an object detection system within a reinforcement learning environment. The object detection system that was incorporated into the reinforcement learning environment is based on Detecto, a Python package that allows building object detection models using a Faster R-CNN architecture, which is composed of two modules: a deep convolutional network processing regions (RPN) and a Fast R-CNN detector that uses these proposed regions.[3]

This detection system allows defining the relations of the agent with the objects in its environment, obtaining relations of the following type expressed as predicates:

$$\begin{gathered} object(x, y, z), \\ wall(w), \\ close(x, y, z, w). \end{gathered}$$

where w, x, y, z are objects in the domain of interest.

We intend to use this relational representation together with the Q-Learning algorithm to check that an agent can learn faster than other RL methods using these relations that describe the environment within the game. In addition, we plan to test to what degree the knowledge acquired in a task can be transferred to similar tasks.

3 Preliminary Findings

The object detection model was trained using a set of 65 images, with the objective to correctly detect and classify the agent and the objects in its environment (trophies, obstacles and walls). Figures 1 and 2 show the graph of the loss function for the detection model and an example of object detection within the environment of an ATARI 2600 game. This detection is performed periodically during the training of the agent.

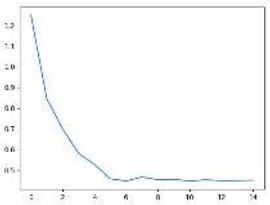


Fig. 1: Loss function

Fig. 2: Object detection

The relations between the agent and surrounding objects are defined based on object detection. Based on Figure 3, one can define the relation of the agent and the nearest object (obstacle) using first-order logic[4] as follows:

$$CloseToObject(a, o), \text{ which is true, if agent } a \text{ is near object } o.$$
$$CloseToWall(a, o), \text{ which is true, if agent } a \text{ is near wall } w.$$

So the first-order logic expression of the relation is:

$$\exists a[Rel(a, obstacle, left, up, \emptyset) \iff CloseToObject(a, obstacle) \vee$$
$$CloseToWall(a, obstacle) \wedge x = left \wedge y = up \wedge w = \emptyset],$$
$$obstacle \in \hat{O}, x \in \hat{X}, y \in \hat{Y}, w \in \hat{W}.$$

Where $\hat{O} = \{\emptyset, trophy, obstacle\}, \hat{X} = \{\emptyset, left, right\}, \hat{Y} = \{\emptyset, up, down\}, \hat{W} = \{\emptyset, L, R, F, B\}$ are the domains of o, x, y and w respectively.

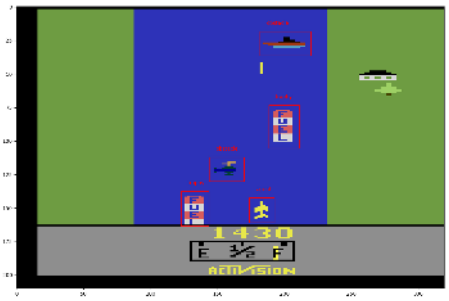


Fig. 3: Example of relational representation through object detection.

With these relations, a set of states was defined, such that the agent is in one of these states depending on the relation it has with the elements of its environment. Using this representation of the states, it was incorporated into the Q-Learning algorithm, with parameters $\epsilon = 0.1$ and $\gamma = 0.6$ in 500 episodes to obtain the preliminary results in Figure 4.

Fig. 4: Episodes vs Rewards

4 Discussion and Interpretation

The results obtained so far show us that the object detection system works adequately under the conditions of the reinforcement learning environment, where detection is performed as a function of the number of actions of the agent.

Furthermore, it can be observed that the agent reaches a new maximum reward approximately every 100 - 150 episodes, so it can be conjectured that given more training time, the agent will find an optimal policy as the number of episodes increases.

5 Future Directions

The next part of the project consists of generating an optimal policy within an ATARI 2600 game and transferring this knowledge obtained to another game, which has similar objectives. In this way, it is intended to prove that the training of the agent in a task using relational representation can be transferred to similar tasks.

The development of the initial ideas of how the relational representation can be used to construct causal models and its incorporation into some reinforcement learning algorithm are left as a basis for future work.

That said, it is expected that the relational representation and transfer of knowledge acquired by this approach can be transferred to similar tasks, thus defining an approach that combines elements of computational vision and relational representation with reinforcement learning.

6 Conclusion

The interdisciplinary nature of reinforcement learning allows different approaches and ideas to be used to further develop this area as well as applications in artificial intelligence. Relational representation and causal models can improve an

agent's ability to understand complex environments, resulting in more efficient decision making.

Preliminary work has provided insight into how it is possible to incorporate an external sensing system into reinforcement learning environments, particularly in ATARI 2600 games, as well as some important initial findings in training the agent from relations.[5]

In this way, it is expected that from these results obtained so far a way of transferring the knowledge acquired by the agent will be obtained, as well as establishing the theoretical bases on the equivalence of relational representations and causal models.

7 References

1. Sutton R., Barto A.: "Reinforcement Learning, An Introduction", 2nd edition, The MIT Press, 2020.
2. E. Morales. Scaling Up Reinforcement Learning with a Relational Representation. In: Proc. of the Workshop on Adaptability in Multi-Agent Systems (AORC - 2003), pp. 15-26.
3. Bi, Alan.: GitHub - alankbi/detecto: Build fully-functioning computer vision models with PyTorch, https://github.com/alankbi/detecto
4. De Raedt, Kersting, Natarjan, Poole, "Statistical Relational Artificial Intelligence: Logic, Probability and Computation", Morgan & Claypool, 2016.
5. V. Mnih, K. Kavukcouglu, D. Silver.: Playing Atari with Deep Reinforcement Learning. https://arxiv.org/abs/1312.5602v1

Learning MDP-ProbLog Programs for Behavior Selection in Self-Driving Cars*

Alberto Reyes[1][0000−0002−8509−6974], Héctor Avilés[2][0000−0001−5310−3474], Marco Negrete[3][0000−0002−5468−2807], Rubén Machucho[2][0000−0002−5731−6677], Karelly Rivera[2][0000−0002−4749−0663], Gloria I. de-la-Garza-Terán[2][0009−0004−2446−6435]

[1] National Institute of Electricity and Clean Energies, Morelos 62490, Mexico, areyes@ineel.mx

[2] Polytechnic University of Victoria, Cd. Victoria Tamaulipas 87138, Mexico, {havilesa,rmachuchoc,1930435,2130071}@upv.edu.mx

[3] Faculty of Engineering, National Autonomous University of Mexico, Mexico City 04510, Mexico, marco.negrete@ingenieria.unam.edu

Abstract. A two-stage scheme to learn MDP-ProbLog programs for self-driving cars is proposed. In a first stage, the transition and reward functions will be learned from simulated driving examples. Both functions will be described as an influence diagram (*ID*). In a second stage, the ID will be converted into a set of probabilistic clauses that will match the syntax of MDP-ProbLog. During this process, non-essential rules will be removed and redundant ones will be merged. The architecture of our self-driving car includes behavior selection, visual perception and control. This proposal is part of an ongoing research to evaluate the suitability of probabilistic logic to model safe autonomous decision-making in self-driving cars.

Keywords: Probabilistic logic, Factored Markov decision processes, self-driving cars.

1 Introduction

Self-driving cars promote potential positive effects for the mobility of humans and goods. An important capability of those type of vehicles is the autonomous selection of behaviors, that is responsible for deciding what reactive or short-term action (e.g., braking, accelerating, or stopping) is more appropriate in a current driving scenario to improve the safety of navigation.

We believe that probabilistic logic[1] is convenient for the selection of driving behaviors because: i) logical rules benefit the interpretation and explainability of decisions in comparison to pure numerical representations, and ii) probability theory is the most widely used framework for dealing with uncertainty. In

* This work was supported by UNAM-DGAPA under grant TA101222 and Consorcio de IA CIMAT-CONACYT

[1] We restrict our attention here to first-order logic clauses (rules and facts) extended with probability values.

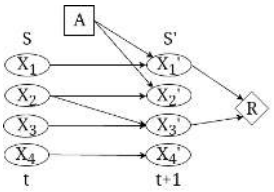


Fig. 1. An example of a influence diagram representing a transition function.

particular, Markov decision processes (*MDPs*) are gaining increasing attention for sequential decision-making in autonomous vehicles due to its capability to generate useful driving policies. Therefore, given that MDP-ProbLog does not include a learning model, we propose a two-stage scheme for learning MDP-ProbLog Programs (*MDP-PL*) [2] to generate action policies useful as a behavior selection scheme for self-driving cars. The numerical transition function and the reward function will be translated into a simpler propositional clause-based representation to construct the policy using MDP-ProbLog.

2 Factored Markov Decision Processes (*FMDPs*)

In Factored MDPs, states of the system are identified through the instances of a set of state random variables. This allows to reduce the number of model parameters in comparison to traditional MDPs, and to explore relationships among state variables. FMDPs require the definition of four main elements: i) the set of n state variables $\mathcal{X} = \{X_i\}_{i=1}^{n}$ along with the set $\mathbf{X}$ of all possible joint value assignments for all the variables in $\mathcal{X}$, ii) the set of possible actions $\mathcal{A}$ a decision maker can choose, iii) the transition function $p(\mathbf{x}'|\mathbf{x}, a)$ (that it is a shorthand notation for $p(X_1' = x_1', ..., X_n' = x_n'|X_1 = x_1, ..., X_n = x_n, a)$), where $\mathbf{x} \in \mathbf{X}$ represents a *pre-action* state, $\mathbf{x}' \in \mathbf{X}$ is a *post-action* state, and $a \in \mathcal{A}$, and iv) the reward model $R(\mathbf{x}, a)$. The transition function can be compactly represented via influence diagrams (*IDs*) [3] by taking advantage of conditional independence assumptions among random variables. A 4-variable ID representing a transition function is depicted in Fig. 1. IDs usually requires more compact conditional probability tables (*CPTs*) to represent probability distributions over post-action state variables, in comparison to a CPT of a transition function without conditional independence suppositions. As rewards are determined by the environment, the reward function $R(\mathbf{x}, a)$ could not be known beforehand by the decision maker, and it could be also learned.

3 Current progress on modeling driving decisions with MDP-PL

In this section, the behavior selection and the control and perception modules of the current self-driving architecture are briefly discussed. More details can be found in **[1]**.

3.1 Behavior selection module

In the selected problem domain, there are one self-driving car and four vehicles traveling on a one-way street with two lanes. Once the self-driving car moves, accordingly to the existence of other cars nearby, it has to decide one of three actions: a) cruise motion, b) overtaking or c) keep distance. The self-driving starts its movement on the right lane. The perceptual system (described in Section 3.2) reports constantly if there are other cars in predefined positions on the right and left lane. Those positions are labeled as North, North-West, West and South-West. Each one of these labels are associated with a Boolean state variable, and hence, the occupancy of the spaces around the self-driving car defines each one of the $2^4 = 16$ possible states of the system (Fig. 2).

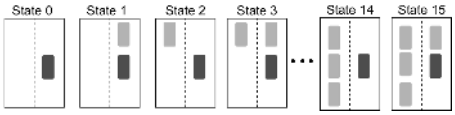


Fig. 2. States of our driving environment. The dark grey rectangle represents the self-driving car and the light grey rectangles are the nearby vehicles.

An early version of an MDP-PL was designed and implemented manually and coded in MDP-ProbLog. The reward function is based on (independent) additive utilities assigned to actions (without regarding on the state in which the actions are performed) and to state variables (and hence, it adds the utility to subsets of states that share the same value of the state variable). However, deterministic reward functions can be defined by including rules that assign utilities to specific state-action pairs. The source code is available in: https://github.com/hector-aviles/CaDis_Workshop. A number of 320 decision trials were performed, from which the car selected the right action in 98.75% of cases.

3.2 Perception and control modules

The perception module is divided in lane detection and lane tracking. Lane detection is based on regions of interest, the Canny edge detection and the Hough Transform to find the right and left limits of the lane, using an RGB camera. Steering is calculated based on the error between the expected and observed lines. The position of other vehicles and the estimation of their speed can be summarized in the following steps: i) a 3D Lidar sensor generates a point cloud that is filtered by distance and height of the vehicles nearby, ii) filtered points are clustered by K-means, and iii) the velocity of the vehicles is estimated by Kalman Filters. The source code of the architecture can be downloaded from: https://github.com/hector-aviles/CodigosEIR22-23.

4 Learning MDP-ProbLog programs

To learn an MDP-PL, data are collected from random driving actions performed by the self-driving car on the environment described in section 3.1. This data will be partitioned and sequentially registered in ordered 4-tuples $d_t(\mathbf{x}, a, \mathbf{x}', r)$ indexed in time $t \in \{1, ..., T\}$, such that $T \in \mathbb{N}$, $\mathbf{x} \in \mathbf{X}$ is the current observed state, $a \in \mathcal{A}$ is the current performed action, $\mathbf{x}' \in \mathbf{X}$ is the resulting state and r is a numerical reward value assigned to $\mathbf{x}$ and a. It's been considered to record from the simulator: timestamps, relative positions of the detected cars and velocities, action selected, and if its execution resulted successful or not. The reward r can be a positive quantity if the state-action pair does not lead to an accident or a negative value if a car crash takes place.

In the first learning stage, the K2 algorithm will be used to learn the transition function $p(\mathbf{x}'|\mathbf{x}, a)$. The reward function will be obtained by using J48 to generate a reward decision tree as detailed in [5]. In the second learning stage, the ID will be converted into an MDP-PL. These probabilistic clauses could be further simplified as referred in [4].

5 Conclusions and future work

A two-stage scheme to learn MDP-ProbLog programs to select driving behaviors in self-driving cars was proposed. The plan is to learn a dynamic Bayes net (factored transition function) and the reward function. We believe that this approach will help select safe driving actions, improve the understandability of the model, and discover causal relationships among variables that represent the entities on the road and the driving decisions to perform safe manouvers.

References

1. Avilés, H., Negrete, M., Machucho, R., Rivera, K., Trejo, D., Vargas, H.: Probabilistic logic markov decision processes for modeling driving behaviors in self-driving cars. In: Advances in Artificial Intelligence–IBERAMIA 2022: 17th Ibero-American Conference on AI, Cartagena de Indias, Colombia, November 23–25, 2022, Proceedings. pp. 366–377. Springer (2023)
2. 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)
3. Darwiche, A., M., G.: Action networks: A framework for reasoning about actions and change under understanding. In: Proc. of the Tenth Conf. on Uncertainty in AI, UAI-94. pp. 136–144. Seattle, WA, USA (1994)
4. Muggleton, S.H.: Duce, an oracle-based approach to constructive induction. In: IJCAI. vol. 87, pp. 274–281. Citeseer (1987)
5. Reyes, A., Ibarguengoytia, P.H., Santamaría, G.: SPI: A software tool for planning under uncertainty based on learning factored Markov Decision Processes. In: Advances in Soft Computing. Lecture Notes in Computer Science Series, vol. 11835. Springer (2019)