Improving Causal Learning Scalability and Performance using Aggregates and Interventions

KANVALY FADIGA, École Polytechnique, Paris, France
ETIENNE HOUZÉ, EDF R&D, Palaiseau, France & and Télécom Paris, Palaiseau, France
ADA DIACONESCU, Télécom Paris, Palaiseau, France
JEAN-LOUIS DESSALLES, Télécom Paris, Palaiseau, France

Smart homes are Cyber-Physical Systems (CPS) where multiple devices and controllers cooperate to achieve high-level goals. Causal knowledge on relations between system entities is essential for enabling system self-adaption to dynamic changes. As house configurations are diverse, this knowledge is difficult to obtain. In previous work, we proposed to generate Causal Bayesian Networks (CBN) as follows. Starting with considering all possible relations, we progressively discarded non-correlated variables. Next, we identified causal relations from the remaining correlations by employing "*do-operations*". The obtained CBN could then be employed for causal inference. The main challenges of this approach included: "non-doable variables" and limited scalability. To address these issues, we propose three extensions: i) early pruning of weakly correlated relations to reduce the number of required do-operations; ii) introducing aggregate variables that summarize relations between weakly-coupled sub-systems; iii) applying the method a second time to perform *indirect do* interventions and handle non-doable relations. We illustrate and evaluate the efficiency of these contributions via examples from the smart home and power grid domain. Our proposal leads to a decrease in the number of operations required to learn the CBN and to an increased accuracy of the learned CBN, paving the way towards large CPS applications.

ACM Reference Format:
Kanvaly Fadiga, Etienne Houzé, Ada Diaconescu, and Jean-Louis Dessalles . 2023. Improving Causal Learning Scalability and Performance using Aggregates and Interventions. 1, 1 (September 2023), 23 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Self-Adaptive Systems (SAS) can modify themselves dynamically to continue achieving their goals when run-time changes occur within their internal resources or external environment [26, 28]. Hence, SAS is a powerful paradigm for implementing Cyber-Physical Systems (CPS) that operate within dynamic contexts (e.g. smart homes, buildings or power grids). Self-adaptation requires knowledge of the system and its environment, so as to determine the causes of identified problems (e.g. failures or conflicts) and to identify the actions that would solve these problems (e.g. predicting the effects of actions). Hence, *causal models* provide suitable knowledge representations to enable reasoning-driven self-adaptation. The essential role of such knowledge representations for SAS has been extensively identified in the literature, either as a key enabler of SAS computational reflection

Authors' addresses: Kanvaly Fadiga École Polytechnique, Paris, France, f.kanvaly@gmail.com; Etienne Houzé EDF R&D, Palaiseau, France & Télécom Paris, Palaiseau, France, etienne.houze@telecom-paris.fr; Ada Diaconescu Télécom Paris, Palaiseau, France, ada.diaconescu@telecom-paris.fr; Jean-Louis Dessalles Télécom Paris, Palaiseau, France, jean-louis.dessalles@telecom-paris.fr.

XXXX-XXXX/2023/9-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

(i.e. referred to as “reference model”, “self-knowledge” or “domain model” [32], [2]) or in more “classic” feedback or control theory (i.e. “system model” in [4]).

However, such causal models are difficult to acquire and maintain, especially within complex, highly-diverse and dynamic environments. Moreover, scalability becomes a difficult issue as the number of relevant system variables and their relations may increase exponentially with the system’s size (e.g. considering an entire smart building or neighbourhood of buildings instead of a unique room drastically increases the number of managed variables and their potential causal relations).

Basic SAS solutions employ hard-coded and often implicit causal models– e.g. as a set of predefined adaptation rules, relying on extensive ontologies of the system and its environment. However, specifying predefined models for systems that feature highly diverse configurations may quickly become too costly or simply unfeasible – e.g. having to manage a wide variety of smart homes, buildings and neighbourhoods with highly heterogeneous devices and interconnections. For instance, when a user experiences temperature anomalies within a room, the cause may be linked to the “smart” devices present within that room and within adjacent rooms. Yet, in one home such devices may consist of classic thermostats controlling heaters and thermometers, while in another home they may also include air-conditioning equipment. Creating the corresponding causal models, customised for each managed environment, via manual and offline processes is both tedious and unfeasible. Online learning is needed instead to overcome this issue by producing causal models automatically and at runtime, within each particular SAS context.

The key role of online learning for acquiring knowledge about the managed system and its environment are at the core of the Self-Aware Computing initiative [25]. Moreover, this initiative also highlights the utility of such knowledge, not only for identifying efficient self-adaption actions, but also for providing *explanations* about such self-adaptations to users. In previous work, we have proposed a solution where abductive reasoning based e.g. on causal models can generate explanations to users on-the-fly [19].

Models@Runtime [35] have also been proposed to represent (mostly structural) knowledge about dynamically adaptable systems; and to update this knowledge automatically when the system changes. A ‘causal’ relation is ensured between the system and its runtime model, meaning that changes in the system are reflected in the model, and vice-versa. Thus, the runtime model allows to both observe and adapt the underlying system. However, the ‘causal’ relation here is between the system and its model– it says nothing about the causality between parts of the system (or of the model).

To address the above issues and enable online knowledge acquisition, recent CPS, such as smart home systems integrate Machine Learning (ML) components to predict the environment’s responses to actions and model its behavior. However, most “classic” ML approaches learn *associations*, or *correlations*, between observable variables based on repeated measurements and statistical analysis. Hence, spurious correlations pose a challenging issue in this kind of approach, as they may often lead to data misinterpretation and erroneous detection of causal relations. This phenomenon is particularly important in high-dimensional data [13], which is the case when considering large building with many variables. As a simple example, consider an ML system that i) correlates a thermometer’s increased temperature display with a heater being switched-on; and, ii) correlates the same thermometer’s increased temperature display with a presence detector’s indication of human presence (this can be the case when users always switch the heater on when entering the house). However, these correlations do not necessarily imply causal relations and hence may be misleading for SAS decision-making. Namely, if the SAS decides to i) indirectly act on the thermometer’s display (e.g. by switching-on another heat source nearby; cf. subsec. 4.3) this will not cause the initially observed heater to switch-on. Similarly, if the SAS decides to ii) indirectly trigger the presence detector (e.g. by activating a smart cleaning robot) this will not cause an increased temperature display. Hence, a SAS that relies on correlations rather than causal relations may carry-out ineffective adaptations.

To tackle this problem, we proposed in previous work [12] an approach for automatically modeling causal relations in SAS and their environment, based on: a) monitoring data of relevant variables; and b) performing "do" operations on available actuators (considering safety, security and performance issues – out of scope here) to distinguish causal relations from mere correlations. This approach aims to be generic so as to be applicable to various domains– e.g. smart homes and buildings, smart grids, autonomous vehicles, industrial systems.

Our proposal uses elements of causality, as introduced by J. Pearl [37]. The main finding we are reusing from Pearl's work is that causality can only be tested via counterfactual reasoning: i.e. the ability to consider alternative courses of events, and answer the question "what if . . . ?". To this extent, according to Pearl, *counterfactual reasoning*, based on models or simulations, is at the top of the "ladder of causation" – i.e. the most potent approach for determining causal relations. At an intermediate level, *intervention* allows verifying causal relations by performing actions on the actual system ("do-operation"). Finally, at the lowest level, *association* allows to correlate variable values and provides the weakest causality-detection method (cf. Table 1). Hence, our proposal aims to climb-up the "ladder of causation" from: a) the most basic level at the bottom – i.e. association, by observation-only, as in most ML approaches (e.g. how would seeing a hot temperature change the belief in the heater being on?); towards b) the middle level – i.e. intervention, by "doing", or actually checking the effects of actions (e.g. if I change the temperature will the heater switch on?).

Level	**Activity**	**Question**	**Example**
Association $P(Y\|X)$	Seeing	How would seeing X change my belief in Y?	What is the probability of someone's presence knowing the light is on ?
Intervention $P(Y\|do(X))$	Doing	What if I do $X = x$?	If I turn the heater on, would the temperature change?
Counterfactual $P(Y_x\|X', Y')$	Imagining	What if things were different?	Had the light been off, would the temperature had changed?

Table 1. The "Ladder of Causation"[5, 38]: most ML models are based on learning association from past data. However, Pearl identifies this as being only the first step towards causal reasoning, the highest objective being counterfactual thinking.

Our original approach [12] focused on investigating the benefits of "do" operations for distinguishing causal relations from correlations. It payed little attention to the costs or overheads induced by such actions. First, we considered a fully connected graph over all observable variables. Then, we performed "do" operations to discard relations where such actions had no observable effect. A final pass eliminated relations that were not coherent with sounder causal influences, i.e. removing arrows that would create cycles in the causal graph. Once the structure was known, we learned the Bayesian probabilities by using the maximum likelihood estimator.

While our initial results were encouraging [12], this method suffers from three major drawbacks. First, performing intervention operations, or *do-operations* on real CPS may raise various issues of safety, security, energy consumption and user nuisance. Second, in large systems, the scalability of our approach is limited by the increasing amount of do operations required. Third, it is possible that some variables are *non-doable* (e.g. too dangerous or impossible to manipulate). The first two issues indicate that the amount of "do" operations should be limited and hence may not support starting from a fully-connected graph. The third issue limits the accuracy of the method because some relations cannot be analyzed further to establish causality.

To tackle these limitations, we propose in this article three possible improvements to our original method. First, we propose to perform an early pruning of uncorrelated variables to reduce the number of do-operations to execute. Second, we consider capitalizing on available knowledge of underlying system organisation to reduce the number of superfluous correlations to verify. Notably, large systems often feature multi-scale designs [10], where the system is composed of several loosely-coupled sub-systems, each one in turn composed of tightly-coupled variables. Here, we propose to introduce aggregate variables that summarize the collective influences that a sub-system's tightly-coupled variables have upon other sub-systems. This allows reducing the number of correlations to verify between the system's weakly-coupled sub-systems. Third, we propose to introduce a second pass of the algorithm, where the previously learnt CBN can be used to indirectly intervene on non-doable variables. This is achieved by acting upon doable variables that were established previously to have causal relations with the non-doable ones, of interest.

The first two improvements tackle the first two issues of our initial method, by considerably limiting the number of required "do" operations. For instance, using early pruning of non-correlated variables reduces the number of relations to analyse from 2550 to 85 connections, hence a 96% reduction, in the theoretical experiments with 10 smart houses interconnected via an electric grid (cf. subsec. 5.1). In addition, the use of aggregate variables reduced the number of relations to analyse from 702 to 206, hence a 60% reduction, in the simulation experiments involving four rooms within a smart house (cf. subsec. 5.2). The third improvement addresses the third issue by capitalising on the accumulated causal knowledge to indirectly act on non-doable actuators.

This paper's contribution is to propose an improved online method for acquiring causal models of CPS, which represent key enables for CPS self-adaptation and self-explanation processes. The paper focuses on presenting and evaluating the proposed causal model generation method via two simulated examples from the smart home and power grid domains. Illustrating how such causal models (and related Bayesian networks) can be employed in various decision-making processes for self-adaptation and self-explanations is out of the paper's scope, as this connection is already supported in the related literature – e.g. [30], [14] for decision-making under uncertainty, [9], [47], [6] for self-diagnosis and autonomic fault management, [11] for online knowledge learning, [19] for self-explanation.

In the rest of this article, we present the theoretical principles behind our approach (sec. 2). Then, we detail the original Causal Learner method (sec. 3), before describing our proposed improvements in detail (sec. 4). We illustrate and evaluate our proposal on a realistic example modeling a neighborhood with several smart homes (sec. 5). Results confirm the advantages of the proposed extensions and improve its applicability to larger-scale systems. We discuss related works in sec. 6 and finally conclude in sec. 7.

2 THEORETICAL BACKGROUND

Albeit causality is a wide-spread and intuitive notion that has been considered in science for centuries ("gravity causes the apple to fall"), its formal definition and proper study has been a topic mostly since the beginning of the 20^{th} century. One of the main principles of this field consists in the close link between causality and counterfactual thinking: an effect E has a cause C if the effect would not have occurred were C not the case.

2.1 Structural Causal Model

A further formalism of causality is achieved by relying on Structural Causal Models (SCM). By definition [39], a SCM is a directed graph over a set of nodes representing random variables X. An edge (C, E) is in the model if and only if there exists a random distribution N_C, an independent random noise distribution N_E and a deterministic function f_E such that $C \equiv N_C$ and $E = f_E(C, N_E)$, i.e. E is the result of a deterministic function and an eventual noise. In the terms of an SCM, we can say that C is the cause of E.

Two aspects are important to note here. First, this definition of causality is consistent with the intuitive notion based on a counterfactual: if C is the SCM-cause for E, then a distribution change over C would result in a distribution change over E. Second, the intuitive non-cyclic aspect of causality is respected. If C is the SCM-cause for E, then there exists f_E and N_E such that $E = f_E(C, N_E)$. However, the opposite must be false: there cannot conversely exist a deterministic f_C and an independent noise N_C such that $C = f_C(E, N_C)$. [21] provides a detailed proof of this assertion and generalizes it to demonstrate that no cycle can exist within an SCM. As a result, SCM can be represented by *acyclic* graphs where vertices are the variables of the model and the edges are the causal relations. The support graph of an SCM is denoted as a *causal graph.*

2.2 Do-operation

The notion of intervention, as described by Pearl [37], mimics the scientific principle of identifying and isolating a variable to measure its influence on the outcome of an experiment. To this extent, intervention differs from the notion of conditional probability on the notion of variable isolation: in an intervention, the operator *forces* the variable to take a given value and *isolates* it from its causes, while conditioned probability is based on observations only.

Formalization of intervention has led to the development of *do-calculus.* Following Pearl's notations, $do(C = x)$ denotes that variable C was forced to take the value x by means of an action that is external to the causal model, isolating it from its intrinsic causes. If, in the causal model, $C \longrightarrow E$ is identified as a causal relation, then the relation $E = f_E(C, N_E)$ remains unchanged by the do-operation. Therefore, the do-operation allows to identify the relation: if $\mathbb{P}(E) \neq \mathbb{P}(E|do(C = x))$, then there is a causal connection $C \longrightarrow E$. In this case, we will use the notation $do(C) \rightsquigarrow E$.

The difference between do-operations and conditional probability can be understood by considering the example in Figure 1. This consists of a room equipped with a heater, a thermometer and a window. An intervention on the room's thermometer will *not* affect the heater's state: if one *forces* the room's thermometer to show $20°C$, this will not affect observations of the heater's state. Hence, $\mathbb{P}(H) = \mathbb{P}(H|do(T = 20°)$. However, the same is not true for conditional probability: as the heater affects room temperature, it is more likely to be switched on if the thermometer indicates a warm temperature. Therefore, $\mathbb{P}(H) \neq \mathbb{P}(H|T = 20°)$.

As the causal framework described by Pearl represents an abstraction, it eludes the question of actually implementing do-operations in a realistic setup. Hence, in real scenarios, some operations may not be applicable, or do-able, onto the actual system (e.g. a self-adaptive system cannot directly force the temperature of a room to an arbitrary level; a critical component cannot be disabled without jeopardising safety). To some extent, one could circumvent this limitation by considering a digital twin [16] of the system. This may be available when updating the causal model of a part of the system that has changed, while relying on a causal model of a part of the system that is already known and unchanged (i.e. the digital twin). In such cases, it could be possible to e.g., disable critical components in simulation, or to set the room temperature to a given value. However, any simulation will be limited in its possible settings. Thus, we introduce the notion of *non-doable* variables, which the system cannot set directly to an arbitrary value.

2.3 Bayesian Networks

Bayesian Networks (BN) are powerful tools for representing knowledge of a system's interactions via conditional probabilities [22]. Formally, a BN is a directed acyclic graph whose vertices are random variables. Each of these vertices X_i is associated to a set of conditional probabilities $\mathbb{P}(X_i \mid par(X_i))$, where $par(X_i)$ denotes the set of parents of the node X_i in the BN. Figure 2 represents a typical Bayesian Network, called "Asia network" that models the probability of developing some diseases based on preliminary factors. It is a case in point to show the efficiency of BNs to infer knowledge: here, the total probability of each node is displayed, based on the information that the person develops dyspnea. For a variable X_i, its total probability is defined as:

$$\mathbb{P}(X_i) = \sum_{(y_1,\ldots,y_k)\in par(X_i)} \mathbb{P}(par(X_i) = (y_1, \ldots, y_k))\mathbb{P}(X_i | par(X_i) = (y_1, \ldots, y_k))\,.$$

The BN integrates both the structure of the model as the topology of the graph, and its parameters as the conditional probability tables defined for each node. Therefore, there are usually two steps in the building of a BN. First, one needs to define the graph of the model, i.e. the variables covered by the network and the topology of the graph. Second, one needs to learn the conditional probability tables for each of the BN's nodes that best fit the data. In many cases, the first step is performed manually, by relying on expert knowledge. Then, in the second step, parameter learning can be automated.

Bayesian networks and causal graphs mainly differ with respect to the meaning of their edges: in a BN, an edge between two variables simply indicates that there exists a correlation between these variables that is expressed in the conditioned probability table of the target node, without any

[1]bayesserver.com/examples/networks/asia

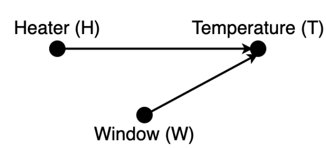


Fig. 1. A simple causal graph: the room's temperature T is affected by both the heater's state H and the window's state W. These relations can be described by physical phenomena (heat equations) that are independent of the state of the variables, hence they correspond to causal links.

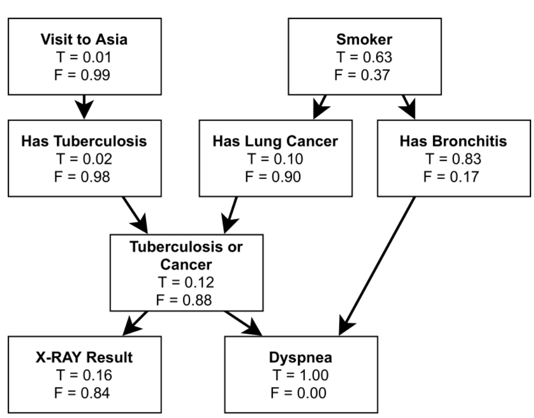


Fig. 2. An example of a Bayesian Network (the "Asia" network from Bayes Server[1]). The total probabilities obtained for each node are shown: they are computed by propagating an observation (here, dyspnea) along the edges of the network according to conditional probabilities. In this example, one may conclude that a person having dyspnea is likely to be a smoker (63%) and to have bronchitis (83%).

implication regarding causality. As such, a BN is merely a model aiming to best fit observed data. For instance, the Asia network shown in Figure 2 can be transformed into its reverse by changing the direction of all edges and re-computing the probabilities accordingly. Both the original and the reversed network would model the same variables with the same performance, and neither one could be deemed "better" than the other. By contrast, a causal model aims to model both the observations and the processes that generate them – hence, it cannot be reverted.

In the rest of this paper, we aim to model the system as a Causal Bayesian Network– i.e. a Bayesian Graph, where vertices represent system variables and edges represent causal influences between these variables. The point is that the system would augment its self-adaptive capabilities by exploiting such inferred causal relations.

3 LEARNING CAUSAL BAYESIAN NETWORKS USING INTERVENTION

In previous work [12], we proposed an automatic approach for learning Causal Bayesian Networks (CBNs) based on the principles introduced above. The main novelty consisted in performing do-operations to identify causal relations between system variables. We illustrate the main principles of our proposed algorithm, based on a small network example, as follows. In subsection 3.1, we start by considering all possible interactions between variables (i.e. a fully connected graph); then we prune the relations that appear incorrect based on two kinds of tests: direct influence (i.e. do-operations) and conditional influence (i.e. conditional do-operations). In Section 3.2, we present the actual algorithm that implements this approach. Finally, in Section 3.3, we present the learning procedure for establishing the CBN parameters. In Section 4 we present the novel extensions that we propose for this method to improve its efficiency and scalability for larger networks.

3.1 Testing causal influence using interventions

Direct Influence tests implement the notion of intervention (i.e. *do* operations). If there exists a direct causal influence of C over another variable E, then $do(C) \rightsquigarrow E$. In practice, to test the existence of this influence, one can use a statistical test, such as the χ^2 test, over the distribution of E when the variable C has been forced to different values, and compare this with the general distribution of E. To illustrate this, let us consider the example case of Boolean variables depicted in Figure 3: it represents a room equipped with a presence sensor P that triggers a light L and a heater H, which, in turn, impacts room temperature T. Here, one would verify the direct influence of P over L by testing the equality between $\mathbb{P}(E \mid do(C = 0))$ and $\mathbb{P}(E \mid do(C = 1))$. In this case, the two distributions clearly differ (see Figure 4a). Conversely, the same test indicates that L is not a cause for P (Figure 4b) as both distributions coincide. Hence, in our general framework, the edge (P, L) is confirmed as a causal link, while the edge (L, P) can be removed from the graph.

We note that in this simple example the two distributions in Figure 4b coincide perfectly, making their comparison trivial. This may not be the case in more complicated scenarios, considering multiple outputs. To handle such cases, the current proposal can be extended to consider the distance between the analysed distributions. Various methods are available in the literature to to evaluate the resemblance between two distributions, e.g. Kullback–Leibler divergence (KL divergence). The most suitable method and associated decision threshold will be studied in future work.

Conditional Influence tests perform interventions under certain conditions. While confirming direct causal influences can be achieved by simply checking a change in the probability distribution following an intervention (*do* operation), determining conditional influences requires a more subtle process. It can be described via the following question: *"Did C have an influence on E, given B?"*, which corresponds to the notation $do(C) \rightsquigarrow E|do(B)$. Contrary to the direct influence, the introduction of a third variable is necessary here. Think, for instance, of the influence that opening the window may have on the indoor temperature: there is a causal influence, but it is conditioned by the outdoor temperature.

Checking if a conditional influence does exist is a heavier process: one needs to check, via a χ^2 test, if there is a change between the distribution $\mathbb{P}(E \mid do(C = 0), do(B = u))$ and $\mathbb{P}(E \mid do(C = 1), do(B = u))$ for a given value u. For instance, considering Figures 4c and 4d, we infer that P has a causal influence on T, conditioned on L, while P has no influence on T conditioned on H. Hence, we can conclude that there is no direct causal influence between P and T: this result corresponds to

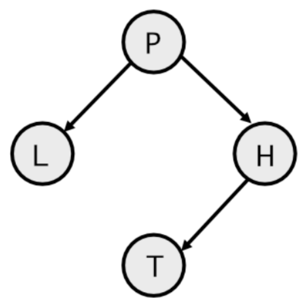
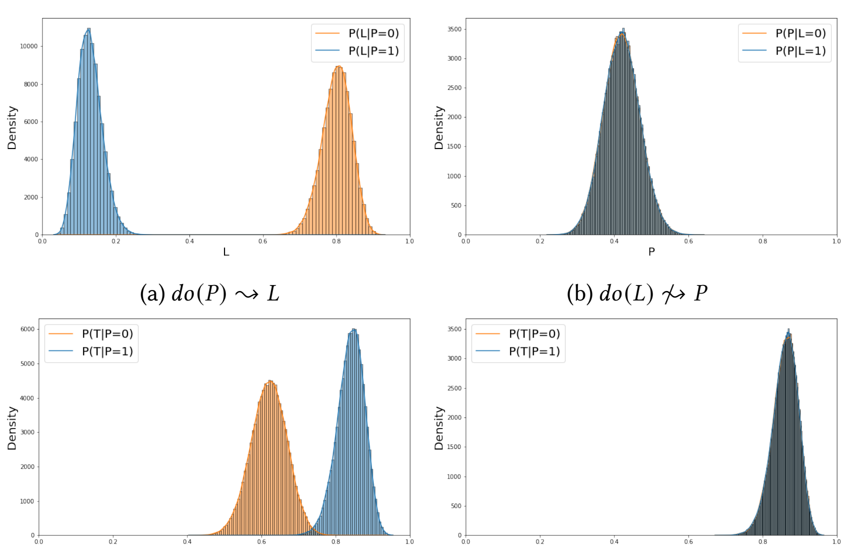


Fig. 3. An illustrative causal diagram over Boolean variables: the user's presence P causes the light L and the heater H to be switched on. This latter, if switched on, can cause the temperature T to become hot.



(a) $do(P) \rightsquigarrow L$

(b) $do(L) \not\rightsquigarrow P$

(c) $do(P) \rightsquigarrow T \mid do(L)$

(d) $do(P) \neg \rightsquigarrow T \mid do(H)$

Fig. 4. Different probability distributions resulting of do-operations. (a) finds a causal influence as the distributions differ with the intervention. This is not the case for the reverse direction (b), showing that L is not a cause of C. (c) an intervention on the presence P, when conditioned on the light L, changes the probability distribution of the temperature T, showing a possible causal influence. (d) however, a new intervention on L, this time conditioned by the heater H, shows that no change occurs in the probability distribution of T. The causal influence between P and T is indirect: $L \longrightarrow H \longrightarrow T$.

the observation from the ground truth causal graph shown in Figure 3. As above, we note that for the simple examples considered, the threshold values for the χ^2 test were easy to determine in an ad-how manner. For more complicated applications, we will need to explore various thresholds and their potential consequences.

3.2 Algorithm

Our method for identifying causal relations in an environment relies on the two previous tests. The overall approach is detailed in Algorithm 1. We initialize the process by considering a fully connected graph, i.e. by default, we consider that all variables influence each other. Then, we progressively remove relations by testing causal influences: first, we check first-order influences, i.e. direct causal influences; and second, we check higher-order influences, i.e. conditioned causal influences. The higher the order, the more variables have to be set to given values, but the less relations have to be tested, as most of them are discarded by the initial edge removals.

Algorithm 1: Extended *do* Causal Learner Algorithm

init
  $\mathcal{G}$ is the fully connected graph over nodes of $\mathcal{X}$;
  $k \leftarrow O$;
while *There are nodes with more than k neighbors in* $\mathcal{G}$ **do**
  for *each such node* X_A*, each of its neighbors* X_B **do**
    for *each subset* $\mathcal{S}$ *of k neighbors of* X_A **do**
      `// Influence test for doable node`
      if X_A *is doable* **then**
        Compute $do(X_A) \rightsquigarrow X_B|do(\mathcal{S})$;
        Remove $A \longrightarrow B$ from $\mathcal{G}$ if needed (if computation above indicates no causality link)
      else
        ;
      `// Independence test for non-doable node`
    Compute $Corr(X_A, X_B|\mathcal{S})$;
    Remove $X_A \longrightarrow X_B$ if variables are independent;
$k \leftarrow k + 1$;
Postprocess $\mathcal{G}$ to turn it into a DAG by following the method presented in Table 2.

Figure 5 illustrates the algorithm for our smart room example. It highlights one of the key limitation we face: some do-operations are not feasible, as they are too difficult or dangerous to enact or simulate. Here, this is the case for the temperature: one cannot simply set a room's temperature to a fixed value without modifying other variables. Therefore, we flag the corresponding node T as a *"non-doable"* (ND) node, and consider all outgoing arrows as *"ND-arrows"*. As the corresponding do-operations cannot be performed, these arrows cannot be removed directly (i.e. via direct causality tests).

However, it is still possible to process them. Similar to the approach using the PC-algorithm [45], we can test simple variable correlations (using chi square independence test) over these arrows to potentially remove them: if no correlation is found, then the ND-arrow can be removed. For the remaining connections, we use the overall topology of the causal graph learned so far. As we

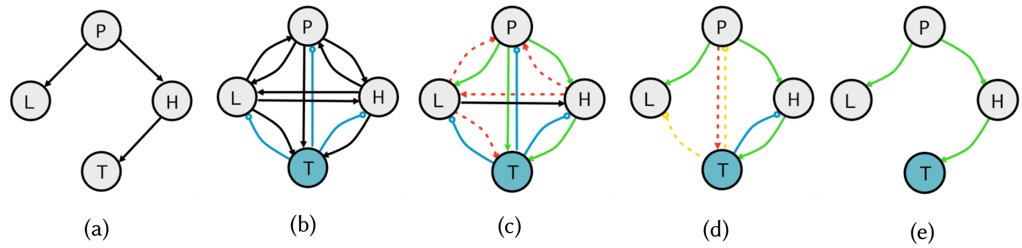


Fig. 5. Principle of our algorithm. (5a) Ground truth causal model. (5b) Initialization to a fully-connected graph over the variables. Non-doable nodes and corresponding arrows are shown in blue. (5c) Causality tests with interventions remove the red arrows. (5d) Arrows are removed, either by independence test (in yellow) or causality test (in red). (5e) The final graph is obtained by removing cycles, prioritizing the removal of non-doable arrows.

know that the causal graph is, by definition, acyclic, it is possible to further trim down some arrows. When considering a cycle, the method is to remove the arrow that we are least confident in, for instance a non-doable arrow that could not be tested via do-operation. The different configurations and the corresponding processing are listed in Table 2.

This postprocessing, which occurs at line 14 in Algorithm 1, returns a directed acyclic graph that represents the causal model compatible with the observations and the interventions performed on the target system. The resulting causal graph can then be used for various kinds of analysis and control functions. For instance, causality analysis can provide the basis for explanation and abductive inference, and can be integrated as such into broader explanatory systems [19]. A second prospect, that we detail, is to learn a Bayesian Network over this causal graph, thus resulting in a Causal Bayesian Network (CBN).

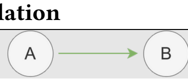
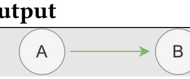
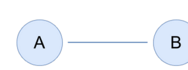
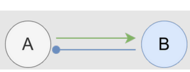
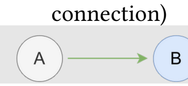
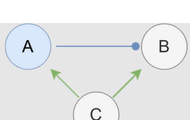
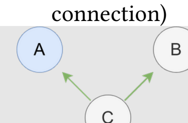
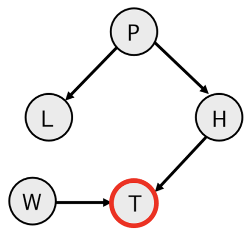
Relation	**Description**	**Output**
	A is the cause of B	
	Exactly one of the following holds: • A is the cause of B • B is the cause of A • No connection (spurious correlation)	 (flagged undirected connection)
	A is the cause of B. The other direction does not need further testing	
	A is **not** the cause of B. Exactly one of the following holds: • B might be the cause of A. • No connection (spurious correlation)	 (flagged directed connection)
	A confounder C is known, so we can delete the correlation between A and B	

Table 2. The different possible configurations for processing the remaining ND-arrows. Causal arrow are shown in green, ND-arrow are in blue. Red arrow represent arrow remove by causality test

3.3 Learning a CBN



W	0	0	1	1
H	0	1	0	1
T=0	p_{00}	p_{01}	p_{10}	p_{11}
T=1	$1-p_{00}$	$1-p_{01}$	$1-p_{10}$	$1-p_{11}$

Fig. 6. Example of a small Bayesian network. The probability table for node T is displayed.

Once the structure is known, learning the Bayesian Network corresponds to learning its parameters [24]. Here, we can use a conventional approach and rely on a maximum likelihood estimator: we estimate the probability of a variable's value based on past observations of this variable depending on its parents' values. For instance, considering the graph in Figure 5e, which corresponds to the output of the algorithm for the room example, we can compute p_{00} ($p_{(W,H)=(0,0)}$) with:

$$p_{00} = \frac{N_{T=0,(W,H)=(0,0)}}{N_{T=0,(W,H)=(0,0)} + N_{T=1,(W,H)=(0,0)}}, \tag{1}$$

where $N_{T=i,(W,H)=(j,k)}$ denotes the number of past occurrences of $(T, W, H) = (i, j, k)$, Figure 6.

However, this conventional approach may face issues with variable states that have few occurrences. The introduction of the do-operation removes this limitation, as it becomes possible, for doable nodes, to generate all kinds of situations required to observe the outcome and estimate missing parameters of the Bayesian Network.

4 EXTENSIONS TO THE ORIGINAL METHOD

This section presents the three contributions we propose for extending our CBN learning method and for improving its performance and accuracy. More precisely, an important issue that may occur when targeting large systems is our method's algorithmic complexity: as it starts from a fully-connected network and then discards relations progressively, the method requires a number of operations proportional to $O(n^2)$ for a system of n variables. To circumvent this, we propose two novel improvements. First, we discard uncorrelated or weakly-correlated relations early-on, before performing costly causality-checking interventions (Cf. 4.1). Second, we propose to capitalize on the system's underlying structure, if available, to further discard improbable relations (cf. 4.2). Another key issue encountered in our initial algorithm is the handling of non-doable nodes: as they do not allow interventions, we cannot infer causal relations for them. To tackle this issue, we propose to act on such non-doable nodes indirectly, via doable nodes that are causally linked to them (Cf. 4.3).

4.1 Early pruning of uncorrelated relations

The original method presented in Algorithm 1 is initialized with a fully connected network. This implies that the number of interventions required during the learning process increases (at least) proportionally with the number of variables (n). This may become overly costly and seriously hamper scalability. Indeed, interventions are resource- and time-consuming operations (e.g. having to simulate and assess system behavior in counterfactual situations). In contrast, finding correlations in observed data is relatively fast. Hence, a first improvement to the speed of our Causal Learner algorithm is to initialize the graph to correlated variables only. Our intention here is that, even if all correlations are not yet confirmed as causal relations, a preliminary pruning (based on correlation strength only) can remove a significant amount of relations, hence improving learning performance.

4.2 Using aggregate variables to scale-up to large systems

As above, considering large systems with more than hundreds or thousands of variables (n) implies having to test a large number of relations between these variables ($\approx n^2$). Moreover, any change in the targeted system implies starting over the entire learning process. Early pruning alone will not suffice to handle these issues. Hence, we propose an additional novel extension to address them. Namely, we propose to rely on pre-existing knowledge on the system's organization to remove unlikely relations. To facilitate viability, we recommend to only rely on those system organisational features that are unlikely to change very often (e.g. the hierarchical structure of smart micro-grids [15]).

We assume here that most large systems will feature a modular or multi-scale structure to ensure viability in ever-changing environments (e.g. [10]). For instance, let us consider a smart micro-grid over an entire neighborhood of smart houses, equipped with various smart devices. This is a large system, with thousands of variables, yet featuring a clear organization: variables within each home may all be correlated, while inter-home influences are scarce, only affecting a handful of variables – and that, often in an aggregate, indirect way. E.g., the individual state of each electric device within a house may only have a negligible effect on the neighboring house; however, the entire state of the house (e.g. all electric devices together) may affect its neighbor (e.g. cause a power outage). This can be modeled by the causal graph as shown in fig. 7a, with several dense sub-networks connected via scarce connections. Hence, based on knowledge of underlying system structure, we introduce aggregate variables (e.g. house power) that summarize all relations between variables of different system sub-graphs (fig. 7b). An aggregate variable can be understood as a low-dimensional representation of the state of its corresponding sub-network. Using this constraint, all influences between sub-networks must go through these aggregate variables. This, in turn, reduces the number of relations between sub-networks and provides a significant improvement over a fully-connected graph.

We note here that even though the resulting graph is simplified, it may still remain significantly complex, as we expect each organisation level, or scale, to feature a non trivial structure. We reiterate the fact that introducing such aggregate variables is only possible when these aggregates can subsume the overall influences that all fine-grained variables which they represent can have on any other external variables to which they are causally related. Hence, we expect domain experts to establish the applicability and concrete instance of this method, case-by-case.

In future work, we may further extend this improvement by developing a multi-scale learning approach. First, we learn each identified dense sub-network using the method described in Section 3.2. As influences between sub-networks are ignored at this point, this step of the process may be performed in parallel, improving performance. In the next step, we introduce aggregate variables to subsume the representation of each sub-network. Then, the relations between sub-networks are determined by repeating the Causal Learner algorithm on the aggregate variables only. This modularity allows to learn different system sub-graphs independently; and then to update and reuse them separately from each other. This would further improve the method's scalability with the number and frequency of system changes that require dynamic re-learning of causal links.

4.3 Refining the causal graph via indirect interventions

We propose using indirect interventions to help determine the causality relations of non-doable variables. To illustrate this approach, consider the two networks shown in Figure 8. In the first

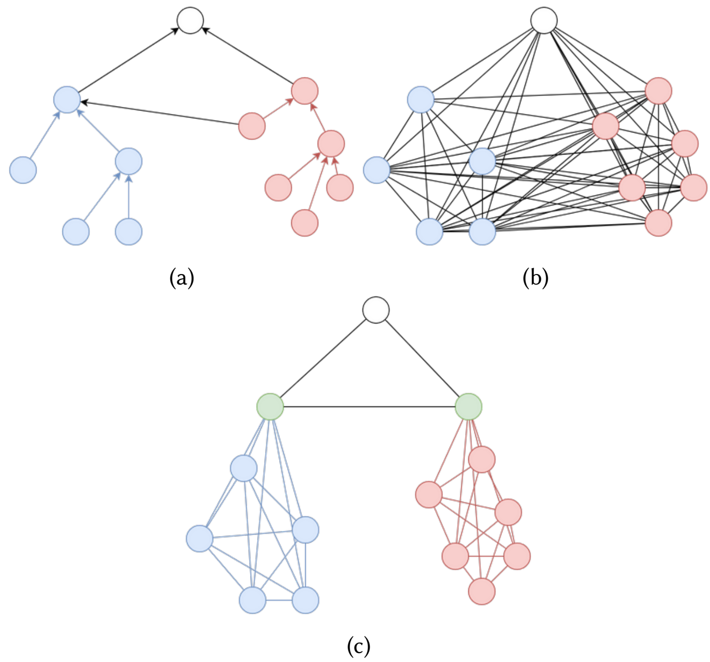
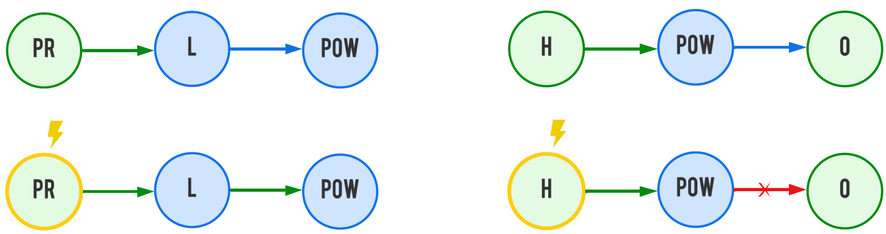


Fig. 7. Handling of large organized systems. (a) The ground truth model of the system is composed of two dense sub-networks with scarce interconnections between them. A naive implementation of our Causal Learner requires to consider the fully-connected graph, which is unnecessarily complex(b). Acknowledging the hierarchical organization of the system by introducing aggregate variables (in green) and limiting full connection *between*

sub-networks limit the number of relations to consider (c).

case (Figure 8a), the original Causal Learner (top) finds the following structure: $Pr \longrightarrow L \longrightarrow Pow$, indicating that a Presence sensor Pr triggers a Light L, which in turn impacts a Power meter Pow. As the Light L is presence-controlled and hence non-doable here, its influence on the power Pow cannot be tested (i.e. blue arrow $L \longrightarrow Pow$). This problem can be fixed with the proposed refined method (bottom): as Pr is doable we can act on it and learn that it is causally related to L. This allows us to later operate on L indirectly by performing a *do* on Pr. Doing this confirms the relation $L \longrightarrow Pow$ (green arrow) and solves our previous problem. In the second case (Figure 8b), a first pass of the algorithm (top) identifies the structure $H \longrightarrow Pow \longrightarrow O$, suggesting that a heater H impacts power consumption Pow, which in turn influences outdoor temperature O. As we cannot act on the power Pow directly, the latter relation is unconfirmed (blue arrow). Again, using indirect intervention on Pow via H (bottom) allows to discard this latter relation from the graph (red arrow $Pow \longrightarrow O$).

We generalize and formalize the proposed method refinement as follows (Algorithm 2). As we learn a Bayesian Model, it is possible to consider indirect interventions. Thus, considering an already established Causal Bayesian Graph $\mathcal{G}$ and a non-doable node $X \in \mathcal{G}$, we define $Anc(X)$ as the subset of doable nodes in G that are causal ancestors of X (this subset is minimal with regards to intervention: for all ancestor Y of X, there is no doable node on the path from Y to X). As influence coefficients are known, it is possible to indirectly intervene on X by intervening on its ancestors.



(a) (b)

Fig. 8. Overview of indirect interventions for the Causal Learner algorithm refinement. Blue nodes and relations represent unconfirmed correlations obtained after a first algorithm pass. In a second pass, based on (indirect) *do* operations (Yellow), they are to be either confirmed (Green) or discarded (Red). (a) *Confirming a link.* Top: the first pass finds the structure $Pr \longrightarrow L \longrightarrow Pow$. Pr is doable and hence $Pr \longrightarrow L$ can be confirmed (green); but L is non-doable hence $L \longrightarrow Pow$ cannot be confirmed (blue). Bottom: we operate on L indirectly via Pr, thus confirming $L \longrightarrow Pow$ (green). (b) *Discarding a link.* Top: a first pass identifies the structure $H \longrightarrow Pow \longrightarrow O$. As before, the latter relation is unconfirmed (blue). Bottom: we act on Pow indirectly via H and consequently discard $Pow \longrightarrow O$ from the graph (red).

Algorithm 2: The refined Causal Learner algorithm: non-doable nodes can be modified indirectly by using the already learned Bayesian Network. The process stops once there is no improvement detected between two iterations. $ND(\mathcal{G}$ denotes the non-doable nodes of $\mathcal{G}$, and $out(X)$ the edges originating from X in $\mathcal{G}$.

init
  $\mathcal{G} \longleftarrow$ First pass of the Causal Learner algorithm
for $X \in ND(\mathcal{G})$ **do**
  $A_X \longleftarrow Anc(X)$;
  if *Intervention on* A_X *can influence* X **then**
    Use Bayesian intervention to indirectly intervene on X;
    for $e \in out(X)$ **do**
      Test e using χ^2 ;
      Remove or confirm e in $\mathcal{G}$
return $\mathcal{G}$

This approach allows to refine the model by confirming causal relations and by discarding spurious correlations.

However, limitations can arise when considering causal structures such as "diamonds" [1], where it may be impossible to properly isolate some variable using this indirect intervention method. This phenomenon limits the possible reach of our current method and will be further studied in future work.

5 EXPERIMENTS AND RESULTS

To illustrate and evaluate the efficiency and efficacy of our proposed contributions, we carried out two main experiments. The first experiment (Section 5.1) relies on a predefined Bayesian Network

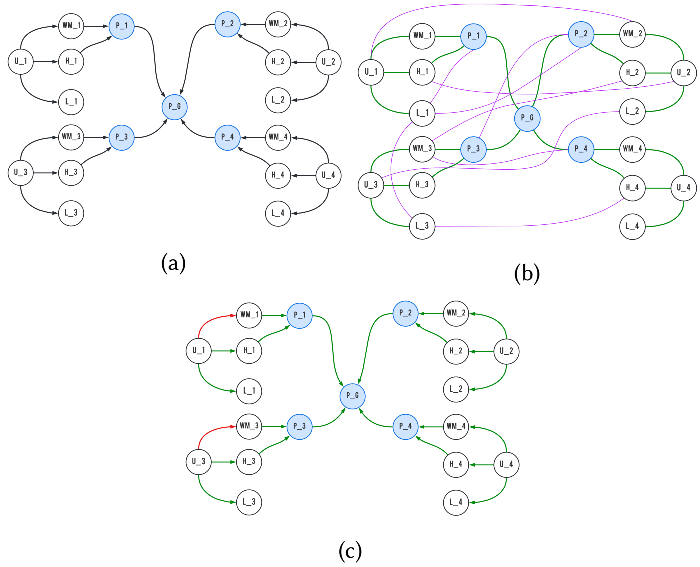


Fig. 9. First experiment based on predefined Bayesian Network. (a) Ground-truth Bayesian network used for data generation, showing 4 houses, with non-doable variables in blue. (b) Initial graph used for the Causal Learner algorithm, after pruning of uncorrelated variables: purple relations correspond to spurious correlations. (c) Result of the Causal Learner algorithm: green arrows are correctly identified relations, the algorithm misses red arrow relations.

and an associated ground truth. It tests our method's ability to confirm causal relations and discard the non-causal ones. Here, we focus on showing the impact that early pruning (Section 4.1) may have on the speed of the learning method, by reducing the number of variable relations to check. The second experiment (Section 5.2) is based on a smart home simulation platform, iCASA[2]. The aim of this experiment is twofold. First, it tests the impact of using aggregate variables (Section 4.2) on the reduction of relations to verify, within a more realistic scenario. Second, it tests the impact of using indirect do operations on the accuracy of the resulting CBN, by allowing to verify the causality of non-doable variables.

5.1 First experiment: A Bayesian Network as the environment

Setup. This experiment is based on data that are generated by means of a Bayesian Model, shown in Figure 9a. This model represents the ground truth, which is hidden to the Causal Learner agent. We flag some of its variables as non-doable nodes, to recreate a realistic setup.

The situation simulated here corresponds to several houses with temperature regulation systems and other electric appliances, which are connected to the same electric grid. I.e., within each house i, the user U_i may impact the state of washing machines WM_i, heaters H_i and lights L_i, which in turn impact power consumption P_i. We note here that the Lights' influences on Power consumption is ignored, for simplicity, as relatively minor compared to the Heaters (Figure 9a). House consumption P_i is an aggregate variable impacting grid consumption P_G. This hierarchical setup helps to show how the introduction of aggregate variables improves learning performance.

[2]iCASA homepage: http://adeleresearchgroup.github.io/iCasa/snapshot/index.html

Results. We conducted experiments on systems composed of 1 to 10 similar houses, each one containing 5 variables.

First, we study the effect of the "early pruning" approach (cf. Section 4.1). A first step to limit the number of required interventions is to discard variables that appear uncorrelated in observation data: this operation alone reduces the number of possible connections in the case of 10 houses from 2550 down to 85 connections (Figure 10). Hence, in this example, the pruning operation enables the algorithm to scale linearly, following the increase in size of the ground-truth model. That is, the number of ground truth connections grows linearly with the number of houses, as the addition of each house only adds a single causal link to the graph: from the aggregate power variable of that new house (P_i) to the micro-grid's global power variable (P_G). The example in Figure 9b illustrates the case of 4 houses. The early pruning allowed to reduce the number of edges to test while avoiding to delete the valid causal edges. There are still some connections that are pure correlations because the test has a low threshold to avoid removing causal edges with weak influence. But they will be deleted later by the algorithm thanks to the interventions on the environment illustrated in Figure 9c.

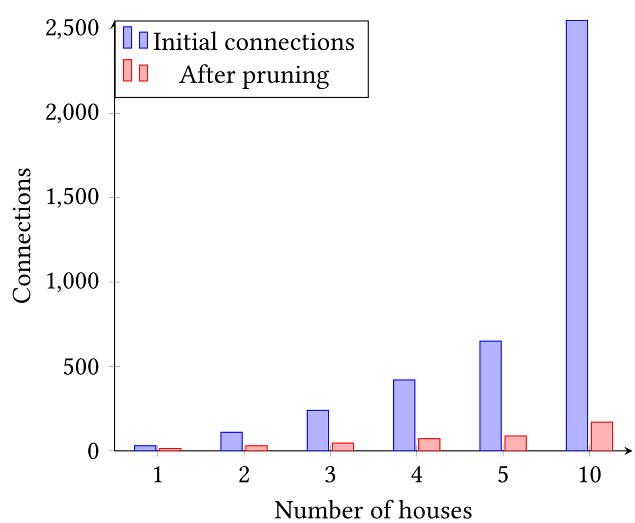


Fig. 10. Effect of correlation pruning on the number of connections. Each house is modeled with 5 variables.

The 4 houses in the example have different behaviors. The red arrows, which indicate where our method failed to identify a causal relation, can be explained by the rarity of the events. In the case of the washing machine, our model implements that, when the user is home, the probability of using the washing machine remains low: $\mathbb{P}(WM) < 0.1$. Due to this low probability, do-operations on the user's presence do not always result in the washing machine being turned on, which makes this relation hard to detect. This effect can be mitigated by increasing the number of interventions we perform, enabling the detection of less frequent interactions.

5.2 Second experiment: data from a Smart Home Simulator

Setup. In a second experiment, we generate more realistic measurements by using a smart home simulator – iCasa [29]. Figure 11a depicts the initial setup of our experimental scenario: a house with four rooms, each room being equipped with a thermostat (i.e. PID controller regulating room temperature around a targeted value). In addition, lights and windows can be controlled

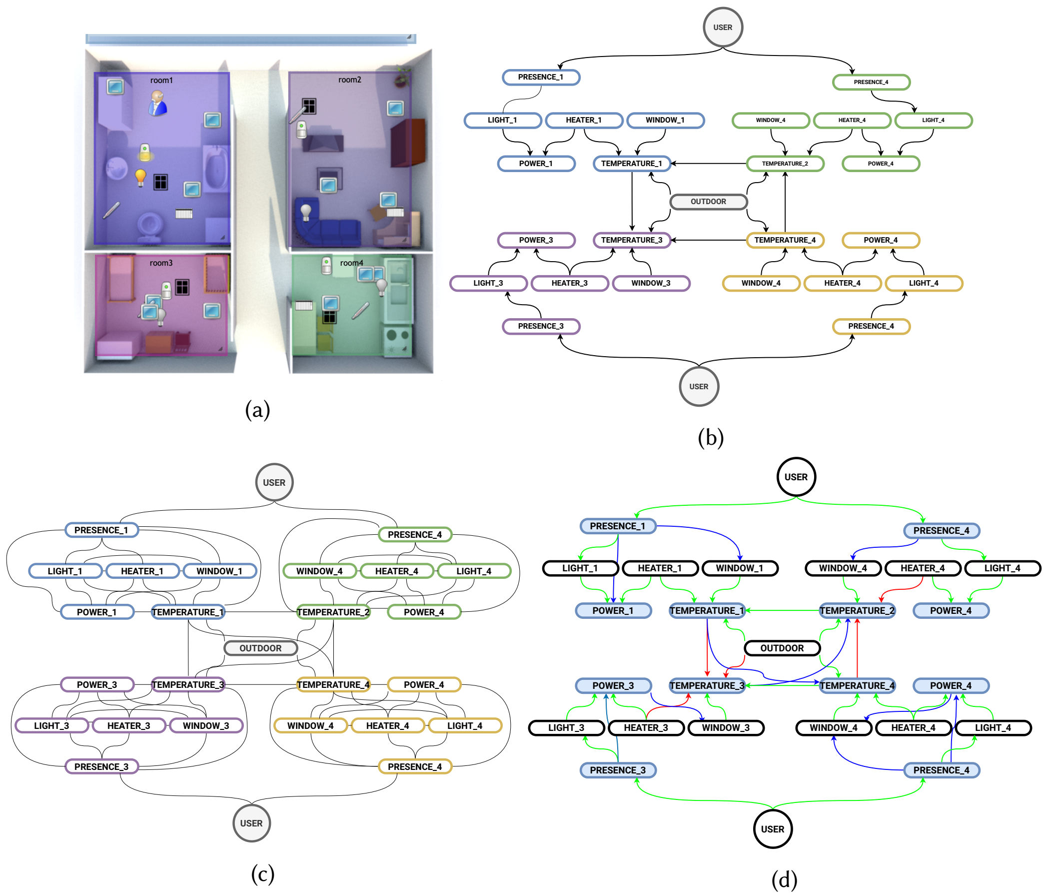


Fig. 11. Second experiment. (a): view of the iCasa web interface showing the setup: four rooms with automated control systems. (b): the corresponding ground truth causal graph, showing the temperature interactions between rooms. The possible division of this graph into four sub-networks, one for each room, is color-coded. (c): using aggregate variables to incorporate this knowledge into the Causal Learner algorithm reduces the number of relations to consider in the subsequent process. (d) output of a first pass of the Causal Learner algorithm, when initialized using aggregate variables, which decreases threefold the number of operations compared to the original method.

and monitored. Temperatures in adjacent rooms influence each other, modeling heat transfer through the walls. The resulting ground-truth model is depicted in Figure 11b, showing inter-room temperature influences. We note here the introduction of room temperatures as aggregate variables (i.e. $Temperature_i$ for room i). We also note that, due to the constraint of CBN being acyclic graphs, inter-room temperature influences are unidirectional. Finally, to implement do and indirect-do operations, we relied on iCASA's REST API, which allows us to intervene on simulated devices and to measure the consequences, at runtime.

Results. Here, using aggregate variables significantly decreases the number of relations to test. As the system contains 26 variables, the fully connected graph that would be used in the original version of the Causal Learner algorithm would contain 702 relations. Instead, we separate rooms

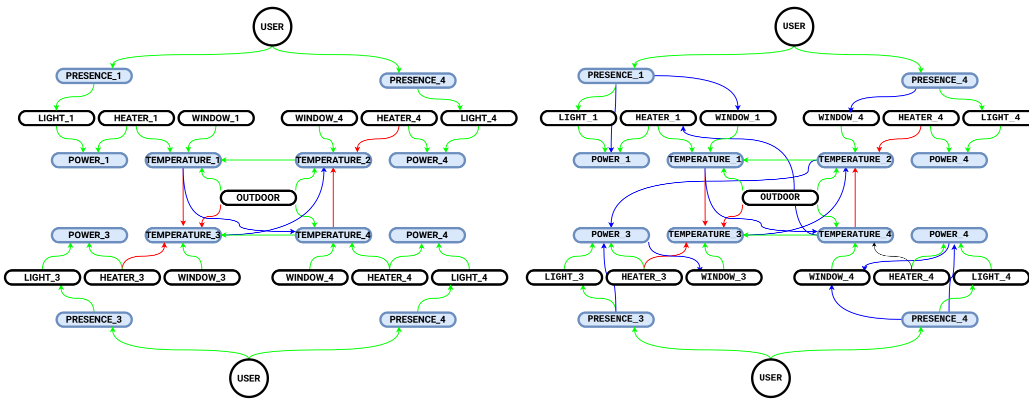


Fig. 12. The refinement operation on the graph from Figure 11d improved the accuracy of the prediction by deleting 7 non-doable arrows (a). When compared to the result of the original method (b), the new method is significantly closer to ground truth and down from 11 to 2 non-doable unconfirmed relations.

intro distinct sub-networks, which are weakly-coupled via aggregate variables (room temperatures) – cf. Figure 11c. This approach reduces the amount of initial relations to check to 206 – hence merely 30% of the initial relations number. This threefold improvement leads to a faster learning process and results in the output shown in Figure 11d. Due to the high proportion of non-doable nodes (in blue), many uncertain non-doable relations remain, shown as blue arrows. Green arrows are correctly identified relations, while red arrows are relations that were not found by the algorithm.

We note here that while some of the relations between aggregate room temperatures were found (i.e. between rooms 1 and 2; and between rooms 3 and 4) other equivalent relations were not (e.g. between rooms 1 and 3; and between rooms 2 and 4). The same occurs for some relations between heaters and the corresponding room temperatures. This is most likely due to temperature variability over time, making it more difficult to determine causal effects. More measurements over larger time extents would probably eliminate these effects (future work). However, the purpose of the presented experiments was mainly to show how the proposed extensions improve the speed of the learning process, while not harming its accuracy. This is the case here, as results obtained with the new method are no less accurate than those of the original method (Figure 12b); and are actually more accurate, as discussed next.

To improve the result, we use the refinement technique of indirect do-operations presented in **??** 2. Figure 12a shows the final output of our new method, after applying the refinement pass to the output from Figure 11d. Thanks to indirect do-operations, 5 non-doable arrows were discarded. This final output is closer to the ground-truth structure in Figure 11b, showing a clear improvement compared to the output of the original method (Figure 12b), while remaining faster overall.

5.3 Discussion

Illustrated advantages. Experimental results illustrate how the proposed improvements to our causality model generation approach may actually help to: on the one hand, decrease the number of required "do" operation significantly; and, on the other hand, allow to act on "non-doable" variables via indirect "do" operations. Namely, the first experiment (subsec. 5.1) shows how early pruning of non-correlated variables (as identified by an ML approach) reduces the number of variable inter-connections to analyse from 2550 to 85, hence to a mere 3.3% of the initial figure – for 10 houses with 5 variables each, specified via a formal graph model. It is important to note here that the scope of our approach is limited to cases where sufficient data is available to establish correct correlations. Otherwise, pruning non-correlated variables may actually remove variables that are causally related. Other causality detection methods may be employed instead to handle cases where correlation is not an option (e.g. abduction technique based on memorable event detection [20]).

The second experiment (subsec. 5.2) shows how the use of domain knowledge – in terms of multi-scale system architecture – allows to introduce aggregate variables and reduce the total number of inter-connections (including interconnections between non-aggregate variables) from 702 to 206, hence merely 30% of the initial figure – for a smart home with four adjacent rooms, containing 26 variables in total and simulated via iCASA platform.

Finally, the second experiment (Section 5.2 also shows how performing indirect "do" operation on non-doable variables allowed removing 5 non-doable arrows from the causality model 12a, hence bringing it closer to the ground truth (Figure 11b than the original method (Figure 12b).

Surely, the exact figures in terms of connection reductions and increased accuracy will depend on the particular characteristics of each targeted system. However, the experiments illustrate the extent to which the proposed improvements can be beneficial, even at a very small system scale. We consider this encouraging and expect such advantages to be amplified when applied to larger-scale systems.

Simulating "do" operations. As previously mentioned, performing "do" operations via real actuators may raise various concerns in terms of safety, security, energy consumption and user nuisance. When possible, performing such operation on a virtual environment, e.g. simulation or digital twin [46], would alleviate this issue. Certainly, in many cases, obtaining such digital twin for highly diverse managed systems like the ones targeted here may be as difficult a challenge as obtaining the causal model. That said, we argue that a digital twin, when available, may not provide the required causal information for SAS decisions or for providing explanations to users [19].

This is the case when simulation models or digital twins represent a low abstraction level, or finer granularity, of the targeted system – e.g. a smaller, simplified, real CPS version of a targeted CPS; or, an analogical model of a CPS, such as a power grid. This does not provide direct knowledge of high-level causal relations between coarser grained system components. For instance, a smart micro-grid modelled via differential equations can predict a blackout when too many consumers are plugged-in, yet cannot say which SAS action may have caused the blackout and what its purpose was. The larger the system, the more cumbersome the high-level causality tracking would become when only based on low-level system models. Higher-level causal models such as the one proposed here may rely on such low-level, forward-tracking digital twins to provide answers more suitable to SAS reasoning (e.g backwards-tracking for fault detection) or to user understanding (e.g. explainability).

The utility of producing causal models by relying on existing digital twins may also occur when these latter are provided as black-box facilities by system providers (e.g. embedded within each smart home by building firms or promoters). As above, the proprietary digital twin may allow for forward-tracking (predictions) but offer no backward-tracking (cannot identify causes to detected anomalies). Providing a causal model as proposed in our approach can extend the digital twin's applicability to third-party SAS or Explanatory AI systems. Surely, online adaptability based on such proprietary digital twins can only go as far as the adaptability support of the digital twin (i.e. if the digital twin does not adapt to runtime changes, the causal model cannot be updated either). Finally, our proposal could use a hybrid approach and rely on partial digital twins whenever available while considering real actions otherwise.

Limitations in terms of cyclical causality chains. A current limitation inherent in the representation of domain knowledge as causal models is the difficulty of handling cyclical relations, or loops. This is exemplified in the second case study (cf. Section 5.2, Figure 11), where adjacent rooms within a smart home should influence each-other mutually in terms of temperature, via bi-direction heat transfers. Such bi-directional causal relation is the smallest instance of a causal cycle. As shown in Figure 11b, our current causal model only supports uni-directional relations. This could suffice in some cases, like the one illustrated here, when some rooms are always warmer than others. Yet, in general, uni-directional causal links necessarily induce limitations. This is especially the case in SAS, where (causal) feedback-loops are the norm.

While out of this paper's scope, future work will aim to address this limitation as follows. One idea is to extend causal models to allow for cyclical causal chains, but only considering one direction at the time when employing the causal model for SAS reasoning. The selected direction would depend on the actual variable in the cycle that is observed and analysed, in terms of its potential causes, at any one time. Here, special attention must be paid to the cause-tracking algorithm so as to avoid entering infinite loops. For instance, reasoning could halt upon detecting a repeating, cyclical causal pattern.

A second, complementary idea would be to detect such causal cycles and replace them by aggregate variables, when suitable, as already proposed in the current contribution (e.g. including a single temperature aggregate variable for the entire house). While this approach does not help track the causality between finer-grade variables within a cycle (e.g. influences between individual room temperatures) it can help track the impact that the entire cycle may have on variables outside the cycle (e.g. electricity consumption or outdoor temperature) – this is especially suitable for cases where the cyclic behaviour leads to a steady state (e.g. all rooms converge to similar temperatures). This is often the purpose of control feedback loops.

6 RELATED WORKS

Causal Learning has gained an increasing popularity over recent years, due to major works in causality, notably by J. Pearl [37] who developed the methods for do-calculus and intervention operations. In general, Reinforcement Learning can be considered as an application of causal theories, as the learning agent is able to intervene on its environment to better understand it, thus reaching the second step of the ladder of causality. This method led to state-of-the-art performance in selected applications, such as games [42], autonomous driving [17], autonomous navigation systems [43], user-centred smart systems [18] or self-adaptive systems in general [40]. However, Causal Learning remains implicit in Reinforcement Learning and the resulting models are not considered interpretable [8].

To this extent, Bayesian Models have been a popular tool for representing complex phenomena [22], and powerful models to predict behavior and gain insights via belief propagation operations [24, 36]. While expert-based structure is still widely used [24], different methods exist. [34] proposes to identify a minimal equivalence class between DAGs that would fit observational data. The result is a P-DAG (Partially Directed Acyclic Graph, i.e. a DAG with additional undirected edges) that can be used for learning Bayesian coefficients. As this method guarantees the minimalism of the proposed graph, it helps generate simple models, which often correlate with ground truth causal models. However, due to its lack of intervention operations, it cannot guarantee that the proposed relations correspond to true causal links. A refinement of this method, described in [33], completes this method by using a few selected intervention operations to disambiguate causal directions.

Structural Causal Models (SCM) already propose to integrate counterfactual reasoning into their learning process [31, 48]: they consider Causal Learner agents that are able to use their current causal model of the environment to test hypotheses and to improve performance. [31] illustrates this in a strategy game environment to propose justifications for the agent's behavior. [41] offers a more realistic classification task setup and shows that allowing do-operations improves the classifier's performance.

Our approach differs from these existing works by considering the existence and impact of non-doable variables, and by tackling the scalability aspect that is of prime importance in real-life cyber-physical systems. Moreover, our approach can capitalise on lessons learned from Interactive Reinforcement Learning [3], [27], by opening the possibility to ask users to perform some of the "do" actions; or asking user permission before performing these automatically. Finally, the proposed approach can be extended in the future to also build upon recent developments in Contextual or Context-aware Reinforcement Learning (e.g. [23], [44], [7]). This would allow to consider context-dependent changes in causal relations– i.e. where correlations that hold across several contexts may or may not be confirmed as causal relations, via do operations, when the context changes.

7 CONCLUSION

Cyber-Physical systems are required to become self-adaptable as they grow in complexity. Part of this goal can be achieved by allowing the system to build a causal model of itself. In this paper, we described a Causal Learner algorithm for constructing a Bayesian Causal Network that models a Cyber-Physical System, based on previous work. The method is based on the notion of interventions (do operations), which allows testing causal relations between variables and differentiating them from mere spurious correlations. We then identified the two main challenges within this initial approach: i) some nodes cannot be intervened upon (non-doable) hence preventing direct tests of their causality links; and ii) the number of relations to consider grows as the squared number of variables in the system, limiting the solution's scalability for large systems.

To tackle these challenges, we proposed three novel extensions to our original Causal Learner algorithm. First, we introduced an early pruning operation based on variable correlations that we perform on the initial graph. Its effect is to remove a significant number of inter-variable relations from the initial fully-connected graph, hence diminishing the number of costly do-operations. Second, we assumed that many large-scale systems feature a hierarchical organization (where dense sub-networks of variables interact scarcely with each other) and we capitalised on the fact that this organization can be known in advance. We proposed to take advantage of this knowledge to avoid initializing the Causal Learner algorithm with a fully-connected graph over all variables. This method can also further reduce the number of links to be analysed after early pruning is performed (hence providing an additional advantage over the first extension proposed). To implement this approach, we introduced aggregate variables that model all interactions between distinct sub-networks, hence reducing the number of relations to test.

Third, we introduced considerations for "indirect do-operations" in the Causal Learner algorithm to improve the quality of predictions and solve cases where the previous algorithm version was unable to determine whether a relation was a causal influence or a mere correlation.

We illustrated and evaluated the efficiency of our three algorithm extensions via two examples from the smart home domain: one theoretical, based on a predefined Bayesian network; and one simulation-based, using iCASA smart home platform. Our results confirm our intuition about the benefits of our proposal. They show a significant reduction in the number of relations to be verified via do operations, when using early pruning and aggregate variables; and an increased accuracy of the obtained CBN results, when employing indirect interventions. These results provide an encouraging basis for future experiments, which will focus on testing larger system scales and assessing the ability to obtain larger CBNs by composing modular sub-CBNs learned separately, in parallel.

REFERENCES

[1] Ergun Akleman, Stefano Franchi, Devkan Kaleci, Laura Mandell, Takashi Yamauchi, and Derya Akleman. A theoretical framework to represent narrative structures for visual storytelling. In *Proceedings of bridges 2015: mathematics, Music, art, architecture, culture*, pages 129–136, 2015.

[2] Jesper Andersson, Rogerio de Lemos, Sam Malek, and Danny Weyns. Reflecting on self-adaptive software systems. In *2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, pages 38–47, 2009.

[3] Christian Arzate Cruz and Takeo Igarashi. *Interactive Reinforcement Learning for Autonomous Behavior Design*, pages 345–375. Springer International Publishing, Cham, 2021.

[4] Karl J. Astrom and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers, 2nd Ed. (Ch3. 'System Modeling', pages 61–103).* Princeton University Press , 2021.

[5] Elias Bareinboim, Juan D Correa, Duligur Ibeling, and Thomas Icard. On Pearl's Hierarchy and the Foundations of Causal Inference. *ACM special volume in honor of Judea Pearl*, pages 507–556, 2020.

[6] Fernando Benayas, Alvaro Carrera, and Carlos A. Iglesias. Towards an autonomic bayesian fault diagnosis service for sdn environments based on a big data infrastructure. In *2018 Fifth International Conference on Software Defined Systems (SDS)*, pages 7–13, 2018.

[7] Carolin Benjamins, Theresa Eimer, Frederik Schubert, Aditya Mohan, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. Contextualize me - the case for context in reinforcement learning. *arXiv preprint arXiv:2202.04500*, 2022.

[8] Davide Castelvecchi. Can we open the black box of AI? *Nature*, 538(7623):20–23, 2016.

[9] Ranganai Chaparadza. Unifaff: a unified framework for implementing autonomic fault management and failure detection for self-managing networks. *International Journal of Network Management*, 19(4):271–290, 2009.

[10] Ada Diaconescu, Louisa Jane Di Felice, and Patricia Mellodge. Exogenous coordination in multi-scale systems: How information flows and timing affect system properties. *Future Generation Computer Systems*, 114:403–426, 2021.

[11] Simon Dobson, Roy Sterritt, Paddy Nixon, and Mike Hinchey. Fulfilling the vision of autonomic computing. *Computer*, 43(1):35–41, 2010.

[12] Kanvaly Fadiga, Etienne Houzé, Ada Diaconescu, and Jean-Louis Dessalles. To do or not to do: finding causal relations in smart homes. In *Proceedings of the 2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, pages 110–119, Online, 2021. IEEE. _eprint: 2105.10058.

[13] Jianqing Fan, Fang Han, and Han Liu. Challenges of big data analysis. *National science review*, 1(2):293–314, 2014. Publisher: Oxford University Press.

[14] N Fenton and M Neil. Making decisions: using bayesian nets and mcda. *Knowledge-Based Systems*, 14(7):307–325, 2001.

[15] Sylvain Frey, Ada Diaconescu, David Menga, and Isabelle Demeure. A generic holonic control architecture for heterogeneous multiscale and multiobjective smart microgrids. 10(2):1–21, 2015.

[16] Michael Grieves. Digital twin: manufacturing excellence through virtual factory replication. *White paper*, 1(2014):1–7, 2014.

[17] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020. Publisher: Wiley Online Library.

[18] Tom Hanika, Marek Herde, Jochen Kuhn, Jan Marco Leimeister, Paul Lukowicz, Sarah Oeste-Reiß, Albrecht Schmidt, Bernhard Sick, Gerd Stumme, Sven Tomforde, and Katharina Anna Zweig. Collaborative interactive learning - A clarification of terms and a differentiation from other research fields. *arXiv:1905.07264*, abs/1905.07264, 2019.

[19] Etienne Houzé, Jean-Louis Dessalles, Ada Diaconescu, David Menga, and Mathieu Schumann. A decentralized explanatory system for intelligent cyber-physical systems. In *Intelligent Systems and Applications. IntelliSys 2021*, pages 719–738, August 2022.

[20] Etienne Houzé, Jean-Louis Dessalles, Ada Diaconescu, and David Menga. What should i notice? using algorithmic information theory to evaluate the memorability of events in smart homes. *Entropy*, 24(346):1–20, 02 2022.

[21] Dominik Janzing and Bernhard Schölkopf. Causal inference using the algorithmic Markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010. Publisher: IEEE.

[22] Finn V. Jensen. *An introduction to Bayesian networks*, volume 210. UCL press, 1996.

[23] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of generalisation in deep reinforcement learning. *arXiv*, abs/2111.09794, 2021.

[24] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

[25] Samuel Kounev, Peter Lewis, Kirstie L Bellman, Nelly Bencomo, Javier Camara, Ada Diaconescu, Lukas Esterle, Kurt Geihs, Holger Giese, Sebastian Götz, and others. The notion of self-aware computing. In *Self-Aware Computing Systems*, pages 3–16. Springer, 2017.

[26] Jeff Kramer and Jeff Magee. A rigorous architectural approach to adaptive software engineering. *Journal of Computer Science and Technology*, 24(2):183–188, 2009.

[27] Samantha Krening and Karen M. Feigh. Effect of interaction design on the human experience with interactive reinforcement learning. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, DIS '19, pages 1089––1100, New York, NY, USA, 2019. Association for Computing Machinery.

[28] Christian Krupitzer, Felix Maximilian Roth, Sebastian VanSyckel, Gregor Schiele, and Christian Becker. A survey on engineering approaches for self-adaptive systems. *Pervasive and Mobile Computing*, 17:184–206, 2015. Publisher: Elsevier.

[29] Philippe Lalanda and Catherine Hamon. A service-oriented edge platform for cyber-physical systems. *CCF Transactions on Pervasive Computing and Interaction*, 2(3):206–217, 2020. Publisher: Springer.

[30] Philippe Lalanda, Julie A. McCann, and Ada Diaconescu. *The Decision Function*, pages 185–215. Springer London, London, 2013.

[31] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2493–2500, 2020. Issue: 03, pp 2493-2500.

[32] Pattie Maes. Concepts and experiments in computational reflection. In *Conference Proceedings on Object-Oriented Programming Systems, Languages and Applications*, OOPSLA '87, pages 147–155, New York, NY, USA, 1987. Association for Computing Machinery.

[33] Stijn Meganck, Philippe Leray, and Bernard Manderick. Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In Vicenç Torra, Yasuo Narukawa, Aïda Valls, and Josep Domingo-Ferrer, editors, *Modeling Decisions for Artificial Intelligence*, pages 58–69, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[34] Bojan Mihaljević, Concha Bielza, and Pedro Larrañaga. Learning Bayesian network classifiers with completed partially directed acyclic graphs. In *International Conference on Probabilistic Graphical Models*, pages 272–283. PMLR, 2018.

[35] Brice Morin, Olivier Barais, Jean-Marc Jezequel, Franck Fleurey, and Arnor Solberg. Models@ run.time to support dynamic adaptation. *Computer*, 42(10):44–51, 2009.

[36] Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3):139–188, 1986. Publisher: Elsevier.

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

[38] Judea Pearl and Dana Mackenzie. *The Book of Why.* Basic Books, New York, 2018.

[39] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms.* MIT press, 2017.

[40] Simon Reichhuber and Sven Tomforde. Active reinforcement learning - A roadmap towards curious classifier systems for self-adaptation. *CoRR*, abs/2201.03947, 2022.

[41] Jonathan G Richens, Ciarán M Lee, and Saurabh Johri. Improving the accuracy of medical diagnosis with causal machine learning. *Nature communications*, 11(3923):1–9, 2020. Publisher: Nature Publishing Group.

[42] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, and others. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017. Publisher: Nature Publishing Group.

[43] Nikita Smirnov and Sven Tomforde. Navigation support for an autonomous ferry using deep reinforcement learning in simulated maritime environments. In Galya Rogova, Alicia Ruvinsky, Tom Ziemke, Giancarlo Fortino, and Mary Freiman, editors, *IEEE Conference on Cognitive and Computational Aspects of Situation Management, CogSIMA 2022, Salerno, Italy, June 6-10, 2022*, pages 142–149. IEEE, 2022.

[44] Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9767–9779. PMLR, 18–24 Jul 2021.

[45] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search.* MIT Press, 01 1993.

[46] Fei Tao, Qinglin Qi, Lihui Wang, and AYC Nee. Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison. *Engineering*, 5(4):653–661, 2019.

[47] Nikolay Tcholtchev and Ranganai Chaparadza. Autonomic fault-management and resilience from the perspective of the network operation personnel. In *2010 IEEE Globecom Workshops*, pages 469–474, 2010.

[48] Sergei Volodin, Nevan Wichers, and Jeremy Nixon. Resolving spurious correlations in causal models of environments via interventions. *arXiv preprint arXiv:2002.05217v2*, 2020.