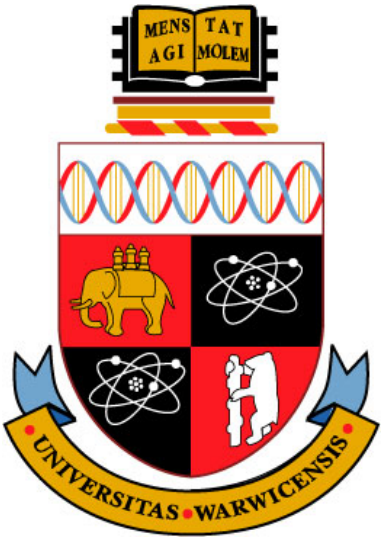


Causal Analysis on Chain Event Graphs for Reliability Engineering

by

Xuewen Yu

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Statistics

December 2021

THE UNIVERSITY OF WARWICK

Contents

List of Tables

List of Figures

List of Algorithms

Acknowledgments

First and foremost I am deeply grateful to my supervisor Prof. Jim Q. Smith for his generous and continuous support throughout my PhD. His advice on my academic research and writing has been very helpful. He has been always patient, caring and encouraging, especially during the pandemic.

I would also like to thank my panel members Prof. Chenlei Leng and Prof. Martyn Plummer who offered valuable comments on my paper and thesis. I want to thank Dr Linda Nichols for helping us get access to the data and understand the data. Special thanks to National Grid for providing the data.

My appreciation also goes out to the Engineering and Physical Sciences Research Council (EPSRC) and the Statistics Department who offered grant for my research.

Finally and most of all I appreciate all the support and encouragement I received from my beloved parents.

Declarations

I hereby declare that this thesis is based on my own research, except when stated otherwise. Some of this work has been published albeit in a different form as follows: the material in Section 2.3.2 and Section 2.3.3 has appeared in the *Proceedings of the International Symposium on Reliability Engineering and Risk Management* under the title "Bayesian Learning of Causal Relationships for System Reliability". This is listed in the bibliography and cited throughout the thesis as Yu et al. [2020]. The material related to the routine intervention and the missingness chain event graphs forms the basis for a second paper "Causal Algebras on Chain Event Graphs with Informed Missingness for System Failure". This has been published by the *Entropy* as a special issue in *Causal Inference for Heterogeneous Data and Information Theory*. This is cited as Yu and Smith [2021*a*] in this thesis.

The material in Section 2.3 has been expanded to form a third paper "Identifying Causal Effects on Chain Event Graphs for Remedial Interventions". This has been submitted to *Journal of Causal Inference* and under review.

I was also engaged in another project which performed a Bayesian decision analysis for the strategies of coronavirus. The content of this project does not appear in this thesis and is not cited. The paper for this project is:

> Strong, P. , Shenvi, A., Yu, X., Wynn, H.P., Papamichail, N. and Smith, J.Q. (2021) "Building a Bayesian Decision Support System for Evaluating COVID-19 Countermeasure Strategies" *Journal of the Operations Research Society* (to appear).

This thesis has not been submitted for examination at any other university.

Abstract

Various graphical models have been utilised in reliability literature to express the qualitative aspect embedded in certain hypotheses about how a system might fail. There is a wide range of research that translates domain expert beliefs to Bayesian networks (BNs), fault trees and so on [Bedford et al., 2001]. However, many conventional tree-structured analyses designed to demonstrate how systems can fail in reliability theory are not embellished with probabilities and conditional independence statements. Here we apply the Chain Event Graph (CEG) which is a probabilistic graphical model derived from an underlying event tree. This class of model retains the advantages of both events trees and BNs. So a CEG can chronologically represent sequences of events along the paths and model conditional independence. In particular, the CEG model generalises the discrete BNs. A BN can be transformed to an equivalent CEG. Compared with BNs, the class of tree-based CEGs have richer semantics for representing context-specific dependencies. For example, given non-extreme weather and temperature, the failure of a system depends on the condition of sub-systems A and B. When having extreme weather, the failure of this system depends only on the temperature. This can be easily represented by a CEG, but it is non-trivial to capture the sample space structure of this scenario by a BN. I show in this thesis that these semantics are rich enough to represent the unfolding of the asymmetric failure processes or deteriorating processes and also to provide a formal framework around which to define the intervention calculus required for this domain.

Over the last 40 years statistical analyses which embed causal reasoning have been shown to improve the predictive inference and the efficiency in decision making in various fields, such as economics, medicine, public health and reliability [Langseth and Portinale, 2007]. There is almost no research relevant to such analyses which use probability trees – widely used in reliability – as the foundational structural framework with which to explore putative causal hypotheses and to define appropriate causal algebras.

Therefore, in Chapter 2, we demonstrate how an event tree can be customised for modelling causes of system failures. A CEG derived from this tree is then constructed which faithfully represents typical classes of model found in reliability as causal probability models. Causal algebras associated with different domains have already been successfully developed for the CEG. However we find that these are

not usually suitable for the types of causal interventions appropriate to reliability theory. Our main contribution here is customising causal algebras for two types of domain-specific interventions – the remedial intervention and the routine intervention – with semantics of CEGs. The former is associated with maintenance performed after observing failures which fixes the root causes of the observed failures. The latter is associated with routine maintenance which is scheduled to prolong the system's lifetime and to prevent failures. We show that the manipulations in response to these domain-specific interventions can be imported into CEGs in a simple and transparent way. We can then use the developed causal algebras to study the effects of such interventions. In particular, we have been able to adapt the algorithms originally developed by Pearl [2009] to determine when certain causal effects are identifiable and produce explicit formulae for these effects as a function of these interventions bespoke to this application and the CEG representing the failure processes. Thwaites [2013] has shown that Pearl's back-door theorem [Pearl, 2009] can be extended on CEGs to identify effects of controlling an event. Here, under the two new types of intervention regime, we have more complicated types of manipulations than controlling a single event. We show that the back-door theorem can still be adapted to estimate effects of these new interventions even when data are only partially observed.

Although there are confidentiality constraints that have precluded me sharing fully its contents, this thesis is informed by a dataset based on engineer reports of the failure and maintenance of electrical transformers. These documents consist of well-structured ordinary data and free texts. The free texts are informative about how engineers believe a system may fail and how the system can be repaired or restored. So we have available to us documentation of how engineers reason causally where this reasoning is encoded within the natural language descriptions. In order to automate the process of causal discovery from these free texts onto a CEG for this system, it is required to design algorithms which enable us to extract and embed these causal hypotheses from the texts. In Chapter 3, we propose a new sequence of algorithms that are able to perform this extraction and provide an innovative hierarchical framework with two levels which can be used to embed them on a CEG. The surface level registers the extracted causal events while the deeper level can be described by a causal CEG. The complexity of the analysis is increased when data is only partially observed or missing in embedding causal dependencies and making predictive inference about causal effects in this domain. However, we show in Chapter 4 that this issue can be successfully addressed with the bespoke causal algebras within the proposed causal framework.

In Chapter 5, we show predictive inference can be improved by incorporating the causal algebras established for the remedial intervention and the routine intervention. We also design an algorithm to map free texts onto a CEG using the hierarchical framework developed in Chapter 3 and evaluate the performance of this algorithm using synthetic experimental data. In the last chapter, we give a brief discussion on possible extensions of the current work.

Abbreviations and Notation

Abbreviations

ABAO	As Bad As Old
AGAN	As Good As New
ARA	Arithmetic Reduction of Age
BDD	Boolean Decision Diagram
BN	Bayesian Network
CAEVO	CAscading EVent Ordering
CEG	Chain Event Graph
CM	Corrective Maintenance
CMC	Causal Markov Condition
DAG	Directed Acyclic Graph
DCEG	Dynamic Chain Event Graph
EM	Expectation-Maximisation
FMEA	Failure Mode and Effects Analysis
GN	Global Net
HMM	Hidden Markov Model
HSMM	Hidden semi-Markov Model
LDA	Latent Dirichlet Allocation
MAP	Maximum A Posteriori
MAR	Missing At Random
MACR	Missing Completely At Random
MCMC	Markov Chain Monte Carlo
M-CEG	Missingness Chain Event Graph
M-GN	Missingness Global Net
MNAR	Missing Not At Random
NER	Name Entity Recognition
NLP	Natural Language Processing

PM	Preventive Maintenance
POS	Part-of-speech
RBN	Recursive Bayesian Network
RCMC	Recursive Causal Markov Condition
RDCEG	Reduced Dynamic Chain Event Graph
RMC	Recursive Markov Condition
SCM	Structural Causal Modeling
SGT	Supergrid Transformer
VN	VerbNet
WN	WordNet
WOD	Work Order Description
WOED	Work Order Extended Description
WRED	Work Order Requested Extended Description

General Notation

$p(\cdot)$	Probability mass function of some variables
$f(\cdot)$	Probability density function
$P(\cdot)$	Cumulative distribution function
$\pi(\cdot)$	Path related probability
$\tilde{\pi}(\cdot)$	Path related probability on the M-CEG
$\hat{\pi}^{\Lambda_x}(\cdot) = \pi(\cdot\|\|\Lambda_x)$	The path probability of Λ_y with a manipulation on Λ_x
$\Gamma(\cdot)$	Gamma function
$B(\cdot)$	Beta function
$exp(\cdot)$	Exponential function
$Q(\cdot;\cdot)$	Log-likelihood score
$lpBF(\cdot,\cdot)$	Log-posterior Bayes factor
pa	Parents
$Dsup$	Direct superiors
ch	Children
nd	Non-descendants
$\mathcal{T}$	Tree
$V_{\mathcal{T}}$	Vertex set of the tree
v	Situations
$E_{\mathcal{T}}$	Edge set of the tree
$E(v)$	Set of edges emanating from the vertex v
$S_{\mathcal{T}}$	Set of situations

$\mathcal{F}$	Floret
$\mathcal{C}$	Chain Event Graph
λ	Path in $\mathcal{T}$ or $\mathcal{C}$
$\lambda(v, v')$	Paths traversing v and v'
$\mu(v, v')$	Subpaths rooted at v and terminating in v'
$\Lambda_{\mathcal{C}}$	Set of root-to-sink paths in $\mathcal{C}$
W	Set of positions
w	Positions
W_λ	Set of vertices traversed by λ
E_λ	Set of edges lying along λ
w_∞^f	Failure terminal node
w_∞^n	Working terminal node
$\Lambda(w)$	Set of paths passing through w
$\Lambda(e)$	Set of paths passing through e
$\boldsymbol{\theta}$	Primitive probabilities
x	D-events
$E(x)$	Set of edges labelled by x
$W(x)$	Set of receiving vertices of $E(x)$
$x(e)$	The d-event labelled on the edge e
$X(w)$	Set of d-events labelled on the edges pointing to w
Λ_x	Set of paths passing along the edges labelled by x
D	Datasets
R	The maintenance variable
A	The unseen maintenance variable
δ	The status indicator
I_e	The intervention indicator of the edge e
E^Δ	Set of edges labelled by root causes
$\boldsymbol{w}^*$	Intervened positions
H	Holding times
$\boldsymbol{\alpha}$	Hyperparemters of $\boldsymbol{\theta}$
$\boldsymbol{\omega}_d$	Sequence of words for document d
$\boldsymbol{u}_d$	Set of extracted core events for document d
$\pi_{\boldsymbol{u}_d}$	Set of partial orders of $\boldsymbol{u}_d$
$G^* = (V^*, E^*)$	Global net with vertex set V^* and edge set E^*
L	Core event variables
$I^\lambda(w)$	Incident variables
$Y^\lambda(w)$	Floret variables
$\mathcal{H}$	An assignment to floret variables and incident variables

Chapter 1

Introduction

In this chapter, we will give a literature review in reliability, Natural Language Processing (NLP) and causal Bayesian networks (BNs), which are the foundations of the established methodology and framework in this thesis. We will explain the motivations in applying the tree-based Chain Event Graphs (CEGs) to model how a system might fail and establishing a causal framework for studying system reliability within this chapter that will then be used in later chapters.

1.1 Reliability models in engineering

1.1.1 Nature of data

Operational risk in system reliability refers to faults, failures, control, or crashes [Fenton and Neil, 2018]. In industry, this forms an essential part of a company's risk management so that they can efficiently use machines and reduce costs.

The analyses developed in this thesis have been informed by defect data for supergrid transformers (SGTs). These are used at substations to change voltage for onward distribution [Nichols et al., 2017]. They are high volume transformers with multiple components including the tap changer, which can change turns operated on the coil, the bushing, which is a type of insulator, the winding temperature indicators, which indicate the winding core temperature, and the noise enclosure, which isolates sounds [Jeude et al., 2015]. The data was collected by a power supply company. Due to commercial sensitivity, we cannot disclose this dataset. We only use a subset of this dataset where there are no disclosure issues and explore the informative and useful features which are enlightening for designing the intervention regimes and building the framework for a causal analysis.

The data collected by engineers often has two types: ordinary data and

work order(WO)	start date	closed date	FP.Equip.No	person name	total cost	top-up units (kg/L)
00000001	2009-01-23	2009-02-20	000000000001	A.B.	5000	20
00000002	2009-02-15	2009-03-01	000000000002	C.D.	40	0
...	...	...	...	...	...	...

Table 1.1: Examples of ordinary data. Due to data confidentiality, the series numbers, names, dates and expenses are all artificial in this table.

subcomponent	symptom	cause	WO extended desc.	WO request desc.
conservator	drycol breather defect	missing or damaged component	FDCS: drycol replaced	REPLACE FAULTY DRYCOL BREATHER
conservator	breather defect	missing or damaged component	Transformer Abnormal reflash. FDCS : SGT 4A breather supply faulty.	Transformer Abnormal reflash. A.B. informs breather flt on SGT4A.
...	...	...	...	...

Table 1.2: Defect scripts. The first three columns are categorical data.

text data. Table 1.1 gives some examples of the ordinary data available in the database. We can find the category of the defect asset, the asset number, the supplier company, the service time, the site location, the field engineer who carried out the maintenance, the equipment fitted date and so on. There are 42 fields in the dataset for the SGTs which are in the form of the well-structured ordinary data. Some of these are numerical, such as the cost of maintenance, while the others are categorical, such as the component types. Such data can be useful in a regression analysis for predicting machine's reliability. On the other hand, text data is in the form of engineer's reports, also referred to as maintenance logs or defect scripts in this thesis. The text data usually provides richer information about the malfunction or failures and the maintenance which is not available from the ordinary data. The field engineers inspect malfunctioned assets and then give a description of the defects, failures or remedial work which has been planned or carried out at the time of inspection. These free texts are, therefore, valuable for analysing reasons or symptoms of different types of system failures. There are three fields in the dataset filled with free texts: "work order description", "work order extended description", "work order request description". Some of these entries maybe empty and some texts are just short phrases. There are spelling mistakes, grammatical mistakes, abbreviations, repetitions, or other errors in these texts. Therefore, it is necessary to clean and preprocess these texts.

Jeude et al. [2015] explored the ordinary data for SGTs. They designed algorithms to group the similar defects and match the defects to the assets. They also found that no correlation between the number of defects and the age of defects,

which was inconsistent with the reliability literature. They therefore suggested that a more detailed analysis of these defects was required. Nichols et al. [2017] carried out this analysis for the SGT data. They cleaned the text data using existing packages in R programme and devised a lookup table for mapping the defect script to a failure mode, for example the dielectric failure.

Both of these pieces of research were motivated by improving the efficiency of the maintenance policy through evaluating the risk of failure or defect for every asset. One popular risk-based approach in reliability engineering is Failure Mode and Effects Analysis (FMEA) [Bedford et al., 2001]. This programme explores the causes and effects of potential failure modes in a system. This coincides with the essence of this thesis – to exploit causal relationships between events that lead to a system failure and to infer the causal effects of different types of maintenance. Unlike FMEA and the analysis performed by Jeude et al. [2015] and Nichols et al. [2017], this thesis focuses on applying a probabilistic graphical model for causal analysis in this domain.

To fulfil this objective, we needed to extract the causally related events from the data. But not all the information in the dataset was relevant. So we first selected only the following 7 fields for constructing the causal framework: "Component", "Cause", "Symptom", "Subcomponent", "Work order description" (WOD), "Work order extended description" (WOED), "Work order requested extended description" (WRED). Note here that WOD, WOED, WRED are in the form of free texts, some of which are missing. Furthermore the content of these three fields for the same work order maybe repetitive. For example, there may exist repeated sentences. The data for the other four fields are ordinary data. However, some of the entries are missing. These 7 fields provide information about the root causes of the failure, the observed faults, the conditions of some components, and the maintenance. These events are crucial for designing the intervention regimes for a causal framework in reliability. Note that unlike local causes, which are usually symptoms, such as leak, the root causes are the initial contributing factor of the failure which may not be routinely recorded and tabulated within database. If the root causes are recorded, they are usually in the form of free texts. Any standard statistical approach depending only on the numerical data provided within these reports will necessarily miss this intrinsic direct information. In Chapter 2, we will give a thorough discussion about how to import this concept to the CEGs for customising the bespoke causal algebras.

Note that in reliability there are two main categories of maintenance commonly used in industry: **corrective maintenance** (CM), which is performed for an observed failure, and **preventive maintenance** (PM), which refers to the main-

tenance scheduled routinely for preventing system failure [*Types of maintenance: The 9 different strategies explained*, n.d.]. For the maintenance logs which record failures or the remedial work, for example, the free texts we have in the dataset, the maintenance is carried out after observing the failure. Therefore, the maintenance here is classified as the first type of maintenance. Notice that, instead of treatment, which stems from medical science and is widely used in causation literature, the terminology **remedial work** or **remedy** is used in reliability. It refers to a collection of acts designed to rectify the root cause of a failure incident and restore the system to full working order. Potential remedies are especially useful objects in this context because we find that they can often be extracted directly from the natural language texts provided by engineers. So this new concept plays a crucial role in the bespoke causal algebras designed for the associated intervention. In Section 2.3, we will differentiate this concept from treatment and demonstrate qualitatively and quantitatively for the intervention designed for this type of maintenance for remedial purpose. In Section 2.4, we design another type of intervention, called the routine intervention, for the routine maintenance for preventive purpose. Since the data for the preventive maintenance is not available to us, we formulate the underlying intervention regime based on the background knowledge in reliability theory. Some of the material for the remedial intervention and the routine intervention has been published in our papers [Yu et al., 2020] and [Yu and Smith, 2021*a*] respectively.

1.1.2 Graphical models in reliability literature

Graphical models provide an intuitive way to visualise and understand the process, so they are not uncommon in reliability. Here, we review some popular reliability graphical models.

In safety engineering and reliability engineering, domain experts often use a **fault tree**. This is a structured top-down logic diagram, to analyse how a system fails [Bedford et al., 2001; Lee et al., 1985]. This helps understand the failure processes and optimise the maintenance policy. The fault tree starts with a top event, which is the critical system event, for example, a type of system failure. This top event is then decomposed successively into intermediate events whose composition or intersection can cause the top event. These intermediate events could be failures or non-failures of subcomponents. Subcomponents are usually defined no finer than the smallest independently maintainable part of a system. In the final step the failure events are - when appropriate - then linked at the final level of exogenous driving events that might happen and so impact on the failure of these subcomponents. The events forming the final level that partition what might happen at this

most refined level, *i.e.* the lowest level of the identified causes, are called the basic events. The events are connected by logic gates which represent the logical relationships between these events, for example AND/OR. Therefore, throughout the recursive construction of the fault tree, the top event is re-expressed as an element of the sigma algebra generated - through the operations of union, intersection and complementation - by the basic events. These basic events form the atoms of the probability space. We can read from the structure of the fault tree which particular combination of failures or faults will cause the whole system to fail.

So basic events of the fault tree define the level of granularity within which the top event is explained and any concept of causation can be embedded. Note that this granularity is normally chosen to that it is sufficiently fine so that the functionality of the system after all intervention events of interest - such as restoring certain combinations of subcomponents to perfect working order - also lie in this sigma algebra. On the other hand for the sake of simplicity and transparency the system is described in no further detail than this.

The representation on the fault tree is simply logical. And unlike an event tree, on which the events are usually ordered chronologically, there is no specific or explicit partial order in the representation that describes any longitudinal information [Shafer, 1996]. Everything is represented cross-sectionally as an instantaneous picture. The description of a fault trees is often supplemented by further information, but this is not formally represented in the tree itself.

Furthermore, all events, including the basic events, within this standard depiction need to be binary. They either have a failed state or not. Clearly, this limits the representation of the failure process. For example, if we need to represent the extent or type of failure, then we have to do this in a contrived way by choosing some arbitrary binary decomposition using artificial constructions. The binary constraint has the technical advantage that Boolean algebra can immediately be used to represent the explanation. In particular the top event binary variable can be represented as a polynomial in the basic events. But this elegance is obtained at a cost. Recent reliability theory has tried to address this shortcoming using multi-state generalisations [Natvig, 1985; Lisnianski et al., 2017], but the Boolean framework behind these developments becomes much more cumbersome so that such methodologies are much less transparent.

The limitation of the flat representation of what happens in a system on the fault tree and the Boolean limitation can be overcome by the representation on the event tree. This temporally orders the events from root to leaf and can be elaborated for a statistical or probability analysis. The CEG - the main model discussed in this

thesis - is derived from the event tree, so inherits the advantages of the event tree representation that can embed a description of not only what happens but how things happen – an intrinsic part of any causal model. In Section 2.1, we will give a formal definition of the event trees and the CEGs.

Another traditional graphical representation in reliability theory is the Boolean Decision Diagram (BDD) [Bryant, 1995; Bedford et al., 2001]. This represents the Boolean function of the events, including the basic events and the intermediate events. The BDD is a rather different but equivalent graphical representation of the fault tree which exploits its Boolean structure more directly and compactly. The events are ordered in the same way as the event tree, where the basic events come first while the top events are the last components, but are not guided by the causal partial order like the CEGs [Cowell et al., 2014]. Given the order of the binary fault indicator variables, rules were proposed by Bryant [1995] to remove duplicate and redundant nodes to obtain canonical and compact form of the BDD so that the graph is simplified and can be asymmetric. The BDD can be generalised to the Multiple-Valued Decision Diagram (MDD) which models multi-states systems [Kostolny et al., 2014]. However, the MDDs are not populated with probabilities, and the order of variables is arbitrary and is not designed for a causal inference. By contrast, the richer semantics of the CEG are more flexible and transparent in representing the causally ordered events. In particular, each path on the CEG represent a causal story of a specific process, see Section 2.1 for details.

Neither a fault tree analysis nor a BDD systematically considers a statistical analysis. Torres-Toledano and Sucar [1998] also criticised the traditional reliability analysis that the deficiency of these models lies in the incapability of representing dependencies between failures and complex systems and suggested instead the use of a BN. The BN is a probabilistic graphical model whose topology is directed acyclic and can characterize and analyse uncertainty in an effective way. Classical fault trees can be transformed into BNs, so Fenton and Neil [2018] argued that the BN provided a more powerful graphical representation. Previous work [Torres-Toledano and Sucar, 1998; Cai et al., 2018] showed that BNs can well represent the system faults and failures and the effects of maintenance. Each node corresponds to a fault variable or a failure indicator, *etc.* And a conditional probability table needs to be specified for each node. Importantly, the advantages of applying BNs in reliability analysis lie in embedding probabilistic knowledge, managing probability propagation, inference and causal reasoning. The dependencies between failures or faults can be read from the structure of the underlying BN. The semantics of the BNs are supportive for causal inference and intervention reasoning [Ruiz-Tagle et al.,

2021]. Although BNs capture all these advantages, they lose things that might be captured by CEGs. We will discuss these weaknesses of BNs later.

In fact, the role of causality in system reliability cannot be ignored. Understanding the causal structure of a system can improve the prediction of reliability [Hund and Schroeder, 2020]. Further, causal reasoning through interventions in risk and reliability analysis can inform the policy makers or the engineers about the potential effects of new policies or actions so that the maintenance strategy can be optimised in an efficient and effective way. Interestingly, causal reasoning is popular in many domains, such as medical science [Mani and Cooper, 1999; Gillies, 2018] and genetics [Krieger and Davey Smith, 2016]. However, not many researchers have systematically studied formal causal analyses as they might be applied to reliability or risk analysis from a causal perspective [Hund and Schroeder, 2020]. Hund and Schroeder [2020] presented how to use the structural causal modeling (SCM) framework for reliability estimation in an engineering application given a set of data and assumptions. Nyberg [2013] described a systematic approach to construct causal BNs for safety analysis. Instead of converting it from a fault tree, the proposed method firstly elicited a failure propagation graph from the architecture of the system and requirements engineering [Hull et al., 2010] which was used to determine dependency between fault variables. Li and Shi [2007] established a method to learn causal networks from observational data with domain knowledge applied for manufacturing systems. However, no previous work has developed an application of Pearl's theory of causal reasoning in this domain. For example, the $do-$calculus [Pearl, 1995, 2009] and the back-door theorem for identifying causal effects [Pearl, 2009]. In the later section of this chapter, we will give a brief review of these theories on BNs and adapt them later in Chapter 2. In this thesis, we do not design intervention calculus on BNs for machines failure data. Instead, we apply CEGs.

Despite the popularity of the BNs framework for exploring causal relationships, the discrete joint probability distributions represented on the BNs are highly symmetric. It is non-trivial to use BNs to represent models with non-symmetric sample space structures [Thwaites, 2008; Freeman, 2010], for example, when the state spaces of some of the random variables are radically different given the values of some other random variables. So the semantics of BNs are not compelling for representing context-specific independence.

Moreover, maintenance is highly likely to be context-specific in reliability theory. For example, a manipulation may force the oil level to be normal but still retain an oil leak when the pipe seal is replaced and the other components are not maintained. Alternatively it may force the oil level to be normal with no oil

leak when the pipe seal, the oil indicator and the breather are all replaced. When modeling this intervention, the manipulation induced by it is likely to be asymmetric, which means it might not simply force a variable to take a specific value like the $do-$operator. Again such manipulations are not simply expressible by the semantics of BNs which are variables based.

Many researchers [Shafer, 1996; Spirtes et al., 2000; Riccomagno and Smith, 2005] have argued that event tree based inference provides an even more flexible and expressive graph than the BN from which to explore causal relationships. The unfolding of the asymmetric failure process can be easily represented by the event trees. The CEG groups the vertices and edges of the underlying event tree and has a simpler structure. The context-specific conditional probabilities can be well-defined on such tree graphs and the effects of asymmetric manipulations can be well-captured by the CEGs. Every BN can be transformed to an equivalent CEG which expresses the same conditional independence properties as the underlying BN [Barclay et al., 2013; Collazo et al., 2018]. So in this sense the discrete BN is a special case of a CEG. A CEG can provide a framework for directly expressing a probability model faithful to an engineer's hypotheses about what might have happened. It can be used to answer and evaluate dependency queries about any malfunction quantitatively. So CEGs are expressive tools for modeling system failures. However, no previous work has exploited the advantages of applying the CEG for reliability analysis. In this thesis, we will develop a causal framework with CEG semantics in this domain. So this is completely novel to either the engineers or the statisticians familiar with causality or CEGs. In particular, given the maintenance logs, our goal is to define a direct mapping from the collections of features found by natural language processors onto a probability model that faithfully represents their explanatory statements, *i.e.* a CEG. We next review the natural language processors which have been devised in literature.

1.2 Natural Language Processing

We have already mentioned when introducing information from a dataset for machines failure that the free texts are required to be cleaned and preprocessed and we aim to extract the causally related events from these natural language descriptions. The artificial intelligence-based computational techniques for processing and learning from texts are referred as **Natural Language Processing** (NLP). The formal definition of it is given below.

Definition 1.2.1 (NLP [Liddy, 2001])**.** *NLP is a theoretically motivated range of*

computational techniques for analysing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications.

NLP emerged in late 1940s as machine translation [Liddy, 2001]. It has gradually gained popularity in a variety of fields, including information retrieval [Nadkarni et al., 2011] and speech recognition [Collobert et al., 2011; Hirschberg and Manning, 2015]. Linguists, machine learners and statisticians have all made efforts to accelerate the development of this technology [Blei et al., 2003; Collobert et al., 2011; Young et al., 2018].

Some common NLP tasks include: tokenization, stemming, part-of-speech (POS) tagging, chunking, named entity recognition and so on [Liddy, 2001; Dudhabaware and Madankar, 2014; Collazo and P.G., 2017]. **Tokenization** breaks the texts into words, phrases or sentences which are defined as tokens [Gupta and Malhotra, 2015]. This is usually the first step for processing texts. For example, the text "The oil level was incorrect." can be tokenized into "The", "oil", "level", "was", "incorrect", ".", in which case the sentence is split into sequence of words and symbols. **Stemming** finds the base form or stem of each word in the texts [Zitouni et al., 2010; Dudhabaware and Madankar, 2014], which can reduce words difference. **POS tagging** technique [Mohamed et al., 2011] was designed to annotate each word in the document by its part of speech. The POS tag depends on the meaning of the word and the context it lies in. With POS tags, one can further perform **chunking** to the annotated texts, which labels segments of a sentence by syntactic constituents [He et al., 2009; Collobert et al., 2011; Dudhabaware and Madankar, 2014]. In addition, the texts can be labelled by the predefined categories such as "DATE", "PERSON", "LOCATION". This is called **named entity recognition** (NER). These techniques are useful in cleaning and preprocessing the free texts in reliability data.

Software has been well developed in various platforms to perform these tasks. The **tm** package [Feinerer et al., 2008] as an R programme provides tools for data cleaning, *e.g.* removing stopwords, and tagging tasks. In Python, a variety of libraries are available for NLP [Schmitt et al., 2019], such as **NLTK** [Bird, 2006] and **SpaCy** [Honnibal and Montani, 2017]. Manning et al. [2014] also designed a Java annotation pipeline framework called the **stanford CoreNLP**.

Some more complex text analyses have been established. For example, sentiment analysis aims to identify positive or negative orientation of texts [Dudhabaware and Madankar, 2014; Hirschberg and Manning, 2015]. Blei et al. [2003] established a three-level hierarchical Bayesian model called **Latent Dirichlet Alloca-**

tion (LDA). Each word in the document has a latent topic and the documents are random mixtures over latent topics. This model has been extended to a dynamic version that models the time evolution of topics [Blei and Lafferty, 2006; Perrone et al., 2017]. Further, Xun et al. [2017] and Blei and Lafferty [2007] considered correlations between topics.

A wide range of research [Mirza and Tonelli, 2016; Pustejovsky et al., 2005] has discovered the importance of the role of temporal information in natural language texts. So algorithms have been designed to automate the process of recognising and extracting temporal and event expressions. For example, Pustejovsky et al. [2005] devised a language called **TimeML** for capturing the temporal and event features. It tags the time expressions, temporal events, and the relationships between these tags. TimeML has been extensively applied and developed [Chambers et al., 2014; Mirza and Tonelli, 2016; Ning et al., 2019]. The temporal extraction is not a simple task because in different domains there may exist temporal expressions or events which are domain specific. For example, in medical science, to identify medical events, medical concepts need to be imported into the corpora, and the temporal expressions are the dates the patient accepts or experiences these events [Tang et al., 2013].

Curating causal relations from texts have drawn more attention recently in understanding the texts and improving predictive tasks. Many researchers [Sorgente et al., 2013; Dasgupta et al., 2018; Hendrickx et al., 2019] have designed architectures for automating the causal relationship extraction using linguistic rule based, supervised and unsupervised machine learning approaches. Causal events extraction can benefit from the temporal information extraction [Ning et al., 2019; Zhao et al., 2017] since **a cause happens before its effects**. However, causality detection is challenging, because some cause and effect events are marked with clear causal connectives, such as "so", while others are not; some documents explicitly state the cause and the effect while some do not [Dasgupta et al., 2018].

Maintenance logs provide crucial information about causal explanations of different failure events because these documents describe what engineers believe might explain malfunction they observe. Therefore, extracting and embedding these causal explanations is a critical aspect of this thesis. To extract the causal relationships of particular interest, in Chapter 3, we will propose a sequence of algorithms for extracting causally related events from maintenance logs. Our proposed method is based on two previous works.

The first is the CAscading EVent Ordering architecture (CAEVO) [Chambers et al., 2014]. This utilises rule-based and machine-learned classifiers to annotate

automatically the temporal relation between event-event pair, event-time expression pair, time-time pair, event-document record time pair, and time-document record time pair. For every pair, there are six possible relations: BEFORE, AFTER, INCLUDES, IS INCLUDED, SIMULTANEOUS, and VAGUE. Here the VAGUE relation means that for the specified pair of events, no clear temporal relation between them is identified given the corpora. We give an example below to show how CAEVO processes a document.

Example 1. *Suppose we have the following raw texts.*

"Oil leak - bleed valve to be replaced : bleed valve has been damaged in the past when removed by the use of wrench. Bleed valve to be replaced."

Inputting this document into CAEVO, the extracted events and temporal relations are shown in Table 1.3. "Bleed valve to be replaced" is repeated twice in the document, but CAEVO is unable to recognise them as the same event. This makes the event "replaced" appear twice in the sequence of ordered events. From the description in the document, we can deduce the real temporal order of these events. We highlight the events whose relations are incorrectly extracted in blue in the table.

event 1	**event 2**	**relation**
damaged	the past	IS INCLUDED
leak	replaced	BEFORE
removed	replaced	BEFORE
leak	the past	IS INCLUDED
removed	the past	BEFORE
leak	removed	AFTER
replaced	removed	AFTER
damaged	replaced	AFTER
removed	use	BEFORE
leak	damaged	AFTER
leak	use	BEFORE
replaced	damaged	AFTER

Table 1.3: Output from CAEVO.

Note that this programme treats verbs as events, so noun phrases cannot be extracted. In this case, when processing "event A causes event B", the causal connective "causes" is recognised as an event by CAEVO. Moreover, for domain-specific data, there may be many VAGUE relations asserted by this programme because reliability concepts were not considered when building this architecture. Therefore, we will not only just rely on the results given by CAEVO for a causal analysis. In Section 3.2, we will explain in detail how we have applied this method in this context and how the results of it are interpreted.

Other authors [Zhao et al., 2017] proposed a hierarchical framework for

causality embedding. Here, linguistic patterns for extracting causally related events were required to be pre-defined. These patterns consist of a set of rules for picking pairs of events whose order are determined by the causal connectives. A causal network can then be constructed using these extracted pairs of events whose nodes correspond to the extracted events. This is then generalised to an abstract causal network via replacing nouns and verbs by words with general meaning and picking frequent events and relations. The causal relations are embedded into a continuous vector space by using a dual cause-effect transition model.

It is important to note that this method was defined for the analysis of news, which has fewer grammar and spelling mistakes than our data. Furthermore we regularly find that our defect scripts do not have causal connectives. Therefore, the free texts we have need to be cleaned and curated before they are inputted into any existing programme. In addition, only some ideas of this model are adopted in our algorithms. For example, we are not interested in using their dual causal-effect transition model, see Section 3.2 for details.

1.3 Causal reasoning through interventions on BNs

In this chapter, we have reviewed the application of BNs in reliability theory and emphasised the role of causality. This section exploits causal effects of interventions with the semantics of BNs [Pearl, 1993, 1995, 2009]. The edges or dependence statements that can be read from the DAGs of BNs are not necessarily causal. However if the variables are ordered to respect their chronological and causal order, and the structure of the BN can be seen as describing the data generating process, then the DAG structure can be asserted as causal [Pearl, 2009]. When a causal BN is not intervened upon we say the model describes the **idle system**, *i.e.* the unmanipulated system.

The typical external intervention explicitly formulated on a causal DAG is the **atomic intervention**. This forces a variable X_i to attain a specific value x_i, denoted by $do(X_i = x_i')$. This is also called the do-operation. Let $pa(X_i)$ denote the parents of X_i in the causal diagram, and pa_i denote the values taken by $pa(X_i)$. Then under an atomic intervention, Pearl [2009] and others convincingly argue that the post-intervention probability satisfies $p(x_i|pa_i) = 1$ for $x_i = x_i'$ and $p(x_i|pa_i) = 0$ for $x_i \neq x_i'$. The joint probability function $p(x_1, \cdots, x_n)$ is now $p(x_1, \cdots, x_n|do(X_i = x_i'))$. Let $\hat{x}_i$ denote the value of variable X_i being forced to

attain a specific value x_i'. Then $p(x_1, \cdots, x_n|do(X_i = x_i')) = p(x_1, \cdots, x_n|\hat{x}_i')$:

$$p(x_1, ..., x_n|\hat{x}_i') = \begin{cases} \frac{p(x_1,...,x_n)}{p(x_i'|pa_i)} & \text{if } x_i = x_i', \\ 0 & \text{if } x_i \neq x_i'. \end{cases} \tag{1.3.1}$$

This intervention can be extended to force a subset X of variables to take on fix values x. The causal BN is then formalised by the following definition based on the formula of the atomic intervention.

Definition 1.3.1 (Causal Bayesian network [Pearl, 2009])**.** *Let $p(v)$ be a probability distribution on a set V of variables. If there is an intervention $do(X = x)$ that sets $X \subseteq V$ to constants x, then let $p_x(v)$ denote the distribution resulting from this intervention. Denote by $\boldsymbol{P}_*$ the set of all interventional distributions $p_x(v) = p(v|\hat{x}) = p(v|do(X = x))$, including $p(v)$, when $X = \emptyset$. A DAG G is said to be a causal Bayesian network compatible with $\boldsymbol{P}_*$ if and only if the following three conditions hold for every $p_x \in \boldsymbol{P}_*$:*

1. *the joint probability function can be decomposed as $p_x(x_1, \cdots, x_n) = \prod_{j\in\{1,\cdots,n\}} p_x(x_j|pa_j)$ given G;*
2. *$p_x(v_i) = 1$ for all $V_i \in X$ whenever v_i is consistent with $X = x$;*
3. *$p_x(v_i|pa_i) = p(v_i|pa_i)$ for all $V_i \notin X$ whenever $pa(v_i)$ is consistent with $X = x$.*

Suppose we are interested in the effect on a set of variables Y, then the causal effect of X on Y is represented by $p(y|\hat{x})$. Causal effects enable us to predict the influence of a hypothetical intervention from the passive observations summarised from $p(v)$ and the causal graph G. When some variables are unobserved, we need to show the identifiability of the causal effects, see definition below.

Definition 1.3.2 (Causal effect identifiability [Pearl, 2009])**.** *The causal effect of X on Y is identifiable from a graph G if the quantity $p(y|do(X = x))$ can be computed uniquely from any positive probability of the observed variables - that is, if $p_{M_1}(y|do(X = x)) = p_{M_2}(y|do(X = x))$ for every pair of models M_1 and M_2 with $p_{M_1}(v) = p_{M_2}(v) > 0$ and $G(M_1) = G(M_2) = G$.*

With partial observations, suppose the controlled variables X and the response variables Y are both well-defined on the causal graph G, Pearl [1993] designed the **back-door criterion** to examine if there is a set of variables $Z \subseteq V$ which is sufficient for identifying $p(y|do(X = x))$. The definition of the criterion is given below.

Definition 1.3.3 (Back-door criterion [Pearl, 1993, 2009]). *A set of variables Z satisfies the back-door criterion relative to an ordered pair of variables X, Y in a DAG G if:*

1. *no node in Z is a descendant of X; and*
2. *Z blocks every path between X and Y that contains an arrow into X.*

The first criterion requires that Z is a subset of non-descendant of X, denoted by $nd(X)$: $Z \subseteq nd(X)$, which implies that

$$Z \perp\!\!\!\perp X | pa(X). \tag{1.3.2}$$

Pearl [1993] augmented the DAG by adding an intervention indicator F_x taking values in $\{do(x), \text{idle}\}$ and linking it to the intervened variable by adding a directed edge from F_x to X. In the augmented graph, let p' denote the distribution over the variables. The marginal probability of every non-descendant of X in the augmented DAG remains the same as that in the original DAG. Since F_x is a parent of X, we have

$$p'(z|F_x) = p'(z) = p(z). \tag{1.3.3}$$

The second criterion ensures that all paths from F_x to Y traverse the children of X and are blocked when conditioning on X. This implies that

$$p'(y|x, z, F_x = do(x)) = p'(y|x, z, F_x = \text{idle}) = p(y|x, z) \tag{1.3.4}$$

and

$$Y \perp\!\!\!\perp pa(X) | (X, Z). \tag{1.3.5}$$

Theorem 1.3.4 (Back-door adjustment [Pearl, 2009]). *If a set of variables Z satisfies the back-door criterion relative to X, Y, then the causal effect of X on Y is identifiable and is given by the formula*

$$p(y|do(X = x)) = \sum_z p(y|x, z)p(z). \tag{1.3.6}$$

Proof.

$$p(y|do(X = x)) = \sum_z p(y|z, do(X = x))p(z|do(X = x)) \tag{1.3.7}$$

$$= \sum_z p(y|z, x, do(X = x))p(z|do(X = x)) \tag{1.3.8}$$

Applying equations 1.3.3 and 1.3.4, we therefore have the formula. □

Apart from the atomic intervention, more complex situations may occur when making policies. The stochastic intervention imposes a new conditional distribution $p^*(x|z)$ for the controlled variable X, which equivalently imposes a functional relationship $do(X = g(z))$. This can be treated as forcing $do(X = x)$ with probability $p^*(x|z)$. The back-door theorem can be extended to test the identifiability in this scenario.

Theorem 1.3.5 (Back-door adjustment for a stochastic intervention [Pearl, 2009])**.** *The effect on Y of a stochastic policy which imposes a new conditional distribution $p^*(x|z)$ is*

$$p(y|p^*(x|z)) = \sum_x \sum_z p(y|\hat{x}, z)p^*(x|z)p(z). \tag{1.3.9}$$

Pearl [2009] also assess whether a cause is genuine by the following definition.

Definition 1.3.6 (Genuine cause [Pearl, 2009])**.** *A variable X has a genuine causal influence on another variable Y if there exists a variable Z such that either:*

1. *X and Y are dependent in any context and there exists a context S satisfying*
   (a) *Z is a potential cause of X,*
   (b) *Z and Y are dependent given S, i.e. $Z \not\perp\!\!\!\perp Y|S$, and*
   (c) *Z and Y are independent given $S \cup X$, i.e. $Z \perp\!\!\!\perp Y|S \cup X$; or*
2. *X and Y are in the transitive closure of the relation defined in the above criterion.*

Note that a **confounding variable** or **confounder** is an unmeasured variable which may distort or mask the effects of the predictors on outcome variables [Pearl, 2009]. So far it has always been assumed that there are no unobserved confounders when studying CEGs, see *e.g.* Cowell et al. [2014].

We can explore causal hypotheses with the semantics of CEGs in a similar way with the semantics of the BNs [Cowell et al., 2014; Thwaites, 2008]. The atomic intervention on BNs reviewed above has been extended to the CEG as the singular intervention by Thwaites et al. [2010] and Thwaites [2013]. The identifiability of the causal effects on CEG for the singular intervention is analogous to the identifiability of the causal effects on the BN for the atomic intervention [Thwaites et al., 2010; Thwaites, 2013], which is an extension of the back-door theorem.

It also has been shown that context-specific manipulations, which may be asymmetric, can be well captured by the CEG semantics. This is another overwhelming advantage of the CEG to be applied in reliability engineering. The causal algebras for the domain-specific interventions in reliability is built upon the foundation of the singular intervention. In Chapter 2, we will demonstrate the formulae for the new intervention regimes.

In addition to the causal BNs, we next introduce the framework developed by Williamson and Gabbay [2005] and Casini et al. [2011] for **recursive Bayesian networks** (RBNs). The conditional independence assumptions for the hierarchical framework proposed in this thesis are made in light of this framework, details see Chapter 3. So here we briefly review the definition of the RBN and the essential terminologies within this model.

Definition 1.3.7 (The RBN [Williamson and Gabbay, 2005; Casini et al., 2011])**.** *The RBN is a special class of BN defined over N variables $V = \{V_1, ..., V_N\}$ where some variables can take BNs as values.*

Within a RBN [Casini et al., 2011; Williamson and Gabbay, 2005], a variable $V_i \in V$ is a **network variable** if its values index a set of BNs, denoted by $G(V_i)$. Otherwise, it is a **simple variable**. The variables V' corresponding to the vertices in $G(V_i)$ are the **direct inferiors** of the network variable V_i and V_i is called the **direct superior** of V'.

Note here that $G(V_i)$ and V are disjoint, where V is the set of variables lying at the deepest level of the RBN so that $V_j \in V$ is either a network variable who has direct inferiors or a simple variable. Let $\mathcal{N} = \{V_{j1}, \cdots, V_{jk}\} \subseteq V$ denote the set of network variables in the RBN. Let $Dsup(V')$ denote the direct superior of V', $Dinf(V_i)$ denote the direct inferiors of V_i, and $NID(V_i)$ denote the non-inferiors or descendents of V_i. Each edge is interpreted causally in the RBN.

Definition 1.3.8 (The flattening [Casini et al., 2011; Williamson and Gabbay, 2005])**.** *Given an assignment of values $n = \{v_{j1}, \cdots, v_{jk}\}$ to the network variables in the RBN, a non-recursive Bayes network called the flattening,denoted by $n^{\downarrow}$, can be constructed.*

The vertex set of the flattening consists of the vertices associated with simple variables and the assignment $v_{j1}, \cdots, v_{jk}$.

There is an edge e_{v_i,v_j} in the flattening if V_i is a direct superior or a parent of V_j in the correponding RBN.

The flattening is defined so that the conditional probabilities and the joint probability can be formally specified for the RBN.

(1) $p(M), p(S|M)$ (2) $p_{m_0}(D), p_{m_0}(R)$ (3) $p_{m_1}(D), p_{m_1}(R|D)$

Figure 1.1: An example of the RBN given by Casini et al. [2011]: (1) plots the upper level BN; (2) plots the lower level BN corresponding to $M = 0$; (3) plots the lower level BN corresponding to $M = 1$. The probability $p_{m_i}(\cdot) = p(\cdot|M = i)$ for $i \in \{0, 1\}$.

Williamson and Gabbay [2005] and Casini et al. [2011] made the following assumptions to better understand the nested causal structure in the RBN.

ASSUMPTION 1.3.9 (Causal Markov Condition (CMC) [Casini et al., 2011; Williamson and Gabbay, 2005])**.** *At each level of the RBN, each variable is independent of its non-descendants, conditional on its parents.*

ASSUMPTION 1.3.10 (Recursive Markov Condition (RMC) [Casini et al., 2011; Williamson and Gabbay, 2005])**.** *Each variable is probabilistically independent of those variables that are neither its inferiors nor peers, conditional on its direct superiors.*

ASSUMPTION 1.3.11 (Recursive Causal Markov Condition (RCMC) [Casini et al., 2011; Williamson and Gabbay, 2005])**.** *For an RBN, every variable $V_i \in V$ is independent of those variables that are neither its descendants nor its inferiors conditional on its parents and its direct superiors. So we have*

$$V_i \perp\!\!\!\perp NID(V_i)|pa(V_i) \cup Dsup(V_i).$$

Figure 1.1 shows a simple example of the RBN given by Casini et al. [2011]. This is a two-layer RBN. The upper level of this toy RBN has two variables $\{M, S\}$, see Figure 1.1(1), where M is a network variable and S is a simple variable. Assume M takes value 1 or 0. When $M = 0$, we have a net m_0 in Figure 1.1(2) corresponding to this value. When $M = 1$, the net m_1 is shown in Figure 1.1(3), where the variable R is dependent on D. Both D and R are direct inferiors of M, and M is the direct superior of D and R.

1.4 Thesis outline

So far, in this chapter, we have introduced the background knowledge in reliability, natural language processing, and statistical causal graphical models, and highlighted the advantages of using CEGs in reliability analysis.

The rest of the thesis is organised as follows. Chapter 2 will demonstrate how to customise a CEG for a system in reliability. This will be followed by designing domain-specific intervention regimes on CEGs for different types of maintenance. The back-door theorem by Pearl [2009] reviewed in the previous section will be adapted for identifying causal effects of these new interventions. Through our customised causal algebras we are then able to make predictive inferences about the effects of a variety of types of domain-specific interventions. Chapter 3 will propose a hierarchical framework for embedding the causal reasoning which are encoded within the maintenance logs. This consists of a sequence of text processing algorithms to extract causally related events, which are informed by the NLP literature we reviewed in Section 1.2, together with the formulation of the casual dependency within the hierarchical model. Specifically, the framework has a causal network called the Global Net (GN) at its surface level and a CEG at its deeper level. The maintenance logs may not provide complete information about a failure or deteriorating process and its maintenance procedures. In Chapter 4, we will extend the formulae for the domain-specific interventions and this hierarchical model to capture the types of missingness which can appear in the reliability data. Chapter 5 will evaluate the methodology by designing various comparative experiments. In the last chapter, we will summarise potential extensions of the contribution made in this thesis.

Chapter 2

CEGs and Causal Algebras

This chapter will demonstrate how to construct a domain-specific tree to model the deteriorating processes and the failure processes of a system and emphasise the advantages of using CEGs in reliability engineering. The semantics of CEGs are extremely expressive in representing the trajectory of events that lead to a failure and the causal relations between events. This chapter will also demonstrate how the CEG is able to accommodate the essential terminologies and concepts in reliability engineering and to provide an inferential framework for a causal analysis. More importantly, we will demonstrate how the novel intervention calculi can be customised for various domain-specific interventions and show the effects of these domain-specific interventions are identifiable on CEGs. By designing causal algebras for different interventions, we can make predictive inferences about the effects of various types of maintenance and so improve the prediction of system failures.

This chapter is organised as follows: Section 2.1 begins with introducing how to elicit a CEG from an event tree and build a domain-specific causal CEG for analysing system failure. From Section 2.2 to Section 2.4, we focus on causal algebras on CEGs. Specifically, Section 2.2 briefly reviews the analogous *do*-operation on CEGs developed by Thwaites et al. [2010] and Thwaites [2013]. A simple example is given to demonstrate how to apply this type of intervention for analysing system failures. In Section 2.3, we define a new type of intervention in light of the "remedial work". This is established by classifying different types of this intervention and devising the bespoke causal algebras. Section 2.4 discusses another new type of intervention designed for the preventive maintenance and demonstrates how to import the corresponding manipulations into the idle (unmanipulated) CEG. On the basis of the discussion in Section 2.3 and Section 2.4, Section 2.5 demonstrates how to integrate the innovative causal algebras we customised for the domain-specific in-

terventions to the learning algorithm. In Section 2.6, we give a concise discussion of modelling the lifetime of the system on the CEG and how the lifetime distributions are affected by the domain-specific interventions.

2.1 A CEG and its semantics

2.1.1 The construction of a CEG

A CEG is derived from an **event tree** $\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T})$ with vertex set $V_\mathcal{T}$ and edge set $E_\mathcal{T}$ [Smith and Anderson, 2008; Freeman, 2010; Collazo et al., 2018]. An event tree is a directed tree that provides an intuitive way for visualising the unfolding of a process over discrete event space. Here we only consider the tree to be finite. The **parents** of a vertex $v \in V_\mathcal{T}$ are the set of vertices in the tree whose emanating edges are received by v. We denote the set of parents as $pa(v) = \{v' \in V_\mathcal{T} : e_{v',v} \in E_\mathcal{T}\}$. The **children** of v are then defined as the set of vertices that receive the outcoming edges of v. We denote the set of children as $ch(v) = \{v' \in V_\mathcal{T} : e_{v,v'} \in E_\mathcal{T}\}$. Let $E(v) = \{e_{v,v'}\}_{v' \in ch(v)}$ denote the set of edges emanating from v. The vertex $v_0 \in V_\mathcal{T}$ with an empty parent set is called the **root** of the tree, while the vertices without children are called the **leaves** of the tree.

The path starting from the root vertex and ending in a leaf of the tree is composed of a sequence of edges in $E_\mathcal{T}$. We call such a path the **root-to-leaf path**. Let $\Lambda_\mathcal{T}$ denote the set of all root-to-leaf paths on the tree, and $E_\lambda \in E_\mathcal{T}$ denote the set of edges lying along the root-to-leaf path $\lambda \in \Lambda_\mathcal{T}$. Every root-to-leaf path depicts a sequence of events with respect to the temporal order of these events. We denote the set of root-to-leaf paths passing through vertices $v, v' \in V_\mathcal{T}$ by $\lambda(v, v') \in \Lambda_\mathcal{T}$, and the subpath starting from v and sinking in v' by $\mu(v, v')$.

Following Smith and Anderson [2008] we shall call the non-leaf nodes of the event tree **situations**, denoted by $S_\mathcal{T} \subset V_\mathcal{T}$. The set of leafs is $V_\mathcal{T} \setminus S_\mathcal{T}$. For every situation $v \in S_\mathcal{T}$, we can define a **floret**, denoted by $\mathcal{F}(v) = (V_{\mathcal{F}(v)}, E_{\mathcal{F}(v)})$. This is a subtree of the event tree $\mathcal{T}$ with vertex set consisting of v and its child vertices $V_{\mathcal{F}(v)} = \{v\} \cup ch(v)$ and edge set connecting v and its children $E_{\mathcal{F}(v)} = \{e_{v,v'} : v' \in ch(v), e_{v,v'} \in E_\mathcal{T}\}$. Let $\mathcal{F}(S_\mathcal{T})$ denote the set of florets that can be defined on the event tree.

Let $\pi(\cdot)$ denote the path related probability so that $\pi(\lambda)$ represents the probability of a unit passing along $\lambda \in \Lambda_\mathcal{T}$ and $\pi(v'|v)$ represents the probability of a unit arrives $v' \in V_\mathcal{T}$ given its current position at $v \in V_\mathcal{T}$. The **primitive probability** vector of $v \in S_\mathcal{T}$ is defined to be $\boldsymbol{\theta}_v = (\theta_{v,v'})_{v' \in ch(v)} = (\theta_e)_{e \in E(v)}$ where $\theta_{v,v'} = \pi(v'|v)$ can be thought of as the (conditional) transition probability along

edge $e_{v,v'}$. Every $\boldsymbol{\theta}_v$ satisfies $\sum_{v' \in ch(v)} \theta_{v,v'} = 1$ and $\theta_{v,v'} \in (0,1)$ for all $v \in S_{\mathcal{T}}$. Let $\boldsymbol{\theta}_{\mathcal{T}} = (\boldsymbol{\theta}_v)_{v \in S_{\mathcal{T}}}$. Note the tree and the set of primitive probabilities fully specify the probability model expressed by the tree. The pair $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is a **probability tree** [Cowell et al., 2014; Collazo et al., 2018] associated with $\mathcal{T}$. Such a tree can embody the asymmetric unfolding of the process by removing the edges whose associated events has probability zero. Note that the probability of a unit passing along $\lambda \in \Lambda_{\mathcal{T}}$ can be factorised as $\pi(\lambda) = \prod_{e \in E_\lambda} \theta_e$.

The probability tree can then be embellished into a **staged tree** [Smith and Anderson, 2008; Görgen and Smith, 2016]. The first step to construct a staged tree is partitioning the situation set $S_{\mathcal{T}}$. A **stage** is a set of situations so that two situations $v_i, v_j \in S_{\mathcal{T}}$ are in the same stage if and only if $\mathcal{F}(v_i)$ and $\mathcal{F}(v_j)$ have the same topology, *i.e.* there exists a one-to-one mapping φ_{ij} between $E_{\mathcal{F}(v_i)}$ and $E_{\mathcal{F}(v_j)}$ so that $\varphi_{ij}(e_{v_i,v_k}) = e_{v_j,v_l}$ when e_{v_i,v_k} and e_{v_j,v_l} represent the same event [Barclay et al., 2013], and $\boldsymbol{\theta}_{v_i} = \boldsymbol{\theta}_{v_j}$ up to a permutation [Collazo et al., 2018] so that when the emanating edge e_{v_i,v_k} represents the same event as e_{v_j,v_l}, $\theta_{v_i,v_k} = \theta_{v_j,v_l}$. Let $\mathbb{U}_{\mathcal{T}} = \{u_0, ..., u_{n_u}\}$ denote the set of stages of the situations on $\mathcal{T}$, where $n_u \in \mathbb{N}^+$. Having the stages, we can colour the tree accordingly. For situations belonging to the same stage, we assign the same unique colour. For edges $e_{v_i,v_k}, e_{v_j,v_l} \in E_{\mathcal{T}}$ and $v_i, v_j \in u_r \in \mathbb{U}_{\mathcal{T}}$, if they have the same label and $\theta_{v_i,v_k} = \theta_{v_j,v_l}$, then they have the same unique colour. The staged tree has the same set of vertices and edges as the event tree but are coloured according to $\mathbb{U}_{\mathcal{T}}$.

The CEG is derived from the staged tree by further partitioning the stage set and transforming the graph accordingly. For each situation $v \in S_{\mathcal{T}}$, let $\mathcal{T}(v)$ denote a subtree of $\mathcal{T}$ that roots at v and sinks in the leaves of $\mathcal{T}$, *i.e.* $\Lambda_{\mathcal{T}(v)} = \{\mu(v, v')\}_{v' \in V_{\mathcal{T}} \setminus S_{\mathcal{T}}}$. We say situations v_i, v_j are in the same **position** if the subtrees $\mathcal{T}(v_i)$ and $\mathcal{T}(v_j)$ are isomorphic, which means these two subtrees have the same structure, *i.e.* $V_{\mathcal{T}(v_i)} = V_{\mathcal{T}(v_j)}$ and $E_{\mathcal{T}(v_i)} = E_{\mathcal{T}(v_j)}$, and colouring. This gives a finer partition of situations than stages. If v_i, v_j are in the same position, then they are in the same stage. However, v_i, v_j being in the same stage cannot imply that they are in the same position. Let $\mathbb{W}_{\mathcal{T}} = \{w_0, ..., w_{n_w}\}$ denote the collection of positions. Then the definition of the CEG is formalised as following.

Definition 2.1.1. *A **Chain Event Graph** (CEG) $(\mathcal{C}, \boldsymbol{\theta}_{\mathcal{C}})$ is a probabilistic graphical model with topology $\mathcal{C} = (V_{\mathcal{C}}, E_{\mathcal{C}})$ and primitive probability vectors $\boldsymbol{\theta}_{\mathcal{C}}$.*

The sink node of the CEG, denoted by w_∞, is elicited from the underlying staged tree by merging all the leaves of it. The vertex set is given by the sink node and the set of positions $V_{\mathcal{C}} = W \bigcup w_\infty$. Every position inherits its colour from the staged tree.

For any two $w, w' \in V_{\mathcal{C}}$, creating an edge for every $v \in w$ and the child node $v' \in ch(v) \in V_{\mathcal{T}}$ which belongs to the position w', labelling it by the same value as $e_{v,v'} \in E_{\mathcal{T}}$ and inheriting the colour and the transition probability $\theta_{v,v'}$ [Barclay et al., 2015].

2.1.2 The causal CEGs for modelling and analysing system failure

In reliability engineering, the central task is to predict the probability of failure and trace the reason of the failure to prevent it reoccurring. Therefore we aim to depict the trajectories of equipment's service life by an event tree and derive a CEG from this event tree.

For a system reliability analysis, the event tree portrays the trajectory of the events that a system would have experienced before maintenance. So it models the failure processes and deteriorating processes of the system. The discrete process being modelled here is represented by a sequence of events which describe how the system gradually deteriorates and eventually fails or operates in a degraded condition. A default order of this process usually starts with a cause, followed by symptoms or faults and terminates with a failure or a worn-out status. The symptoms or faults can be split into the primary fault, the secondary fault [Bedford et al., 2001] and so on, depending on the system. The event tree for analysing system failures is constructed with respect to this order so that the edges emanating from the root node are usually labelled by root causes. In this way, the events are chronologically ordered on the tree.

In causal analysis, a cause is defined to happen before its effects. Then by asserting the order of events being causal in the idle system, we can further explore putative causal hypotheses on the CEG. We explain the role a CEG plays for a causal analysis later in Section 2.2.

Following this order, the last event modelled on the event tree is either a failure or an operational condition. In other words, the floret $\mathcal{F}(v)$ satisfying $ch(v) \subseteq V_{\mathcal{T}} \backslash S_{\mathcal{T}}$ corresponds to a failure indicator. Then the leaves of the event tree represent the status of the system just before maintenance. Under this setting, every root-to-leaf path of the tree either sinks in a vertex representing a failed status or sinks in a vertex representing an operational status. Accordingly, we can extend the structure of a CEG defined in Definition 2.1.1 to better fit a machine's failure data. Instead of having a single sink node w_∞, we replace it by two nodes: a **failure sink node** w_∞^f and a **working sink node** w_∞^n. All the leaves in the staged tree representing a failure status are then merged into w_∞^f, while the rest of the leaves are then merged into w_∞^n. The root-to-sink path now refers to the path that starts from the root node

$w_0 \in W$ and ends at w_∞^f or w_∞^n. Depending on whether a path $\lambda \in \Lambda_\mathcal{C}$ terminates at w_∞^f or w_∞^n, we accordingly classify the set of root-to-sink paths into two disjoint subsets. Let W_λ denote the set of nodes traversed by the root-to-sink path λ. If a path $\lambda \in \Lambda_\mathcal{C}$ satisfies $w_\infty^f \in W_\lambda$, *i.e.* w_∞^f is traversed by λ, then it is called a **failure path** and classified into $\lambda \in \Lambda_\mathcal{C}^f$. If a path $\lambda \in \Lambda_\mathcal{C}$ satisfies $w_\infty^n \in W_\lambda$, then it is called a **deteriorating path** and classified into $\lambda \in \Lambda_\mathcal{C}^n$. Then $\Lambda_\mathcal{C}^f \cup \Lambda_\mathcal{C}^n = \Lambda_\mathcal{C}$ and $\Lambda_\mathcal{C}^f \cap \Lambda_\mathcal{C}^n = \emptyset$. A failure path portrays a failure process of equipment while a deteriorating path portrays a deteriorating process of equipment.

We next introduce a new concept into the CEGs – the **d-events**. A d-event is the event that a particular unit in the population passes along a particular edge on the tree. For example, if a CEG is constructed for modelling different types of failures of a conservator, see Figure 2.3, then the population under study is a set of conservators. Let a d-event be oil leak and alarm due to varying temperature or breather defect. Every operational conservator may experience this d-event. For any operational conservator under study, if this d-event happens, then the path representing the deteriorating or failure process of the conservator passes along the edge e_{w_1,w_3} associated with this d-event. Let $\mathbb{X}_\mathcal{C}$ denote the set of unique d-events labelled on $E_\mathcal{C}$. For any $x \in \mathbb{X}_\mathcal{C}$, let $E(x) \subseteq E_\mathcal{C}$ denote the set of edges associated with x. This means once the d-event x is observed, the unit must traverse an edge $e \in E(x)$. As for which edge in $E(x)$ is traversed, it depends on the current position of this unit. We represent the set of receiving vertices of the edges in $E(x)$ by $W(x) \subseteq V_\mathcal{C}$. Given an edge $e \in E_\mathcal{C}$, there must be a d-event associated with it. Denote this by $x(e) \in \mathbb{X}_\mathcal{C}$. This d-event is unique for every edge. If w is the receiving node of e, then w is also associated with $x(e)$. However, in a CEG there could be more than one edge pointing to w, so there could be multiple d-events associated to w. Let $X(w)$ denote the set of d-events associated with w so that $X(w) = \{x(e_{w',w})\}_{w' \in pa(w)}$. The d-events can also be defined on the event tree in the same way. Let $\mathbb{X}_\mathcal{T}$ denote the set of d-events that can be discovered on $\mathcal{T}$. Then $\mathbb{X}_\mathcal{T} = \{x(e)\}_{e \in E_\mathcal{T}}$. The concept of d-event is essential for constructing the hierarchical causal model, we will explain this later in Chapter 3.

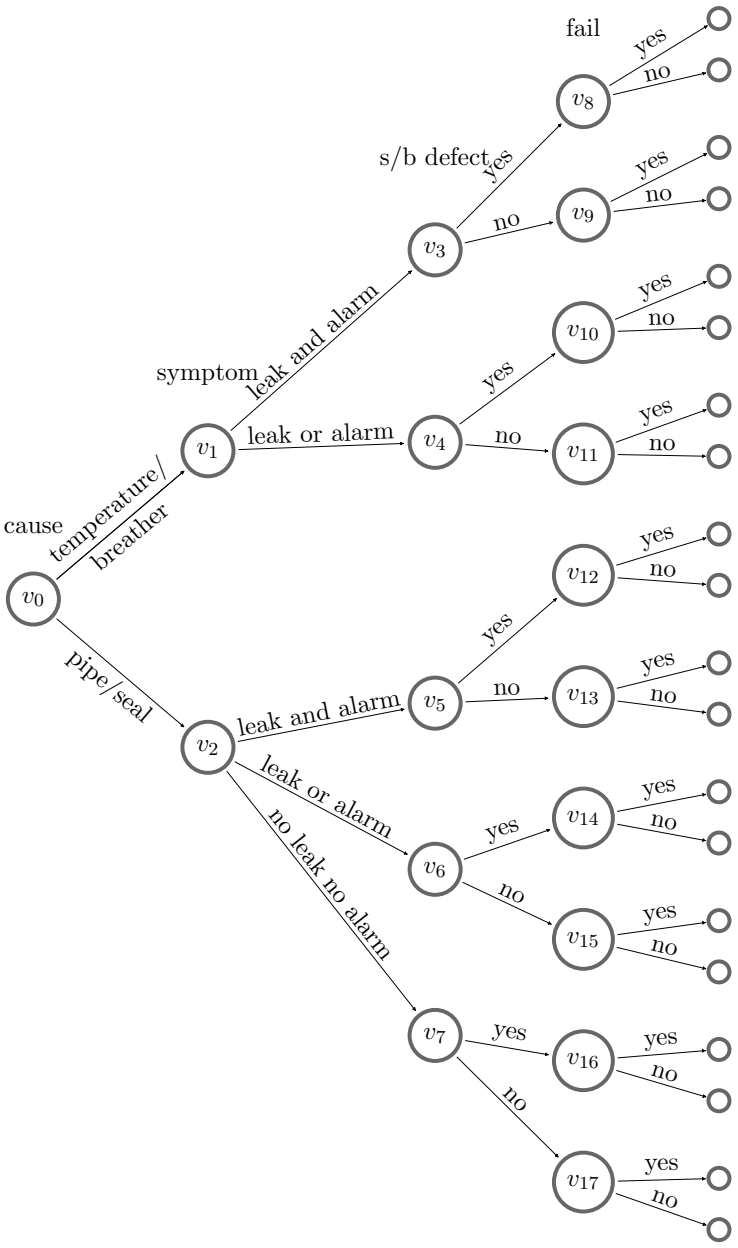


Figure 2.1: The event tree for the conservator system. This structure is elicited from engineers reports with assumptions.

Example 2. *Figure 2.1 is an example of the event tree constructed for a conservator system. This models the failure processes and the deteriorating processes that can happen within the conservator system. The events are chronologically ordered on the tree. In particular, on each root-to-leaf path, the initial event is the root cause, which is followed by the symptom and the defect, and the last event is either a failure or an operational condition. There are two root causes represented on*

the tree: {temperature/breather, pipe/seal}. So this event tree portrays failures or degradation either caused by the temperature change or breather defect, or caused by the faults in pipes or seals. The floret in Figure 2.1 associated with root causes is $\mathcal{F}(v_0)$. Three symptoms {oil leak and active alarm, oil leak or active alarm,no oil leak and no alarm} are modelled by the event tree. Florets $\mathcal{F}(v_1), \mathcal{F}(v_2)$ are associated with symptoms. Here, to give an example of asymmetric processes, we assume that a leakage or alarming is rendered to be unavoidable as long as temperature changes. Then $p(\text{"no oil leak and no alarm"}|\mu(v_0, v_1)) = 0$, i.e. $\mathcal{F}(v_1)$ only represents two categories: oil leak and active alarm, oil leak or active alarm. This context-specific conditional probability can be translated onto the event tree. Note however that to represent such information using BNs we need to first reconstruct the random variables, e.g. splitting the root cause variable with two levels {temperature/breather, pipe/seal} into an indicator for temperature/breather and an indicator for pipe/seal.

Whether there is a defect in the sight glass or the buchholz conditional on different causes and symptoms are represented by the florets $\mathcal{F}(v_3), ..., \mathcal{F}(v_7)$. The florets $\mathcal{F}(v_8), \cdots, \mathcal{F}(v_{17})$ are associated with failure indicators.

To elicit a staged tree, we make the following assumptions on the context-specific conditional independence. We assume:

1. *whether the equipment fails or not depends only on the root causes;*
2. *whether there is a sight glass or buchholz defect is independent of the root causes conditional on the symptoms.*

Under these two assumptions we are able to define the stages for the underlying event tree. Note that these assumptions are made to give a simple demonstration of the stages and positions. More realistic assumptions are required for modelling real data. The root node itself is a stage, denoted by $u_0 = \{v_0\}$. No hypotheses have been made about the relationship between symptoms and root causes. Then we have $u_1 = \{v_1\}$ and $u_2 = \{v_2\}$. Following the second assumption, v_3 and v_5 are in the same stage, while v_4 and v_6 are in the same stage. Let $u_3 = \{v_3, v_5\}$, $u_4 = \{v_4, v_6\}$ and $u_5 = \{v_7\}$. The first assumption implies that $u_6 = \{v_8, \cdots, v_{11}\}$ and $u_7 = \{v_{12}, \cdots, v_{17}\}$. Thus the set of stages on this tree is $\mathbb{U}_{\mathcal{T}} = \{u_0, u_1, \cdots, u_7\}$. Figure 2.2 depicts this hypothesised staged tree.

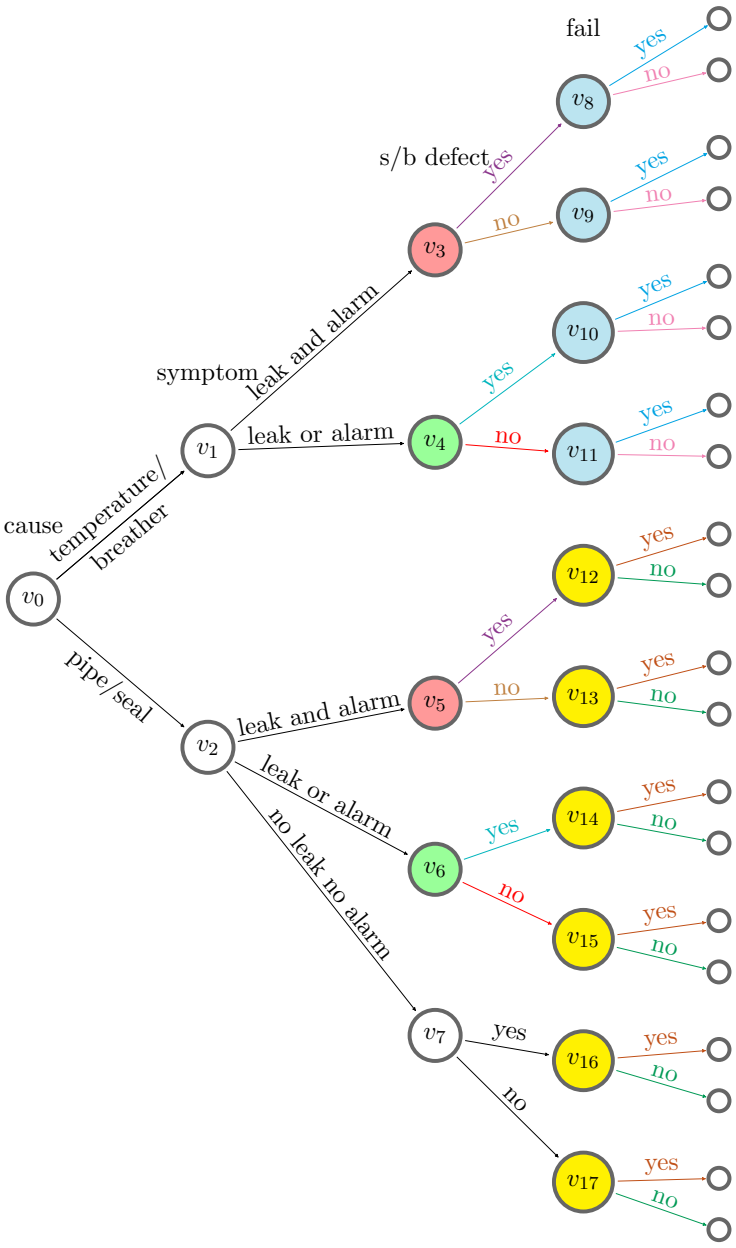


Figure 2.2: The staged tree elicited for the conservator system. Vertices with the same colour are in the same stage. Different stages are coloured differently. Each of the uncoloured vertex constitutes a single stage.

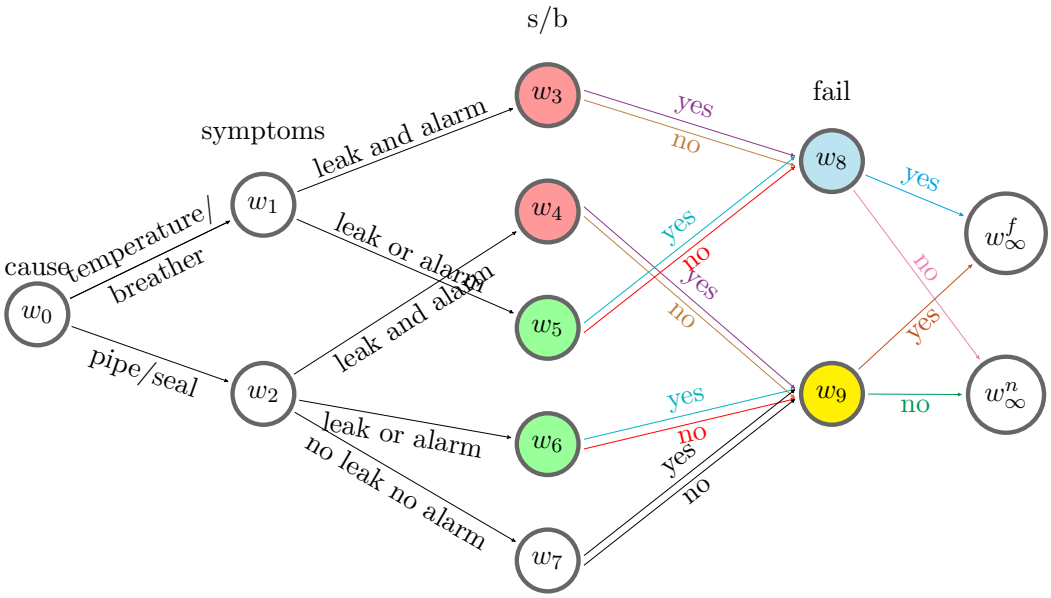


Figure 2.3: The CEG transformed from the staged tree in Figure 2.2

This staged tree can be transformed into a CEG by the method described above, see Figure 2.3. This gives 10 positions exclusive of the sink nodes. If this CEG is ordinal [Barclay et al., 2014], then $w_3, ..., w_7$ *are ranked from top to bottom so that* $p($*defect in sight glass or buchholz* $|\mu(w_0, w))$ *for* $w \in \{w_3, \cdots, w_7\}$ *is ranked in descending order. So the edge emanating from* w_3 *is predicted to have the highest conditional probability of observing a sight glass or buchholz defect.*

The set of d-events for the CEG in Figure 2.3 is $\mathbb{X}_{\mathcal{C}} = \{$*fail, not fail, temperature/breather, pipe/seal, oil leak and active alarm, oil leak or active alarm, no oil leak and no alarm, s/b defect, no s/b defect*$\}$*. For simplicity, we can annotate the d-event "fail" by* $x_{f,1}$*, "not fail" by* $x_{f,0}$*, "temperature change" by* $x_{c,1}$*, "pipe/seal" by* $x_{c,2}$*, "oil leak and active alarm" by* $x_{s,1}$*, "oil leak or active alarm" by* $x_{s,2}$*, "no oil leak and no alarm" by* $x_{s,3}$*, "s/b defect" by* $x_{d,1}$*, and "no s/b defect" by* $x_{d,0}$*. Then* $\mathbb{X}_{\mathcal{C}} = \{x_{f,1}, x_{f,0}, x_{c,1}, x_{c,2}, x_{s,1}, x_{s,2}, x_{s,3}, x_{d,1}, x_{d,0}\}$*. If we observe both oil leak and alarming, then we can find the associated edges* $E(x_{s,1}) = \{e_{w_1,w_3}, e_{w_2,w_4}\}$ *and the associated positions* $W(x_{s,1}) = \{w_3, w_4\}$*.*

Note that the CEG introduced here is acyclic. It represents the deteriorating processes and the failure processes which might happen before performing any maintenance. But the dynamic CEG (DCEG) established by Barclay et al. [2015] and the reduced CEG (RDCEG) proposed by Shenvi and Smith [2018] for modelling the dynamic process can be cyclic. We can model a dynamic process so that the deteriorating/failure process and the maintenance take place in turns with finite number of

times by introducing new nodes to the CEG [Freeman and Smith, 2011b], for example the seal failed for the fifth time. In this case, the CEG remains acyclic, although the topology of it becomes bigger. However, if like a time homogeneous Markov process an event keep happening indefinitely, then provided we have a Markov assumption we can depict this too with a cycle [Barclay et al., 2015; Collazo and P.G., 2017]. We will discuss this in the final chapter of this thesis.

2.1.3 A conjugate analysis on a CEG

In a Bayesian setting, consider to estimate for a random sample of identically distributed units – units within a given population are modelled on the same CEG. Conjugate inference is available for learning the CEG [Barclay et al., 2013; Collazo et al., 2018], where Dirichlet distribution is usually used as the prior distribution for each stage vector $\boldsymbol{\theta}_u = (\theta_{u1}, ..., \theta_{um_u})$ and $u \in \mathbb{U}_{\mathcal{T}}$. Let

$$\boldsymbol{\theta}_u \sim \text{Dirichlet}(\boldsymbol{\alpha}_u), \tag{2.1.1}$$

where the concentration parameters are $\boldsymbol{\alpha}_u = (\alpha_{u1}, \cdots, \alpha_{umu})$ and $\alpha_{uj} > 0$ for $j \in \{1, \cdots, mu\}$. The probability density function is:

$$f(\boldsymbol{\theta}_u) = \frac{\prod_{j\in\{1,\cdots,mu\}} \Gamma(\alpha_{uj})}{\Gamma(\sum_{j\in\{1,\cdots,mu\}} \alpha_{uj})} \prod_{j\in\{1,\cdots,mu\}} \theta_{uj}^{\alpha_{uj}}, \tag{2.1.2}$$

where $\Gamma(\cdot)$ is the Gamma function.

Then we write the prior over the staged tree as

$$f(\boldsymbol{\theta}|\mathcal{T}) = \prod_{u\in\mathbb{U}_{\mathcal{T}}} \frac{\Gamma(\sum_{j=1}^{m_u} \alpha_{uj})}{\prod_{j=1}^{m_u} \Gamma(\alpha_{uj})} \prod_{j=1}^{m_u} \theta_{uj}^{\alpha_{uj}}. \tag{2.1.3}$$

Let $\alpha_u = \sum_{j=1}^{m_u} \alpha_{uj}$ so that the equivalent sample size is $\alpha = \sum_{u\in\mathbb{U}_{\mathcal{T}}} \sum_{j=1}^{m_u} \alpha_{uj}$.

Given observations D,the posterior can be computed in a closed form due to Dirichlet-multinomial conjugacy.

$$f(\boldsymbol{\theta}|D, \mathcal{T}) = \prod_{u\in\mathbb{U}_{\mathcal{C}}} f(\boldsymbol{\theta}_u|D, \mathcal{T}) = \prod_{u\in\mathbb{U}_{\mathcal{T}}} \frac{\Gamma(\sum_{j=1}^{m_u} \alpha_{uj+})}{\prod_{j=1}^{m_u} \Gamma(\alpha_{uj+})} \prod_{j=1}^{m_u} \theta_{uj}^{\alpha_{uj+}}, \tag{2.1.4}$$

where $\boldsymbol{\alpha}_{u+} = \boldsymbol{\alpha}_u + \boldsymbol{n}_u$ is the updated parameter vector.

2.2 Singular intervention

Given a causal CEG, we can derive causal hypotheses from the structure and estimate causal effects under different hypothesised underlying causal mechanisms. The semantics of CEG represent causal hypotheses in an analogous fashion to the semantics of a causal BN [Thwaites et al., 2010; Collazo et al., 2018]. So we can use this graph to estimate the effects of different types of interventions by designing experiments and collecting the underlying data. Since events are causally ordered on the tree, we can explore the effect of an intervention on the events which lie downstream of the controlled events along the root-to-sink paths. To show this explicitly, we begin with reviewing an external intervention on the CEG first established by Thwaites et al. [2010] and Thwaites [2013]. This is analogous to Pearl's *do*-operator established for BNs [Pearl, 1993, 2009].

Since we defined the new concept "d-events", here we demonstrate this causal algebra in terms of d-events. Suppose there is a manipulation on a d-event $x \in \mathbb{X}_{\mathcal{C}}$ so that x is forced to be observed. Importing this external intervention into the CEG is equivalent to manipulating the associated edges $E(x) \subseteq E_{\mathcal{C}}$ so that $\theta_e = 1$ for $e \in E(x)$ and $\theta_{e'} = 0$ for $e' \in E(pa(W(x))) \setminus E(x)$. This is called the **singular manipulation** [Thwaites, 2013; Thwaites et al., 2010] on the CEG. In other words, the edges representing the d-event x are forced to be passed along for any unit arriving at $w \in pa(W(x))$ while the other edges emanating from the same vertex set as $E(x)$ are assigned probability 0. This is analogous to the atomic intervention $do(X = x)$ on the BN that forces a variable X to take value x.

Let Λ_x denote the set of root-to-sink paths passing along the edges representing x. We call this set of paths the **manipulated paths**. Then $\Lambda_x = \Lambda(E(x)) = \bigcup_{e \in E(x)} \Lambda(e)$, where $\Lambda(e)$ represents the collection of root-to-sink paths passing along $e \subseteq E_{\mathcal{C}}$. A manipulated CEG, denoted by $\hat{C}^{\Lambda_x}$, can be constructed by removing the root-to-sink paths which are not in Λ_x, *i.e.* $\Lambda_{\mathcal{C}} \setminus \Lambda_x$. Let $\hat{\pi}^{\Lambda_x}$ denote the post-intervention path related probability given the manipulation on Λ_x. Then for $\lambda \in \Lambda_x$,

$$\hat{\pi}^{\Lambda_x}(w_j|w_i) = \frac{\sum_{\lambda \in \Lambda_x} \pi(\lambda, \Lambda(e_{w_i,w_j}))}{\sum_{\lambda \in \Lambda_x} \pi(\lambda, \Lambda(w_i))}. \tag{2.2.1}$$

If we are interested in the effect of the singular intervention on the d-event y, then on the CEG the associated path set is $\Lambda_y = \Lambda(E(y)) = \bigcup_{e \in E(y)} \Lambda(e)$. Let $\pi(\Lambda_y||\Lambda_x)$ denote the probability of a unit traversing any path in Λ_y given a singular manipulation on Λ_x. Pearl [2009] designed a graphical test called the **back-door criterion** to check whether observing the variable Z is sufficient for identifying

the causal effects of $do(X = x)$ on $Y = y$. Thwaites et al. [2010] and Thwaites [2013] extended this theorem onto the CEG to find the **back-door partition Λ_z** which partitions $\Lambda_{\mathcal{C}}$ that is sufficient to identify the causal effects of the singular intervention $\pi(\Lambda_y||\Lambda_x)$, see below.

Definition 2.2.1. *The partition $\{\Lambda_z\}$ of $\Lambda_{\mathcal{C}}$ is a back-door partition if*

$$\hat{\pi}^{\Lambda_x}(\Lambda_y) = \pi(\Lambda_y||\Lambda_x) = \sum_z \pi(\Lambda_y|\Lambda_x, \Lambda_z)\pi(\Lambda_z), \tag{2.2.2}$$

where

$$\hat{\pi}^{\Lambda_x}(\Lambda_y|\Lambda_z) = \pi(\Lambda_y|\Lambda_x, \Lambda_z), \tag{2.2.3}$$

$$\hat{\pi}^{\Lambda_x}(\Lambda_z) = \pi(\Lambda_z). \tag{2.2.4}$$

Theorem 2.2.2. *For any $w \in pa(W(x))$, and $e_{w,w^*} \in E(x)$, if*

$$\pi(\Lambda_z|\Lambda(w)) = \pi(\Lambda_z|\Lambda(e_{w,w^*})) \tag{2.2.5}$$

$$\pi(\Lambda_y|\Lambda_x, \Lambda_z) = \pi(\Lambda_y|\Lambda(w), \Lambda_x, \Lambda_z) = \pi(\Lambda_y|\Lambda(e_{w,w^*}), \Lambda_z) \tag{2.2.6}$$

holds for every element of $\{\Lambda_z\}$, then $\{\Lambda_z\}$ is the back-door partition.

As demonstrated by Thwaites [2013], equation (2.2.5) is a direct result of the analogue of the conditional independence property in BNs: $Z \perp\!\!\!\perp X|pa(X)$. Equation (2.2.6) is a direct result of the analogue of $Y \perp\!\!\!\perp pa(X)|(X, Z)$.

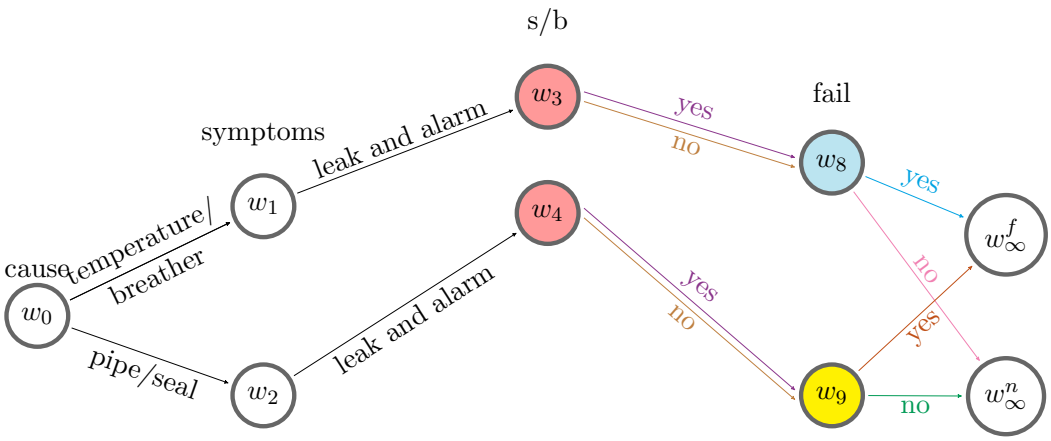


Figure 2.4: The manipulated CEG for Example 3.

Example 3. *We give an example of a singular manipulation on the CEG in Figure 2.3 for a conservator system. If we are only interested in the failure after seeing oil*

leak and alarming, and there is some way to intervene the conservator system for this purpose, then equivalently we force the d-event $x_{s,1}$ to happen. The associated edge which is forced to be traversed by the unit with probability 1 after this intervention is $E(x_{s,1}) = \{e_{w_1,w_3}, e_{w_2,w_4}\}$. For the purpose of this intervention alone, the probability of traversing either e_{w_1,w_5}, or e_{w_2,w_6}, or e_{w_2,w_7}, is assumed to be 0. The manipulated paths are $\Lambda_{x_{s,1}}$. By removing $\Lambda_{\mathcal{C}} \setminus \Lambda_{x_{s,1}}$ from the idle CEG, we construct the manipulated CEG $\hat{\mathcal{C}}^{\Lambda_{x_{s,1}}}$, see Figure 2.4.

If we are interested in how system failure would be affected by this intervention, then we need to calculate $\pi(\Lambda_{x_{f,1}}||\Lambda_{x_{s,1}})$, where the d-event $x_{f,1}$ indicates a failure. The choice of the back-door partition $\{\Lambda_z\}$ is flexible. In this example, we partition $\Lambda_{\mathcal{C}}$ by letting $\{\Lambda_{z_1}, \Lambda_{z_2}\}$, where $\Lambda_{z_1} = \Lambda(w_3) \cup \Lambda(w_5)$, $\Lambda_{z_2} = \Lambda(w_4) \cup \Lambda(w_6) \cup \Lambda(w_7)$. Note that $\Lambda(w_5), \Lambda(w_6), \Lambda(w_7)$ are not included in the manipulated CEG. Since we estimate the effects of this intervention from the observable data, which are not the intervened data, $\{z\}$ should partition the whole dataset and $\{\Lambda_z\}$ should partition $\Lambda_{\mathcal{C}}$. Therefore although $\Lambda(w_5), \Lambda(w_6), \Lambda(w_7)$ are not part of the manipulated CEG, these paths are still be included in $\{\Lambda_z\}$.

For Λ_{z_1}, we have

$$\pi(\Lambda_{z_1}|\Lambda(w_1)) = \pi(\Lambda(w_3), \Lambda(w_5)|\Lambda(w_1)) = 1, \tag{2.2.7}$$

and

$$\pi(\Lambda_{z_1}|\Lambda(e_{w_1,w_3})) = \pi(\Lambda(w_3), \Lambda(w_5)|\Lambda(e_{w_1,w_3})) = \pi(\Lambda(w_3)|\Lambda(e_{w_1,w_3})) = 1, \tag{2.2.8}$$

so

$$\pi(\Lambda_{z_1}|\Lambda(w_1) = \pi(\Lambda_{z_1}|\Lambda(e_{w_1,w_3})). \tag{2.2.9}$$

Similarly, we can check for Λ_{z_2}.

$$\pi(\Lambda_{z_2}|\Lambda(w_2)) = \pi(\Lambda(w_4), \Lambda(w_6), \Lambda(w_7)|\Lambda(w_2)) = 1, \tag{2.2.10}$$

$$\pi(\Lambda_{z_2}|\Lambda(e_{w_2,w_4})) = \pi(\Lambda(w_4), \Lambda(w_6), \Lambda(w_7)|\Lambda(e_{w_2,w_4})) = \pi(\Lambda(w_4)|\Lambda(e_{w_2,w_4})), \tag{2.2.11}$$

therefore,

$$\pi(\Lambda_{z_2}|\Lambda(w_2)) = \pi(\Lambda_{z_2}|\Lambda(e_{w_2,w_4})). \tag{2.2.12}$$

Thus, the criterion stated in equation (2.2.5) is satisfied for both Λ_{z_1} and Λ_{z_2}. From

the graph, we can deduce that

$$\begin{aligned}\pi(\Lambda_{x_{f,1}}|\Lambda_{x_{s,1}},\Lambda_{z_1}) &= \pi(\Lambda(w_\infty^f)|\Lambda(w_1),\Lambda_{x_{s,1}},\Lambda(w_3)) \\ &= \pi(\Lambda_{x_{f,1}}|\Lambda(e_{w_0,w_3}),\Lambda(w_3)),\end{aligned} \tag{2.2.13}$$

and

$$\begin{aligned}\pi(\Lambda_{x_{f,1}}|\Lambda_{x_{s,1}},\Lambda_{z_2}) &= \pi(\Lambda(w_\infty^f)|\Lambda(w_2),\Lambda_{x_{s,1}},\Lambda(w_4)) \\ &= \pi(\Lambda_{x_{f,1}}|\Lambda(e_{w_2,w_4}),\Lambda(w_3)).\end{aligned} \tag{2.2.14}$$

So the second criterion in equation (2.2.6) is also satisfied. Therefore we can evaluate $\pi(\Lambda_y||\Lambda_x)$ *using the partition we defined.*

2.3 Remedial intervention

Now we focus on formulating the domain-specific interventions on CEGs. The causal algebras designed for the new interventions encapsulate the domain-specific concepts in system engineering.

2.3.1 Remedy vs treatment

In a causal inferential framework, for either a graphical model or a functional causal model [Pearl, 2009; Dawid, 2000; Pearl, 1995], the causality is tied to an action for each unit in the population. This action is referred as **treatment** [Rubin, 2003]. This term stems from medical science and has been adopted in the literature of causality. The unit being treated can be a person, or other objects. In engineering reports, many use the term "**remedial work**" when recording the maintenance for some defects or failures. And "**remedy**" is a more familiar terminology in reliability engineering than "treatment". Similarly to a treatment, a remedy is treated as an intervention here and we are interested in evaluating its effects.

The unit upon which the remedy operates is the machine or the subsystem of the machine. There is no heterogeneity in the units themselves. So here we assume no covariate is added to the model which determines whether or not a remedial act takes place. The engineer decides a remedial act when he has observed the failure. So the unit has been failed before the remedy takes place. However, this is not the case for a treatment. For example, the medicine is given to a patient when some symptoms has appeared. This is to alleviate the symptoms and avoid death. The treatment is not given to a dead patient who has been censored in the study.

The remedy depends on the observed failure and it strives to find and fix

the root cause of the observed failure in a process driven by a mechanism which is well understood [Bedford et al., 2001]. Unlike with the human body, where we are unsure *e.g.* how a cancer comes about, here with the constructed system we understand well how and where each failure might happen. The remedy could be a single act or a sequence of actions. It normally forms a part of the data we consider as a record of what has been done, which is usually in the form of defect reports or failure reports. But it can also be missing. Whatever the remedial acts are, they all intervene on the root cause of the failure. There is no mediating variable between any pair of remedial acts and there is no confounding variable. In terms of the effects of the remedy, the remedy perfectly corrects the underlying root cause, that has happened prior to the remedy, and restores the faulty system to the status that is the same as a completely new system. From the direct effect of the intervention we can trace and deduce the underlying root cause.

By contrast, sequential treatments are regarded as a sequence of decisions. Previous work augmented the DAG to an influence diagram by adding a regime indicator [Dawid, 2002; Didelez et al., 2012; Dawid and Didelez, 2005] to each intervened variable. Each regime indicator is a decision variable and it does not have marginal distribution. The exact regime imposed on the system maybe conditional on some covariates or mediating variables. The effect of the treatment conditional on these covariates may vary.

In light of the concept of the remedy, we define a novel intervention customised for reliability engineering, called the **remedial intervention**. This type of external intervention is inspired by the concept of remedies and aims to prevent the same defect or failure reoccurring by exploring the root cause of a fault that has occurred and correcting it [Yu et al., 2020]. Importing the remedial intervention into the idle CEG and analysing its effects from the CEG semantics provide an inferential framework to improve the prediction of future failures. We devise novel and bespoke causal algebras through the semantics of CEGs which capture the features of different types of remedies.

2.3.2 Three types of remedial intervention

Next, we address how a system would response to an intervention as suggested by Pearl [1993]. However, this is different from the one considered by Pearl [1993] because of the particular demands of this setting. We explain this in detail below.

Consider a repairable system, if after the maintenance the status of the system is returned to the status the same as a new one and the system has the same failure rate as a new system, then the maintenance is **perfect** and the status after

maintenance is called **as good as new** (AGAN) [Iung et al., 2005; Borgia et al., 2009; Bedford et al., 2001]. If the status of the system returns to an operational condition just prior to failure, then the maintenance is **minimal** and the status after maintenance is called **as bad as old** (ABAO) [Iung et al., 2005; Borgia et al., 2009]. If the status of the system after maintenance is somewhere between ABAO and AGAN, then the maintenance is **imperfect** [Iung et al., 2005]. In light of these classifications and the remedial work recorded in the engineers reports, we carefully define three types of remedies for the remedial intervention.

Before formalising the concepts of the remedial intervention, we firstly make the following assumptions.

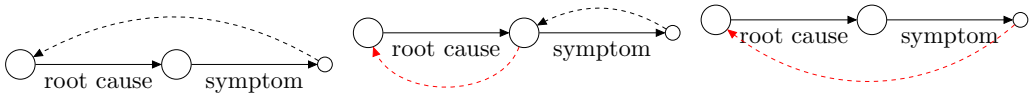
ASSUMPTION 2.3.1. *The underlying CEG or the event tree is faithfully constructed with respect to the domain knowledge of a particular system so that every failure process or deteriorating process that may happen in this system can be identified on the tree and every root cause and symptom are well-captured by the semantics of the tree.*

ASSUMPTION 2.3.2. *The system modelled by the CEG is repairable, and the AGAN status is attained when the root cause of the failure is completely fixed.*

For illustrative purpose, we augment the failure path to a new graphical framework which integrates the failure process with the process of maintenance and it is used to demonstrate the differences between various scenarios of the remedial intervention. We call such a graph the **status monitor**.

Definition 2.3.3. *The status monitor is a coloured cyclic tree-graph* $\Upsilon = (V_\Upsilon, E_\Upsilon)$.

- *The edge set* $E_\Upsilon = \{E_f, E_r\}$ *consists of the solid directed edges* E_f *and the dashed directed edges* E_r.
- *The directed path connected by the directed solid edges is the* ***failure path****.*
- *The directed path connected by the directed dashed edges is the* ***recovery path****; the edges associated with unobserved events are coloured in red.*
- *For a vertex, if the outgoing edge is* $e \in E_f$ *and the incoming edge is* $e \in E_r$*, then it is the* ***AGAN vertex*** v_0 *representing an AGAN status of the system; if the outgoing edge is* $e \in E_f$ *and the incoming edge is* $e \in E_r$*, then it is the* ***failed vertex*** v_f *representing a failed status of the system; the other vertices are the interior nodes of either the failure path or the recovery path. The status represented on the vertex degrades along the failure path.*



(a) The perfect remedy (b) The imperfect remedy (c) The uncertain remedy

Figure 2.5: The status monitors for the three types of remedies.

The failure path in the status monitor can be any failure path defined on the event tree or the CEG. The recovery path is absent in the CEG. Unlike the augmented DAG or the influence diagram [Pearl, 1993, 2009; Dawid, 2000] which adds a node corresponding to the intervention (regime) indicator to the graph representing the idle system, the status monitor is not designed to be integrated to the idle CEG and there is no conditional probability assigned to any edge or node of the status monitor. If we are interested in the effects of a remedy on the status of the failed system, we can use the recovery path as a reference. So the status monitor only provides an intuitive way for distinguishing different remedies.

We define a remedy to be **perfect** if the root cause of the failure is identified and successfully fixed by the observed maintenance so that the post-intervention status of the component is AGAN [Yu et al., 2020; Yu and Smith, 2021*c*] by Assumption 2.3.2. Figure 2.5a shows a status monitor under the perfect remedy. Here we simplify the failure path by using only two edges: the initial one represents a root cause, the edge following it represents a symptom. The recovery path is fully observed in this scenario. The observed maintenance returns a failed condition to full working order.

If the observed maintenance that aims to fix a root cause only returns the status of the failed system to somewhere between failure and AGAN, then we define it as the **imperfect remedy**. The observed maintenance only fixed secondary or intermediate faults [Bedford et al., 2001]. A graphical interpretation of the status change under this maintenance is shown in Figure 2.5b. The observed recovery path no longer terminates in the AGAN vertex but in the interior node of the failure path. Then to fully restore or remedy the system, in addition to the observed maintenance, more actions are required to be taken. The subsequent actions are unseen which therefore give rise to the uncertainty associated with the imperfect remedy [Yu and Smith, 2021*c*]. The purpose of implementing these subsequent actions is obvious: perfectly fixing the root cause of the observed failure. So there is a recovery path associated with these subsequent but unseen actions which rooting from the sink node of the observed recovery path and terminating in the root node of the failure path so that the system is AGAN after these actions.

If no maintenance is recorded in maintenance logs, then this leads to an **uncertain remedy**. We do not adopt the concept of the minimal maintenance because no maintenance is observed in this scenario. The whole maintenance process is unknown means there is no observed recovery path, see Figure 2.5c. However, for remedial purpose, to restore the failed equipment, some maintenance should be scheduled in the near future although this is not possible to retrieve from the maintenance logs provided to us. So we depict the red dashed path from the failed vertex to the AGAN vertex to represent the unobserved recovery process.

The events associated with maintenance can be extracted from engineers reports using the natural language processing algorithms proposed in the next chapter. Given these events, we can then deduce whether the root cause is fixed so that we can classify the type of the remedial intervention accordingly. Let R denote a variable with sample space $\mathbb{U}_R = \{r_0, r_1, \cdots, r_{n_R}\}$. Each state $r_i \in \mathbb{U}_R$ represents an observable maintenance event. Let r_0 represent the special event "no observation". We also define a variable A with state space $\mathbb{A} = \{a_1, \cdots, a_{n_A}\}$ to denote the unseen maintenance. Assume that $\mathbb{A}$ is a known and finite set. Each state $a_i \in \mathbb{A}$ can represent a single action or a sequence of actions. Both the observed maintenance events and the unobserved actions are remedial acts because they are performed to remedy the root causes.

Here we will assume that the root causes of a specific defect or failure may not be unique and could be multiple, but are well-defined. Note that a remedial intervention allows multiple root causes to be corrected simultaneously. Therefore in causal jargon we are considering here interventions that need not be singular [Thwaites, 2013].

To express the effects of different types of the remedial intervention, we next introduce some new definitions and notations [Yu and Smith, 2021*c*].

Definition 2.3.4. *A **status indicator** δ is a binary variable so that:*

$$\delta = \begin{cases} 1, & \textit{if the status is AGAN after maintenance,} \\ 0, & \textit{otherwise.} \end{cases} \tag{2.3.1}$$

For a CEG representing a repairable system, there are a set of edges labelled by possible root causes. Let $E^{\Delta} = \{e_{l_1}, \cdots, e_{l_n}\}$ denote such edges. Given a maintenance event, we are interested in which root causes are fixed by it and which edges represent these fixed root causes. So we give the following definition.

Definition 2.3.5. *For any edge representing a root cause $e_{l_i} \in E^{\Delta}$, the **intervention indicator** of it, denoted by $I_{e_{l_i}}$, is a binary variable indicating whether the*

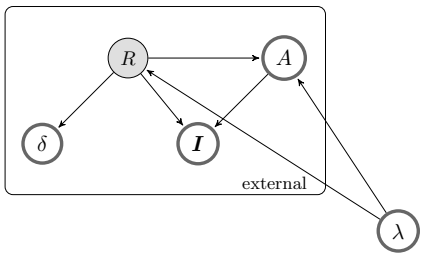


Figure 2.6: Demonstration of the external force of the remedial intervention. The variables lying in the black box are outside of the idle CEG.

labelled root cause is fixed by the observed maintenance events:

$$I_{e_{l_i}} = \begin{cases} 1, & \textit{if the root cause represented on } e_{l_i} \textit{ is fixed by the maintenance,} \\ 0, & \textit{otherwise.} \end{cases} \tag{2.3.2}$$

Then we can define a vector of intervention indicators over E^{Δ}: $\boldsymbol{I} = (I_{e_{l_1}}, \cdots, I_{e_{l_n}}) = \boldsymbol{I}_{E^{\Delta}}$. We can now express this vector in terms of positions:

$$\boldsymbol{I} = (\boldsymbol{I}_{w_{j_1}}, \cdots, \boldsymbol{I}_{w_{j_n}}), \tag{2.3.3}$$

where $\boldsymbol{I}_{w_{j_k}} = \boldsymbol{I}_{E(w_{j_k})} = (I_{e_{w_{j_k}1}}, \cdots, I_{e_{w_{j_k}m}})$. Each $\boldsymbol{I}_{w_{j_k}}$ is defined over the emanating edges of w_{j_k}, *i.e.* $E(w_{j_k}) = \{e_{w_{j_k}1}, \cdots, e_{w_{j_k}m}\}$ and $E(w_{j_k}) \subseteq E^{\Delta}$. Let $W^{\Delta} = \{w_{j_1}, \cdots, w_{j_n}\}$ denote the set of positions whose emanating edges are labelled by various root causes.

When there is no remedy, $\boldsymbol{I} = \boldsymbol{0}$. This refers to the observational or idle regime when there is no intervention. In this case, the status indicator δ is undefined.

When a remedy is performed, we have an intervened regime so that $\boldsymbol{I} \neq \boldsymbol{0}$. There is at least one edge $e \in E^{\Delta}$ so that $I_e = 1$ and there is at least one position $w \in W^{\Delta}$ so that $\boldsymbol{I}_w \neq \boldsymbol{0}$.

Unlike the regime indicator [Dawid, 2002] which is associated with a decision variable and has no marginal distribution, the intervention indicator and the status indicator have a well-defined conditional probability distribution. Given an observation $r \in \mathbb{U}_R$ and the failure process, we are interested in inferring the value of the intervention indicator vector from it. The failure process can be represented by a failure path $\lambda \in \Lambda_{\mathcal{C}}^{f}$ on the CEG. This means the intervention indicators can

be inferred from:

$$\begin{aligned} p(\boldsymbol{I}|r,\lambda) &= \sum_{\delta\in\{0,1\}} p(\boldsymbol{I},\delta|r,\lambda) \\ &= p(\boldsymbol{I}|\delta=1,\lambda,r)p(\delta=1|r,\lambda) + p(\boldsymbol{I}|\delta=0,\lambda,r)p(\delta=0|r,\lambda). \end{aligned} \tag{2.3.4}$$

ASSUMPTION 2.3.6. *The status indicator is independent of the failure process given the observed maintenance.*

Following this assumption, we can write

$$p(\delta=1|r,\lambda) = p(\delta=1|r) \tag{2.3.5}$$

and

$$p(\delta=0|r,\lambda) = p(\delta=0|r). \tag{2.3.6}$$

In a Bayesian setting, this is a posterior of δ given the observation r. To estimate the posterior, we can define a prior $p(\delta)$ for δ.

Now we go back to equation (2.3.4) with a focus on $p(\boldsymbol{I}|\delta,\lambda,r)$. We make the following assumptions for the sequential remedial acts (r,a) that constitute a remedy of an observed failure.

ASSUMPTION 2.3.7. *There are no mediating variables between any two remedial acts that are conducted for the same failure.*

ASSUMPTION 2.3.8. *Given a perfect remedy, the observed maintenance provides sufficient information about the root causes of the observed failure. Given an imperfect remedy or an uncertain remedy, the observed maintenance r and the unobserved maintenance a provide sufficient information about the root causes of the failure.*

Following this assumption, we have

$$\boldsymbol{I} \perp\!\!\!\perp \lambda|(R,A). \tag{2.3.7}$$

When $\delta=1$, the remedy is perfect. The observed maintenance r is perfect if and only if the root causes are correctly identified and fixed. Therefore, in this case, the value of $\boldsymbol{I}$ is known and we denote it by $\boldsymbol{I}(r)$. So the probability $p(\boldsymbol{I}|\delta=1,\lambda,r)$ is deterministic.

$$p(\boldsymbol{I}|\delta=1,\lambda,r) = p(\boldsymbol{I}|r,\delta=1) = \begin{cases} 1, & \boldsymbol{I}=\boldsymbol{I}(r), \\ 0, & \boldsymbol{I}\neq\boldsymbol{I}(r). \end{cases} \tag{2.3.8}$$

When $\delta = 0$, the remedy is imperfect or uncertain and there will be uncertainty associated with the unobserved additional remedial acts. We make one more assumption about the unobserved maintenance A.

ASSUMPTION 2.3.9. *The unseen additional maintenance depends only on the observed failure process and the observed maintenance.*

Thus,

$$A \perp\!\!\!\perp \delta | (\lambda, R). \tag{2.3.9}$$

Given assumptions 2.3.7-2.3.9, we can write

$$\begin{aligned} p(\boldsymbol{I}|\delta = 0, \lambda, r) &= \sum_{a \in \mathbb{A}} p(\boldsymbol{I}|r, a, \lambda, \delta = 0) p(a|r, \lambda, \delta = 0) \\ &= \sum_{a \in \mathbb{A}} p(\boldsymbol{I}|r, a) p(a|r, \lambda). \end{aligned} \tag{2.3.10}$$

Equation (2.3.4) can now be simplified as:

$$p(\boldsymbol{I}|r, \lambda) = p(\boldsymbol{I}|r) p(\delta = 1|r) + \sum_{a \in \mathbb{A}} p(\boldsymbol{I}|r, a) p(a|r, \lambda) p(\delta = 0|r). \tag{2.3.11}$$

Based on all the assumptions made in this section, we draw a DAG representing the relationships between the variables we mentioned above, see Figure 2.6. Note that λ is read from the CEG while the other variables lying in the plate are external to the CEG.

2.3.3 Manipulations on CEGs

With the essential domain-specific concepts, such as root causes and remedial work, and the definitions of different remedies, we next accommodate all this information into the causal analysis on CEGs and design the bespoke causal algebras for the remedial intervention corresponding to the remedies. In this section, we define a map transforming the idle system to the manipulated system given the intervention indicator vector $\boldsymbol{I}$.

The objective behind correcting a root cause is to prevent the fault or failure caused by it reoccurring. This implies that after a remedial intervention, the probability distribution over root causes needs to be transformed from what it was in the idle system. In particular, if the maintenance remedied a root cause, then it is less likely that the next defect is still caused by it. Therefore, when importing this idea into the idle CEG system, this equivalently forces the probability distributions over

some of the florets representing root causes to be reassigned. We call such manipulations the **stochastic manipulations** on the CEG. We give a formal definition of this type of manipulation below.

Note that not all florets representing root causes are affected if the asymmetric structure exists for florets representing root causes. Let $\boldsymbol{w}^*$ denote the set of vertices whose corresponding florets $\mathcal{F}(\boldsymbol{w}^*)$ are assigned new probability distributions under a given remedial intervention. The **intervened positions** $\boldsymbol{w}^*$ can be formally identified by the following rule. We have defined $W^{\Delta} = \{w_{k_1}, \cdots, w_{k_m}\}$ to denote the set of positions whose florets represent root causes. For $w \in W^{\Delta}$, if $\boldsymbol{I}_w = \{I_{e_{w,w'}}\}_{w' \in ch(w)}$ and there exists at least one edge emanating from w satisfying $I_{e_{w,w'}} = 1$, then $w \in \boldsymbol{w}^*$.

The set of root-to-sink paths on the CEG passing through any position in the intervened positions $w \in \boldsymbol{w}^*$ are the **manipulated paths**. We denote this set of paths $\Lambda(\boldsymbol{w}^*)$. The probability of traversing any of the manipulated paths $\lambda \in \Lambda(\boldsymbol{w}^*)$ is manipulated under the remedial intervention. We will specify the post-intervention path probabilities after giving the formal definition of the stochastic manipulation.

Definition 2.3.10. *A manipulation on a CEG C is **stochastic** if there exists a set of positions $\boldsymbol{w}^* \subseteq W$ such that,*

1. *for each $w \in \boldsymbol{w}^*$, there is a well-defined map $\digamma$ updating the primitive probabilities vector $\boldsymbol{\theta}_w = (\theta_{w,w'})_{w' \in ch(w)}$*

$$\digamma : (\boldsymbol{\theta}_w, \boldsymbol{I}_w) \mapsto \hat{\boldsymbol{\theta}}_w, \tag{2.3.12}$$

   where $\hat{\boldsymbol{\theta}}_w = (\hat{\theta}_{w,w'})_{w' \in ch(w)}$ denotes the post-intervention primitive probabilities vector,

2. *the new primitive probabilities vector $\hat{\boldsymbol{\theta}}_w$ satisfies $\hat{\boldsymbol{\theta}}_w \neq \boldsymbol{\theta}_w$, $\sum_{w' \in ch(w)} \hat{\theta}_{w,w'} = 1$ and $\hat{\theta}_{w,w'} \in (0,1)$ for $w' \in ch(w)$,*

3. *for position $w \in W_{\Lambda(\boldsymbol{w}^*)} \setminus \boldsymbol{w}^*$, i.e. the position that lies on any of the paths passing through $\boldsymbol{w}^*$ and is not an intervened position, the corresponding primitive probabilities vector remains the same as the pre-intervention: $\hat{\boldsymbol{\theta}}_w = \boldsymbol{\theta}_w$,*

4. *for position $w' \in ch(pa(\boldsymbol{w}^*)) \setminus \boldsymbol{w}^*$, i.e. the position which shares the same parents with $\boldsymbol{w}^*$ but is not an intervened position, $\hat{\theta}_{pa(w'),w'} = 0$.*

According to the second condition in the definition, we distinguish the stochastic manipulation from the singular manipulation since the former does not include

the special case that forces an edge to be passed with probability 1. This is analogous to the stochastic policy defined by Pearl [2009] on BNs but is more flexible. Wilkerson [2020] has defined a set of vertices W' to be a **fine cut** if and only if

$$\bigcup_{w \in W'} \Lambda(w) = \Lambda_C. \tag{2.3.13}$$

The set of targeted positions $\boldsymbol{w}^*$ defined above is not necessarily a fine cut because $\bigcup_{w \in \boldsymbol{w}^*} \Lambda(w)$ is not force to equal to the whole collection of the root-to-sink paths $\Lambda_{\mathcal{C}}$. The stages of the CEG may be changed by a stochastic manipulation. This is because for w, w' being in the same stage it is allowed that $\mathcal{F}(w)$ is stochastically manipulated while $\mathcal{F}(w')$ is not affected by the intervention.

If the manipulated probabilities $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$ are known, then the post-intervention path related probabilities can be revised accordingly. Let $\hat{\pi}(\cdot)$ denote the post-intervention path related probability. Then, for $\lambda \in \Lambda_{\mathcal{C}}$,

$$\hat{\pi}(\lambda) = \begin{cases} \frac{\prod_{e \in E_\lambda} \theta_e}{\prod_{e' \in E(\boldsymbol{w}^*) \cap E_\lambda} \theta_{e'}} \times \prod_{e' \in E(\boldsymbol{w}^*) \cap E_\lambda} \hat{\theta}_{e'} & \text{if } \lambda \in \Lambda(\boldsymbol{w}^*), \\ 0 & \text{otherwise,} \end{cases} \tag{2.3.14}$$

where $E(\boldsymbol{w}^*)$ denotes the set of emanating edges of all positions $w \in \boldsymbol{w}^*$, and E_λ denotes the set of edges traversed by λ.

Let $\hat{\mathcal{C}}^{\Lambda(\boldsymbol{w}^*)}$ denote the topology of the conditioned CEG (after intervention) [Thwaites, 2013] constructed with respect to $\Lambda(\boldsymbol{w}^*)$. We can evaluate the post-intervention probability of traversing a root-to-sink path conditional on $\Lambda(\boldsymbol{w}^*)$. Let $\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\cdot)$ denote the path related probability of $\hat{\mathcal{C}}^{\Lambda(\boldsymbol{w}^*)}$. Then, for position $w_i \in W_{\Lambda(\boldsymbol{w}^*)}$ and $w_j \in ch(w_i)$,

$$\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(w_j|w_i) = \frac{\sum_{\lambda \in \Lambda(\boldsymbol{w}^*)} \hat{\pi}(\lambda, \Lambda(e_{w_i,w_j}))}{\sum_{\lambda \in \Lambda(\boldsymbol{w}^*)} \hat{\pi}(\lambda, \Lambda(w_i))}. \tag{2.3.15}$$

Therefore the probability of a unit traversing a root-to-sink path on the conditioned CEG $\mathcal{C}^{\Lambda(\boldsymbol{w}^*)}$ is given by the factorisation of the revised conditional probabilities:

$$\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\lambda) = \prod_{w_i, w_j \in W_\lambda, w_i = pa(w_j)} \hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(w_j|w_i). \tag{2.3.16}$$

Definition 2.3.11. *The manipulated CEG of a remedial intervention with respect to $\Lambda(\boldsymbol{w}^*)$ has a topology $\hat{\mathcal{C}} = \hat{\mathcal{C}}^{\Lambda(\boldsymbol{w}^*)} = (\hat{V}, \hat{E})$ and transition probabilities $\hat{\boldsymbol{\theta}}$.*

- *The vertex set is* $\hat{V} = W_{\Lambda(\boldsymbol{w}^*)}$*;*
- *The edge set is* $\hat{E} = E_{\Lambda(\boldsymbol{w}^*)}$*;*
- *The primitive probabilities are evaluated via equation (2.3.14), equation (2.3.15) and the specification in Definition 2.3.10.*

We can formalise a stochastic manipulation on the CEG by specifying this by a map:

$$\zeta : (\mathcal{C}, \boldsymbol{\theta}) \mapsto (\hat{\mathcal{C}}, \hat{\boldsymbol{\theta}}). \tag{2.3.17}$$

The domain of this map is the topology and primitive probabilities of the CEG: $V \bigotimes E \bigotimes \Theta$. This map is well-defined if and only if the transformation of the primitive probabilities in equation (2.3.12) is well-defined for all $w \in \boldsymbol{w}^*$.

What we have discussed so far assumes $\boldsymbol{\theta}_w$ is known, so given $\digamma$ that we can define the components $\hat{\boldsymbol{\theta}}_w$. However, we can also consider an inferential setting, where these parameters need to be estimated from observations. Thus, within a Bayesian model, let $f(\boldsymbol{\theta}_w; \boldsymbol{\alpha}_w)$ denote the prior distribution of $\boldsymbol{\theta}_w$ with parameters vector $\boldsymbol{\alpha}_w$. Let $\hat{f}(\cdot)$ denote the post-intervention prior of $\boldsymbol{\theta}_w$ and $G(\cdot)$ denote the transformation that updates the distribution over the florets $\mathcal{F}(\boldsymbol{w}^*)$ under the remedial intervention so that

$$\hat{f}(\boldsymbol{\theta}_w) = G[f(\boldsymbol{\theta}_w)] \tag{2.3.18}$$

for $w \in \boldsymbol{w}^*$. There is no default way to define G for a remedial intervention. However, we propose an approach that transforms the distribution by updating the hyperparameters. Let $\hat{\boldsymbol{\alpha}}_w$ denote the post-intervention hyperparameters of $\hat{f}$. Then we define a generic function κ to transform $\boldsymbol{\alpha}_w$ for $w \in \boldsymbol{w}^*$ [Yu and Smith, 2021*c*]:

$$\kappa : (\boldsymbol{\alpha}_w, \boldsymbol{I}_w) \mapsto \hat{\boldsymbol{\alpha}}_w. \tag{2.3.19}$$

The map κ can take different forms, such as a linear transformation, and domain experts could play an important role here to decide to what extent the root causes are affected.

In Section 2.1.3, we give an example of Dirichlet priors for the primitive probabilities vectors. Thus we can update the concentration parameters $\boldsymbol{\alpha}_w$ via κ. For example,

$$\hat{\boldsymbol{\alpha}}_w = \kappa(\boldsymbol{\alpha}_w, \boldsymbol{I}_w) = \boldsymbol{\alpha}_w + \boldsymbol{\omega}_w(1 - \boldsymbol{I}_w), \tag{2.3.20}$$

where $\boldsymbol{\omega}_w = (\omega_{e_{w,w'}})_{w \in ch(w)}$, and $\omega_{e_{w,w'}} > 0$, is introduced to control the effect of the remedial intervention on the root causes. In particular, if $I_{e_{w,w'}} = 1$, then

the corresponding $\alpha_{e_{w,w'}}$ is unchanged. When $I_{e_{w,w'}} = 0$, the corresponding $\alpha_{e_{w,w'}}$ rises by $\omega_{e_{w,w'}}$ and so the average weight of the remedied root cause is reduced by this transformation. The value of $\boldsymbol{\omega}_w$ can be advised by the domain experts in order to better measure and adjust the possibility of root causes for prediction purpose. Note that such transformations have been proposed elsewhere, albeit in rather different contexts. For example, Eaton and Murphy [2007] introduced a similar idea of assuming a deterministic linear function to increase the likelihood of entering a state of a variable and called it a "soft" intervention on BNs.

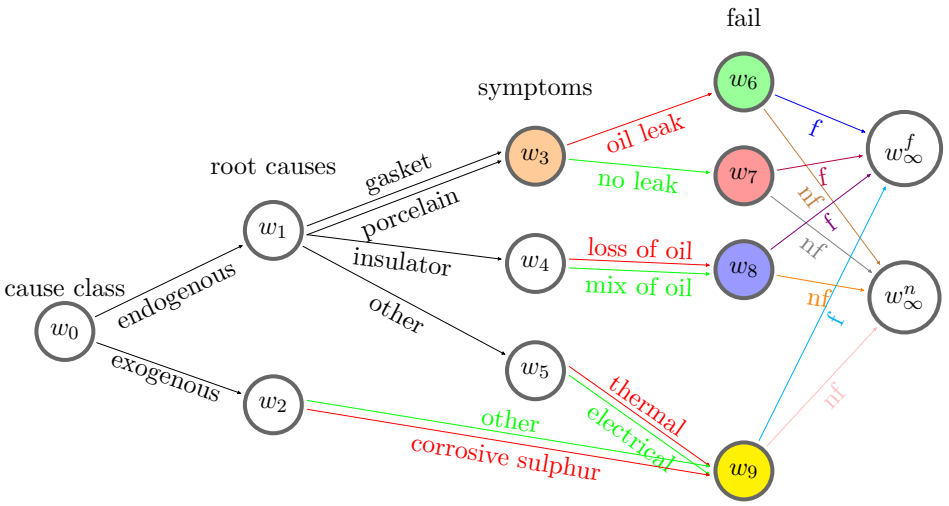


Figure 2.7: The causal CEG constructed for a bushing system. This structure is elicited from the description in [Al Abri et al., 2017] with appropriate conditional independence assumptions. Some of the labelled d-events are simplified to fit the figure.

Example 4. *Now we demonstrate the manipulation imported to the idle CEG in response to a remedial intervention using a example of a bushing*[1] *system. From the investigation report provided by Al Abri et al. [2017] which performed different diagnostic tests to check the root causes of a bushing's failure, we construct a simple but transparent event tree for this system and elicit a hypothesised idle CEG for causal analysis, see Figure 2.7.*

This first component modelled on this tree classifies the root causes into endogenous or exogenous causes. The d-events x_{en}, x_{exo} are associated with these two classes respectively. Following the root causes classifier, the root causes, symptoms and failure indicators are modelled. There are 6 root causes considered here, and we represent the associated d-events by $x_{c,1}, \cdots, x_{c,6}$. The d-events $x_{s,1}, \cdots, x_{s,6}$ are

[1]Bushing is an insulator in the transformer.

associated with the six types of symptoms.

d-event notation	exact d-event	associated edges
x_{en}	endogenous cause	e_{w_0,w_1}
x_{exo}	exogenous cause	e_{w_0,w_2}
$x_{c,1}$	failed or aging gasket	$e^1_{w_1,w_3}$
$x_{c,2}$	seal/axial movement of porcelain	$e^2_{w_1,w_3}$
$x_{c,3}$	cracked insulator	e_{w_1,w_4}
$x_{c,4}$	other endogenous reasons	e_{w_1,w_5}
$x_{c,5}$	corrosive sulphur	$e^2_{w_1,w_9}$
$x_{c,6}$	lightening and other exogenous reasons	$e^1_{w_1,w_9}$
$x_{s,1}$	oil leak	e_{w_3,w_6}
$x_{s,2}$	no oil leak and no other faults	e_{w_3,w_6}
$x_{s,3}$	loss of oil	$e^1_{w_4,w_8}$
$x_{s,4}$	mix of oil	$e^2_{w_4,w_8}$
$x_{s,5}$	thermal runaway	$e^1_{w_4,w_9}$
$x_{s,6}$	electrical discharge	$e^2_{w_4,w_9}$
$x_{f,1}$	fail	$e_{w_6,w^f_\infty}, e_{w_7,w^f_\infty}, e_{w_8,w^f_\infty}, e_{w_9,w^f_\infty}$
$x_{f,2}$	no fail	$e_{w_6,w^n_\infty}, e_{w_7,w^n_\infty}, e_{w_8,w^n_\infty}, e_{w_9,w^n_\infty}$

Table 2.1: The d-events for the CEG in Figure 2.7 for Example 4.

From Figure 2.7, we can see that root causes are represented on the set of edges $E^\Delta = \{e^1_{w_1,w_3}, e^2_{w_1,w_3}, e_{w_1,w_4}, e_{w_1,w_5}, e^1_{w_2,w_9}, e^2_{w_2,w_9}\}$. *There are two edges between* w_1 *and* w_3 *and two edges between* w_2 *and* w_9. *The superscripts index the order of the two edges appearing in the CEG from top to bottom. Here* $e^1_{w_1,w_3}$ *represents the d-event "aging gasket" and* $e^1_{w_1,w_3}$ *represents "seal/axial movement of porcelain".*

If the root cause $x_{c,1}$ *is perfectly remedied, for example when the gasket used to seal and prevent oil leak is replaced by a new one, then we can find the corresponding edge on the tree, which is* $e(x_{c,1}) = e^1_{w_1,w_3}$. *Under this intervention, the intervention indicator vector defined over* E^Δ *is* $\boldsymbol{I} = \{1,0,0,0,0,0\}$ *and* $\boldsymbol{w}^* = \{w_1\}$. *So only the floret* $\mathcal{F}(w_1)$ *is directly affected by the remedial intervention. The floret* $\mathcal{F}(w_2)$ *also represents root causes but stays unaffected. So we see here the semantics of CEG are expressive in representing the context-specific manipulations.*

The remedy fully restores the gasket to AGAN. But this does not mean that this gasket will not deteriorate and not cause a failure again. Instead what we mean specifically here is that this intervention could reduce the probability of observing failure or a defect caused by the gasket at the next maintenance. This is why we say the conditional probability mass functions defined over $\mathcal{F}(w_1)$ *are stochastically manipulated. This yields a manipulated CEG whose topology is plotted in Figure 2.8.*

The underlying staged tree of the CEG is depicted in Figure 2.9. The position w_9 *in the idle system contains a single stage* u_7 *that consists of situations* $\{v_7, v_8, v_{15}, v_{16}\}$. *However in the manipulated CEG, only* $\{v_{15}, v_{16}\}$ *are still in this*

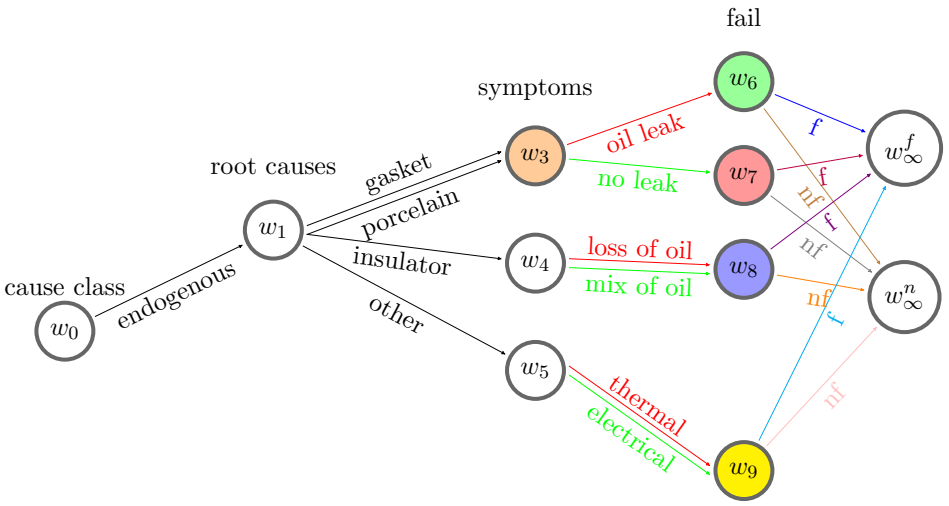


Figure 2.8: The manipulated CEG for Example 4.

position since v_7, v_8 lie downstream of the paths passing along the edge representing exogenous causes. Therefore, although the topology of the CEG conditional on $\Lambda(\boldsymbol{w}^*)$ retains the topology of the subtree consisting of $\Lambda(\boldsymbol{w}^*)$, the composition of the stages or positions may change.

In this example, the conditional probability $\hat{\pi}^{\Lambda(w_1)}(w_1|w_0) = 1$ in the manipulated CEG because in this case exogenous causes are forced to be ruled out.

Here, the root floret $\mathcal{F}(w_0)$ is associated with the root cause classifier, and the following florets $\mathcal{F}(w_1)$ and $\mathcal{F}(w_2)$ are associated with root causes. The positions w_1 and w_2 are not in the same stage. For system reliability, only the florets representing root causes are manipulated as a result of the remedial intervention. Note that the concept of a direct remedy is not usually appropriately applied to an exogenous root cause, such as lightening. So we still adopt the singular intervention established by Thwaites et al. [2010] to estimate the effects of the exogenous root cause with observational data from the partially observed system.

2.3.4 A back-door criterion

For a singular intervention with respect to Λ_x, Thwaites [2013] denotes its effect on the probability of observing event y by $\pi(\Lambda_y||\Lambda_x)$ on a CEG. Here we encounter a more complicated situation. Instead of starting with the manipulated paths Λ_x, we start with the observed maintenance (remedy).

When observing maintenance r, it is equivalently to impose externally a *do*-operator $do(R = r)$ onto the idle CEG. To examine the effects of r on y, we estimate $\pi(\Lambda_y|do(R = r))$.

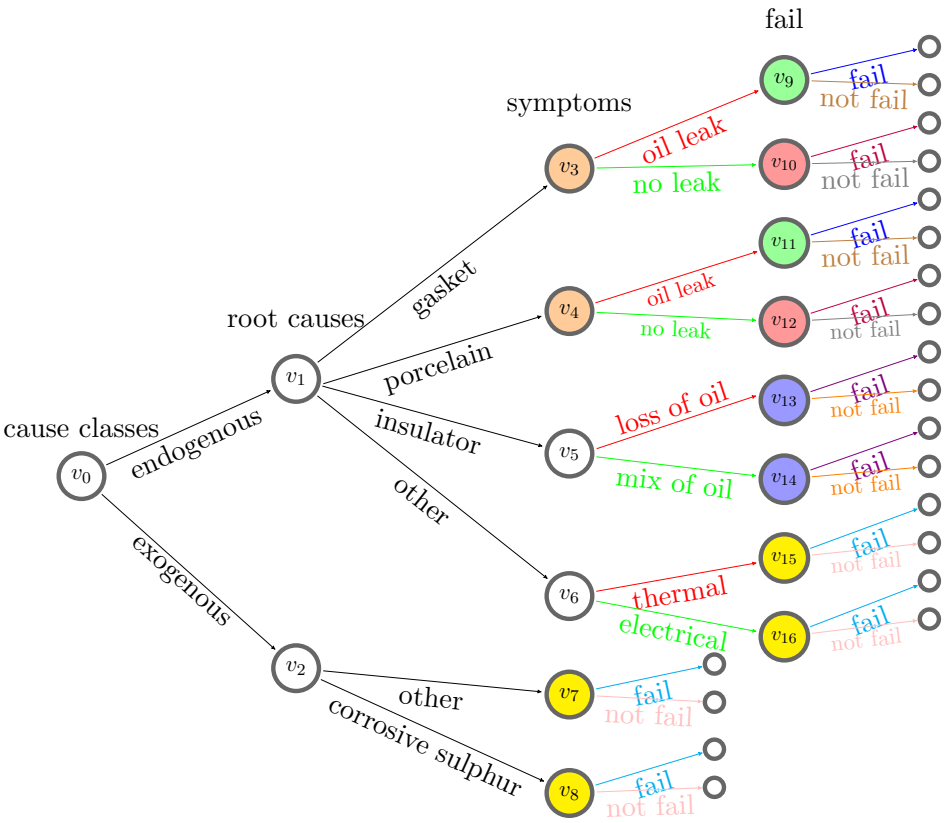


Figure 2.9: The staged tree for Example 4. Some of the labelled d-events are simplified to fit the figure.

Note that $\pi(\Lambda_y)$ is defined on the CEG while the operator $do(R = r)$ is external to the CEG. So we may decide to rewrite this expression by transforming the external control onto the CEG so that the controlled events are also represented on the CEG.

As discussed in the previous section, given the observed maintenance, we can infer the value of the intervention indicators, from which we can deduce the intervened positions and edges. When the remedial intervention is perfect, then $\boldsymbol{I}(r)$ is known and so $\boldsymbol{w}^*$ is known. Assume F is known. We can then express the causal query in terms of $\boldsymbol{I}_{\boldsymbol{w}^*}(r) = (I_e(r))_{e \in E(\boldsymbol{w}^*)}$, which is the intervention indicator vector defined over edges emanating from $\boldsymbol{w}^*$. Therefore, we can write the causal query as:

$$\pi(\Lambda_y | do(r), \delta = 1) = \pi(\Lambda_y || F(\boldsymbol{\theta}_{\boldsymbol{w}^*}, \boldsymbol{I}_{\boldsymbol{w}^*}(r))) = \pi(\Lambda_y || \hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}). \tag{2.3.21}$$

The right hand side expression is the probability of a unit traversing Λ_y when

intervening on $\boldsymbol{w}^*$ by forcing a change of probability distributions over $\mathcal{F}(\boldsymbol{w}^*)$ to $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$.

Otherwise, when $\delta = 0$, the intervention indicators vector $\boldsymbol{I}$ cannot be perfectly informed from the observed maintenance r. Following equation (2.3.4), we estimate the stochastic effects of r using the formulae as follows.

$$\begin{aligned}\pi(\Lambda_y|do(r),\delta=0) &= \sum_a \pi(\Lambda_y,a|do(r),\delta=0)\\ &= \sum_a \pi(\Lambda_y|do(r),a)p(a|do(r),\delta=0)\\ &= \sum_{\boldsymbol{I}}\sum_a \pi(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})p(\boldsymbol{I}|a,r)p(a|r,\delta=0)\end{aligned} \tag{2.3.22}$$

where the intervened positions $\boldsymbol{w}^*$ is determined by the value of $\boldsymbol{I}$.

Thus, we have an immediate proposition as follows.

PROPOSITION 2.3.12. *Given $\boldsymbol{w}^*$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$, a necessary and sufficient condition for the effects of the remedial intervention r to be identified is that $\pi(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})$ is identifiable on the causal CEG C.*

We have mentioned that the intervened positions $\boldsymbol{w}^*$ may not form a fine cut [Wilkerson, 2020], especially when the manipulations are asymmetric and the processes modelled on the idle CEG are asymmetric. For example, the intervention we specified in Example 4 only directly manipulated endogenous root causes, in which case $\Lambda(\boldsymbol{w}^*) = \Lambda(w_1)$ cannot partition $\Lambda_{\mathcal{C}}$. So $\boldsymbol{w}^*$ cannot "cut" the CEG fully. So here we construct a corresponding conditioned CEG from the idle CEG with respect to the intervened paths for identifying the underlying effects. This is similar to how we define the manipulated CEG but here we use the pre-intervention conditional probabilities since we aim to estimate the causal effects from the partially observed system.

Definition 2.3.13. *Let $\mathcal{C}^{\Lambda(\boldsymbol{w}^*)} = (V^*, E^*)$ denote the topology of the idle CEG conditional on $\Lambda(\boldsymbol{w}^*)$, then*

- *the vertex set is $V^* = W_{\Lambda(\boldsymbol{w}^*)}$;*
- *the edge set is $E^* = E_{\Lambda(\boldsymbol{w}^*)}$;*
- *the primitive probabilities are $\boldsymbol{\theta}^* = \{\boldsymbol{\theta}^*_w\}_{w\in W_{\Lambda(\boldsymbol{w}^*)}}$, where $\theta^*_{w,w'} = \pi^{\Lambda(\boldsymbol{w}^*)}(w_j|w_i)$ is evaluated as:*

$$\pi^{\Lambda(\boldsymbol{w}^*)}(w_j|w_i) = \frac{\sum_{\lambda\in\Lambda(\boldsymbol{w}^*)}\pi(\lambda,\Lambda(e_{w_i,w_j}))}{\sum_{\lambda\in\Lambda(\boldsymbol{w}^*)}\pi(\lambda,\Lambda(w_i))}. \tag{2.3.23}$$

For a remedial intervention, the stochastic manipulation $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$ directly affects the paths $\Lambda(\boldsymbol{w}^*)$ and the stochastically controlled d-events are $x(E(\boldsymbol{w}^*))$, which are the d-events labelled on the emanating edges of the intervened positions $E(\boldsymbol{w}^*)$.

Under a remedial intervention, the controlled event $x \in x(E(\boldsymbol{w}^*))$ is either a root cause fixed by the remedy or a root cause represented in the same floret as the fixed root cause. Then the probability of a unit passing along the paths Λ_x needs to be revised given a stochastic manipulation on $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$ since $\Lambda_x \subseteq \Lambda(\boldsymbol{w}^*)$. The post-intervention probability on the manipulated CEG is evaluated by equation (2.3.16) which combines the results of equation (2.3.14) and equation (2.3.15). Thus,

$$\pi(\Lambda_x||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}) = \hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\Lambda_x). \tag{2.3.24}$$

Enlightened by how Pearl [2009] proved the identifiability of the effects of a stochastic policy on BNs, here we adopt the similar idea to import the manipulated conditional probabilities $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$ into the idle system. Specifically, on the CEG, we force each event $x \in x(E(\boldsymbol{w}^*))$ to be controlled with probability $\pi(\Lambda_x||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})$. Unlike the stochastic policy defined by Pearl [2009], the imposed new probabilities are determined externally by domain experts to reflect impacts of maintenance on root causes and not dependent on other observations. Here we only discuss the case that the imposed new probabilities are known. When we are uncertain about how to update the probability distribution, we may use time series model or other approaches to predict the change of probabilities from historic data.

Since $x \in x(E(\boldsymbol{w}^*))$ and $\boldsymbol{w}^*$ may not form a fine cut as discussed earlier, $\sum_{x \in x(E(\boldsymbol{w}^*))} \pi(\Lambda_x)$ may not be equal to 1 unless conditional on $\Lambda(\boldsymbol{w}^*)$. Therefore, we rewrite the causal query as:

$$\begin{aligned} \pi(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}) &= \sum_{x \in x(E(\boldsymbol{w}^*))} \pi(\Lambda_y||\Lambda_x, \hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})\pi(\Lambda_x||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}) \\ &= \sum_{x \in x(E(\boldsymbol{w}^*))} \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x)\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\Lambda_x). \end{aligned} \tag{2.3.25}$$

The manipulated CEG is defined to be conditional on $\Lambda(\boldsymbol{w}^*)$ and the conditioned idle CEG is also defined with respect to $\Lambda(\boldsymbol{w}^*)$ in Definition 2.3.13. So estimation of this causal query is well supported by the semantics of these graphs.

Now we only need to show that $\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x)$ can be estimated. For this, we have the following proposition.

PROPOSITION 2.3.14. *For a remedial intervention that reshapes the probability distributions over florets $\mathcal{F}(\boldsymbol{w}^*)$, the effects of this intervention are identified*

if and only if $\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x)$ *can be estimated for every* $x \in x(E(\boldsymbol{w}^*))$ *given the observations and the CEG.*

Proof. From equation (2.3.25), it is straightforward to see that if $\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x)$ is identifiable for all $x \in x(E(\boldsymbol{w}^*))$, then $\pi(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})$ can be estimated, since $\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\Lambda_x)$ is determined externally. So it is a sufficient condition for the identifiability of $\pi(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})$.

If there exist a d-event $x \in x(E(\boldsymbol{w}^*))$ so that $\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x)$ cannot be estimated uniquely from the observable events, then the effects of the manipulation of $\hat{\theta}_{w,w(x)}$ are not identifiable. Since $e_{w,w(x)} \in E(\boldsymbol{w}^*)$ and $w \in \boldsymbol{w}^*$, so $\hat{\theta}_{w,w(x)} \in \hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$. This means the effects of $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$ are not fully identifiable so that $\pi(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})$ cannot be estimated. Therefore, the identifiability of $\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x)$ for every $x \in x(E(\boldsymbol{w}^*))$ is a necessary condition for the identifiability of $\pi(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})$.

□

The causal query $\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x)$ refers to identifying the effects of a singular manipulation on Λ_x within the conditioned CEG $\mathcal{C}^{\Lambda(\boldsymbol{w}^*)}$ whose topology is a subgraph of $\mathcal{C}$. This means we need to check whether the effects of this singular intervention are identifiable on the subtree constructed with respect to $\Lambda(\boldsymbol{w}^*)$.

Two criteria are specified for choosing the back-door partition [Thwaites, 2013], see equation (2.2.5) and equation (2.2.6). These criteria can be simply adapted for the conditioned CEG so that for all $x \in x(E(\boldsymbol{w^*}))$, the effects of the manipulation on Λ_x on Λ_y can be identified. The intervened edges $e_{w,w'} \in e(x)$ are the edges emanating from the intervened positions $w \in \boldsymbol{w}^*$. The intervened positions $\boldsymbol{w}^*$ form a fine cut of $\Lambda(\boldsymbol{w}^*)$. Their children $ch(\boldsymbol{w}^*)$ also form a fine cut of $\Lambda(\boldsymbol{w}^*)$. The following theorem is a restricted version of Theorem 2.2.2 on the conditioned idle CEG $\mathcal{C}^{\Lambda(\boldsymbol{w}^*)}$.

Theorem 2.3.15. *For any* $w \in \boldsymbol{w}^*$*, for all* $x \in x(E(\boldsymbol{w}^*))$ *and any* $e_{w,w'} \in e(x)$ *if*

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z|\Lambda(w)) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z|\Lambda(e_{w,w'})) \tag{2.3.26}$$

and

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda(w), \Lambda_x, \Lambda_z) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda(e_{w,w'}), \Lambda_z). \tag{2.3.27}$$

hold for every element of $\{\Lambda_z\}$*, then* $\{\Lambda_z\}$ *is the back-door partition for identifying the effects of a remedial intervention.*

When the root floret of the CEG represents a root cause variable, then the manipulated paths $\Lambda(\boldsymbol{w}^*) = \Lambda(w_0) = \Lambda_{\mathcal{C}}$ are the whole set of root-to-sink paths of

the idle CEG. This is a special case, where the back-door partition in this case only need to satisfy Theorem 2.2.2.

Finding a back-door partition satisfying Theorem 2.2.2 is sufficient for it being a back-door partition in our case. However, it is not a necessary condition for find a back-door partition for the conditioned CEG.

When a back-door partition can be identified, we can estimate $\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x)$ from the partially observed system. We can then decompose the expression in equation (2.3.25) as follows.

Theorem 2.3.16. *The effects of a stochastic manipulation are identifiable whenever a back-door partition $\{\Lambda_z\}$ can be found so that*

$$\pi(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}) = \sum_{x\in x(E(\boldsymbol{w}^*))} \sum_z \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda_x,\Lambda_z)\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z)\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\Lambda_x). \tag{2.3.28}$$

This holds when

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\hat{\Lambda}_x,\Lambda_z) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda_x,\Lambda_z), \tag{2.3.29}$$

here $\hat{\Lambda}_x$ denotes that the singular intervention acts on Λ_x only, not on Λ_z, and

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z||\Lambda_x) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z). \tag{2.3.30}$$

Proof. If we can find a partition $\{\Lambda_z\}$ of $\Lambda(\boldsymbol{w}^*)$ then the singular intervention causal query conditional on $\Lambda(\boldsymbol{w}^*)$ can be written as:

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x) = \sum_z \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\hat{\Lambda}_x,\Lambda_z)\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z||\Lambda_x). \tag{2.3.31}$$

If we estimate this intervened quantity from the partially observed system, then

$$\begin{aligned}
\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x) &= \sum_{\substack{w\in\boldsymbol{w}^* \\ e(w,w')\in e(x)}} \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda(w))\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda(w')) \\
&= \sum_{\substack{w\in\boldsymbol{w}^* \\ e(w,w')\in e(x)}} \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda(w))\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda(e_{w,w'})) \\
&= \sum_{\substack{w\in\boldsymbol{w}^* \\ e(w,w')\in e(x)}} \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda(w)) \sum_z \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda(e_{w,w'}),\Lambda_z)\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z|\Lambda(e_{w,w'}))
\end{aligned} \tag{2.3.32}$$

By the two criteria specified in Theorem 2.3.15 for the back-door partition $\{\Lambda_z\}$, we

can replace the last two terms in the last step by the following:

$$\begin{aligned}\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x) &= \sum_{\substack{w\in\boldsymbol{w}^* \\ e(w,w')\in e(x)}} \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda(w)) \sum_z \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda(w),\Lambda_x,\Lambda_z)\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z|\Lambda(w)) \\ &= \sum_z \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda_x,\Lambda_z)\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z).\end{aligned} \tag{2.3.33}$$

Comparing with equation (2.3.31), we therefore have the following two equivalent expressions.

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\hat{\Lambda}_x,\Lambda_z) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda_x,\Lambda_z), \tag{2.3.34}$$

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z||\Lambda_x) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z). \tag{2.3.35}$$

□

Note that though $\{\Lambda_z\}$ is only required to partition $\Lambda(\boldsymbol{w}^*)$, the partition must satisfy the criteria in Theorem 2.3.15 for all controlled events x. Similar to previous work [Thwaites, 2013], $\boldsymbol{z}$ can be chosen flexibly, for example a set of stages, positions or edges. Let $\Lambda_{\boldsymbol{z}} = \{\Lambda_{z_1},\cdots,\Lambda_{z_l}\}$. Each Λ_{z_i} is a collection of root-to-sink paths passing through the edges $E(z_i)$ or positions $W(z_i)$, where $W(z_i)$ are the receiving nodes of $E(z_i)$. Note that for different $w \in \boldsymbol{w}^*$, the emanating edges of w may represent different set of d-events.

Each controlled event x may correspond to more than one edges or just exactly one edge. Then to find $\{\Lambda_z\}$ for each of x, we can let each set $W(z_i)$ include at least one position traversed by the path $\lambda \in \Lambda(E(x))$ for every x. Otherwise, if we consider edges $E(z_i)$, then each set of edges $E(z_i)$ contains at least one edge traversed by the path $\lambda \in \Lambda(E(x))$ for every x. We give an example below.

Example 5. *Continue with the manipulation we discussed in Example 4 with the manipulated CEG in Figure 2.8. Suppose that the effect event is "fail". Then* $\Lambda_y = \Lambda(E(x_{f,1}))) = \Lambda(w_\infty^f)$ *is the set of failure paths sinking in* w_∞^f*. The stochastic manipulation is forced on* $\mathcal{F}(\boldsymbol{w}^*) = \mathcal{F}(w_1)$*. We have plotted the manipulated CEG in Figure 2.8, whose topology is constructed with respect to* $\Lambda(w_1)$*. The controlled events are* $x(E(w_1)) = \{x_{c,1}, x_{c,2}, x_{c,3}, x_{c,4}\} = \{$*failed or aging gasket, seal/axial movement of porcelain, cracked insulator, other endogenous reasons*$\}$*. The associated edges are* $e^1_{w_1,w_3}, e^2_{w_1,w_3}, e_{w_1,w_4}, e_{w_1,w_5}$*.*

We can create a partition $\{\Lambda_{z_1},\Lambda_{z_2}\}$ *so that* $E(z_1) = \{e_{w_3,w_6}, e^1_{w_4,w_8}, e^1_{w_5,w_9}\}$ *and* $E(z_2) = \{e_{w_3,w_7}, e^2_{w_4,w_8}, e^2_{w_5,w_9}\}$*. Here,* z_1 *refers to the d-events: oil leak, loss of*

oil, thermal runaway; z_2 refers the d-events: no oil leak and no other faults, mix of oil, electrical discharge. By this partition, when forcing a singular manipulation on $x_{c,1}$, the edges $e_{w_3,w_6} \in E(z_1)$ and $e_{w_3,w_7} \in E(z_2)$ lie along the paths $\Lambda_{x_{c,1}}$. These two edges are also passed along by the paths $\Lambda_{x_{c,2}}$. When forcing a singular manipulation on $x_{c,3}$, $e^1_{w_4,w_8} \in E(z_1)$ and $e^2_{w_4,w_8} \in E(z_2)$ lie along $\Lambda_{x_{c,3}}$. When forcing a singular manipulation on $x_{c,4}$, the edges $e^1_{w_5,w_9} \in E(z_1)$ and $e^2_{w_5,w_9} \in E(z_2)$ are traversed by the paths in $\Lambda_{x_{c,4}}$. Note that these events partition the paths $\Lambda(w_1)$ but do not partition all the root-to-sink paths in the idle CEG $\Lambda_{\mathcal{C}}$. This is because none of the edges $E(z_1), E(z_2)$ lies along any path in $\Lambda(w_2)$.

We can now check whether the criteria specified in Theorem 2.3.15 are satisfied. When the controlled event is $x_{c,1}$, for example,

$$\pi^{\Lambda(w_1)}(\Lambda(w^f_\infty)|\Lambda(e_{w_3,w_6}), \Lambda_{x_{c,1}}) = \pi^{\Lambda(w_1)}(\Lambda(w^f_\infty)|\Lambda(e_{w_3,w_6}), \Lambda(e^1_{w_1,w_3})) \tag{2.3.36}$$

is obviously true, so

$$\begin{aligned}
\pi^{\Lambda(w_1)}(\Lambda(w^f_\infty)|\Lambda(E(z_1)), \Lambda_{x_{c,1}}) &= \pi^{\Lambda(w_1)}(\Lambda(w^f_\infty)|\Lambda(E(z_1)) \cap \Lambda_{x_{c,1}}) \\
&= \pi^{\Lambda(w_1)}(\Lambda(w^f_\infty)|\Lambda(e_{w_3,w_6}) \cap \Lambda_{x_{c,1}}) \\
&= \pi^{\Lambda(w_1)}(\Lambda(w^f_\infty)|\Lambda(e_{w_3,w_6}), \Lambda(e^1_{w_1,w_3})) \\
&= \pi^{\Lambda(w_1)}(\Lambda(w^f_\infty)|\Lambda(E(z_1)), \Lambda(e^1_{w_1,w_3})).
\end{aligned} \tag{2.3.37}$$

Thus the first criterion in equation (2.3.26) is satisfied for $E(z_1)$. It is straightforward to check this condition for $E(z_2)$ and all the controlled events in the same way.

Now we check the second criterion. Since the edges $e^1_{w_4,w_8}, e^1_{w_5,w_9}$ are not traversed by any path in $\Lambda_{x_{c,1}}$,

$$\pi^{\Lambda(w_1)}(\Lambda(E(z_1))|\Lambda_{x_{c,1}}) = \pi^{\Lambda(w_1)}(\Lambda(e_{w_3,w_6})|\Lambda(e^1_{w_1,w_3})). \tag{2.3.38}$$

This conditional path probability can be easily evaluated from the conditional idle CEG:

$$\pi^{\Lambda(w_1)}(\Lambda(e_{w_3,w_6})|\Lambda(e^1_{w_1,w_3})) = \frac{\theta^*_{w_0,w_1}\theta^{1*}_{w_1,w_3}\theta^*_{w_3,w_6}}{\theta^*_{w_0,w_1}\theta^{1*}_{w_1,w_3}} = \theta^*_{w_3,w_6}. \tag{2.3.39}$$

We can also compute:

$$\pi^{\Lambda(w_1)}(\Lambda(e(z_1))|\Lambda(w_1)) = \frac{\theta^*_{w_0,w_1}(\theta^{1*}_{w_1,w_3} + \theta^{2*}_{w_1,w_3})\theta^*_{w_3,w_6} + \theta^*_{w_0,w_1}\theta^*_{w_1,w_4}\theta^{1*}_{w_4,w_8} + \theta^*_{w_0,w_1}\theta^*_{w_1,w_5}\theta^{1*}_{w_5,w_9}}{\theta^*_{w_0,w_1}}$$
$$= (\theta^{1*}_{w_1,w_3} + \theta^{2*}_{w_1,w_3})\theta^*_{w_3,w_6} + \theta^*_{w_1,w_4}\theta^{1*}_{w_4,w_8} + \theta^*_{w_1,w_5}\theta^{1*}_{w_5,w_9}. \tag{2.3.40}$$

In our example, $e_{w_3,w_6}, e^1_{w_4,w_8}, e^1_{w_5,w_9}$ are coloured the same, so they have the same transition probabilities. We also have $E(w_1) = \{e^1_{w_1,w_3}, e^2_{w_1,w_3}, e_{w_1,w_4}, e_{w_1,w_5}\}$ so $\theta^{1*}_{w_1,w_3} + \theta^{2*}_{w_1,w_3} + \theta^*_{w_1,w_4} + \theta^*_{w_1,w_5} = 1$. Thus, the above equation can be simplified to

$$\pi^{\Lambda(w_1)}(\Lambda(E(z_1))|\Lambda(w_1)) = \theta^*_{w_3,w_6}. \tag{2.3.41}$$

So,

$$\pi^{\Lambda(w_1)}(\Lambda(e_{w_3,w_6})|\Lambda(e^1_{w_1,w_3})) = \pi^{\Lambda(w_1)}(\Lambda(e_{w_3,w_6})|\Lambda(w_1)). \tag{2.3.42}$$

The second criterion is satisfied. Using the same method, it is easy to validate that the second criterion is satisfied for all $x \in x(E(w_1))$ and $\{\Lambda(E(z_1)), \Lambda(E(z_2))\}$.

Example 6. *The remedial intervention we described in Example 4 and Example 5 imposed a special case of stochastic manipulation where e_{w_0,w_1} is forced to be passed with probability 1. So there is a singular manipulation underlying this stochastic manipulation. The causal query $\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_x)$ in this example is $\pi^{\Lambda_{x_{en}}}(\Lambda_{x_{f,1}}||\Lambda_x)$ for every $x \in x(E(w_1))$. This expression can be rewritten as $\pi(\Lambda_{x_{f,1}}||\Lambda_{\boldsymbol{x}^*_r})$ where $\boldsymbol{x}^*_r = (x_{en}, x_r)$ for $x_r \in x(E(w_1))$. Then the stochastic manipulation can be treated as forcing a composite singular manipulations of x_{en} and x_r with probability $\pi(\Lambda_{x_r}||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})$.*

The manipulated CEG in this case is still the tree plotted in Figure 2.8. However, we do not estimate the causal effects from the CEG conditioned on $\Lambda(w_1)$. Instead, we estimate $\pi(\Lambda_{x_{f,1}}||\Lambda_{\boldsymbol{x}^*_r})$ for all $x_r \in x(E(w_1))$ from the non-intervened $\mathcal{C}$ depicted in Figure 2.7. Then the back-door partition $\{\Lambda_{z_1}, \Lambda_{z_1}\}$ must partition the whole set of the root-to-sink paths $\Lambda_{\mathcal{C}}$.

Now we redefine Λ_{z_1} and Λ_{z_2}. We add one more edge to the edge set associated with z_1 compared to Example 5 so that $E(z_1) = \{e_{w_3,w_6}, e^1_{w_4,w_8}, e^1_{w_5,w_9}, e^2_{w_2,w_9}\}$. We also add one more edge to $E(z_2)$ so that $E(z_2) = \{e_{w_3,w_7}, e^2_{w_4,w_8}, e^2_{w_5,w_9}, e^1_{w_2,w_9}\}$. Then it is easy to check that the criteria defined in Theorem 2.2.2 for the singular manipulation are all satisfied.

2.4 Routine intervention

In this section, we focus on another type of domain-specific intervention which is designed for the preventive maintenance (PM). The PM often refers to scheduled maintenance that identifies defects, replaces worn-out components, and aims to prevent fatal problems during the routine inspection before any failure happens [Bedford et al., 2001]. The main advantages of conducting such scheduled maintenance are reducing the chance of breakdown and extending the lifetime and reliability of a particular component of the system. In light of this definition, we call such a new intervention scheme a **routine intervention**.

There are different forms of maintenance and tests that can be arranged on a regular basis, such as cleaning, lubrication, replacing, repairing parts, partial or complete overhauls [*What is Preventive Maintenance?*, n.d.]. Therefore, the manipulation on the CEG given a routine intervention could vary for different maintenance actions.

Note that the difference between a routine intervention and a remedial intervention is that the remedial intervention is devised based on remedies and root causes, so the remedial intervention is essential for analysing equipment failures or breakdowns. On the other hand, the routine intervention is designed for PM, which is not restricted by fixing root causes of failures that have already happened. The maintenance discussed in this section is scheduled to prevent failures and is not dependent on the observed deterioration or failure. The target of the maintenance is not to rectify any defect, but to extend the lifetime of the repairable system. Some of the material we present in this section has appeared in our recent publication [Yu and Smith, 2021*a*].

2.4.1 The stochastic manipulation under the routine intervention

There are many types of checks or work practices that may be performed for PM. For example, the routine inspection and repair of transformers include monitoring the operating conditions, cleaning, checking the oil level in the conservator tank and oil gauge, checking for loose connection, checking pipe cracks and leakage, sealing leakage, checking the Buchholz relay for gas collection, and replacing the silica gel *etc* [*Maintenance Tips for Electrical Transformers*, 2020; Gautam, 2021]. For the conservator system we give in Example 3, if the routine maintenance controls oil leak, then we can simply identify its effects by treating it as a singular manipulation on e_{w_2,w_7}. When repairing or replacing a piece of equipment, however, the maintenance may affect a set of d-events, all of which are related to the maintained equipment.

This cannot simply be formalised as a singular manipulation.

Depending on the repaired part and the degree of repair, multiple florets can be influenced separately at the same time. Consider a floret $\mathcal{F}(w)$ which contains events being affected by the routine intervention, denoted by $x(E(w))$. Then some of the d-events in $x(E(w))$ or all d-events in $x(E(w))$ are faults or symptoms of the maintained part. Unlike for a remedial intervention, no specific event $x \in x(E(w))$ is targeted to be fixed which therefore induces a reduction in the corresponding transition probability.

Another feature of a routine intervention that differentiates it from a remedial intervention is that we assume that the result of inspections should be perfectly known since such maintenance is scheduled and should be well-documented. In comparison, the remedy can be uncertain. The root cause then need to be inferred when the remedy is not perfect, so the intervention indicator vector in this case is uncertain and needs to be drawn from a distribution conditional on the observed failure path and the partially observed remedy. However, given a routine intervention, the controlled events $\boldsymbol{x}$ are assumed to be perfectly informed from the observed and scheduled action. Here, for consistency, we still use $\mathcal{F}(\boldsymbol{w}^*)$ to denote the set of florets that are manipulated, where $\boldsymbol{w}^* = pa(W(\boldsymbol{x}))$.

The status of the maintained part depends on the degree of repair. Therefore, a routine intervention brings more uncertainty to the probability distribution over the events being affected by the maintenance. The post-intervened probability distribution assigned to $\mathcal{F} \in \mathcal{F}(\boldsymbol{w}^*)$ is required to be updated accordingly. This is a stochastic manipulation, although the distributions are manipulated in a different way from that for a remedial intervention. For $w \notin \boldsymbol{w}^*$, the distributions of the primitive probabilities retain the original distribution.

We define a new parameter $\phi \in (0, 1]$ to measure the effect of the routine intervention. Assume that the value of the discount parameter ϕ can be assessed and informed by the domain experts. The post-intervention distribution for every affected florets is assumed to be a function of the pre-intervention distribution and the discount parameter. Here we still use $\hat{f}(\cdot)$ to denote the post-intervened distribution, $G(\cdot)$ to denote the transformation that updates the distributions over the affected florets under a routine intervention. Then for $w \in \boldsymbol{w}^*$,

$$\hat{f}(\boldsymbol{\theta}_w) = G[f(\boldsymbol{\theta}_w)]. \tag{2.4.1}$$

The transformation G preserves the properties of the primitive probabilities so that $\sum_{e \in e(w)} \theta_e = 1$ and $\theta_e > 0$, and embodies the features of the manipulation we have

discussed above.

Smith [1990, 1992] have shown various ways to model the drift of information, *i.e.* an increasing uncertainty. One can map distributions to distributions through non-linear state space models. A possible transformation is the *power steady transformation* [Attwell and Smith, 1991; Smith, 1979] which takes the form:

$$\hat{f}(\boldsymbol{\theta}_w) \propto f(\boldsymbol{\theta}_w)^{\phi}. \tag{2.4.2}$$

The information is lost up to a proportionality that depends on the extent to which the equipment is repaired:

$$\mathbb{E}[\log \hat{f}(\boldsymbol{\theta}_w)] = \phi \mathbb{E}[\log f(\boldsymbol{\theta}_w)] + c, \tag{2.4.3}$$

for some constant c. Note that Freeman and Smith [2011*b*] also adopted this transformation explicitly for florets in CEGs for a different purpose from us. Instead of using it in a causal setting, the power steady transformation to model increasing entropy over time rather than because of an intervention.

Smith [1979] gave an example of this drift of information as applied to Beta distributions. Here we extend it to Dirichlet distribution. If the pre-intervention prior of $\boldsymbol{\theta}_w$ is $Dirichlet(\boldsymbol{\alpha}_w)$, then we can update each concentration parameter by

$$\hat{\alpha}_{w,w'} - 1 = \phi(\alpha_{w,w'} - 1), \tag{2.4.4}$$

where $w' \in ch(w)$ and $\hat{\alpha}_{w,w'}$ denotes the post-intervention hyperparameter so that $\hat{\boldsymbol{\alpha}}_w = (\hat{\alpha}_{w,w'})_{w' \in ch(w)}$. By this transformation, the mode of the distribution, denoted by $\hat{\boldsymbol{\vartheta}}_w = (\hat{\vartheta}_{w,w'})_{w'(w)}$, remains the same:

$$\hat{\vartheta}_{w,w'} = \frac{\phi(\alpha_{w,w'} - 1)}{\sum_{w_i \in ch(w)} \phi(\alpha_{w,w_i} - 1)} = \frac{\alpha_{w,w'} - 1}{\sum_{w_i \in ch(w)} (\alpha_{w,w_i} - 1)} = \vartheta_{w,w'}. \tag{2.4.5}$$

The information drifts in a way so that equation (2.4.3) is satisfied.

The collection of root-to-sink paths that are affected under such intervention is the set of paths passing through the manipulated florets $\mathcal{F}(\boldsymbol{w}^*)$. The collection of the controlled d-events of a routine intervention is the set of d-events labelled on the edges lying in $\mathcal{F}(\boldsymbol{w}^*)$, which is $x(E(\boldsymbol{w}^*))$. Then $\Lambda(\boldsymbol{w}^*)$ is the set of root-to-sink paths that contains edges whose transition probabilities are manipulated. When the post-intervention probabilities $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$ are known, we can estimate the effects of it through $\pi(\Lambda_y || \hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})$. The identifiability of the stochastic manipulation is the same as that discussed for the remedial intervention.

2.4.2 Composite manipulations on CEGs and identifiability

We have discussed in the previous section that the routine intervention can import singular manipulations or stochastic manipulations to system depending on what actions have been taken during the scheduled maintenance. Interestingly, if the field engineer detects a failure during the scheduled maintenance, then he need to fix this problem through diagnosing root causes. In this case we have both a remedial intervention and a routine intervention.

Here we discuss different scenarios in terms of the manipulations imported to the idle system by the intervention. We list the possibilities below:

1. a singular manipulation;
2. stochastic manipulations on $\mathcal{F}(\boldsymbol{w}^*)$, where the intervened positions $\boldsymbol{w}^*$ can contain only one position or a set of positions;
3. composite singular manipulations;
4. composite singular and stochastic manipulations.

We have shown that the identifiability of the effects for the first two scenarios. We next extend the formulae of the back-door theorem for the last two scenarios.

Composite singular manipulations. When the PM perfectly repairs some parts of some equipment, suppose this external intervention forces the d-events x_{r1} and x_{r2} to occur. Then each of the controlled d-event can be identified on a set of edges on the tree, denoted by $E(x_{r1}) \subseteq E_{\mathcal{C}}$ and $E(x_{r2}) \subseteq E_{\mathcal{C}}$ respectively. In this case, every edge in the set $e \in E(x_{r1})$ or $e \in E(x_{r2})$ is directly intervened by the routine PM and is forced to be passed through with probability 1: $\hat{\theta}_e = 1$. Then this routine intervention combines two singular manipulations simultaneously. If during the maintenance, more perfect repairs have been attained so that more d-events are directly intervened, then we force multiple separate singular manipulations simultaneously. We now give a general definition for this type of manipulations.

Definition 2.4.1. *Given the idle CEG $\mathcal{C}$, if there is a **composition of multiple separate singular manipulations** with controlled events $\boldsymbol{x} = \{x_1, ..., x_n\}$ imported into the idle system, then the controlled paths on the CEG can be identified as follows:*

$$\Lambda_{\boldsymbol{x}} = \Lambda(e(\boldsymbol{x})) = \bigcap_{x_i \in \boldsymbol{x}} \Lambda_{x_i} = \bigcap_{x_i \in \boldsymbol{x}} \Lambda(e(x_i)), \tag{2.4.6}$$

where $\Lambda_{\boldsymbol{x}} \subseteq \Lambda_{\mathcal{C}}$. The manipulated CEG of these composite manipulations is the conditioned CEG [Thwaites, 2013] $\mathcal{C}^{\Lambda_{\boldsymbol{x}}}$ where the transition probabilities along the

edges take value:

$$\pi_e^{\Lambda_{\boldsymbol{x}}}(w_j|w_i) = \frac{\sum_{\lambda\in\Lambda_{\boldsymbol{x}}}\pi(\lambda,\Lambda(e_{w_i,w_j}))}{\sum_{\lambda\in\Lambda_{\boldsymbol{x}}}\pi(\lambda,\Lambda(w_i))}. \tag{2.4.7}$$

When there are two d-events singularly manipulated, this is the simplest case of this type of composite manipulations. We continue with the example given above, where the composite d-events that are controlled under the routine intervention are $\boldsymbol{x} = \{x_{f,0}, x_r\}$. On the idle CEG, the controlled collection of paths are $\Lambda_{\boldsymbol{x}} = \Lambda_{x_{f,0}} \cap \Lambda_{x_r} = \Lambda(E(x_{f,0})) \cap \Lambda(E(x_r))$. The manipulated CEG is the CEG conditioned on $\Lambda_{\boldsymbol{x}}$. The path probability on the manipulated CEG can be evaluated using the formula

$$\hat{\pi}(\lambda) = \begin{cases} \frac{\prod_{e\in E_\lambda}\theta_e}{\theta_{e(x_{f,0})}\theta_{e'}} & \text{if } \lambda \in \Lambda(E(x_{f,0})) \cap \Lambda(e'), \text{ for } e' \in E(x_r), \\ 0 & \text{otherwise,} \end{cases} \tag{2.4.8}$$

Extending the back-door theorem for singular manipulations on the CEG that discussed in Section 2.2, we can show the effects of the composite singular manipulations are identifiable. Through finding the appropriate back-door partition $\Lambda_{\boldsymbol{z}}$, we can estimate the effect on Λ_y from the partially observed system

$$\pi(\Lambda_y||\Lambda_{\boldsymbol{x}}) = \sum_z \pi(\Lambda_y|\Lambda_{\boldsymbol{x}},\Lambda_z)\pi(\Lambda_z). \tag{2.4.9}$$

The partition $\Lambda_{\boldsymbol{z}}$ must satisfy the following criterion. For any $w_1 \in pa(W(x_{r1}))$, $w_2 \in pa(W(x_{r2}))$, the manipulated edges are $e_{w_1,w_1^*} \in E(x_{r1})$ and $e_{w_2,w_2^*} \in E(x_{r2})$. If

$$\pi(\Lambda_z|\Lambda(w_1),\Lambda(w_2)) = \pi(\Lambda_z|\Lambda(e_{w_1,w_1^*}),\Lambda(e_{w_2,w_2^*}))) \tag{2.4.10}$$

$$\pi(\Lambda_y|\Lambda_{\boldsymbol{x}},\Lambda_z) = \pi(\Lambda_y|\Lambda(w_1),\Lambda(w_2),\Lambda_{\boldsymbol{x}},\Lambda_z) = \pi(\Lambda_y|\Lambda(e_{w_1,w_1^*}),\Lambda(e_{w_2,w_2^*}),\Lambda_z) \tag{2.4.11}$$

hold for every element of $\{\Lambda_z\}$, then $\{\Lambda_z\}$ is the back-door partition.

Composite singular and stochastic manipulations. Suppose the routine intervention leads to a combination of a singular manipulation on x_{r1} and a stochastic manipulation on $x(E(\boldsymbol{w}^*))$ with new probabilities $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$. The controlled d-events are $\boldsymbol{x} = \{x_{r1}, x(E(\boldsymbol{w}^*))\}$. The manipulated paths on the CEG are $\Lambda_{\boldsymbol{x}} = \Lambda_{x_{r1}} \cap \Lambda_{x(E(\boldsymbol{w}^*))} = \Lambda_{x_{r1}} \cap \Lambda(\boldsymbol{w}^*)$. The manipulated CEG here is the CEG conditional on $\Lambda_{x_{r1}} \cap \Lambda(\boldsymbol{w}^*)$, denoted by $\mathcal{C}^{\Lambda_{\boldsymbol{x}}}$. The causal effects of a routine intervention with a singular manipulation and a stochastic manipulation can be shown to be identifiable on the CEG by adapting the back-door theorems for the singular manipulation and the stochastic manipulation. By forcing $\Lambda_{x_{r1}}$ to be passed along

and importing new probabilities $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$, we aim to estimate its effect on Λ_y by:

$$\begin{aligned}\pi(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*},\Lambda_{x_{r1}}) &= \sum_{x'\in x(E(\boldsymbol{w}^*))} \pi(\Lambda_y||\Lambda_{x_{r1}},\Lambda_{x'},\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})\pi(\Lambda_{x'}||\Lambda_{x_{r1}},\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}) \\ &= \sum_{x'\in x(E(\boldsymbol{w}^*))} \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y||\Lambda_{x'},\Lambda_{x_{r1}})\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\Lambda_{x'}||\Lambda_{x_{r1}}) \\ &= \sum_{x'\in x(E(\boldsymbol{w}^*))}\sum_{z} \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda_{x'},\Lambda_{x_{r1}},\Lambda_z)\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z)\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\Lambda_{x'}||\Lambda_{x_{r1}}).\end{aligned} \tag{2.4.12}$$

The back-door partition $\Lambda_{\boldsymbol{z}}$ must satisfy the following criteria. For any $w_1 \in pa(W(x_{r1}))$, $w_2 \in \boldsymbol{w}^*$, the manipulated edges are $e_{w_1,w_1^*} \in E(x_{r1})$ and $e_{w_2,w_2^*} \in E(\boldsymbol{w}^*)$. For any w_1, w_2 lying along the same path, if

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z|\Lambda(w_1),\Lambda(w_2))) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_z|\Lambda(e_{w_1,w_1^*}),\Lambda(e_{w_2,w_2^*})) \tag{2.4.13}$$

and

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda(w_1),\Lambda(w_2),\Lambda_{x'},\Lambda_{x_{f,0}},\Lambda_z) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_y|\Lambda(e_{w_1,w_1^*}),\Lambda(e_{w_2,w_2^*}),\Lambda_z) \tag{2.4.14}$$

hold for every element of $\{\Lambda_z\}$ and for all $\boldsymbol{x} = \{x_{f,0}, x(E(\boldsymbol{w}^*))\}$, then $\{\Lambda_z\}$ is the back-door partition.

We also need to ensure the probability $\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\Lambda_{x'}||\Lambda_{x_{r1}})$ can be estimated given the newly assigned probablities $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$ which is assumed to be known. This is equivalent to imposing a singular manipulation on x_{r1} on the conditioned CEG $\mathcal{C}^{\Lambda(\boldsymbol{w}^*)}$ with $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$. Therefore we also need a back-door theorem for identifying $\hat{\pi}^{\Lambda(\boldsymbol{w}^*)}(\Lambda_{x'}||\Lambda_{x_{r1}})$. Let $\{\Lambda_u\}$ denote the back-door partition that partitions $\Lambda(\boldsymbol{w}^*)$. Then $\{\Lambda_u\}$ must satisfy:

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_u|\Lambda(w_1)) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_u|\Lambda(e_{w_1,w_1^*})) \tag{2.4.15}$$

and

$$\pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_{x'}|\Lambda(w_1),\Lambda_{x_{r1}},\Lambda_u) = \pi^{\Lambda(\boldsymbol{w}^*)}(\Lambda_{x'}|\Lambda(e_{w_1,w_1^*}),\Lambda_u) \tag{2.4.16}$$

for $w_1 \in pa(W(x_{r1}))$ and $e_{w_1,w_1^*} \in E(x_{r1})$.

Thus, to identify the effects of the routine intervention which leads to a singular manipulation and a stochastic manipulation from the partially observed system, it is necessary to find the back-door partitions $\{\Lambda_z\}$ and $\{\Lambda_u\}$ separately satisfying the criteria specified above.

2.5 Learning the CEG from the intervened data

So far we have customised the intervention regimes and the bespoke intervention calculi for different types of maintenance. All these development can be applied in real data analysis so that the estimated effects are credible and consistent with the enormous amount of domain knowledge extant in this research area. In this section, we demonstrate an application of these causal algebras within the reliability domain.

A model selection algorithm can be designed to learn from data about the structure of the CEG that is most consistent with the data. The maximum a posterior (MAP) model selection is the most used Bayesian model selection method, although it may not necessarily be considered the best model. Barclay et al. [2013] and Cowell et al. [2014] formulated a MAP selection of CEGs analogously to that of BNs [Jaeger et al., 2006].

The model selected by the MAP algorithm best explains data. But there is no way simply from the observation we can deduce that this is a causal graph in the sense of Pearl [2009] and Thwaites [2013]. In our case, unless we assume a ground truth CEG perfectly informed by the domain experts, there is no exhaustive dataset from which to estimate probabilities with almost certainty. So we still need to learn the structure of CEGs. Following previous work [Cooper and Yoo, 2013; Cowell et al., 2014; Pensar et al., 2020], causal discovery could be cast as a Bayesian model selection problem and the MAP algorithm is one of the tool which has been well-developed and is easy to implement. Assuming that there are no unobserved confounders [Cowell et al., 2014], then we can make a further assumption that the best scoring model selected by the MAP algorithm is the CEG in idle mode when the system is not intervened and is causal. This then enables us to further perform causal analysis on the causal CEG.

If we let the prior of the primitive probability vector be Dirichlet, see equation (2.1.3). The log-likelihood score for a CEG can be computed explicitly in a closed form due to Dirichlet-Multinomial conjugacy.

$$\log Q(\boldsymbol{\theta};\mathcal{C}) = \sum_{u\in\mathbb{U}_{\mathcal{T}}} \log Q_u(\boldsymbol{\theta};\mathcal{C}) = \sum_{u\in\mathbb{U}_{\mathcal{T}}} (\log\Gamma(\alpha_u) - \log\Gamma(\alpha_{u+}) - \sum_{j=1}^{m_u} (\log\Gamma(\alpha_{uj}) - \log\Gamma(\alpha_{uj+}))). \tag{2.5.1}$$

We then use the log-posterior Bayes factor to compare any two candidate CEGs to find a better CEG structure with higher score. For any pair of candidate structure $\mathcal{C}_i, \mathcal{C}_j$, the log-posterior Bayes factor is [Collazo et al., 2018]:

$$lpBF(\mathcal{C}_i,\mathcal{C}_j) = \log q(\mathcal{C}_i) - \log q(\mathcal{C}_j) + \log Q(\mathcal{C}_i) - \log Q(\mathcal{C}_j), \tag{2.5.2}$$

where $\log q(\mathcal{C}_i)$ denotes the log prior.

In Section 2.3 and Section 2.4, we gave examples of transforming prior distributions of primitive probabilities as a result of either the remedial intervention or the routine intervention. We can accommodate these ideas when learning the best topology of a CEG and estimating the parameters using a Bayesian conjugate analysis when we have the intervened data.

We still compute the log-posterior Bayes factor using equation (2.5.2) to compare different models. We only need to incorporate the stochastic manipulations induced by either remedial interventions or routine interventions for model selection. When using the Dirichlet-Multinomial conjugacy, we first identify the set of situations $\boldsymbol{v}^* \in S_{\mathcal{T}}$ in the intervened position $\boldsymbol{w}^*$. Recall that the distributions over the florets $\mathcal{F}(\boldsymbol{w}^*)$ are manipulated in response to either a remedial or a routine intervention. We replace the priors of the manipulated primitive probabilities by the post-intervention priors. For example, we transform the hyperparameters of the Dirichlet priors as specified in equation (2.3.19) for a remedial intervention or equation (2.4.4) for a routine intervention. Then we recompute the posteriors using equation (2.1.4). The post-intervention posterior over the tree can be expressed as:

$$\begin{aligned}\hat{f}(\boldsymbol{\theta}) &= \prod_{u\in\mathbb{U}_{\mathcal{T}}} \frac{\Gamma(\sum_{j=1}^{m_u}\hat{\alpha}_{uj+})}{\prod_{j=1}^{m_u}\Gamma(\hat{\alpha}_{uj+})}\prod_{j=1}^{m_u}\theta_{uj}^{\hat{\alpha}_{uj+}} \\ &= \prod_{u\in U^*} \frac{\Gamma(\sum_{j=1}^{m_u}\hat{\alpha}_{uj+})}{\prod_{j=1}^{m_u}\Gamma(\hat{\alpha}_{uj+})}\prod_{j=1}^{m_u}\theta_{uj}^{\hat{\alpha}_{uj+}} \times \prod_{u\in\overline{U}} \frac{\Gamma(\sum_{j=1}^{m_u}\alpha_{uj+})}{\prod_{j=1}^{m_u}\Gamma(\alpha_{uj+})}\prod_{j=1}^{m_u}\theta_{uj}^{\alpha_{uj+}}.\end{aligned} \tag{2.5.3}$$

Here $\boldsymbol{v}^*$ are in the stages denoted by U^* and we let $\overline{U} = \mathbb{U}_{\mathcal{T}} \setminus U^*$. The log-likelihood score is revised by:

$$\begin{aligned}\log\hat{Q}(\boldsymbol{\theta};\mathcal{C}) = &\sum_{u\in U^*}(\log\Gamma(\hat{\alpha}_u) - \log\Gamma(\hat{\alpha}_{u+}) - \sum_{j=1}^{m_u}(\log\Gamma(\hat{\alpha}_{uj}) - \log\Gamma(\hat{\alpha}_{uj+})))+ \\ &\sum_{u\in\overline{U}}(\log\Gamma(\alpha_u) - \log\Gamma(\alpha_{u+}) - \sum_{j=1}^{m_u}(\log\Gamma(\alpha_{uj}) - \log\Gamma(\alpha_{uj+}))).\end{aligned} \tag{2.5.4}$$

2.6 Modelling time to failure

In this context, it is essential for various reliability specific modelling features to be included before the calculus we define above could be made practically implementable. The analysis of failure and maintenance data also concerns about mod-

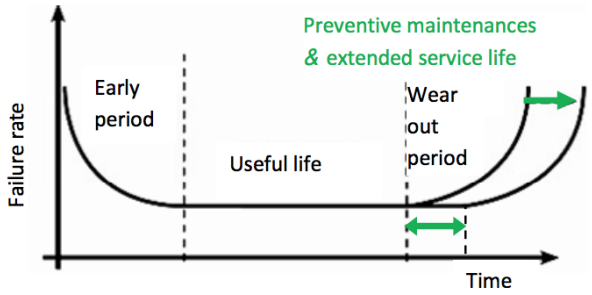


Figure 2.10: The bathtub curve [Bicen, 2015]

elling failure rates or lifetime [Bedford et al., 2001]. So we next demonstrate how the semantics of CEGs can be applied for this purpose and how the lifetime is affected by various interventions.

2.6.1 The semi-Markov process

The CEG is flexible in modelling the failure time of the machine. Barclay et al. [2015] introduced holding time distributions to the CEG so that the holding time depends only on the current and the receiving positions. In an event tree, let h_{wm} denote the holding time for situation $w \in S_{\mathcal{T}}$ just before transitioning to a child vertex of it along the m^{th} emanating edge of w. Note that not every edge in the tree has an associated transitioning time. Some florets represent classifications, such as component names, then the associated edges are not assigned holding time distribution. If the m^{th} emanating edge of w has holding time, denoted by $e_{wm} \in E^{\dagger}$, then we assume that this is drawn from a distribution with parameters $\boldsymbol{\rho}_{wm}$: $h_{wm} \sim f_{wm}(h; \boldsymbol{\rho}_{wm})$.

Under the assumption of the holding time being dependent only on the current and the next state, the CEG links directly to a semi-Markov process [Barclay et al., 2015]. Let $Q_{wm}(h)$ represent the renewal kernel by with transition probabilities θ_{wm} and holding times distribution $f_{wm}(h, \boldsymbol{\rho}_{wm})$:

$$Q_{wm}(h) = \theta_{wm} f_{wm}(h; \boldsymbol{\rho}_{wm}). \tag{2.6.1}$$

There are various choices for holding time distributions. Previous work [Shenvi and Smith, 2018; Shenvi et al., 2018] used Weibull holding time and considered Weibull-Inverse-Gamma conjugacy in a Bayesian setting. Then for $e \in E^{\dagger}$, let $h_e \sim Weibull(\beta_e, \eta_e)$, where β_e is the shape parameter and η_e is the scale parameter.

According to Bedford et al. [2001], the **bathtub effect** should be taken

care of when modelling the failure time. This theory divides the life of equipment into three periods depending on the characteristics of the failure rate: early failures, random failures, and worn-out failures [Lienig and Bruemmer, 2017]. The first phase is the early life of new equipment. New equipment usually starts with a very high failure rate associated with poor manufacture or poor installation. The random failures refer to the period with constant failure rate. The failure rate then starts to rise again when the equipment suffers significantly from aging and degradation when approaching the end of service life. The failure rate of equipment during its service life can then be plotted, see Figure 2.10, which has a shape like a bathtub.

The Weibull distribution is flexible for modelling a varying hazard [Bedford et al., 2001]. Lienig and Bruemmer [2017] also indicated that the summation of three Weibull functions for the three periods respectively can approximate the bathtub curve. We can let $\beta_e < 1$ for early failures, $\beta_e = 1$ for random failures and $\beta_e > 1$ for worn-out failures. In contrast, an exponential distribution can only model constant failure rate for equipment that does not experience wearing out until long after the expected life [Lienig and Bruemmer, 2017].

In reliability engineering, however, we are more interested in time-to-failure. From engineers reports, we can only observe the time between two consecutive failures, time between two consecutive maintenance and the maintenance time. We do not learn directly about the time spent in each phase of lifetime. If the CEG portrays the system of a single piece of equipment, then the total holding time assigned to the edges along a failure path represents the time-to-failure for the equipment. If the root node represents age zero, then the total holding time modelled on λ is the lifetime of the equipment. For any deteriorating path $\lambda \in \Lambda_{\mathcal{C}}^n$, the total holding time modelled on this path is the time-to-maintenance. Let $E_\lambda^\dagger \subseteq E_\lambda$ denote the set of edges lying along λ with holding time. Then the total time along path λ is $h_\lambda = \sum_{e \in E_\lambda^\dagger} h_e$.

However, the distribution of the sum of Weibull random variables does not have a closed form. So if we assign each edge a Weibull holding time, then it is hard to compute the probability density function of the holding time of a whole path. Here we have two alternative ways to solve this problem:

1. assign a Weibull density to each root-to-leaf path instead of a single edge, where we set $h_\lambda \sim Weibull(\beta_\lambda, \eta_\lambda)$;

2. use a Gamma holding time for each edge for computation simplicity so that the sum of the holding time along a path has a closed-form density.

For the first proposal, the probability density function is

$$f(h_\lambda) = \frac{\beta_\lambda}{\eta_\lambda} h_\lambda^{\beta_\lambda - 1} exp(-\frac{h_\lambda^{\beta_\lambda}}{\eta_\lambda}). \tag{2.6.2}$$

To perform a conjugate analysis, we adopt the assumption made by [Shenvi and Smith, 2018] to fix the shape parameter β_λ and assign an Inverse-Gamma prior to the scale parameter η_λ so that $\eta_\lambda \sim InverseGamma(\zeta_\lambda, \mu_\lambda)$, where the μ_λ is the scale parameter and ζ_λ is the shape parameter,

$$f(\eta_\lambda) = \frac{\mu_\lambda^{\zeta_\lambda}}{\Gamma(\zeta_\lambda)\eta_\lambda^{\zeta_{\lambda+1}}} exp(-\frac{\mu_\lambda}{\eta_\lambda}). \tag{2.6.3}$$

It then follows that the posterior of η_λ is $InverseGamma(\zeta_{\lambda+}, \mu_{\lambda+})$. The hyper-parameters are updated by $\zeta_{\lambda+} = \zeta_\lambda + n_\lambda$, where n_λ is the number of units whose failure or deteriorating processes are modeled on λ, and $\mu_{\lambda+} = \mu_\lambda + \sum_{l=1}^{n_\lambda} h_{\lambda l}^{\beta_\lambda}$, where $h_{\lambda l}$ is the time-to-failure or time-to-maintenance of the l^{th} unit in the n_λ units.

For the second proposal, a Gamma distribution is popular in modelling the lifetime in either reliability engineering or health sciences. We can assign a Gamma holding time to $e \in E_\lambda^\dagger$ for $\lambda \in \Lambda_\mathcal{C}$ with shape parameter ξ_e and rate parameter ρ_λ, $h_e \sim Gamma(\xi_e, \rho_\lambda)$. The sum of Gamma variables is still Gamma, then $h_\lambda \sim Gamma(\xi_\lambda, \rho_\lambda)$ where $\xi_\lambda = \sum_{e \in E_\lambda^\dagger} \xi_e$. A Gamma-Gamma conjugate inference can be employed here to estimate the parameters. Assume that ξ_e is known for every $e \in E^\dagger$ and ρ_λ has a $Gamma$ prior $\rho_\lambda \sim Gamma(g_1, g_2)$. Then the posterior of it is still Gamma. Let g_{1+}, g_{2+} denote the hyperparameters. Then $g_{1+} = g_1 + \sum_{\lambda \in \Lambda_\mathcal{C}} n_\lambda \xi_\lambda$ and $g_{2+} = g_2 + \sum_{l=1}^{n_\lambda} h_{\lambda l}$. In this case, we can still focus on any edge's holding time and could manipulate its distribution easily. Note that $\xi_e > 1$ models a increasing failure rate, while $\xi_e < 1$ models a decreasing failure rate and $\xi_e = 1$ models a constant failure rate. Therefore we can still approximate the bathtub curve by assigning appropriate shape parameters of edges along the root-to-sink paths. In this case, we should have the sum of the shape parameters $\xi_\lambda > 1$.

Shenvi and Smith [2018] revised the definition of position when holding time distributions are assigned to edges. Two edges are defined to be in the same cluster $c \in \mathbb{C}$ if the two edges have the same holding time distribution, where $\mathbb{C}$ denotes the set of clusters. Edges in the same cluster are coloured the same. Then for situations $v_i, v_j \in \mathcal{S}_\mathcal{T}$ in the same stage, if $\mathcal{T}(v_i)$ and $\mathcal{T}(v_j)$ are isomorphic in terms of structure and colouring of both edges and vertices. We can simply adopt this idea whilst modeling holding times with a Gamma distribution instead. When we only model the time along the whole root-to-sink path on the tree by the Weibull,

the paths that share the same Weibull time distribution can be coloured the same. The definition of positions by Shenvi and Smith [2018] can then be directly used to determine the topology of the CEG.

2.6.2 Effects of an intervention on time to failure

For a repairable system, Kijima [1989] and Guessoum and Aupied [2010] have shown that the PM impacts equipment's aging. In particular, the maintained equipment is "rejuvenated". When modelling the lifetime of a repairable system by the bathtub curve, we can directly visualise the effect of such maintenance on the failure rate and lifetime. This is shown in Figure 2.10 [Bicen, 2015]. The curve representing the worn-out period is shifted towards right as a result of the PM so that the rising failure rate is decelerated and the worn-out period before failure is extended. This plot shows that the scheduled inspection often takes place before wearing out to prevent accelerated failure rate.

Here we have adapted the **Arithmetic Reduction of Age** (ARA) model [Doyen and Gaudoin, 2004] for modelling the residual lifetime of the system after maintenance. This class of model is designed to evaluate the efficiency of the maintenance in terms of the **virtual age** of the system, which is a positive function of the observed real age [Kijima, 1989; Doyen and Gaudoin, 2004]. The ARA model assumes that the virtual age of the maintained equipment is discounted due to the PM.

Let T_s represent the failure time of an equipment with observed age s. Let the cumulative distribution function of the failure time of an AGAN equipment be

$$F(t) = P(T_0 \leq t). \tag{2.6.4}$$

Then the survival function is

$$P(T_s > t) = \frac{1 - F(s+t)}{1 - F(s)}. \tag{2.6.5}$$

In an idle CEG system, when there is no intervention and the machine is AGAN and not degraded, the real age of the machine is 0. The failure time for the idle system is then T_0. The total holding time H_λ of a root-to-sink path $\lambda \in \Lambda_{\mathcal{C}}$ associated to the equipment has the same distribution as T_0 conditional on the particular failure process represented by λ, denote it by T_0^λ. This means $H_\lambda \stackrel{d}{=} T_0^\lambda$, or equivalently,

$$P(H_\lambda > t) = P(T_0^\lambda > t) = 1 - F_\lambda(t), \tag{2.6.6}$$

where $F_\lambda(t)$ is the reliability distribution given failure process λ. In this section, we simply explain the effect on Weibull lifetime.

Suppose the routine maintenance is scheduled to take place at times $\{m_1, m_2, ...\}$ for an equipment. Assume that at each time, a set of units of this equipment are inspected and maintained. Suppose at time m_i when the i^{th} maintenance is conducted, the age of the targeted equipment is τ. We introduce a new parameter $\xi \in [0, 1]$ for the system to represent the degree of repair after the routine intervention [Kijima, 1989; Guessoum and Aupied, 2010]. Kijima [1989] defined the degree of repair so that $\xi = 1$ corresponding to a minimal repair while $\xi = 0$ corresponding to a perfect repair. The former returns the status of the maintained part to a functioning condition just prior to the repair while the latter returns the status of the maintained part to AGAN [Barlow and Proschan, 1996]. If the equipment is remedied, the status of the equipment is returned to AGAN. So after the remedial intervention, $\xi = 0$. If the equipment is worn-out but not maintained, then $\xi = 1$. If there is a routine intervention, then ξ could be any value between 0 and 1. For the sequence of maintenance, we can define the degree of repair $\xi_i \in [0, 1]$ for the maintenance taking place at m_i. In this thesis, we only focus on the one-time maintenance and its corresponding ξ. In the last chapter, we will briefly discuss the extension to dynamic process.

The preventive maintenance aims to extend the lifetime of the equipment instead of fixing the defect of the equipment and it is scheduled in advance. We assume that the degree of repair does not depend on the observed process, *i.e.* ξ is independent of λ. Kijima [1989] and Guessoum and Aupied [2010] defined the virtual age after the maintenance to be a function of the real age τ and ξ. Specifically, after the maintenance, the equipment is rejuvenated with a virtual age $\xi\tau$. The post-intervened time to failure distribution, *i.e* the residual lifetime distribution, is:

$$P(T^\lambda_{\xi\tau} > t) = P(H_\lambda > t | H_\lambda > \xi\tau). \tag{2.6.7}$$

Let $\hat{H}_\lambda$ denote the post-intervened time to failure for failure process λ. It then has the same distribution as $T^\lambda_{\xi\tau}$: $\hat{H}_\lambda \overset{d}{=} T^\lambda_{\xi\tau}$. We can evaluate the reliability of the maintained equipment by:

$$P(\hat{H}_\lambda > t) = P(T^\lambda_{\xi\tau} > t) = \frac{1 - F_\lambda(t + \tau\xi)}{1 - F_\lambda(\tau\xi)}. \tag{2.6.8}$$

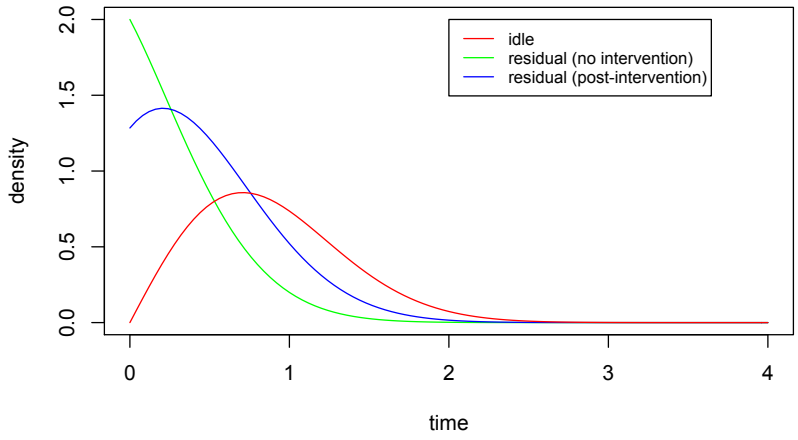


Figure 2.11: Comparing failure time density between idle system and the intervened system on a Weibull toy example.

If we use a Weibull lifetime for each path, then the reliability is

$$P(\hat{H}_\lambda > t) = exp(-\frac{(t+\xi\tau)^{\beta_\lambda}}{\eta_\lambda} + \frac{(\xi\tau)^{\beta_\lambda}}{\eta_\lambda}). \tag{2.6.9}$$

The post-intervened density of failure time is then

$$\hat{f}_\lambda(t) = \frac{\beta_\lambda}{\eta_\lambda}(t+\tau\xi)^{\beta_\lambda - 1} exp(-\frac{(t+\tau\xi)^{\beta_\lambda}}{\eta_\lambda} + \frac{(\xi\tau)^{\beta_\lambda}}{\eta_\lambda}). \tag{2.6.10}$$

This is still conjugate to the InverseGamma distribution so that the hyperparameters in posterior are $\zeta_{\lambda+} = \zeta_\lambda + n_\lambda$ and $\mu_{\lambda+} = \mu_\lambda + \sum_{l=1}^{n_\lambda}(h_{\lambda l} + \xi\tau)^{\beta_\lambda} - (\xi\tau)^{\beta_\lambda}$.

Figure 2.11 gives a toy example on time-to-failure modelled by $Weibull(2,1)$. The red line plots the density of holding time for idle system. Assume the engineer arranges a routine repair on this equipment when it is 1 year old. Let the degree of repairing be $\xi = 0.5$. The virtual age is shortened to 0.5 after the repair. The residual lifetime density using formula in equation (2.6.2) is plotted by the blue curve. If no intervention takes place, then the residual reliability is conditional on the real age 1, whose density is plotted in green in the figure. Due to the gain of life from the routine intervention, the residual life density curve shifts towards right compared to the no-intervention curve and the failure rate is decelerated by the routine intervention.

2.6.3 Learning the CEG taking account of lifetime

The log-likelihood score for a CEG $\mathcal{C}$ can be decomposed into the score of transitions along the paths and the score of holding time:

$$\log Q(\mathcal{C}) = \underbrace{\sum_{u\in\mathbb{U}_{\mathcal{T}}} \log Q_{u_i}(\boldsymbol{\theta};\mathcal{C})}_{\text{score of transition}} + \underbrace{\sum_{\lambda\in\Lambda_{\mathcal{C}}} \log Q_{\lambda_i}(\boldsymbol{h};\mathcal{C})}_{\text{score of time}}. \tag{2.6.11}$$

If we have Weibull time assigned to the whole path, then the log-likelihood score of a CEG has a closed form due to Weibull-InverseGamma conjugacy. The score of time has the expression:

$$\sum_{\lambda\in\Lambda_{\mathcal{C}}} \log Q_{\lambda_i}(\boldsymbol{h};\mathcal{C}) = \sum_{\lambda\in\Lambda_{\mathcal{C}}} \zeta_\lambda \log \mu_\lambda - \zeta_{\lambda+} \log \mu_{\lambda+} + \log\Gamma(\zeta_{\lambda+}) - \log\Gamma(\zeta_\lambda). \tag{2.6.12}$$

We can learn the CEG from the intervened data with log-likelihood score:

$$\log \hat{Q}(\mathcal{C}) = \log \hat{Q}(\boldsymbol{\theta};\mathcal{C}) + \log \hat{Q}(\boldsymbol{h};\mathcal{C}), \tag{2.6.13}$$

We have demonstrated how to take into account the stochastic manipulations in Section 2.5. We can also use the residual lifetime distribution after intervention introduced in Section 2.6.2 when learning the CEG with holding times. For the example of the Weibull lifetime, we have

$$\log \hat{Q}(\boldsymbol{h};\mathcal{C}) = \sum_{\lambda\in\Lambda_{\mathcal{C}}} \hat{\zeta}_\lambda \log \hat{\mu}_\lambda - \hat{\zeta}_{\lambda+} \log \hat{\mu}_{\lambda+} + \log\Gamma(\hat{\zeta}_{\lambda+}) - \log\Gamma(\hat{\zeta}_\lambda), \tag{2.6.14}$$

where $\hat{\mu}_\lambda = \mu_\lambda$, $\hat{\mu}_\lambda = \mu_\lambda$, $\hat{\zeta}_{\lambda+} = \hat{\zeta}_\lambda + n_\lambda$, $\hat{\mu}_{\lambda+} = \hat{\mu}_\lambda + \sum_{l=1}^{n_\lambda} (h_{\lambda l} + \xi\tau)^{\beta_\lambda} - (\xi\tau)^{\beta_\lambda}$.

In summary, so far we have developed causal algebras for two novel types of interventions in the domain of reliability: the remedial intervention and the routine intervention. We have demonstrated how the lifetime of the machine is modelled on the CEG and how it is affected by the domain-specific interventions. On the basis of these developments, we have shown how to incorporating the causal algebras of the domain-specific intervention and the casual effects on lifetime to improve the learning algorithm of the CEGs. Next, we will focus on the text data and work on applying the CEG for embedding the causal relationships which can be extracted from the free texts.

Chapter 3

A Hierarchical Causal Model

This chapter will provide a fruitful discussion on the application of the CEG to system reliability. In Chapter 1, we gave an example of the engineering reports. The insights gained from data exploration provoked us into designing generic methodology to systematically extract and embed the causal dependencies which are implicitly encoded within the engineering reports. In this chapter, we will develop an innovative hierarchical model, called the Global net-Chain Event Graph (GN-CEG) model, to fulfil this task. This framework has a causal network called the Global Net (GN) at its surface level and a causal CEG at the deepest level.

This research focuses on one time maintenance activities which were demonstrated in the previous chapter. However the proposed model can be extended to model the dynamic process of maintenance and failures by having a reduced dynamic CEG [Shenvi and Smith, 2018] at the deepest level. We will briefly discuss this in the last chapter of this thesis.

Section 3.1 draws up a blueprint for projecting natural language texts onto a causal CEG and gives an overview of the two steps required for building up the proposed GN-CEG model: preprocessing and causality embedding. We also establish the new concepts that are essential in this theoretical framework to transform the text embeddings for causal inference. Section 3.2 decomposes the preprocessing step to show how the causally related events can be extracted from each document. We introduce a transparent and simple way to construct the GN and formalise the link between the GN and the documents. In Section 3.3, we explain how a CEG can be applied to provide a platform for further embedding the causal relationships from the GN.

In this chapter, we restrict the discussion to the data in absence of missingness. This equivalently assumes that every event or variable is observable.

3.1 The GN-CEG model

In Chapter 2, we emphasised the advantages of using CEGs in representing the unfolding of failure processes and deteriorating processes for system reliability. Here we proceed to design a new framework to automate the process of causal discovery from maintenance logs on a CEG so that the domain-specific interventions defined in the previous chapter can be well supported [Yu and Smith, 2021*c*]. This consists of a hierarchical model with two layers which we call it the **GN-CEG model**.

Figure 3.1 demonstrates the architecture of this framework. Given the text descriptions in the maintenance logs, we first construct the surface layer of the model by the preprocessing step. This is a causal network called the **Global Net** (GN). The deeper layer of the model is a causal CEG.

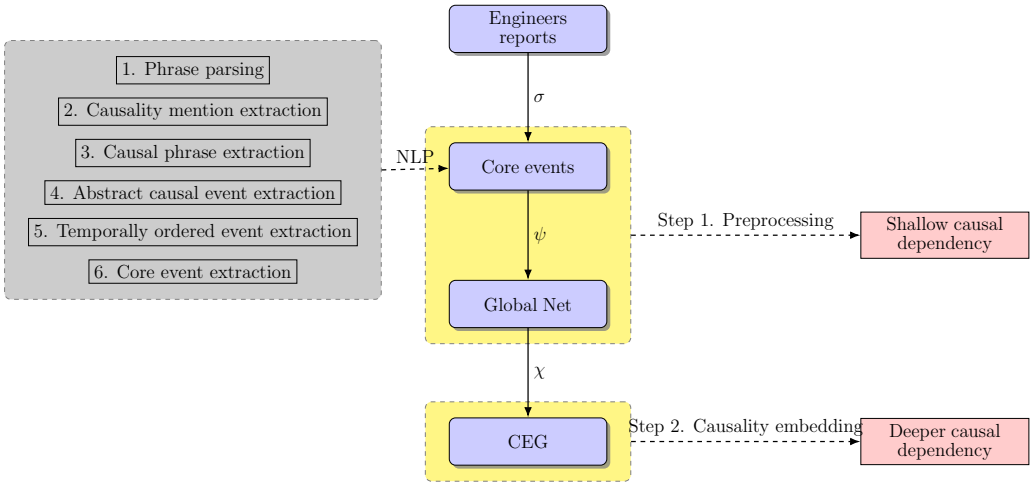


Figure 3.1: The proposed hierarchical causal framework.

Let D_0 represent all documents we pick for a specific context, for example the faults related to a specific system. We select documents from D_0 that have implicit or explicit causal patterns and implement basic text cleaning on them to obtain a dataset D, such as removing grammar mistakes, replacing abbreviations and spelling mistakes *etc.* Let N_D denote the number of documents in D. If the d^{th} document consists of a sequence of N_d words, then we denote this document by $\boldsymbol{\omega}_d = (\omega_{d1}, ..., \omega_{dN_d}) \in D$, where ω_{dj} denotes a word in this document. For each selected document, we make the following assumption for the model we built.

ASSUMPTION 3.1.1. *Each document describes a single failure process or deteriorating process of equipment. Each document may have more than one sentence.*

We have defined in the previous chapter that each root-to-sink path on the CEG portrays either a failure process or a deteriorating process depending on whether it terminates in w_∞^f or w_∞^n. Therefore, on the basis of this feature and Assumption 3.1.1, we can expect each document $\boldsymbol{\omega}_d$ to be associated with a root-to-sink path $\lambda \in \Lambda_{\mathcal{C}}$ on the CEG. We express this as a map

$$\Delta : (\boldsymbol{\omega}_d, \Omega) \mapsto \lambda. \tag{3.1.1}$$

The required parameters set Ω will be carefully defined in the next section. First, in what follows, we give an overview of how to decompose this map according to the procedures depicted in Figure 3.1.

We first propose a new natural language processing (NLP) algorithm to preprocess the text descriptions in the engineering reports. Section 3.2 will explain this NLP algorithm step by step. By employing this text mining technique, we aim to extract the causally related events from texts using linguistic patterns. The extracted events here are called the **core events**, denoted by $\boldsymbol{u}_d = \{u_{d,1}, ..., u_{d,n_{u_d}}\}$ for document d and $n_{u_d} \in \mathbb{Z}^+, n_u > 1$. Let $\pi_{\boldsymbol{u}_d}$ denote the partial ordering of the extracted core events. Note that the core events include both the events that have happened and eventually led to a failure and the maintenance that is undertaken. Let $\mathbb{U}_D$ denote the set of all core events for dataset D and $\boldsymbol{\Pi}_D$ denote all possible orderings over these core events. Using the proposed NLP algorithm, we can extract a set of ordered core events for the d^{th} document, denoted by $(\boldsymbol{u}_d, \pi_{\boldsymbol{u}_d})$. This procedure can be represented by a function:

$$\sigma : (\boldsymbol{\omega}_d, \Omega_{NL}) \mapsto (\boldsymbol{u}_d, \pi_{\boldsymbol{u}_d}), \tag{3.1.2}$$

where $\Omega_{NL} \subset \Omega$ is a subset of parameters that are required for core events extraction. The codomain of σ is $\mathbb{U}_D \times \boldsymbol{\Pi}_D$.

To order these core events in a consistent way, we design a new causal graphical framework, the GN, to register the partial orderings. The structure of the GN is assumed to be a directed acyclic graph (DAG) in this thesis, denoted by $G^* = (V^*, E^*)$. The vertex set V^* corresponds a set of variables constructed from the core events. We call these variables the **core event variables**, and denote them by $\boldsymbol{L} = \{L_1, \cdots, L_{nL}\}$. Each directed edge $e \in E^*$ connects a cause $V_i \in V^*$ to its effect $V_j \in V^*$. The causal relationships embedded within a GN are based on linguistic patterns. We therefore call such causal relations the **shallow causal dependency**. We will establish a systematic way to match each document $\boldsymbol{\omega}_d$ to $\boldsymbol{l}_d = \{l_{d1}, \cdots, l_{dm}\}$, which are the values of the core event variables

$\boldsymbol{L}_d = \{L_{d1}, \cdots, L_{dm}\} \subseteq \boldsymbol{L}$. We can find the vertices on the GN corresponding to $\boldsymbol{L}_d$, denoted by V_d. Let $G_d = (V_d, E_d)$ denote a subgraph of the GN. The edge $e_{v,v'}$ is in E_d if $v, v' \in V_d$ and $e_{v,v'} \in E^*$. From this graph we can read the order of the core event variables $\boldsymbol{L}_d$ from G_d. Let

$$\Gamma : (\boldsymbol{\omega}_d, \Omega_{NL}) \mapsto \boldsymbol{l}_d. \tag{3.1.3}$$

To properly define this map, given the map σ, we need to formally define a map from $(\boldsymbol{u}_d, \pi_{\boldsymbol{u}_d})$ to $\boldsymbol{l}_d$:

$$\psi : (\boldsymbol{u}_d, \pi_{\boldsymbol{u}_d}) \mapsto \boldsymbol{l}_d. \tag{3.1.4}$$

Then the map Γ can be written as a composition $\Gamma = \psi \circ \sigma$.

Though causal semantics represented on the GN embed the shallow causal relations, some more complex relations between events raised in reliability cannot be well-modelled by a GN alone. Therefore in our proposed framework we have one more layer to embed the causally related core events further onto a more refined CEG in order to better understand the causal dependency beyond the causal relations deduced from syntactic or semantic patterns. We call the causal dependency that can be read from the CEG the **deeper causal dependency** to distinguish these from the shallow causal dependency represented on the GN. In Section 3.3, we will specify a method to transform the shallow causal dependency to the deeper causal dependency. In particular, we aim to match every $\boldsymbol{l}_d$ to a root-to-sink path on the CEG by

$$\chi : (\boldsymbol{l}_d, \Omega_{NC}) \mapsto \lambda, \tag{3.1.5}$$

where Ω_{NC} denotes the set of parameters required here for projecting a subgraph G on the GN onto a path on the CEG. Then $\Omega = \{\Omega_{NL}, \Omega_{NC}\}$ and the function to map a document onto a latent path is $\Delta = \chi \circ \Gamma$.

Note that the GN is constructed and extracted from observations so it is observations based and therefore explicitly acknowledged as a pre-processing step. The conditional independence relationships between core event variables are extracted from engineers texts. By contrast, at the deepest layer, according to how much domain knowledge is provided, we may have

1. a CEG whose topology is completely known,
2. a set of candidate structures of the CEG, but which best fits the data needs to be inferred,
3. an event tree and we need to design an algorithm to select the best structure

of the CEG.

Suppose we need to learn the path λ on the CEG for $\boldsymbol{l}_d$ by designing an algorithm given the hierarchical architecture, then the first case is the simplest because we can skip the steps to compare structures, which is required by the second case, or to search over all possible structures, which is required by the third case. This thesis only focuses on the simplest scenario to lay the foundation for the future research. The first scenario assumes the full topology of the CEG is completely known, so the following information is required to be informed from background knowledge or be provided by domain experts:

1. the sequences of d-events that might lead to a system failure or happen before maintenance when the failure has not been observed yet,

2. the conditional independence relationships between these d-events.

The first piece of information allows us to build the event tree and we assume this is the ground truth event tree for the selected system. The second piece of information listed above allows us to elicit the stages and so the CEG can be constructed. Then we assume a priori that this elicited CEG is causal through which the causal relations between conditional events are given by the structure. Having a ground truth structure lying at the deeper layer of the model enables us to avoid the model being completely dependent on the observations, especially when the dataset is insufficient for providing the information about every possible failure or deteriorating process of the selected system. So in this case, unlike the GN, the conditional independence relationships defined on the CEG are not extracted from engineers reports. Note here that the ground truth tree can encapsulate those unobserved events and the events have not happened yet when collecting the dataset D which are known to domain experts. Therefore, the use of CEG cannot only provide richer semantics for representing the causal relationships but also fill the gap between what we observe and what could happen. This is one of the reasons that this proposal is supported by two layers.

In Section 3.2, we shows how to transform each document $\boldsymbol{w}_d$ to $\boldsymbol{l}_d$. In Section 3.3, we formalise the probability $p(\boldsymbol{l}_d|\lambda)$ where $\boldsymbol{l}_d$ is treated as observations while the edges or the positions traversed by the root-to-sink path λ are treated as hidden states.

Table 3.1: Notations for embedding shallow causal dependency

$\boldsymbol{\omega}_d$	the d^{th} document
$\boldsymbol{u}_d$	the core events extracted for the d^{th} document
$\pi_{\boldsymbol{u}}$	the partial causal order of the core events $\boldsymbol{u}$
$\boldsymbol{L}$	the core event variables constructed from the core events extracted from D
Δ	the map projecting from a document onto a root-to-sink path on the CEG
Ω	the set of parameters required for learning latent paths on the CEG for documents
Ω_{NL}	the set of parameters required for extracting causally ordered core events from documents
Ω_r	the set of grammar rules for phrase parsing
Ω_s	the set of linguistic patterns for extracting causality mentions
Ω_{NC}	the set of parameters required for learning latent paths on the CEG for subgraphs on the GN
σ	the map projecting from a document onto a set of ordered core events
Γ	the map projecting from a document onto a set of values of the core event variables represented on the GN
ψ	the map projecting from each document's ordered core events onto a set of values of the core event variables represented on the GN
χ	the map projecting from a a set of values of the core event variables represented on the GN onto a root-to-sink path on the CEG
α	the map implementing phrase parsing
β	the map implementing causality mention extraction
γ	the map implementing causal phrase extraction
ι	the map implementing abstract causal events extraction
μ	the map implementing temporally ordered events extraction
ϕ	the map implementing core events extraction
Ξ	the map that finds a core event variable for each core event
$\boldsymbol{A}$	word indices of phrases
$\boldsymbol{B}$	word indices of causality mentions
$\boldsymbol{C}$	word indices of causality phrases
$\boldsymbol{K}$	word indices of abstract causal events
$\boldsymbol{\xi}$	word indices of temporally ordered events
$\boldsymbol{\Sigma}$	word indices of core events
$\boldsymbol{s}$	sentence indices
$\boldsymbol{\tau}$	part-of-speech tags

3.2 Shallow causal dependency

In this section, we decompose the map Γ to show how to match a document to the values of a set of core event variables depicted on the GN. Beginning with defining the NLP algorithm devised for core events extraction using the function σ in Section 3.2.1, we then give an example of the GN and demonstrate how to construct this in Section 3.2.2. This is followed by formulating the function ϕ that maps the core events of each document to a set of ordered core event variables. The notations used in this section are listed in Table 3.1.

3.2.1 Core events extraction

We first specify the parameters Ω_{NL} required for the functions Γ and σ for text processing. Let $\Omega_{NL} = \{\Omega_r, \Omega_s\}$, where Ω_r denotes the grammar rules and Ω_s denotes a set of linguistic causal patterns. Specifically, Ω_r is a set of grammar rules defined in order to parse a sentence into noun phrases or verb phrases. For example, a noun phrase can be define as "determinator (if exists) + adjective + noun(s)". The linguistic causal patterns Ω_s are defined as a set of R tuples $\{(\boldsymbol{\omega}_C, < \boldsymbol{\omega}_A, \boldsymbol{\omega}_B >, r_{\boldsymbol{\omega}_C}(\boldsymbol{\omega}_A, \boldsymbol{\omega}_B))\}$. Here $\boldsymbol{\omega}_C = (\omega_{C_1}, \cdots, \omega_{C_{n_C}})$ is a vector of words that represents a causal connective, where the subscript C is a set of word indices $C = \{C_1, \cdots, C_{n_C}\}$. The pair $< \boldsymbol{\omega}_A, \boldsymbol{\omega}_B >$ is a pair of events in a sentence that are connected by $\boldsymbol{\omega}_C$, where $\boldsymbol{\omega}_A = (\omega_{A_1}, \cdots, \omega_{A_{n_A}})$ with index set $A = \{A_1, \cdots, A_{n_A}\}$, and $\boldsymbol{\omega}_B = (\omega_{B_1}, \cdots, \omega_{B_{n_B}})$ with index set $B = \{B_1, \cdots, B_{n_B}\}$. The causal relationship between $\boldsymbol{\omega}_A$ and $\boldsymbol{\omega}_B$ is determined by $r_{\boldsymbol{\omega}_C}(\boldsymbol{\omega}_A, \boldsymbol{\omega}_B)$. For example, the causal connective $\boldsymbol{\omega}_C$ is the word "after" in a sentence "After leakage, the system is out-of service." Then $\boldsymbol{\omega}_A$ is "leakage" and $\boldsymbol{\omega}_B$ is "the system is out-of service". In this case we can let $r_{\boldsymbol{\omega}_C}(\boldsymbol{\omega}_A, \boldsymbol{s}_B)$ be the rule that $\boldsymbol{\omega}_A$ is a cause of $\boldsymbol{\omega}_B$ as long as $\boldsymbol{\omega}_A$ does not start with a number and express the ordered events as $\boldsymbol{\omega}_A \prec \boldsymbol{\omega}_B$. Note that we always assume that **a cause happens before its effects**. This assumption is also critical in our proposed NLP algorithm.

Now we establish an innovative algorithm, represented by the function σ, for extracting causally ordered core events. This algorithm has developed from two established methods [Chambers et al., 2014; Zhao et al., 2017] for extracting ordered events that have been briefly reviewed in Section 1.2. Our algorithm adapts some ideas of Zhao et al. [2017] to parse a sentence into events depending on simple causal patterns and combines these ideas with the CAEVO programming developed by Chambers et al. [2014] to extract temporally ordered events. The innovation of our algorithm lies in supplementing the causal relations encoded within a sentence

with the temporal relations that are implicitly causal [Yu and Smith, 2021*c*]. The algorithm can be decomposed into a sequence of steps:

1. phrase parsing,
2. causality mention extraction,
3. causal phrase extraction,
4. abstract causal event extraction,
5. temporally ordered event extraction,
6. core event extraction.

Now we demonstrate each step in detail.

Step 1. Phrase parsing. Given the predefined grammar rules Ω_r, for each $\boldsymbol{\omega}_d = (\omega_{d1}, \cdots, \omega_{dN_d})$, we split the document into sentences. Then we tokenize each sentence into word tokens and find the part-of-speech of each word token in each sentence. Next we check whether there is a match between the parts of speech and the grammar rules. Let $\boldsymbol{\omega}_{\boldsymbol{A}^d} = \{\boldsymbol{\omega}_{A_1^d}, ..., \boldsymbol{\omega}_{A_n^d}\}$ denote the set of n noun phrases or verb phrases extracted from each document. Each phrase $\boldsymbol{\omega}_{A_j^d}$, $j \in \{1, \cdots, n\}$, is a subvector of $\boldsymbol{\omega}_d$. The subscript A_j^d is a subset of consecutive word indices in the d^{th} document so that $A_j^d = \{a_{j1}, \cdots, a_{jm}\} \subseteq \{d1, \cdots, dN_d\}$. Then $\boldsymbol{A}^d = \{A_1^d, ..., A_n^d\}$ is a set of phrases indices.Our next step is to define a map α to parse a document into phrases:

$$\alpha : (\boldsymbol{\omega}_d, \Omega_r) \mapsto (\boldsymbol{\omega}_{\boldsymbol{A}^d}, \boldsymbol{s}_{\boldsymbol{A}^d}, \boldsymbol{\tau}_{\boldsymbol{A}^d}). \tag{3.2.1}$$

The output of this map $\boldsymbol{s}_{\boldsymbol{A}^d} = \{s_{A_1^d}, \cdots, s_{A_n^d}\}$ is a set of sentence indices, where $s_{A_j^d}$ is the sentence index for the phrase $\boldsymbol{w}_{A_j^d}$. This function also returns the part-of-speech (POS) tag for each word in each extracted phrase. Denote the set of POS tags for the d^{th} document by $\boldsymbol{\tau}_{\boldsymbol{A}^d} = \{\boldsymbol{\tau}_{A_1^d}, \cdots, \boldsymbol{\tau}_{A_n^d}\}$, where $\tau_{A_j^d} = \{\tau_{j1}, \cdots, \tau_{jm}\}$ is the set of POS tags of word tokens in $\boldsymbol{\omega}_{A_j^d}$. The pseudo-code for this step is given in Algorithm 1.

Algorithm 1 Phrase parsing using α

Input: $\boldsymbol{w}_d$, Ω_r

Output: $\boldsymbol{\omega}_{\boldsymbol{A}^d}, \boldsymbol{s}_{\boldsymbol{A}^d}, \boldsymbol{\tau}_{\boldsymbol{A}^d}$

Split the document $\boldsymbol{\omega}_d$ into sentences $(\boldsymbol{\omega}_{d1}, ..., \boldsymbol{\omega}_{dJ})$
$\nu = 1$
for $i = 1$ to J **do**
  Tokenize $\boldsymbol{\omega}_{di}$ into words sequence $(\omega_{i1}, ..., \omega_{in_i})$
  Find the part-of-speech tags associated with $\boldsymbol{\omega}_{di}$, denoted by $(\tau_{i1}, \cdots, \tau_{in_i})$
  $l = 1$
  while $l \leq n_i$ **do**
    if the consecutive tags $(\tau_l, \tau_{l+1}, ..., \tau_{l+u}) \subseteq (\tau_{i1}, \cdots, \tau_{in_i})$ match a rule in Ω_r, **then**
      $(\omega_l, ..., \omega_{l+u})$ is a chunk of words that compose a noun/verb phrase
      $A^d_\nu = (l, ..., l+u)$, $s_{A^d_\nu} = i$, $\boldsymbol{\tau}_{A^d_\nu} = (\tau_l, \tau_{l+1}, ..., \tau_{l+u})$
    end if
    $l \leftarrow l + u + 1$
    $\nu \leftarrow \nu + 1$
  end while
end for
$\boldsymbol{A}^d = \{A^d_1, ..., A^d_n\}$, where $n = \nu - 1$
$\boldsymbol{\omega}_{\boldsymbol{A}^d} = \{\boldsymbol{\omega}_{A^d_1}, ..., \boldsymbol{\omega}_{A^d_n}\}$
$\boldsymbol{s}_{\boldsymbol{A}^d} = \{s_{A^d_1}, ..., s_{A^d_n}\}, \boldsymbol{\tau}_{\boldsymbol{A}^d} = \{\boldsymbol{\tau}_{A^d_1}, ..., \boldsymbol{\tau}_{A^d_n}\}$

Example 7. *Here we give an example of extracting the core events from a single document that has one sentence:* $\boldsymbol{\omega}$ = *"Environment or pollution caused coating defect in the low-voltage bushing showing blue phase - low-voltage bushing and high-voltage bushing requires painting."*

The phrases parsed from $\boldsymbol{\omega}$ *by the function* α *are:* $\boldsymbol{\omega}_{\boldsymbol{A}}$ = {*environment, pollution, coating defect, the low-voltage bushing with blue phase, low-voltage bushing, high-voltage bushing requires painting*}. *There are 5 phrases extracted from this document, where* $A_1 = \{1\}, A_2 = \{3\}, A_3 = \{5, 6\}, A_4 = \{8, 9, 10, 11, 12, 13\}, A_5 = \{14, 15\}, A_6 = \{17, 18, 19, 20\}$.

Step 2. Causality mention extraction. Inspired by Zhao et al. [2017], we next use the predefined linguistic patterns Ω_s to extract cause-effect event pairs. We call such causally related events the **causality mentions** [Zhao et al., 2017]. If

there is more than one pattern in Ω_s satisfied within a sentence, then multiple pairs of causality mentions are extracted for this sentence.

Let $\boldsymbol{\omega}_{\boldsymbol{B}^d} = \{(\boldsymbol{\omega}_{B^d_{1,1}}, \boldsymbol{\omega}_{B^d_{1,2}}), \cdots, (\boldsymbol{\omega}_{B^d_{m,1}}, \boldsymbol{\omega}_{B^d_{m,2}})\}$ denote the set of paired causality mentions extracted from the d^{th} document, where $\boldsymbol{B}^d = \{(B^d_{1,1}, B^d_{1,2}), \cdots, (B^d_{m,1}, B^d_{m,2})\}$ is the set of word indices of the extracted causality mentions. The causality mentions within a parenthesis are ordered as `(cause,effect)`. Each causality mention $\boldsymbol{\omega}_{B^d_{j,k}}$, $j \in \{1, \cdots, m\}$ and $k \in \{1, 2\}$, is a subvector of $\boldsymbol{\omega}_d$ with word indices $B^d_{j,k} = \{b_j, \cdots, b_{jl}\} \subseteq \{d1, \cdots, dN_d\}$. Let β be a map implementing this procedure. We then have that

$$\beta : (\boldsymbol{\omega}_d, \Omega_s) \mapsto (\boldsymbol{\omega}_{\boldsymbol{B}^d}, \boldsymbol{s}_{\boldsymbol{B}^d}, \boldsymbol{\tau}_{\boldsymbol{B}^d}). \tag{3.2.2}$$

As for α, this map also outputs the sentence index for each causality mention, denoted by $\boldsymbol{s}_{\boldsymbol{B}^d} = \{(s_{B^d_{1,1}}, s_{B^d_{1,2}}), \cdots, (s_{B^d_{m,1}}, s_{B^d_{m,2}})\}$, and the corresponding POS tags $\boldsymbol{\tau}_{\boldsymbol{B}^d} = \{(\boldsymbol{\tau}_{B^d_{1,1}}, \boldsymbol{\tau}_{B^d_{1,2}}), \cdots, (\boldsymbol{\tau}_{B^d_{m,1}}, \boldsymbol{\tau}_{B^d_{m,2}})\}$. Algorithm 2 below gives the pseudo-code for β.

Algorithm 2 Causality mention extraction using β

Input: $\boldsymbol{\omega}_d$, Ω_s

Output: $\boldsymbol{\omega}_{\boldsymbol{B}^d}, \boldsymbol{s}_{\boldsymbol{B}^d}, \boldsymbol{\tau}_{\boldsymbol{B}^d}$

Split the document $\boldsymbol{\omega}_d$ into sentences $(\boldsymbol{\omega}_{d1}, ..., \boldsymbol{\omega}_{dJ})$
$\nu = 1$
for $j = 1$ to J **do**
  Tokenize $\boldsymbol{\omega}_{dj}$ to $(\omega_{j1}, \cdots, \omega_{jm_j})$
  Tag the parts of speech by $(\tau_{j1}, \cdots, \tau_{jn_j})$
  for $l = 1$ to R **do**
    if the l^{th} pattern $(\boldsymbol{\omega}_{C,l}, < \boldsymbol{\omega}_{A,l}, \boldsymbol{\omega}_{B,l} >, r_{\boldsymbol{\omega}_{C,l}}(\boldsymbol{\omega}_{A,l}, \boldsymbol{\omega}_{B,l}))$ in Ω_s is matched **then**
      Extract the two events: $\boldsymbol{\omega}_{A,l} = (\omega_{jl_1}, \cdots, \omega_{jl_A})$ with POS tags $\boldsymbol{\tau}_{A,l} = (\tau_{jl_1}, \cdots, \tau_{jl_A})$ and $\boldsymbol{\omega}_{B,l} = (\omega_{jl_1}, \cdots, \omega_{jl_B})$ with POS tags $\boldsymbol{\tau}_{B,l} = (\tau_{jl_1}, \cdots, \tau_{jl_B})$
      if $r_{\boldsymbol{\omega}_{C,l}}(\boldsymbol{\omega}_{A,l}, \boldsymbol{\omega}_{B,l}) = \boldsymbol{\omega}_{A,l} \prec \boldsymbol{\omega}_{B,l}$ **then**
        $\boldsymbol{\omega}_{B^d_{\nu,1}} = \boldsymbol{\omega}_{A,l}$,$\boldsymbol{\omega}_{B^d_{\nu,2}} = \boldsymbol{\omega}_{B,l}$, $\boldsymbol{\tau}_{B^d_{\nu,1}} = \boldsymbol{\tau}_{A,l}$,$\boldsymbol{\tau}_{B^d_{\nu,2}} = \boldsymbol{\tau}_{B,l}$
      else
        $\boldsymbol{\omega}_{B^d_{\nu,1}} = \boldsymbol{\omega}_{B,l}$,$\boldsymbol{\omega}_{B^d_{\nu,2}} = \boldsymbol{\omega}_{A,l}$, $\boldsymbol{\tau}_{B^d_{\nu,1}} = \boldsymbol{\tau}_{B,l}$,$\boldsymbol{\tau}_{B^d_{\nu,2}} = \boldsymbol{\tau}_{A,l}$
      end if
      $s_{B^d_{\nu,1}} = s_{B^d_{\nu,2}} = j$
      $\nu \leftarrow \nu + 1$
    end if
  end for
end for
$\boldsymbol{\omega}_{\boldsymbol{B}^d} = \{(\boldsymbol{\omega}_{B^d_{1,1}}, \boldsymbol{\omega}_{B^d_{1,2}}), \cdots, (\boldsymbol{\omega}_{B^d_{m,1}}, \boldsymbol{\omega}_{B^d_{m,2}})\}$, where $m = \nu - 1$
$\boldsymbol{\tau}_{\boldsymbol{B}^d} = \{(\boldsymbol{\tau}_{B^d_{1,1}}, \boldsymbol{\tau}_{B^d_{1,2}}), \cdots, (\boldsymbol{\tau}_{B^d_{m,1}}, \boldsymbol{\tau}_{B^d_{m,2}})\}$
$\boldsymbol{s}_{\boldsymbol{B}^d} = \{(s_{B^d_{1,1}}, s_{B^d_{1,2}}), \cdots, (s_{B^d_{m,1}}, s_{B^d_{m,2}})\}$

Example 8. *Continue with the document in Example 7,* $\boldsymbol{\omega}$ = *"Environment or pollution caused coating defect in the low-voltage bushing showing blue phase - low-voltage bushing and high-voltage bushing requires painting."*

The causal connectives in this sentence are "caused", the dash, and "requires". The linguistic causal patterns that should be defined are $\{$(*caused,*$< \boldsymbol{\omega}_A, \boldsymbol{\omega}_B >, \boldsymbol{\omega}_A \prec \boldsymbol{\omega}_B$ *if "caused" is a past tense*), $(-, < \boldsymbol{\omega}_A, \boldsymbol{\omega}_B >, \boldsymbol{\omega}_A \prec \boldsymbol{\omega}_B)$, (*requires,*$< \boldsymbol{\omega}_A, \boldsymbol{\omega}_B >, \boldsymbol{\omega}_A \prec \boldsymbol{\omega}_B$)$\}$.

Applying the above algorithm gives paired causality mentions $\boldsymbol{\omega}_{\boldsymbol{B}} = \{$*(Environment or pollution, coating defect in the low-voltage bushing showing blue phase - low-voltage bushing and high-voltage bushing requires painting),(Environment*

or pollution caused coating defect in the low-voltage bushing showing blue phase - low voltage bushing and high-voltage bushing, painting),(Environment or pollution caused coating defect in the low-voltage bushing showing blue phase,low voltage bushing and high-voltage bushing requires painting)}.

Step 3. Causal phrase extraction. In this step, we extract **causal phrases** by combining the results from the previous two steps and refining the causality mentions by the extracted noun/verb phrases. These causal phrases are refined pairs of cause-effect events, denoted by $\boldsymbol{\omega}_{\boldsymbol{C}^d} = \{(\boldsymbol{\omega}_{C^d_{1,1}}, \boldsymbol{\omega}_{C^d_{1,2}}), ..., (\boldsymbol{\omega}_{C^d_{w,1}}, \boldsymbol{\omega}_{C^d_{w,2}})\}$ with word indices $\boldsymbol{C}^d = \{(C^d_{1,1}, C^d_{1,2}), \cdots, (C^d_{w,1}, C^d_{w,2})\}$. Let γ be the map that returns the causal phrases:

$$\gamma : (\boldsymbol{\omega}_{\boldsymbol{A}^d}, \boldsymbol{s}_{\boldsymbol{A}^d}, \boldsymbol{\tau}_{\boldsymbol{A}^d}, \boldsymbol{\omega}_{\boldsymbol{B}^d}, \boldsymbol{s}_{\boldsymbol{B}^d}, \boldsymbol{\tau}_{\boldsymbol{B}^d}) \mapsto (\boldsymbol{\omega}_{\boldsymbol{C}^d}, \boldsymbol{\tau}_{\boldsymbol{C}^d}). \tag{3.2.3}$$

The POS tags of the causal phrases are denoted by $\boldsymbol{\tau}_{\boldsymbol{C}^d}$. By this step, if a causality mention $\boldsymbol{\omega}_{B^d_{j,1}}$ consists of phrases in $\boldsymbol{\omega}_{\boldsymbol{A}^d}$ so that there exist $\{A^d_{j1}, ..., A^d_{j\nu}\} \subseteq B^d_j$, then we collect causal phrases with indices $\{(C^d_{l1,1}, C^d_{l1,1})..., (C^d_{l\nu,1}, C^d_{l\nu,2})\} = \{(A^d_{j1}, B^d_{j,2}), ..., (A^d_{j\nu}, B^d_{j,2})\}$. If there exists no phrase in $\boldsymbol{\omega}_{\boldsymbol{A}^d}$ that can be used to refine a causality mention $\boldsymbol{\omega}_{B^d_{j,1}}$, then the corresponding causal phrase is just $\boldsymbol{\omega}_{B^d_{j,1}}$. The implementation of this map is explained in Algorithm 3.

Algorithm 3 Causal phrase extraction using γ

Input: $\boldsymbol{\omega}_{\boldsymbol{A}^d}, \boldsymbol{s}_{\boldsymbol{A}^d}, \boldsymbol{\tau}_{\boldsymbol{A}^d}, \boldsymbol{\omega}_{\boldsymbol{B}^d}, \boldsymbol{s}_{\boldsymbol{B}^d}, \boldsymbol{\tau}_{\boldsymbol{B}^d}$

Output: $\boldsymbol{\omega}_{\boldsymbol{C}^d}, \boldsymbol{\tau}_{\boldsymbol{C}^d}$

p=1
for sentence $j = 1$ to J **do**
  if $s_{B^d_{i,1}} = s_{B^d_{i,1}} = j$ for some $i \in \{1, ..., m\}$ **then**
    Let $\boldsymbol{A}_{s=j} = \{A^d_{jl}, ..., A^d_{j\nu}\}$ be the sets of word indices of phrases in the j^{th} sentence
    Let $\boldsymbol{B}_{s=j} = \{(B^d_{jk,1}, B^d_{jk,2}), ..., (B^d_{j\iota,1}, B^d_{j\iota,2})\}$ be the sets of word indices of causality mentions in the j^{th} sentence
    for $(B^d_{o,1}, B^d_{o,2}) \in \boldsymbol{B}_{s=j}$ **do**
      $\boldsymbol{A}_{j,o,1} = \emptyset$,$\boldsymbol{A}_{j,o,2} = \emptyset$
      for $A_l \in \boldsymbol{A}_{s=j}$ **do**
        if $A_l \in B^d_{o,1}$ **then**
          $\boldsymbol{A}_{j,o,1} \leftarrow \{\boldsymbol{A}_{j,o,1}, A_l\}$
        end if

if $A_l \in B^d_{o,2}$ **then**
$\boldsymbol{A}_{j,o,2} \leftarrow \{\boldsymbol{A}_{j,o,2}, A_l\}$
end if
end for
if $\boldsymbol{A}_{j,o,1} = \emptyset, \boldsymbol{A}_{j,o,2} = \emptyset$ **then**
$(C^d_{p,1}, C^d_{p,2}) = (B^d_{o,1}, B^d_{o,2})$
$p \leftarrow p + 1$
end if
if $\boldsymbol{A}_{j,o,1} \neq \emptyset, \boldsymbol{A}_{j,o,2} = \emptyset$ **then**
for $A \in \boldsymbol{A}_{j,o,1}$ **do**
$(C^d_{p,1}, C^d_{p,2}) = (A, B^d_{o,2})$
$p \leftarrow p + 1$
end for
end if
if $\boldsymbol{A}_{j,o,1} = \emptyset, \boldsymbol{A}_{j,o,2} \neq \emptyset$ **then**
for $A \in \boldsymbol{A}_{j,o,2}$ **do**
$(C^d_{p,1}, C^d_{p,2}) = (B^d_{o,1}, A)$
$p \leftarrow p + 1$
end for
end if
end for
end if
end for
$\boldsymbol{\omega}_{\boldsymbol{C}^d} = \{(\boldsymbol{\omega}_{C^d_{1,1}}, \boldsymbol{\omega}_{C^d_{1,2}}), ..., (\boldsymbol{\omega}_{C^d_{w,1}}, \boldsymbol{\omega}_{C^d_{w,2}})\}$, $w = p - 1$
$\boldsymbol{\tau}_{\boldsymbol{C}^d} = \{(\boldsymbol{\tau}_{C^d_{1,1}}, \boldsymbol{\tau}_{C^d_{1,2}}), \cdots, (\boldsymbol{\tau}_{C^d_{w,1}}, \boldsymbol{\tau}_{C^d_{w,2}})\}$

Some examples of the causal phrases extracted from the phrases given in Example 7 and the causality mentions given in Example 8 are shown in the table in Figure 3.2.

	cause	effect
0	environment	coating defect
1	environment	the low-voltage bushing showing blue phase
2	environment	low-voltage bushing
3	environment	high-voltage bushing requires painting
4	pollution	coating defect

Figure 3.2: Some causal phrases extracted by the map γ.

Step 4. Abstract causal event extraction. Following the idea pro-

posed by Zhao et al. [2017], we discover more general expressions of the paired causal phrases by using the existing lexical database **WordNet** (WN) [Miller, 1995] and **VerbNet** (VN) [Schuler, 2005] via the established natural language toolkit **NLTK** [Loper and Bird, 2002]. The nouns or verbs in causal phrases obtained from the previous step are picked and replaced by words with a more general meaning. Specifically the nouns are replaced by their hypernyms in WN and the verbs are replaced by their classes in VN respectively. The other words are removed when constructing the abstract causal events.

Assume here that the corpus WN or VN are rich enough for our dataset so that we can always find a replacement for a noun or a verb. Let $\boldsymbol{v}_{\boldsymbol{K}^d} = \{(\boldsymbol{v}_{K^d_{1,1}}, \boldsymbol{v}_{K^d_{1,2}}), \cdots, (\boldsymbol{v}_{K^d_{w,1}}, \boldsymbol{v}_{K^d_{w,2}})\}$ denote the set of paired abstract causal events so that the word indices are $\boldsymbol{K}^d = \{(K^d_{1,1}, K^d_{1,2}), \cdots, (K^d_{w,1}, K^d_{w,2})\}$, where $K^d_{i,j} = \{k_{l1}, \cdots, k_{lb}\} \subseteq C^d_{i,j}$ for any $i \in \{1, \cdots, w\}$ and $j \in \{1, 2\}$. We represent this step by the map

$$\iota : (\boldsymbol{\omega}_{\boldsymbol{C}^d}, \boldsymbol{\tau}_{\boldsymbol{C}^d}) \mapsto \boldsymbol{v}_{\boldsymbol{K}^d} \tag{3.2.4}$$

See Algorithm 4 for the pseudo-code.

Algorithm 4 Abstract causal event extraction using ι

Input: $\boldsymbol{\omega}_{\boldsymbol{C}^d}, \boldsymbol{\tau}_{\boldsymbol{C}^d}$

Output: $\boldsymbol{\omega}_{\boldsymbol{K}^d}$

for $i \in \{1, \cdots, w\}$ **do**
  for $j \in \{1, 2\}$ **do**
    Let $\boldsymbol{\omega}^{*j}$ denote the words in $\boldsymbol{\omega}_{C_{i,j}}$ that are nouns or verbs. Let $K_{i,j}$ denote the word indices of $\boldsymbol{\omega}^{*j}$ and $K_{i,j} \subseteq C_{i,j}$
    for $k \in K_{i,j}$ **do**
      if $\tau_k = \{\text{noun}\}$ **then**
        $v_k = hypernym(\omega_k)$ // this is to find the hypernym of the noun from WN
      else
        $v_k = class(\omega_k)$ // this is to find the class of the verb from VN
      end if
    end for
  end for
end for
$\boldsymbol{v}_{\boldsymbol{K}} = \{(\boldsymbol{v}_{K_{1,1}}, \boldsymbol{v}_{K_{1,2}}), \cdots, (\boldsymbol{v}_{K_{w,1}}, \boldsymbol{v}_{K_{w,2}})\}$

Some examples of the abstract causal events are displayed in Figure 3.3. These are extracted from the causal phrases examples in Figure 3.2 by picking

nouns and verbs and replacing them by their hypernyms and classes respectively.

	cause	effect
0	environment	coating defect
1	environment	bushing, color phase
2	environment	bushing
3	environment	bushing, painting
4	pollution	coating defect

Figure 3.3: Some abstract causal phrases extracted by the map ι.

Step 5. Temporally ordered event extraction. Chambers et al. [2014] developed a methodology CAEVO to extract temporally ordered events. As reviewed in the first chapter, this programme returns pairs of verbs together with labels of the temporal relation between each pair. Note here that we refine its results by only selecting those pairs whose relations are either annotated as "BEFORE" or "AFTER". We also replace these verbs by their corresponding classes in VN as we did in the previous step. We define a map to perform this step that outputs the temporally ordered pairs of events $\boldsymbol{v}_{\boldsymbol{\xi}^d} = \{(v_{\xi^d_{1,1}}, v_{\xi^d_{1,2}}), \cdots, (v_{\xi^d_{r,1}}, v_{\xi^d_{r,2}})\}$ where $\xi^d_{i,j} \in \{d1, \cdots, dN_d\}$ for $i \in \{1, \cdots, r\}$ and $j \in \{1, 2\}$:

$$\mu : \boldsymbol{\omega}_d \mapsto \boldsymbol{v}_{\boldsymbol{\xi}^d}. \tag{3.2.5}$$

Algorithm 5 describes the construction of this map.

Algorithm 5 Temporally ordered event extraction using CAEVO

Input: $\boldsymbol{\omega}_d$

Output: $\boldsymbol{v}_{\boldsymbol{\xi}^d}$

1: Run $CAEVO(\boldsymbol{\omega}_d) = \{(\omega_{f^d_{i,1}}, \omega_{f^d_{i,2}}), \epsilon_i, s_i\}_{i \in \{1,\ldots,q\}}$, where $\omega_{f^d_{i,1}}, \omega_{f^d_{i,2}}$ are a pair of events with word indices $(f^d_{i,1}, f^d_{i,2})$ whose relation is given by ϵ_i. The sentence index of this pair of events is s_i.
2: $r = 0$
3: **for** $i = 1, \cdots, q$ **do**
4: **if** $\epsilon_i =$ BEFORE **then**
5: $r \leftarrow r + 1$
6: $\xi^d_{r,1} = f^d_{i,1}, \xi^d_{r,2} = f^d_{i,2}, (v_{\xi^d_{r,1}}, v_{\xi^d_{r,2}}) = (class(\omega_{\xi^d_{r,1}}), class(\omega_{\xi^d_{r,2}}))$
7: **end if**
8: **if** $\epsilon_i =$ AFTER **then**

9: $r \leftarrow r+1$

10: $\xi^d_{r,1} = f^d_{i,2}, \xi^d_{r,2} = f^d_{i,1}, (v_{\xi^d_{r,1}}, v_{\xi^d_{r,2}}) = (class(\omega_{\xi^d_{r,1}}), class(\omega_{\xi^d_{r,2}}))$

11: **end if**

12: **end for**

13: $\boldsymbol{v}_{\boldsymbol{\xi}^d} = \{(v_{\xi^d_{1,1}}, v_{\xi^d_{1,2}}), \cdots, (v_{\xi^d_{r,1}}, v_{\xi^d_{r,2}})\}$

Step 6. Core events extraction. The final step is to refine the causally ordered events $\boldsymbol{v}_{\boldsymbol{K}^d}$ by the temporally ordered event pairs $\boldsymbol{v}_{\boldsymbol{\xi}^d}$. Let $\boldsymbol{v}_{\boldsymbol{\Sigma}^d} = \{(\boldsymbol{v}_{\Sigma_{1,1}}, \boldsymbol{v}_{\Sigma_{1,2}}), \cdots, (\boldsymbol{v}_{\Sigma_{g,1}}, \boldsymbol{v}_{\Sigma_{g,2}})\}$ denote the output of this step. So the set $\boldsymbol{v}_{\boldsymbol{\Sigma}^d}$ represents the set of pairs of core events, where each pair of core events are causally ordered. Given any $(v_{\xi^d_{i,1}}, v_{\xi^d_{i,2}}) \in \boldsymbol{v}_{\boldsymbol{\xi}^d}$, if this temporal relation does not contradict any extracted causal relation in $\boldsymbol{v}_{\boldsymbol{K}^d}$, then we assert $(v_{\xi^d_{i,1}}, v_{\xi^d_{i,2}})$ are causally ordered and $(v_{\xi^d_{i,1}}, v_{\xi^d_{i,2}}) \in \boldsymbol{v}_{\boldsymbol{\Sigma}^d}$. Here we assume the transitivity of causation is valid so that if A causes B and B causes C, then A is an indirect cause of C. Note that there may exist a pair of events $(v_{\xi^d_{i,1}}, v_{\xi^d_{i,2}}) \in \boldsymbol{v}_{\boldsymbol{\xi}^d}$ whose temporal ordering given by $CAEVO$ contradicts their causal ordering given by Step 4. In this case, $(v_{\xi^d_{i,1}}, v_{\xi^d_{i,2}}) \notin \boldsymbol{v}_{\boldsymbol{\Sigma}^d}$, *i.e.* this temporal relation will not be output as a causal relation. This is because a temporal ordering does not necessarily imply a causal relation between two events. The core events for the d^{th} document are $\boldsymbol{u}_d = \{u_{d,i}\}_{i=\{1,\cdots,n_{u_d}\}}$, where each core event $u_{d,i}$ simply corresponds to a sequence of word tokens $\boldsymbol{v}_{\Sigma_{j,k}} \in \boldsymbol{v}_{\boldsymbol{\Sigma}^d}$ and $\boldsymbol{u}_d$ denotes the set of the unique core events which can be define from $\boldsymbol{v}_{\boldsymbol{\Sigma}^d}$. The set of partial orderings of these core events is denoted by $\pi_{\boldsymbol{u}_d}$. This can be directly read from $\boldsymbol{v}_{\boldsymbol{\Sigma}^d}$. Then $\boldsymbol{v}_{\boldsymbol{\Sigma}^d} = (\boldsymbol{u}_d, \pi_{\boldsymbol{u}_d})$. We let

$$\phi : (\boldsymbol{v}_{\boldsymbol{K}^d}, \boldsymbol{v}_{\boldsymbol{\xi}^d}) \mapsto (\boldsymbol{u}_d, \pi_{\boldsymbol{u}_d}). \tag{3.2.6}$$

Details of how to extract the causally ordered core events from the abstract causal events and the temporally ordered events are clarified in Algorithm 6.

Algorithm 6 Core event extraction using ϕ

Input: $\boldsymbol{v}_{\boldsymbol{K}^d}, \boldsymbol{v}_{\boldsymbol{\xi}^d}$

Output: $\boldsymbol{u}_d, \pi_{\boldsymbol{u}_d}$

1: $\boldsymbol{v}_{\boldsymbol{\Sigma}^d} = \boldsymbol{v}_{\boldsymbol{K}^d}$

2: **for** $i = 1, \cdots, r$ **do**

3: **if** $(\xi_{i,1} \in K_{j,1}) \& (\xi_{i,2} \in K_{j,2}) = FALSE$ for every $j \in \{1, \cdots, w\}$ **then**

4: **if** there does not exist $(\xi_{i,1} \in K_{j,1})$ and $\xi_{i,2} \in K_{l,2}$ so that $K_{j,2} \prec K_{l,1}$ or $K_{l,1} \prec K_{j,2}$ by transitivity for $j, l \in \{1, \cdots, w\}$ **then**

5: $\boldsymbol{v}_{\boldsymbol{\Sigma}^d} \mapsto \{\boldsymbol{v}_{\boldsymbol{\Sigma}^d}, (v_{\xi_{i,1}}, v_{\xi_{i,2}})\}$
6: **end if**
7: **end if**
8: **end for**
9: $\boldsymbol{v}_{\boldsymbol{\Sigma}^d} = \{(\boldsymbol{v}_{\Sigma_{1,1}}, \boldsymbol{v}_{\Sigma_{1,2}}), \cdots, (\boldsymbol{v}_{\Sigma_{g,1}}, \boldsymbol{v}_{\Sigma_{g,2}})\}$
10: $i = 1, u_{d,1} = \boldsymbol{v}_{\Sigma_{1,1}}$
11: **for** $\boldsymbol{v}' \in \boldsymbol{v}_{\boldsymbol{\Sigma}^d}$ **do**
12: $i \leftarrow i + 1$
13: **if** $\boldsymbol{v}' \neq u_{d,j}$, for all $j < i$ **then**
14: $u_{d,i} = \boldsymbol{v}'$
15: **end if**
16: **end for**
17: $\pi_{\boldsymbol{u}_d}$ is the set of paired relations given in $\boldsymbol{v}_{\boldsymbol{\Sigma}^d}$

Example 9. *For the document given in Example 7, the CAEVO software only outputted one pair of events:* `(caused,requires)`*. The relation between these events was "VAGUE". So this pair of events were not added to the abstract causal phrases as a cause-effect pair and were not treated as a pair of core events. The core events* $\boldsymbol{u}$ *were just defined from the abstract causal events.*

Recall that the map σ is defined for extracting causally ordered core events from a document. We can now write this map as a composition of the functions defined above:

$$\sigma(\boldsymbol{\omega}_d, \Omega_{NL}) = \phi(\iota(\gamma(\alpha(\boldsymbol{\omega}_d, \Omega_r), \beta(\boldsymbol{\omega}_d, \Omega_s))), \mu(\boldsymbol{\omega}_d)). \tag{3.2.7}$$

PROPOSITION 3.2.1. *The mapping σ is well-defined but not invertible.*

Proof. (1) To prove σ is well-defined, we need to show that:

- there exists a set of causally ordered core events associated with every document,
- this set of causally ordered core events is unique.

To validate the first condition, we show that for any document $\boldsymbol{\omega}_d \in D$, there exists $\boldsymbol{u} \in \boldsymbol{U}_D$ and $\pi_{\boldsymbol{u}} \in \boldsymbol{\Pi}_D$ so that $(\boldsymbol{\omega}_d, \boldsymbol{u}, \pi_{\boldsymbol{u}}) \in \sigma$.

We first consider the first three steps described above. The causal phrases are the ordered noun phrases or verb phrases, which are extracted from the paired causality mentions. So as long as the phrases $\boldsymbol{\omega}_{\boldsymbol{A}^d}$ can be parsed from the document

and the set of causality mentions $\boldsymbol{\omega}_{\boldsymbol{B}^d}$ are not empty, then the causal phrases $\boldsymbol{\omega}_{\boldsymbol{C}^d}$ are always identifiable. Thus,

$$\boldsymbol{A}^d \neq \emptyset, \boldsymbol{B}^d \neq \emptyset \implies \boldsymbol{C}^d \neq \emptyset. \tag{3.2.8}$$

Every selected document has causal patterns, so $\boldsymbol{B}^d \neq \emptyset$. And each document contains at least one full sentence, so $\boldsymbol{A}^d \neq \emptyset$. Then $\boldsymbol{C}^d \neq \emptyset$ for every document. Therefore, we can always extract causal phrases from each document.

Any extracted causal phrase is either a noun phrase or a verb phrase. So by reading the POS tags, we can always identify the nouns and verbs from the extracted causal phrases. Since we assume that the corpus of WN and VN are sufficiently large for our dataset so that the nouns and verbs appearing in the extracted causal phrases can be identified in these two databases, the abstract causal events $\boldsymbol{\omega}_{\boldsymbol{K}^d}$ can always be extracted using Algorithm 4 . Hence,

$$\boldsymbol{C}^d \neq \emptyset \implies \boldsymbol{K}^d \neq \emptyset.$$

When no temporally ordered events are returned by Algorithm 5, then the index set $\boldsymbol{\xi}^d$ is empty. In this case, by Algorithm 6, the causally ordered core events are the ordered abstract causal events $\boldsymbol{v}_{\boldsymbol{\Sigma}^d} = \boldsymbol{v}_{\boldsymbol{K}^d}$. So $\boldsymbol{\Sigma}^d$ is not empty as long as $\boldsymbol{K}^d$ is not empty. When the index set $\boldsymbol{\xi}^d$ is non-empty, then the ordered events obtained from CAEVO might be added to the abstract causal events $\boldsymbol{v}_{\boldsymbol{K}^d}$ to form $\boldsymbol{v}_{\boldsymbol{\Sigma}^d}$. Following Algorithm 6, the extracted core events $\boldsymbol{u}_d$ and the order $\pi_{\boldsymbol{u}_d}$ can be well defined from $\boldsymbol{v}_{\boldsymbol{\Sigma}^d}$. Since $\boldsymbol{\Pi}_D$ is the space of all possible relations of the core events that can be extracted from D, we have $\pi_{\boldsymbol{u}} \in \boldsymbol{\Pi}_D$. Therefore, $(\boldsymbol{\omega}_d, \boldsymbol{u}, \pi_{\boldsymbol{u}}) \in \sigma$ and the first condition is satisfied.

We next validate the second condition by showing that for any document $\boldsymbol{\omega}_d \in D$, if the mapping σ returns two sets of casually ordered core events $(\boldsymbol{u}', \pi_{\boldsymbol{u}'})$ and $(\boldsymbol{u}'', \pi_{\boldsymbol{u}''})$, then $\boldsymbol{u}' = \boldsymbol{u}''$, $\pi_{\boldsymbol{u}'} = \pi_{\boldsymbol{u}''}$. Let $\boldsymbol{v}_{\boldsymbol{K}'}$ and $\boldsymbol{v}_{\boldsymbol{K}''}$ denote the associated abstract causal events. Note that the following functions are deterministic: α for phrase parsing, β for causality mention extraction, γ for causal phrase extraction, and ι for abstract causal event extraction. In particular, α, β, γ are injective and ι is a multiple-to-one map. So for the same document $\boldsymbol{\omega}_d$, the extracted set of ordered abstract causal events is unique. Thus,

$$\boldsymbol{v}_{\boldsymbol{K}'} = \boldsymbol{v}_{\boldsymbol{K}''}. \tag{3.2.9}$$

The sequence of classifiers in CAEVO give coherent temporal relations for

the input dataset. Therefore, CAEVO outputs the same temporal relations when inputting the same sentence $\boldsymbol{w}_d$. It follows that the temporally ordered events $\boldsymbol{v}_{\boldsymbol{\xi}^d}$ are unique for $\boldsymbol{w}_d$. Now we only need to check whether in the core event extraction step $\phi(\boldsymbol{v}_{\boldsymbol{K}'}, \boldsymbol{v}_{\boldsymbol{\xi}^d}) = \phi(\boldsymbol{v}_{\boldsymbol{K}''}, \boldsymbol{v}_{\boldsymbol{\xi}^d})$ is true. When $\boldsymbol{v}_{\boldsymbol{K}'} = \boldsymbol{v}_{\boldsymbol{K}''}$, if there exists $(v_{\xi_i}, v_{\xi_j}) \in \boldsymbol{v}_{\boldsymbol{\xi}^d}$ satisfying the rules set up in Algorithm 6 so that this pair of events can be treated as a pair of causally ordered core events in addition to $\boldsymbol{v}_{\boldsymbol{K}'}$, then this pair should also added to $\boldsymbol{v}_{\boldsymbol{K}''}$. If $(v_{\xi_i}, v_{\xi_j}) \in \boldsymbol{v}_{\boldsymbol{\xi}^d}$ contradicts an existing causal relationship given by $\boldsymbol{v}_{\boldsymbol{K}'}$, then it also contradicts the same causal relationship given by $\boldsymbol{v}_{\boldsymbol{K}''}$. Therefore, we also have $\phi(\boldsymbol{v}_{\boldsymbol{K}'}, \boldsymbol{v}_{\boldsymbol{\xi}^d}) = \phi(\boldsymbol{v}_{\boldsymbol{K}''}, \boldsymbol{v}_{\boldsymbol{\xi}^d})$ and $\boldsymbol{u}' = \boldsymbol{u}''$, $\pi_{\boldsymbol{u}'} = \pi_{\boldsymbol{u}''}$ as required. Hence σ is well defined.

(2) Now we show σ is non-invertible. Suppose σ is invertible, then the inverse function $\iota^{-1}(\boldsymbol{v}_{\boldsymbol{K}^d})$ must exist. Recall that ι is defined for extracting abstract causal events. For a noun, ι^{-1} maps a hypernym to its hyponyms[1]. This is a one-to-multiple correspondence, which is not a well-defined function. For a noun, it maps a class to its class members, which also makes ι^{-1} a one-to-multiple map. This gives a contradiction. Hence, σ is not invertible. □

3.2.2 The construction of a GN

Having extracted the core events for all the documents, we aim to match these events and the associated partial order to a corresponding CEG so as not to lose any extracted causal relationships. It is not easy to implement when the size of the output of the NLP algorithm is large. We are aware of no automised way of performing this matching extant in the literature. What we propose here is to first refine the causal relationships that are extracted based on linguistic patterns before embedding these causal dependencies into a CEG. So we add an extra but necessary preprocessing step that groups the extracted core events and registers an implicit partial order of these core events. This is done by constructing a Global Net (GN). This step helps us find more concise and general shallow causal patterns through the GN than can be read directly from $\{(\boldsymbol{u}_d, \pi_{\boldsymbol{u}_d})\}_{d \in N_D}$

In Section 3.1, we defined the GN to be a DAG to embed the shallow causal dependencies without restricting the choices of this DAG. One can build an event tree or derive a CEG to embed the shallow causal dependency at this level as a GN where the core events can be simply labelled on the edges or vertices. Then we would have two levels of CEGs, where the upper level tree is more dense. However when there are a large number of core events, it is computational expensive to build such a tree. This is especially so when some data is missing, the problem is scaled

[1]Hyponyms have more specific meaning compared to its hypernym.

up and the complexity of computation cannot be ignored. An alternative choice is to build the GN with BN semantics. We have discussed the weakness of BNs in the previous two chapters. However, here we are not using the BN alone so these problems can be simply addressed by the CEG introduced at the deeper layer.

Learning a BN is transparent and easy to implement with the current available software [Scutari, 2009; Scutari and Ness, 2012; Scutari, 2014]. So in this thesis we focus on using the BN semantics as the GN. However, the best scoring BN may not be causal, which would then mean that the definition of the GN is not satisfied. Therefore, we need to elicit genuine causal relations from the BN to construct a proper GN.

The vertices of the GN V^* are associated with a set of variables $\boldsymbol{L} = \{L_1, \cdots, L_{nL}\}$, which are called the core event variables. We have defined the GN to be a causal network. It therefore follows that there is an edge $e_{v_i,v_j} \in E^*$ pointing from $v_i \in V^*$ to $v_j \in V^*$ whenever the associated variable of v_i, denoted by L_i, is a genuine cause [Pearl, 2009] of the associated variable of v_j, denoted by L_j. Accordingly, the two essential steps to construct such a GN are

1. the construction of core event variables,

2. the extraction of genuine cause-effect relationships.

We next explain how to attain these two tasks in detail.

Firstly, the core event variables are constructed by clustering the extracted core events $\boldsymbol{u}$ so that each core event variable corresponds to a set of core events. In particular, we group the core events excluding the maintenance events. This is because maintenance corresponds to an external intervention whose effects are identifiable on the CEG as shown in the previous chapter. If we represent the maintenance variable on the GN, then it cannot be matched to any position or edge of the CEG lying at the deeper layer. In other words, the GN is constructed with respect to the idle CEG alone. We therefore construct the core event variable only from the extracted core events representing various causes, symptoms or failures.

Expert judgements or assumptions are required to determine how to construct the core event variables. For example, we can group the core events that which have similar meanings and treat each of them as a state of a categorical variable L_i. We can also group the core events with the same attribute, such as the core events describing the symptom of a specific defect. We can also create a binary variable for a core event to indicate whether it is present or absent. For example, we create a binary variable to indicate whether oil leak has been observed.

Let $\Xi : u \mapsto L$ denote the correspondence between a core event and a core event variable. To ensure that core event variables are identifiable for each document, we restrict the construction of core event variables by the following assumptions.

ASSUMPTION 3.2.2. *When constructing a set of core event variables from the core events $\mathbb{U}_D$ extracted from a dataset D, we assert,*

1. *each core event $u \in \mathbb{U}_D$ can only correspond to a state of a single core event variable;*

2. *any core event variable $L_i \in \boldsymbol{L}$ taking values in $\mathbb{L}_i = \{l_{i1}, \cdots, l_{in_i}\}$ can correspond to more than one core event; the state of it $l_{ij} \in \mathbb{L}_i$ can correspond to multiple core events;*

3. *the value of any core event variable $L_i \in \boldsymbol{L}$ is observable in at least one document of the whole dataset.*

4. *any two core events $u_{d_{k1}}, u_{d_{k2}}$ of a document cannot be associated with two different values of the same core event variable.*

Following these assumptions, $\Xi : u \mapsto L$ is a multiple-to-one correspondence. It is well-defined, surjective but not injective. We therefore can always find a unique core event variable for each core event by Ξ.

Now we turn to the second task to discover the genuine causal relationships between the core event variables and embed them within the topology of the GN. Pearl [2009] implicitly brought the idea of the genuine cause that there might be some relationships that have a clear causal direction if indeed this exists. This step is carried out by applying a well-developed package called **bnlearn** [Scutari, 2009] to fit a best scoring BN with respect to $\boldsymbol{L}$. Let $G^\dagger = (V^\dagger, E^\dagger)$ denote the best scoring topology. This DAG has the same vertex set as the vertex set of the GN, $V^\dagger = V^*$, so that each vertex of it corresponds to a core event variable. The edges in the edge set $E^\dagger$ are not necessarily causal since the structural learning algorithm of the BNs alone cannot ensure that every edge represents a putative causal dependency. We add two further steps to elicit the desired GN G^* from the selected BN.

We first create a list of directed edges that must be present in the GN and a list of directed edges that should never appear in the GN. The bnlearn package provides a facility to add these constraints when learning the best structure. To form the first list, we summarise the likely causal relationships between the constructed variables from the causally ordered core events and collect the frequent ones. Specifically, we set a number $n^\star$ so that if a cause-effect pair appears more than $n^\star$ times

in the extracted pairs of core events from all documents then we add the edge connecting the corresponding variables to the list. The second list includes only the edges violating the cause-before-effect temporal relations. Expert judgement can be helpful when creating this list. Given the core event variables, if we provide a list of potential cause-effect pairs to the domain experts and ask them to annotate those whose causal relations are invalid in the selected system, then we can use their annotated pairs to form the second list. When inputting this two lists into the bnlearn, the output DAG may have a different structure from $G^{\dagger}$ which is learned without these constraints. Let $G^{\ddagger} = (V^{\ddagger}, E^{\ddagger})$ denote the revised DAG, where the vertex set satisfies $V^{\ddagger} = V^{*} = V^{\dagger}$. This step enables us to preserve the putative cause-effect relations extracted from the sequence of text processing algorithms based on naive linguistic causal patterns which are valid for the given domain conditioning on the causal hypotheses being plausible or meaningful to domain expert judgements.

We have mentioned above that not every edge in $E^{\dagger}$ is causal, so we wish to remove the edges which may not represent causal relationships. In order to do this, we first find the essential graph [Smith, 2010] of $G^{\ddagger}$ and then remove the undirected eges in the essential graph. An essential graph is a mixed graph with undirected edges. To derive an essential graph, the pattern [Smith, 2010] should be constructed from $G^{\ddagger}$ by keeping the directionality of the edges that connect the unmarried parents to the common child. Specifically, by the definition [Smith, 2010], suppose $e \in E^{\ddagger}$ points from $v_i \in V^{\ddagger}$ to $v_j \in V^{\ddagger}$, the directionality of e is removed if and only if there does not exist $v_k \in pa(v_j)$ satisfying $e_{v_i,v_k} \notin E^{\ddagger}$. The essential graph is obtained from the pattern by keeping the configuration of the unmarried parents unchanged. The undirected edges in the essential graph can be given multiple equally well supported interpretations of causal directionality. This is because adding either direction to the undirected edge does not change the equivalence class of the DAG. Since we have defined every edge of the GN to be causal, the final GN topology G^{*} is obtained from the essential graph by removing the undirected edges.

For example, Figure 3.4 is the structure extracted from bnlearn with the constraints on causal edges. Then the pattern of it can be derived using the above-mentioned method, see Figure 3.5. Figure 3.6 plots the essential graph of $G^{\ddagger}$. By removing the undirected edges, we can draw G^{*} in Figure 3.7.

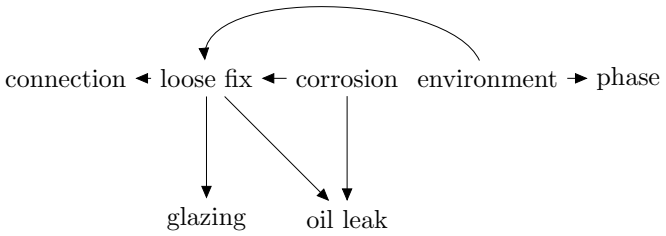


Figure 3.4: The extracted DAG $G^{\ddagger}$.

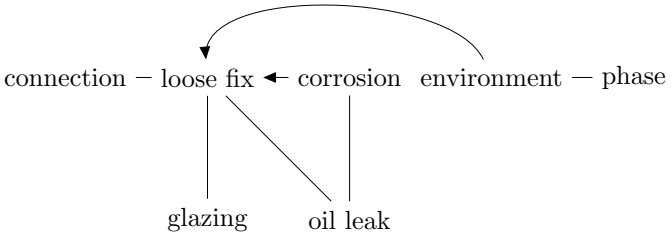


Figure 3.5: The pattern derived from the extracted $G^{\dagger}$ in Figure 3.4

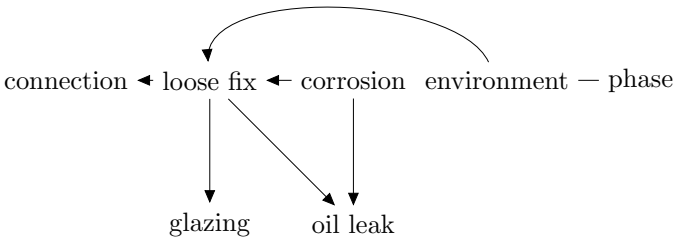


Figure 3.6: The essential graph derived from Figure 3.5

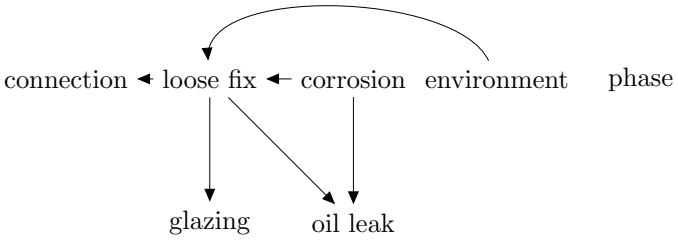


Figure 3.7: The GN derived from Figure 3.6

To summarise, for each document d with extracted core events $\boldsymbol{u}_d$, we can find a set of core event variables corresponding to this document by Ξ. This is guaranteed by the assumptions we made for the construction of core event variables.

In the beginning of this chapter, we let $\boldsymbol{L}_d = \{L_{d1}, \cdots, L_{dm}\}$ denote the set of core event variables whose values $\boldsymbol{l}_d = \{l_{d1}, \cdots, l_{dm}\}$ are determined by the core events $\boldsymbol{u}_d$. The relations between variables in $\boldsymbol{L}_d$ can be directly read from the GN. Then the subgraph $G_d = (V_d, E_d)$ associated with $\boldsymbol{u}_d$ can be simply identified. The vertex set V_d is a subset of V^* consisting of vertices correspond to $\boldsymbol{L}_d$. The edge set $E_d \subseteq E^*$ consists of the edges in the GN connecting vertices in V_d. Since Assumption 3.2.2 ensures a unique set $\boldsymbol{l}_d$ associated with the core events $\boldsymbol{u}_d$, the mapping $\psi : (\boldsymbol{u}_d, \pi_d) \mapsto \boldsymbol{l}_d$ can be well-defined. This function is non-injective since $\Xi : u \mapsto L$ is non-injective.

An immediate implication of this property and Proposition 3.2.1 is that the mapping $\Gamma : (\boldsymbol{w}_d, \Omega_{NL}) \mapsto \boldsymbol{l}_d$ which returns a unique set $\boldsymbol{l}_d$ for a document is well-defined but not injective. In this case, if we can further show that we can find a unique path on the CEG associated with $\boldsymbol{l}_d$ by making proper assumptions, then with the GN-CEG model we can map each document to a unique root-to-sink path on the CEG for a causal analysis. In the following section, we will show how this is possible.

3.3 Causality embedding

In this section, we establish the theory that validates the mapping from the GN to the CEG. This shall explain how the shallow causal dependency can be translated to the deeper causal dependency. Note that within our model, every edge in the GN has a causal interpretation and every edge in the CEG represents a causal dependency. This, therefore, forms a two-level causality structure. And importantly the causal relations are nested. In the domain of natural language understanding, causality is often expressed using a nested structure [Chen et al., 2020]. In our hierarchical model, the nested structure can be understood in a similar way: a cause-effect pair on the GN could be associated with a d-event labelled on some edges on the CEG, which is the cause of another d-event labelled on some other edges on the CEG.

In the introduction chapter, we have reviewed the definition of the **recursive Bayes nets** (RBNs) [Williamson and Gabbay, 2005; Casini et al., 2011] and the essential terminologies within this framework. In light of this framework, we define new concepts and make appropriate assumptions within our hierarchical causal model in an analogous fashion. This will make the transformation of causal relationships between the GN and the CEG possible. In the end of this section, the method of learning a root-to-sink path $\lambda \in \Lambda_{\mathcal{C}}$ for $\boldsymbol{l}_d$, *i.e.* the map χ, should have been clarified.

3.3.1 Linking the GN to the CEG

Each set of values $\boldsymbol{l}_d$ are associated with a set of vertices V_d on the GN. If we treat the values of the core event variables as the observations, then given G_d we have a sequence or multiple sequences of ordered observations. In contrast to these observations, the root-to-sink path on the CEG associated with $\boldsymbol{l}_d$ is hidden, we call it the **latent path** of document d. Then the positions along the path can be treated as **latent (hidden) states** of the observed core event variables. These latent states cannot be directly observed but can be inferred. This motivates us to find a generic approach to group the core event variables given their causal order so that each group associates with a latent state on the CEG.

We can treat the GN as the **observation layer** and the CEG as the **latent layer**. Although our model is similar to the Hidden semi-Markov model (HSMM) [Yu, 2010] in a way that the semi-Markov process can be embedded within the CEG and we trace the process by finding the latent states, our model focuses more on the causal interpretation on both the observation layer and the latent layer. But following the ideas of the HSMM, we can formulate the relationship between every latent state and its corresponding observations in our hierarchical model. In order to do this, we first introduce some new terminologies for the latent layer and the observation layer separately.

The latent states on the deeper level. For each $w \in V_{\mathcal{C}}$, we can define an **incident variable** $I^{\lambda}(w)$ to indicate whether a path $\lambda \in \Lambda_{\mathcal{C}}$ passes through w [Wilkerson, 2020]. Note that an incident variable can be defined on the sink nodes of the CEG. Let

$$I^{\lambda}(w) = \begin{cases} 1, & \text{if } w \in \lambda, \\ 0, & \text{if } w \notin \lambda. \end{cases} \tag{3.3.1}$$

The value of the variable depends on the path λ.

We can always define a measurable variable over each floret on a CEG [Wilkerson, 2020]. This is called a **floret variable**. For each position $w \in W$, we construct the floret variable $Y^{\lambda}(w)$ for the underlying floret $\mathcal{F}(w)$ given a root-to-sink path $\lambda \in \Lambda_{\mathcal{C}}$. Let $e_{w,w'} \in E(w)$ be an emanating edge of w so that $w' \in ch(w)$. Then given a path $\lambda \in \Lambda_{\mathcal{C}}$,

$$Y^{\lambda}(w) = \begin{cases} y_{w,w'}, & \text{if } e_{w,w'} \in \lambda, \\ 0, & \text{if } e_{w,w'} \notin \lambda. \end{cases} \tag{3.3.2}$$

This means each edge corresponds to a state of the floret variable. The floret variable

takes value 0 when the unit traverses a path not passing through w, *i.e.* $I^\lambda(w) = 0$. If $Y^\lambda(w)$ is instantiated, *i.e.* $I^\lambda(w) = 1$, then its value depends on which child of w which is traversed through by λ.

Given the values of floret variables or incident variables we can identify which edges or vertices are traversed by the latent path.

The observations corresponding to a latent state. On the GN layer, *i.e.* the observation layer, suppose that we can find a latent path on the CEG for the observed core events variables lying in the subgraph G_d. Then each latent state can be associated with a subset of these variables causally ordered in G_d. Following this idea, each floret variable $Y^\lambda(w_i)$, $w_i \in W$, is the latent variable of a set of core event variables. We call this set of core event variables the **community** of $Y^\lambda(w_i)$ and denote it by $\boldsymbol{L}_i$. This is a subset of the core event variables, $\boldsymbol{L}_i \subseteq \boldsymbol{L}$. We say a set of variables are instantiated when every variable in this set takes some value in its state space which is known from observations. When $I^\lambda(w_i) = 0$, *i.e.* w_i is not passed through by the latent path, then $Y^\lambda(w_i)$ is not instantiated [Wilkerson, 2020], and so the community $\boldsymbol{L}_i$ is not instantiated. This does not mean no core event variable in $\boldsymbol{L}_i$ is observed. For example, if a core event variable L' lies in two communities $\boldsymbol{L}_i$ and $\boldsymbol{L}_j$ for $i \neq j$ and w_j is traversed by the latent path, then the value of L' is still observed. In this case, instead of $\boldsymbol{L}_i$, the community $\boldsymbol{L}_j$ is instantiated.

The causal relationships between every pair of core event variables in $\boldsymbol{L}_i$ have already been determined by the GN. Let $G_i = (V_i, E_i)$ be a subgraph of the GN which is associated with $\boldsymbol{L}_i$. We call it the **area** of the community $\boldsymbol{L}_i$. The vertex set V_i consists of vertices corresponding to the core event variables $\boldsymbol{L}_i$. An edge $e_{w,w'}$ is included in the edge set E_i if and only if $w, w' \in V_i$ and $e_{w,w'} \in E^*$.

Since we have a fixed and known topology of the GN from the preprocessing step, the value of the latent floret variable only determines the members in the community but not the relations between the members. This means the area of a community is only determined by the variables in the community given the GN. This is different from the idea of the network variable in the RBN [Williamson and Gabbay, 2005], though the floret variable plays a role similar to the network variable.

If $Y^\lambda(w_i) = y_{w_i,w_k} \neq 0$, then there is a **sub-community** corresponding to this value, denoted by $\boldsymbol{L}_{i,k} \subseteq \boldsymbol{L}_i$. The **sub-area** of this sub-community is a subgraph of the area G_i, denoted by $G_{i,k} = (V_{i,k}, E_{i,k})$, where the vertex set $V_{i,k} \subseteq V_i$ corresponds to $\boldsymbol{L}_{i,k}$ and the edge set satisfies $E_{i,k} \subseteq E_i$.

Link between a floret variable and its community. Now a question arises: how to specify the community of a floret variable? Return to the definition

of a floret variable. Each non-zero value of a floret variable can be associated with an edge that is traversed by the latent path. Note that as defined in the previous chapter, each edge is labelled by a d-event $x(e_{w_i,w_k})$ for $e_{w_i,w_k} \in E_{\mathcal{C}}$. Therefore, once the d-event $x(e_{w_i,w_k})$ is observed, the corresponding path on the CEG must pass along an edge labelled by this d-event. This means the floret variable takes value $Y^{\lambda}(w_i) = y_{w_i,w_k}$. It follows that the d-event $x(e_{w_i,w_k})$ will be associated with the corresponding sub-community $\boldsymbol{L}_{i,k}$. Assume that the association between a d-event and the core event variables is known from domain knowledge or learning through algorithms, which will be discussed later in Section 5.2, we then know the sub-community of this d-event. In this case we assume that it is legitimate to conjecture that $x(e_{w_i,w_k})$ happens if the core event variables in $\boldsymbol{L}_{i,k}$ have been observed to be $\boldsymbol{l}_{i,k}$ with respect to their causal ordering.

However, not every floret variable has an associated community. Given a path $\lambda \in \Lambda_{\mathcal{C}}$, for any position traversed by this path $w \in W_{\lambda}$, we have defined the floret variable indexed by this path, denoted by $Y^{\lambda}(w)$, in equation (3.3.2). Note that the last floret variable along any path for a system designed for causal analysis in the domain of system reliability corresponds to a failure indicator. If our data is extracted from failure reports, then the value of this floret variable is known. This is because all the observations recorded in the texts are conditional on a failure has occurred. We denote the set of positions whose floret variables are associated with failure indicators by $W^{\star}$ so that $ch(w^{\star}) = \{w_{\infty}^{f}, w_{\infty}^{n}\}$ for $w^{\star} \in W^{\star}$. Then for any path, $Y^{\lambda}(w^{\star})$ is not associated with any core event variable, *i.e.* it does not have an underlying community. For other positions $w \in W \setminus W^{\star}$, $Y^{\lambda}(w)$ has a corresponding community.

For all paths, every core event variable is in at least one community of a floret variable. But for a given path, not every core event variable is assigned to a community associated with the floret variables instantiated by this path. So to be clear, under our setting, not every subset of the core event variables can constitute a community and it is not necessary to have a one-to-one correspondence between a floret variable and a core event variable given a root-to-sink path.

Then is it possible to identify communities for floret variables? There are two possible scenarios here.

1. When the number of core event variables $\boldsymbol{L}$ is not huge, the CEG does not have a complicated topology and we have sufficient domain knowledge so that the association between a floret variable and its community is known, then we do not need to infer the latent variables for the observations. Note that this first setting necessarily requires human involvement.

2. Alternatively, in any situation where we would like to minimise the amount of human interaction with the inference. Further, in a more complicated scenario, we may not know which set of core event variables are associated with a floret variable. In this second setting, we need to design algorithms to automatically learn the relations between floret variables and the core event variables. If we are able to do this then we may treat this process as a semi-supervised learning problem. In this second case we observe the values of the core event variables for each document and through the algorithm we learn the latent states. This may need us to specify the potential d-events for each core event variable and split each $\boldsymbol{l}_d$ into a potential set of sub-communities. In Section 5.2, we will provide more details about the general process for learning the associations between the observations and the latent states.

Apart from the florets associated with failure indicator, there could also exist a set of florets which are associated with classifications, for example, classifying root causes to be endogenous or exogenous, or classifying the categories of components. We assume that when modelling transition times on the CEG, the edges associated with such classification are not assigned with holding time. We have given a brief discussion of this in the previous chapter. It is also non-trivial to chronologically order the variables associated with classifications with other variables such as root causes or symptoms. Let $W^\dagger$ denote the set of positions so that for any $w^\dagger \in W^\dagger$ and any path λ, $Y^\lambda(w^\dagger)$ is a variable for classification. Let $W^\diamond = W \setminus (W^\dagger \cup W^\star)$. When formalising the ordering of communities, we only consider the communities associated with floret variables defined over $w \in W^\diamond$.

We next formalise the ordering of communities and sub-communities. When a unit traverses a root-to-sink path on the CEG, a sequence of floret variables are instantiated and so the corresponding communities are instantiated. We first make the following assumption.

ASSUMPTION 3.3.1. *The instantiated communities for $Y^\lambda(w)$, where $w \in W^\diamond$, are disjoint with each other and they are ordered consistently with their floret variables. Given the values of the instantiated floret variables, the corresponding sub-communities are ordered consistently with their floret variables.*

This assumption gives us an intuition of how the causal order modelled on the deeper layer CEG is translated to the causal order of the core event variables modelled on the upper layer GN. Specifically, given a root-to-sink path λ, for any two positions traversed by this path, denoted by $w_i, w_j \in W_\lambda$, if the corresponding floret variables $Y^\lambda(w_i)$ and $Y^\lambda(w_j)$ have the associated communities, *i.e.* $w_i, w_j \in W^\diamond$,

then the partial order between communities is inherited from the corresponding partial order between $Y^{\lambda}(w_i)$ and $Y^{\lambda}(w_j)$. For example, if $Y^{\lambda}(w_i) \prec Y^{\lambda}(w_j)$, then we say the community of $Y^{\lambda}(w_i)$, denoted by $\boldsymbol{L}_i$, precedes the community of $Y^{\lambda}(w_j)$, denoted by $\boldsymbol{L}_j$. This relation is indexed by the path λ, so we represent the order by $\boldsymbol{L}_i \prec^{\lambda} \boldsymbol{L}_j$. If $Y^{\lambda}(w_i) = y_{w_i,w_k}$ and $Y^{\lambda}(w_j) = y_{w_j,w_l}$, then we can also order the corresponding sub-communities accordingly. Let $\boldsymbol{L}_{i,k}$ and $\boldsymbol{L}_{j,l}$ denote the sub-communities corresponding to $Y^{\lambda}(w_i)$ and $Y^{\lambda}(w_j)$ respectively, and we represent such an order between sub-communities by $\boldsymbol{L}_{i,k} \prec^{\lambda} \boldsymbol{L}_{j,l}$. Note that the ordering of core event variables within each community or sub-community is fixed by the topology of the GN. To ensure a consistent ordering of the communities and the core event variables, the following assumption must be satisfied in our setting.

ASSUMPTION 3.3.2. *If the core event variables and the communities satisfy $L' \in \boldsymbol{L}_i$, $L'' \in \boldsymbol{L}_j$ and $\boldsymbol{L}_i \prec^{\lambda} \boldsymbol{L}_j$, then $L'' \not\prec L'$, i.e. L'' cannot precede L' on the GN.*

Example 10. *Figure 3.8 gives a hypothesised GN corresponding to the CEG plotted in Figure 2.7 which is assumed to be causal. It is assumed to be constructed from the core events extracted from the failure reports of a bushing system. So the GN is constructed conditional on the event that a failure has been observed.*

The state spaces of the core event variables are: $\mathbb{L}_1 = \{$failed gasket, aging gasket$\}$, $\mathbb{L}_2 = \{$seal crack, axial crack$\}$, $\mathbb{L}_3 = \{$crack, no crack$\}$, $\mathbb{L}_4 = \{$oil level low, leak, normal oil level, loss of oil, transformer oil and bushing oil$\}$, $\mathbb{L}_5 = \{$loose connection, connection ok$\}$, $\mathbb{L}_6 = \{$ oxidant contact, contact resistance$\}$, $\mathbb{L}_7 = \{$thermal, electrical$\}$, $\mathbb{L}_8 = \{$lightening, weather$\}$, $\mathbb{L}_9 = \{$temperature, nitrogen blanketed$\}$.

Based on the d-events and the core events which are grouped as the values for the core event variables, we can define a reasonable assignment of communities to the latent states. The community of the floret variable $Y^{\lambda}(w_0)$ is $\boldsymbol{L}_0 = \{L_1, L_2, L_3, L_5, L_6, L_8, L_9, L_{10}\}$, which consists of all possible root causes. The floret variable $Y^{\lambda}(w_1)$ is associated with endogenous root causes, so its community is defined to be $\boldsymbol{L}_1 = \{L_1, L_2, L_3, L_5, L_6\}$. The floret variable $Y^{\lambda}(w_2)$ is associated with exogenous root causes, so its community is $\boldsymbol{L}_2 = \{L_8, L_9, L_{10}\}$. The floret variable $Y^{\lambda}(w_3)$ is associated with oil leak, so its community is $\boldsymbol{L}_3 = \{L_4\}$. The community of $Y^{\lambda}(w_4)$ is $\boldsymbol{L}_4 = \{L_4\}$. The community of $Y^{\lambda}(w_5)$ is $\boldsymbol{L}_5 = \{L_7\}$.

When $Y^{\lambda}(w_1) = y_{w_1,w_5}$, i.e. when the failure is caused by an endogenous fault which is not related to gasket, porcelain or insulation, then we can determine the sub-community associated with this value to be $\boldsymbol{L}_{1,5} = \{L_5, L_6\}$. We can find the sub-area of $\boldsymbol{L}_{1,5}$ from the GN. We denote this sub-area by $G_{1,5} = (V_{1,5}, E_{1,5})$.

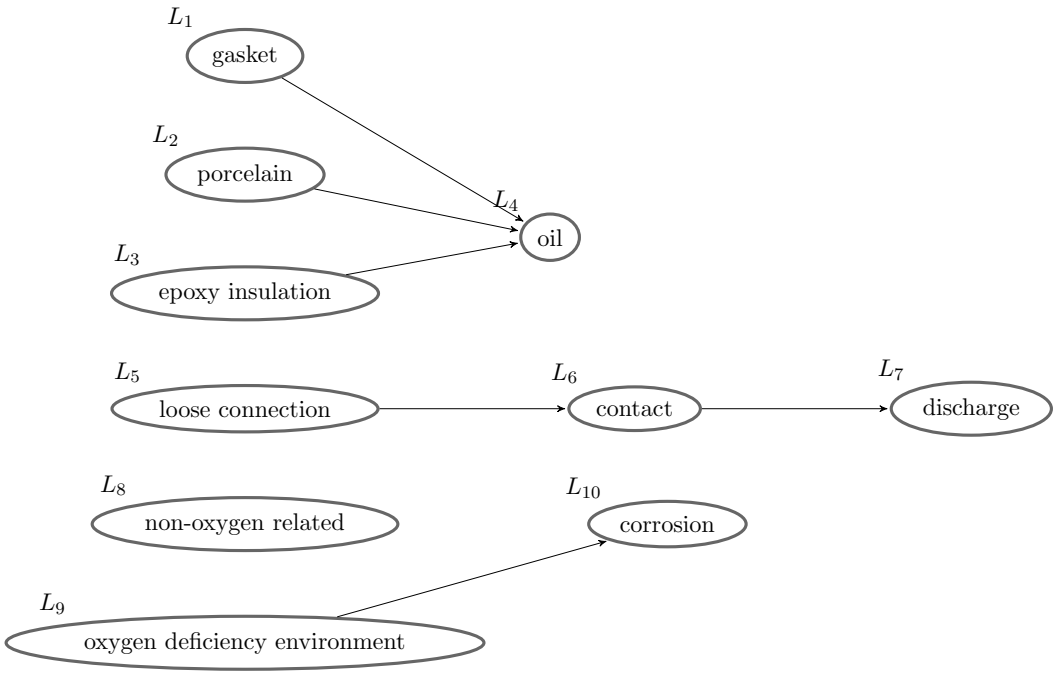


Figure 3.8: The hypothesised GN for Example 10.

As defined above, the vertices of the sub-area, $V_{1,5}$, correspond to $\boldsymbol{L}_{1,5}$. So $V_{1,5}$ are associated with L_5 and L_6. And the edge set $E_{1,5}$ satisfies that $e_{v,v'} \in E_{1,5}$ if and only if $v, v' \in V_{1,5}$ and $e_{v,v'}$ lies in the GN. There is one edge connecting from L_5 to L_6, so $E_{1,5} = e_{L_5,L_6}$. Similarly, we can find the area of $Y^{\lambda}(w_5)$, denoted by $G_5 = (V_5, E_5)$ where V_5 is associated with L_7 because $\boldsymbol{L}_5 = \{L_7\}$ and E_5 is empty.

The positions $W^{\dagger} = \{w_6, w_7, w_8, w_9\}$ do not have associated communities because the associated floret variables correspond to failure indicators, see Figure 2.7. However, the values of their corresponding floret variables are known: $Y^{\lambda}(w) = y_{w,w^f_{\infty}}$ for $w \in W^{\dagger}$, where λ is a failure path.

3.3.2 Conditional independence assumptions

Here we borrow the terminologies in the RBN to demonstrate the conditional independence relationships within the GN-CEG model. Recall that a variable V_j is a direct inferior of another variable V_i if the value of V_i indexes a set of BNs and V_j lies in any of these BNs [Williamson and Gabbay, 2005; Casini et al., 2011]. And in this case V_i is called the direct superior of V_j. For each floret variable $Y^{\lambda}(w_i)$, the core event variables in its community $\boldsymbol{L}_i$ are treated as the direct inferiors of it. For each $L_j \in \boldsymbol{L}_i$, $Y^{\lambda}(w_i)$ is treated as the direct superior of L_j. The definitions given in the previous section allow one core event variable to lie in more than one

communities. If this is the case, then there is more than one floret variable which can be treated as its direct superiors. This means there is more than one position on the CEG corresponding to L_j as its latent state. Denote this set of positions by W_j and

$$Y^{\lambda}(W_j) = \{Y^{\lambda}(w_{j_l})\}_{w_{j_l} \in W_j} = Dsup(L_j). \tag{3.3.3}$$

A position satisfies $w_{j_l} \in W_j$ if the community of the floret variable $Y^{\lambda}(w_{j_l})$ includes L_j, $L_j \in \boldsymbol{L}_{j_l}$.

Here we not only need to specify the conditional independence relations on the CEG, but also the conditional independence relations between the core event variables defined on the GN and the floret or incident variables defined on the CEG. Smith and Anderson [2008] showed that different conditional independence properties can be read from the CEG. Furthermore, Wilkerson [2020] proved analogous d-separation theorem on the CEG. Let $nd(Y^{\lambda}(w))$ denote the floret variables defined over the florets that do not lie downstream of the position $w \in W$. Then this set of variables can be treated as the non-descendants of $Y^{\lambda}(w)$ on the CEG. Note that the non-descendants and the parents of a variable in our model only refer to the non-descendants and the parents lying in the same layer of this variable.

The constitution of the observation layer varies depending on which path is traversed by the unit. Consider an assignment of values to the floret variables and the incident variables defined over the CEG,

$$\mathcal{H} = \{y(w)\}_{w \in W} \cup \{i(w)\}_{w \in V_{\mathcal{C}}}. \tag{3.3.4}$$

We call $\mathcal{H}$ a **consistent** assignment with respect to the GN-CEG model if it corresponds to a unit traversing a single root-to-sink path on the CEG. Given a consistent assignment $\mathcal{H}$, an ordered sequence of the corresponding communities are instantiated. Let $\boldsymbol{L}^{\mathcal{H}}$ denote the instantiated communities, then $\boldsymbol{L}^{\mathcal{H}} = \{\boldsymbol{L}_i\}_{w_i \in \lambda^{\mathcal{H}}}$, where $\lambda^{\mathcal{H}}$ denotes the root-to-sink path traversed by the unit given $\mathcal{H}$. Then the instantiated areas of the instantiated communities can be discovered and well-defined on the GN. Let $G^{\mathcal{H}} = (V^{\mathcal{H}}, E^{\mathcal{H}})$ denote the instantiated areas. The vertex set $V^{\mathcal{H}}$ is associated with $\boldsymbol{L}^{\mathcal{H}}$. The edge $e_{w,w'}$ is included $E^{\mathcal{H}}$ if $e_{w,w'} \in E^*$ points from $w \in V^{\mathcal{H}}$ to $w' \in V^{\mathcal{H}}$. Note that the edge set $E^{\mathcal{H}}$ includes the edges lying in the instantiated areas and the edges connecting variables which lie in different instantiated communities if they exist in the GN. Therefore, $\bigcup_{w_i \in \lambda^{\mathcal{H}}} E_i \subseteq E^{\mathcal{H}}$.

PROPOSITION 3.3.3. *Given a consistent assignment $\mathcal{H}$ to the floret variables and the incident variables, the areas of the instantiated communities on the GN are causally consistent.*

Proof. Consider a path $\lambda \in \Lambda_{\mathcal{C}}$, if there exist positions $w_i, w_j, w_k \in W_\lambda$ and the corresponding edges satisfy $e_{w_i,w_j}, e_{w_j,w_k} \in E_{\mathcal{C}}$, then the floret variables $Y^\lambda(w_i), Y^\lambda(w_j), Y^\lambda(w_k)$ are instantiated and the corresponding communities are instantiated. In particular, since $Y^\lambda(w_i) = y_{w_i,w_j}$, $Y^\lambda(w_j) = y_{w_j,w_k}$, the sub-communities $\boldsymbol{L}_{i,j}$ and $\boldsymbol{L}_{j,k}$ are instantiated with sub-areas $G_{i,j} = (V_{i,j}, E_{i,j})$ and $G_{j,k} = (V_{j,k}, E_{j,k})$ respectively.

We first construct a causal graph denoted by $G_{i,j,k} = (V_{i,j,k}, E_{i,j,k})$. The vertices are constructed with respect to $\boldsymbol{L}_{i,j}$ and $\boldsymbol{L}_{j,k}$. If $e \in E_{i,j}$ or $e \in E_{j,k}$, then $e \in E_{i,j,k}$. Following Williamson and Gabbay [2005], we need to show $G_{i,j,k}$ is a causal supergraph of $G_{i,j}$ and $G_{j,k}$.

Let $G_{\lfloor \boldsymbol{L}_{i,j}} = (V_{\lfloor \boldsymbol{L}_{i,j}}, E_{\lfloor \boldsymbol{L}_{i,j}})$ be a restricted graph of $G_{i,j,k}$ with respect to $\boldsymbol{L}_{i,j}$ so that the vertex set $V_{\lfloor \boldsymbol{L}_{i,j}}$ corresponds to $\boldsymbol{L}_{i,j}$. There is an edge between two vertices $v, v' \in V_{\lfloor \boldsymbol{L}_{i,j}}$ if and only if $e_{v,v'} \in G_{i,j,k}$ or there is a directed path from v to v' in $G_{i,j,k}$ while the interior node on this path lies in $V_{i,j,k} \setminus V_{\lfloor \boldsymbol{L}_{i,j}}$. The latter case should never happen by the assumptions and rules we asserted when defining the communities. Then the edge set $E_{\lfloor \boldsymbol{L}_{i,j}}$ should be exactly the same as the edge set $E_{i,j}$. Thus, this restricted graph is simply the area of the community $\boldsymbol{L}_{i,j}$, *i.e.* $G_{\lfloor \boldsymbol{L}_{i,j}} = G_{i,j}$. Similarly, we can construct another restricted graph with respect to $\boldsymbol{L}_{j,k}$, denoted by $G_{\lfloor \boldsymbol{L}_{j,k}} = (V_{\lfloor \boldsymbol{L}_{j,k}}, E_{\lfloor \boldsymbol{L}_{j,k}})$. And this graph is the same as $G_{j,k}$. Therefore, following the definition of the causal supergraph given by Williamson and Gabbay [2005], $G_{i,j,k}$ is a causal supergraph of $G_{i,j}$ and $G_{j,k}$. Based on the definition of causal consistency given in the previous work for the RBN [Williamson and Gabbay, 2005], we can conclude that the existence of such a graph implies that the instantiated areas are causally consistent.

□

Given a consistent assignment $\mathcal{H}$, for every core event variable $L_i \in \boldsymbol{L}^{\mathcal{H}}$, let $nd^{\mathcal{H}}(L_i)$ denote the non-descendants of L_i that lying on the same level of L_i, $pa^{\mathcal{H}}(L_i)$ denote the parent variables of L_i represented on the GN, and $Dsup^{\mathcal{H}}(L_i)$ denote the direct superiors of L_i. Here we add the superscript $\mathcal{H}$ to the notations we defined earlier to annotate that these variables are instantiated given the assignment $\mathcal{H}$.

The analogous d-separation theorem on CEGs [Wilkerson, 2020] implies that:

$$Y^{\mathcal{H}}(w) \perp\!\!\!\perp nd(Y^{\mathcal{H}}(w)) | I^{\mathcal{H}}(w). \tag{3.3.5}$$

Here $nd(Y^{\mathcal{H}}(w))$ denotes the floret variables and incident variables which are defined for the positions not lying downstream of w on the root-to-sink path $\lambda^{\mathcal{H}}$. This conditional independence relation is valid for any assignment $\mathcal{H}$.

In a BN, we always assume that a variable is conditionally independent of its non-descendants given its parents. This, however, can not be simply applied on the GN since the value of each core event variable is dependent on its direct superior which is defined on the deeper level CEG. But we can extend this assumption and the causal Markov condition (CMC) assumption that has been made for RBNs by Williamson and Gabbay [2005].

Here we are interested in answering the following two questions : (1) what is the conditional independence relation between a core event variable and the other core event variables lying at the same level as it? (2) what is the conditional independence relation between a core event variable and the variables defined on the deeper level CEG?

To answer the first question, we assume an analogous causal Markov condition for the GN-CEG model. Assume that **given a consistent $\mathcal{H}$, the core event variable L_i is conditionally independent of its non-descendants which lie on the GN given its direct superior on the CEG and its parents on the GN.** This assumption can be written as

$$L_i \perp\!\!\!\perp nd^{\mathcal{H}}(L_i) | pa^{\mathcal{H}}(L_i), Y^{\mathcal{H}}(W_i). \tag{3.3.6}$$

Now we turn to the second question. Williamson and Gabbay [2005] defined the recursive Markov condition (RMC) to link a variable to its non-inferiors and the variables not lying on the same layer as it. This can be mirrored to our setting as follows. We assume that **given a consistent assignment $\mathcal{H}$, the core event variable is conditionally independent of the instantiated variables defined on the CEG excluding its direct superior given its parents which lie on the GN and its direct superior.** We express this assumption as follows.

$$L_i \perp\!\!\!\perp (Y^{\mathcal{H}}(\overline{W_i}), I^{\mathcal{H}}(V_{\mathcal{C}})) | Y^{\mathcal{H}}(W_i), pa^{\mathcal{H}}(L_i), \tag{3.3.7}$$

where $\overline{W_i} = W/W_i$.

On straightforward implication of the conditional independence statements 3.3.6 and 3.3.7 is the following proposition.

PROPOSITION 3.3.4. *Given a consistent assignment $\mathcal{H}$ and the conditional independence assumptions in statements (3.3.6) and (3.3.7), the following conditional independence statement is true within the GN-CEG model:*

$$L_i \perp\!\!\!\perp (nd^{\mathcal{H}}(L_i), Y^{\mathcal{H}}(\overline{W_i}), I^{\mathcal{H}}(V_{\mathcal{C}})) | Y^{\mathcal{H}}(W_i), pa^{\mathcal{H}}(L_i). \tag{3.3.8}$$

Given the conditional independence assumptions, we now move to specify the probability distributions over the GN-CEG model. Within the CEG, the primitive probabilities have been specified for each floret in the previous chapter. Here we specify the conditional probabilities and the joint probability distribution for the whole model.

It is useful next to define a set called a **flattened parent** set analogous to the sets defined by other authors. As reviewed in Chapter 1, Williamson and Gabbay [2005] and Casini et al. [2011] constructed a non-recursive BN from the RBN given an assignment which was called the flattening, see Section 1.3 for the definition. Recall that the vertex set in the flattening is equal to the vertex set of the underlying RBN and edges are added to connect from a direct superior to its inferior or from a parent to its child.

For a core event variable L_i which is instantiated given a consistent assignment $\mathcal{H}$, it has a direct superior $Dsup^{\mathcal{H}}(L_i)$ and a set of parents lying on the GN $pa^{\mathcal{H}}(L_i)$ which can be found from the instantiated area $G^{\mathcal{H}}$. Following the conditional independence statement given in Proposition 3.3.8, we define the set of **flattened parents** for L_i, denoted by $pa^{\mathcal{H}\downarrow}(L_i)$. This is the set of "new" parents of L_i when considering both layers. The flattened parents of L_i consist of the parents of L_i and its direct superior:

$$pa^{\mathcal{H}^{\downarrow}}(L_i) = \{pa^{\mathcal{H}}(L_i), Dsup^{\mathcal{H}}(L_i)\}. \tag{3.3.9}$$

For every $L_i \in \boldsymbol{L}$, we should specify a distribution for

$$p(l_i|pa^{\mathcal{H}^{\downarrow}}(l_i)) = p(l_i|pa^{\mathcal{H}}(l_i), y^{\mathcal{H}}(W_i)) \tag{3.3.10}$$

for learning the structure and the parameters of the model. In a Hidden Markov model (HMM)[Eddy, 2004] or a Hidden semi-Markov model (HSMM) [Yu, 2010], the probability that a hidden state generates a single or a sequence of observations is called the **emission probability**. We can treat l_i as an observation emitted from its latent state $pa^{\mathcal{H}^{\downarrow}}(l_i)$, so we borrow the terminology and call $p(l_i|pa^{\mathcal{H}^{\downarrow}}(l_i))$ the emission probability. In a Bayesian setting, one possibility is to assume a Dirichlet prior to be defined over all states of L_i for each possible value of the flattened parents $pa^{\mathcal{H}^{\downarrow}}(L_i)$. Then we can perform a Dirichlet-Multinomial conjugate inference. The

joint distribution over the model can be factorised as follows:

$$\begin{aligned} p^{\mathcal{H}}(\boldsymbol{l}, \boldsymbol{y}, \boldsymbol{i}) &= \prod_{l_j \in \boldsymbol{l}} p(l_j | pa^{\mathcal{H}}(l_j), Dsup^{\mathcal{H}}(l_j)) \prod_{w \in \lambda^{\mathcal{H}}} p(\Lambda(e_{w,w'}) | \Lambda(w)) \\ &= \prod_{l_j \in \boldsymbol{l}} p(l_j | pa^{\mathcal{H}^{\downarrow}}(l_j)) \prod_{w \in \lambda^{\mathcal{H}}} \pi^{\mathcal{H}}(w' | w), \end{aligned} \tag{3.3.11}$$

where $p^{\mathcal{H}}(\cdot)$ and $\pi^{\mathcal{H}}(\cdot)$ denote the probabilities defined given the assignment $\mathcal{H}$. Both $p(l_j | pa^{\mathcal{H}^{\downarrow}}(l_j))$ and $\pi^{\mathcal{H}}(w'|w)$ have Dirichlet priors.

The map $\chi : (\boldsymbol{l}_d, \Omega_{NC}) \mapsto \lambda$ defined earlier in this chapter returns a latent path for a set of core event variables whose values are observed. This is to learn the latent states of the observations and we can achieve this goal as long as the conditional probabilities in equation (3.3.11) can be well-specified. In particular, we need to infer the posterior probability $p(\lambda | \boldsymbol{l}_d)$ for the d^{th} document. By Bayes rule,

$$p(\lambda | \boldsymbol{l}_d) \propto \pi(\lambda) p(\boldsymbol{l}_d | \lambda). \tag{3.3.12}$$

The first component can be factorised as $\pi(\lambda) = \Pi_{e \in \lambda} \theta_e$, where the primitive probabilities can be estimated by a Dirichlet-Multinomial conjugate analysis as explained in the previous chapter. For the likelihood $p(\boldsymbol{l}_d | \lambda)$, notice that given a path λ to be traversed by the unit, we have an underlying consistent assignment $\mathcal{H}_\lambda$. Then,

$$p(\boldsymbol{l}_d | \lambda) = \prod_{l_{d_j} \in \boldsymbol{l}_d} p(l_{d_j} | pa^{\mathcal{H}_\lambda}(l_{d_j}), Dsup^{\mathcal{H}_\lambda}(l_{d_j})). \tag{3.3.13}$$

This actually is the first term on the right hand side of equation (3.3.11). By specifying each $p(l_{d_j} | pa^{\mathcal{H}_\lambda}(l_{d_j}), Dsup^{\mathcal{H}_\lambda}(l_{d_j}))$, the probability $p(\boldsymbol{l}_d | \lambda)$ can be uniquely estimated and $p(\lambda | \boldsymbol{l}_d)$ can be inferred by Bayes rule. Therefore, the function χ is well-defined if we input the estimated mean posterior probabilities to the function. These parameters are denoted by Ω_{NC}. Under this setting, we could learn a unique latent path for $\boldsymbol{l}_d$. This immediately implies the following proposition.

PROPOSITION 3.3.5. *When there is no core event variable that is unobservable, then the mapping $\Delta : (\boldsymbol{\omega}_d, \Omega) \mapsto \lambda$ returns a unique latent path for each document.*

Proof. Suppose that the latent path returned by Δ is not unique for document $\boldsymbol{\omega}_d \in D$. Since the mapping Γ returns a unique set $\boldsymbol{l}_d$ for $\boldsymbol{\omega}_d$, this means for the same set of observations $\boldsymbol{l}_d$, there exists a path $\lambda_1 \in \Lambda_{\mathcal{C}}$ and another path $\lambda_2 \in \Lambda_{\mathcal{C}}$ corresponding to $\boldsymbol{l}_d$ and $\lambda_1 \neq \lambda_2$. This is not possible since χ uniquely determines a latent path for $\boldsymbol{l}_d$. We have a contradiction. So Δ uniquely determines a latent path for a document.

□

Given the hierarchical framework specified earlier in this chapter, in this section, we have clarified how the core event variables lying on a GN and the floret or incident variables lying on a CEG can be associated and made conditional independence assumptions about the relationships between the core event variables and the floret or incident variables. This enables us to transform the causal dependency between the two layers. The connection between the two layers makes it possible to find a unique path for the set of core events extracted from each document. So a unique latent path for each document can be identified on the CEG. Detailed specifications of the conditional probabilities required for learning a latent path for the set of core events associated to a document will be given in the experimental chapter with an example of an implementable algorithm, see Chapter 5.

Chapter 4

Missingness in the GN-CEG

In this chapter, we discuss two types of missingness: floret-dependent missingness and event-dependent missingness. The former requires a reconstruction of CEG while the latter requires a reconstruction of GN.

The floret-dependent missingness refers to the missingness of the information represented within some florets on the CEG. This is also called the informed missingness [Barclay et al., 2014]. Different arbitrary types of missingness mechanisms can be read from the CEG with informed missingness, including missing at random (MAR), missing completely at random (MCAR) and missing not at random (MNAR) [Rubin, 1976]. In a similar way, from a given M-CEG, we can analyse how an event depends on the missingness of another event that precedes it. In this case, the CEG we defined in Chapter 2 is no longer sufficient for modelling the missing data. Instead, we construct a new event tree with informed missingness and derive a CEG from this tree. In Section 4.1, we will formalise this type of missingness and demonstrate how to construct the corresponding tree. The CEG with this new topology embeds the floret-dependent missingness. We aim to estimate the effects of the remedial intervention and the routine intervention on this new topology. We therefore extend the back-door theorem to identify the causal effects on the CEG with informed missingness. The adapted back-door criterion for different types of manipulations will be given in Section 4.2.

The event-dependent missingness refers to the missing values of core event variables. The floret-dependent missingness is a special type of the event-dependent missingness when the values of the core event variables lying in the community of a floret variable are all missing. In Section 4.3, we will provide guidelines for embedding this type of missingness on the GN.

4.1 Floret-dependent missingness

When a community $\boldsymbol{L}_i$ is unobservable, i.e. all the sub-communities $\boldsymbol{L}_{ij}$ can be completely missing, then the associated floret variable $Y(w_i)$ is an unobservable variable. Specifically, when the value of every core event variable in the community $L \in \boldsymbol{L}_i$ is missing, then the value of the corresponding floret variable $Y(w_i)$ is missing. In this case, if we still use the CEG formalised in Chapter 3, we might not be able to determine which root-to-sink path is associated with the observations. This is why we do not index the floret variable by a path here. For this partially observed document, let $W^{'}$ denote the set of positions whose corresponding incident variables are instantiated. There might be multiple root-to-sink paths on the CEG traversing the positions $W^{'}$, which are associated with the partially observed document. Then the map Δ, which is defined to identify a latent path on the CEG, is not well-defined since there might be multiple latent paths matched to a single document. Furthermore, the assignment $\mathcal{H}$ is not consistent any more because $\lambda^{\mathcal{H}}$ is a set of more than one path. Therefore, in presence of this type of missingness, some setups and assumptions made in Chapter 3 are violated. Therefore, this motivates us to augment the CEG so that the rules defined for the hierarchical model can still be followed. This augmented CEG is similar to the CEG with informed missingness introduced by Barclay et al. [2014], but the missingness is modelled in a more general way. The augmented CEG is still derived from the underlying event tree. Thus, we need to first augment the underlying event tree for the missing data.

4.1.1 The m-tree

First of all, we classify the d-events into $\mathbb{X}_{\mathcal{T}}^{M}$ and $\mathbb{X}_{\mathcal{T}}^{O}$, representing the unobservable d-events and the d-events that are always observed respectively. These two subsets satisfy $\mathbb{X} = \mathbb{X}_{\mathcal{T}}^{M} \cup \mathbb{X}_{\mathcal{T}}^{O}$ and $\mathbb{X}_{\mathcal{T}}^{M} \cap \mathbb{X}_{\mathcal{T}}^{O} = \emptyset$. A d-event x is an element in $\mathbb{X}_{\mathcal{T}}^{M}$ if the associated core events are all missing at least once in any data set, otherwise $x \in \mathbb{X}_{\mathcal{T}}^{O}$. Knowing which d-events are unobservable, we can find which floret variables defined on the even tree or the CEG are unobservable.

For a floret $\mathcal{F}(v) \in \mathcal{F}(S_{\mathcal{T}})$ and $v \in S_{\mathcal{T}}$, if every emanating edge of v, $e \in E(v)$, is labelled by an unobservable d-event $x(e) \in \mathbb{X}_{\mathcal{T}}^{M}$, then this floret is classified into $\mathcal{F}(v) \in \mathcal{F}^{M}$ and the corresponding floret variable $Y(v)$ is an unobservable floret variable. Otherwise, this floret variable is always observed and the corresponding floret is classified into $\mathcal{F}(v) \in \mathcal{F}^{O}$. Then $\mathcal{F}^{M}$ and $\mathcal{F}^{O}$ are collections of florets corresponding to the unobservable floret variables and the fully observed floret variables respectively. The set of florets defined on the event tree is $\mathcal{F}(S_{\mathcal{T}})$. The two subsets

$\mathcal{F}^M$ and $\mathcal{F}^O$ satisfy $\mathcal{F}(S_\mathcal{T}) = \mathcal{F}^M \cup \mathcal{F}^O$ and $\mathcal{F}^M \cap \mathcal{F}^O = \emptyset$.

Note here that we do not consider the case that for a situation v, the d-event $x(e)$ is missing for some emanating edge $e \in E(v)$ but not all the emanating edges of v. This may complicate the problem by inducing dependencies between edges in $E(v)$. This scenario is beyond the scope of this thesis but could be studied in the future.

For every floret $\mathcal{F}(v_i) \in \mathcal{F}^M$, we define a missingness indicator for it. We call this type of missingness the **floret-dependent missingness** and the missingness indicator the **missing floret indicator**. The missing floret indicator for $\mathcal{F}(v) \in \mathcal{F}^M$ is defined as:

$$B_{\mathcal{F}(v_i)} = \begin{cases} 1 & \text{if the value of } Y(v_i) \text{ is missing,} \\ 0 & \text{otherwise.} \end{cases} \tag{4.1.1}$$

Then $B_{\mathcal{F}(v_i)}$ represents the conditional missingness so that

$$p(B_{\mathcal{F}(v_i)} = 1) = p(\mathcal{F}(v_i) \text{ missing}) = p(Y(v_i) \text{ missing} | \mu(v_0, v_i)). \tag{4.1.2}$$

Let $p(B_{\mathcal{F}(v_i)} = 1) \in (0, 1)$. Let $\boldsymbol{B} = \{B_\mathcal{F}\}_{\mathcal{F} \in \mathcal{F}^M}$ denote the set of missing floret indicators. For each $B_{\mathcal{F}(v_i)}$, we construct a floret representing this indicator, denoted by $\mathcal{F}(B_{\mathcal{F}(v_i)})$. We call it the **missingness indicator floret**. We import such florets onto the original event tree $\mathcal{T}$. To distinguish the original event tree and the event tree with missingness indicator florets, we call the former the **fact event tree** and the latter the **missingness event tree** (m-tree). Let $\mathcal{T}^M$ denote the topology of the m-tree. The missingness indicator florets compose a new class of florets on the m-tree, denoted by $\mathcal{F}^{MI}$. Then $\mathcal{F}^{MI} = \mathcal{F}(\boldsymbol{B})$.

When importing each missing floret indicator $B_{\mathcal{F}(v_i)}$ to the event tree, we create a set of situations $V(B_{\mathcal{F}(v_i)})$ whose florets $\mathcal{F}(V(B_{\mathcal{F}(v_i)})) = \mathcal{F}(B_{\mathcal{F}(v_i)})$ are missingness indicator florets. For each of the new situation $v_j \in V(B_{\mathcal{F}(v_i)})$, there are two edges emanating from it $E(v_j) = \{e_{v_j 1}, e_{v_j 2}\}$. One represents $b_{\mathcal{F}(v_i)} = 1$. Let the d-event labelled on this edge be $b_{\mathcal{F}(v_i),1}$, then $x(e_{v_j 1}) = b_{\mathcal{F}(v_i),1}$. The other edge $e_{v_j 2}$ represents $b_{\mathcal{F}(v_i)} = 0$. Let the d-event labelled on this edge be $x(e_{v_j 2}) = b_{\mathcal{F}(v_i),0}$. The state space of d-events is enlarged by adding $b_{\mathcal{F}(v_i),1}$ and $b_{\mathcal{F}(v_i),0}$ to $\mathbb{X}_\mathcal{T}$ for all $\mathcal{F}(v_i) \in \mathcal{F}^M$. Let $\mathbb{X}_{\mathcal{T}^M}$ denote the state space of d-events labelled on the m-tree.

Assume $B_{\mathcal{F}(v_i)}$ precedes $Y(v_i)$, denoted by $B_{\mathcal{F}(v_i)} \prec Y(v_i)$. On the m-tree, the edge $e_{v_j 1}$ emanates from $v_j \in V(B_{\mathcal{F}(v_i)})$ is received by a vertex $v_k \in V_{\mathcal{T}^M}$ whose emanating edges $E(v_k)$ are labelled by the d-events $x(E(v_k))$. Since $e_{v_j 1}$ represents $\mathcal{F}(v_i)$ is missing, the floret appended to v_k cannot represent the same information as $\mathcal{F}(v_i)$. Specifically, $\mathcal{F}(v_k)$ should represent the same information as $\mathcal{F}(ch(v_i))$,

which is the set of florets of the children of v_i on the fact event tree. On the fact event tree, the edges of the children of v_i, *i.e.* $E(ch(v_i))$, are labelled by the d-events $x(E(ch(v_i)))$. Here let $x(E(ch(v_i))) = \bigcup_{e \in E(ch(v_i))} x(e)$ be the set of unique d-events labelled on edges $E(ch(v_i))$. Then $x(E(ch(v_i)))$ should be labelled on the edges emanating from v_k so that $x(E(v_k)) = x(E(ch(v_i)))$. Let $v_l \in V_{\mathcal{T}^M}$ be the receiving node of $e_{v_j 2}$. Then the edges emanating from this node, denoted by $E(v_l)$, are labelled by same d-events as $E(v_i)$ on the fact event tree. Therefore, by following the order $B_{\mathcal{F}(v_i)} \prec Y(v_i)$, we are informed by $\mathcal{F}(B_{\mathcal{F}(v_i)})$ about whether the d-events $x(E(v_i))$ are missing. This is called the **informed missingness** [Barclay et al., 2014].

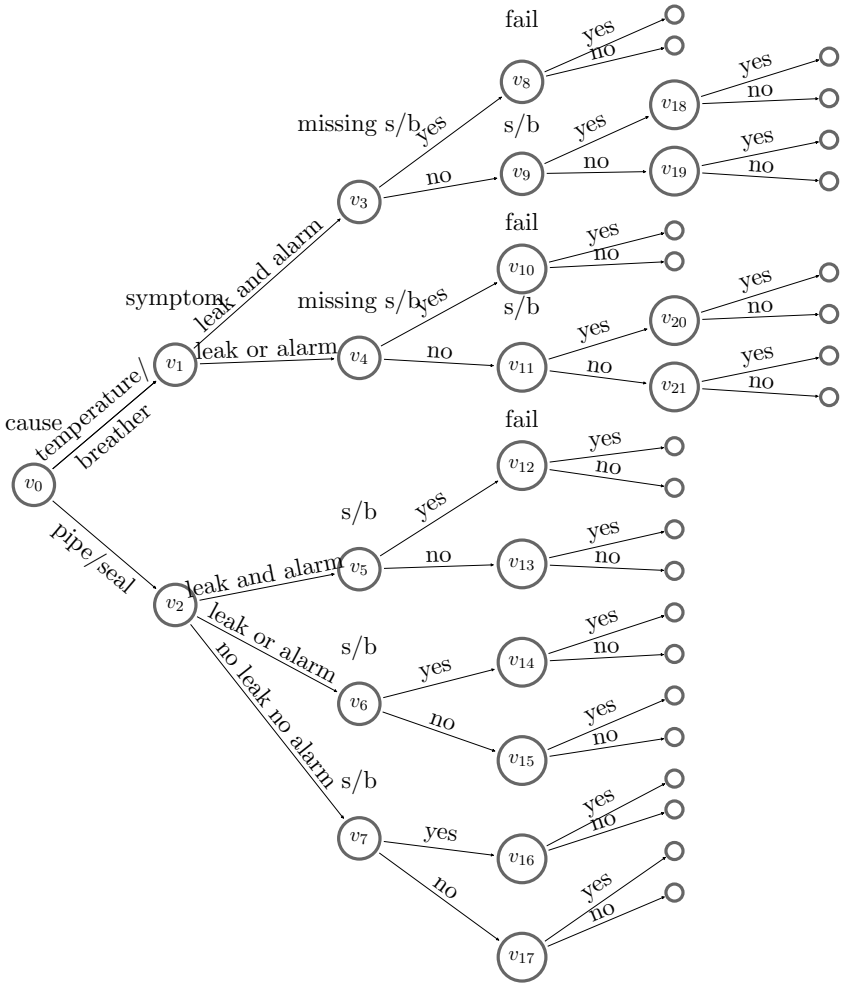


Figure 4.1: The m-tree elicited from the fact tree in Figure 2.1 for Example 11.

Example 11. *We give an example of an m-tree constructed from the fact event tree in Figure 2.1. Assume that the sight glass or buchholz defect are unobserv-*

able conditional on the root cause being temperature change or breather defect. The "conditional" missingness can be easily represented by the semantics of the event tree which shows the advantage of using tree graphs to represent such asymmetric processes. In the fact tree, $\mathcal{F}^M = \{\mathcal{F}(v_3), \mathcal{F}(v_4)\}$. Accordingly, missing floret indicators $\{B_{\mathcal{F}(v_3)}, B_{\mathcal{F}(v_4)}\}$ are created for the unobservable floret variables. Then we construct the corresponding m-tree in Figure 4.1 following the rules we specified above. The missingness indicator florets on the m-tree are $\mathcal{F}^{MI} = \{\mathcal{F}(v_3), \mathcal{F}(v_4)\}$, where $v_3, v_4 \in V_{\mathcal{T}^M}$.

4.1.2 The M-CEG

Having constructed a m-tree, we can elicit or learn a CEG from it in the same way as deriving a CEG from a fact event tree. We call such a CEG the **missingness CEG** (M-CEG). Let $C^M = (V_{C^M}, E_{C^M})$ denote the topology of the M-CEG. The vertex set is $V_{C^M} = S_{\mathcal{C}^M} \cup w_\infty^f \cup w_\infty^n$, where the non-sink vertices are $S_{\mathcal{C}^M} = V^M \cup V^O \cup V^{MI}$. A position w is in V^O if there is no missingness indicator associated with it. Then we classify its floret as $\mathcal{F}(w) \in \mathcal{F}^O$. For $w \in V^M$, there is a missing floret indicator associated with it and we let $\mathcal{F}(w) \in \mathcal{F}^M$. For $w \in V^{MI}$, the floret $\mathcal{F}(w)$ is associated with the missing floret indicator. So we let $\mathcal{F}(w) \in \mathcal{F}^{MI}$.

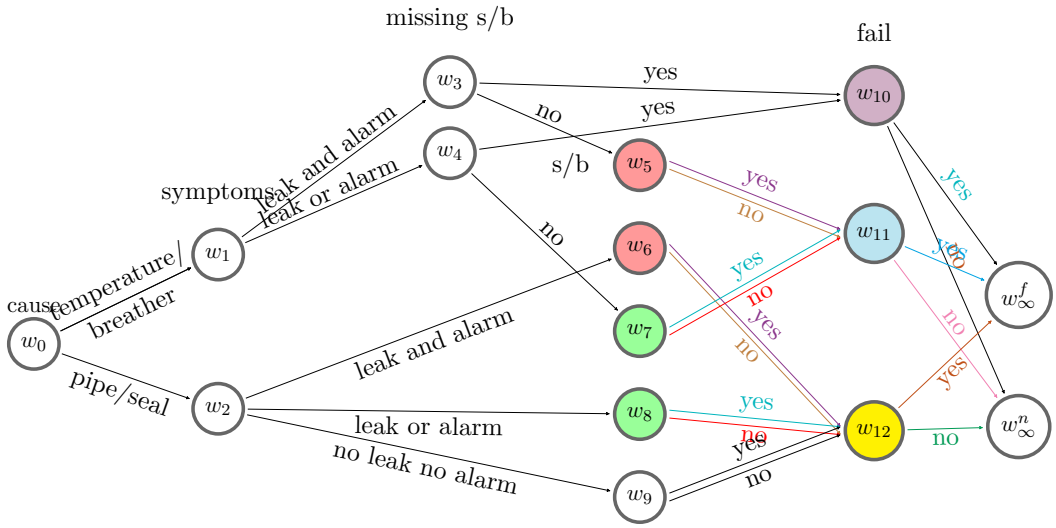


Figure 4.2: A M-CEG for the conservator data.

Example 12. *Figure 4.2 depicts an example of the M-CEG which can be derived from the m-tree in Figure 4.1. Assume it is an ordinal M-CEG[Barclay et al., 2014]. We see the position w_{10} is aligned higher than w_{11} and w_{12} on the tree. So we can deduce that given the missing s/b defect, the probability of failure is predicted to be*

higher than the probability of failure when s/b defect is not missing. We can also conclude that the missingness indicators depend on the observed symptoms since w_3 and w_4 are not in the same stage.

Notice that Heckerman [2008] indicated that the conjugacy of Dirichlet-Multinomial no longer tends to hold in the absence of some data when learning a BN. This leads to an intractable posterior so that the MAP scores do not have a closed form and are not separable. However the scores can still be numerically estimated although the whole search process then becomes much slower. On an m-tree, we assume that, for any floret, the parameters of primitive probability vector are independent, and the vectors of primitive probabilities associated with each stage are mutually independent. This ensures a model search based on product of independent Dirichlet priors over the model parameters and a closed-form conjugate analysis [Freeman and Smith, 2011*a*]. We reassign the priors to $\boldsymbol{\theta}_v$ for all $v \in S_{\mathcal{T}^M}$. Barclay et al. [2014] have shown that the standard Bayesian selection algorithm can then be employed with scores in a closed form in order to search for a best CEG with informed missingness. This, however, has not been established formally in previous work. Here we specify it for selecting the M-CEG from a given m-tree.

In Chapter 2, we have illustrated how conjugate inference can be performed on a CEG and shown the MAP algorithm for model selection [Freeman and Smith, 2011*a*]. We next specify the two perspectives which should be considered and added to the MAP algorithm for learning the M-CEG.

First of all, for situations $v_i, v_j \in S_{\mathcal{T}}$ and their florets satisfy $\mathcal{F}(v_i), \mathcal{F}(v_j) \in \mathcal{F}^M$, if $\mathcal{F}(v_i)$ and $\mathcal{F}(v_j)$ represent the same information, *i.e.* $x(E(v_i)) = x(E(v_j))$, then the situations in $V(B_{\mathcal{F}(v_i)})$ and $V(B_{\mathcal{F}(v_j)})$ on the m-tree can be grouped into the same stage. If $v_a, v_b \in V(B_{\mathcal{F}(v_i)}) \cup V(B_{\mathcal{F}(v_j)})$, and $\boldsymbol{\theta}_{v_a} = \boldsymbol{\theta}_{v_b}$ so that the edges e_{v_a,v_k} and e_{v_b,v_l} with the same label have the same primitive probability, then v_a, v_b are in the same stage.

Secondly, the log-likelihood score for a M-CEG $\mathcal{C}^M$ can be decomposed into local scores associated with the stages corresponding to missing floret indicators $\boldsymbol{B}$, denoted by $\mathbb{U}^{MI}$, and local scores associated with the stages corresponding to observable events, denoted by $\overline{\mathbb{U}}^{MI}$. Let $\mathbb{U}_{\mathcal{T}^M}$ denote the collection of stages of $\mathcal{T}^M$. Then $\mathbb{U}_{\mathcal{T}^M} = \mathbb{U}^{MI} \cup \overline{\mathbb{U}}^{MI}$ and $\mathbb{U}^{MI} \cap \overline{\mathbb{U}}^{MI} = \emptyset$.

$$\log Q(\boldsymbol{\theta}; \mathcal{C}^M) = \sum_{u_i \in \overline{\mathbb{U}}^{MI}} \log Q_{u_i}(\boldsymbol{\theta}; \mathcal{C}^M) + \sum_{u_j \in \mathbb{U}^{MI}} \log Q_{u_j}(\boldsymbol{\theta}; \mathcal{C}^M). \tag{4.1.3}$$

The log-likelihood can be computed explicitly in a closed form as shown in equation (2.5.1).

4.2 Identifying causal effects on the M-CEG

Suppose we have the M-CEG for a causal analysis instead of the CEG without the informed missingness, and we aim to show that the effects of the domain-specific interventions specified in Chapter 2 can be estimated given the M-CEG when data are corrupted by missing values. In this section, we establish an adapted back-door criterion for identifying effects on the M-CEGs for different types of manipulations and interventions we have formulated. This extends the adjustment criterion proposed by Saadati and Tian [2019] for "recovering" $p(y|do(x))$. Now we begin with a review of these authors' work.

4.2.1 A review of m-graphs

Mohan et al. [2013]; Mohan and Pearl [2014]; Mohan [2017]; Mohan and Pearl [2021] have augmented a DAG to a missingness graph, called an m-graph, to explicitly indicate the missingness mechanisms that exist in a given dataset. The m-graph has topology $G = (V, E)$. The vertex set V consists of $V^O \cup V^M \cup V^{MI} \cup V^*$. The vertices V^O correspond to variables $\boldsymbol{R}^O$ which are always observed; V^M correspond to variables $\boldsymbol{R}^M$ which are partially observed; V^{MI} correspond to missingness indicators $\boldsymbol{B}$; V^* correspond to proxy variables. Previous work [Mohan et al., 2013; Mohan and Pearl, 2014; Mohan, 2017; Mohan and Pearl, 2021; Saadati and Tian, 2019] have demonstrated the **recoverability** of the conditional probabilities, the joint probability, and the casual query of a *do*-operator $p(y|do(x))$ on the m-graph.

A joint probability distribution is recoverable whenever it can be consistently estimated for any missingness process. As shown by Mohan et al. [2013], the joint distribution is always recoverable on the m-graph when data are missing completely at random (MCAR) or missing at random (MAR). The recoverability of the joint distribution is complicated to analyse when data are missing not at random (MNAR) because the missingness indicators are not independent of $\boldsymbol{R}^O$ and $\boldsymbol{R}^M$. However, Mohan and Pearl [2014] have shown that the probability $p(y|do(x))$ is always estimable from the dataset with missing data given the m-graph. The recoverability of this probability on the m-graph is a sufficient condition for the identifiability of the causal query on the graph whose vertex set consists of the vertices corresponding to $\boldsymbol{R}^O$ and $\boldsymbol{R}^M$ [Mohan and Pearl, 2021].

Saadati and Tian [2019] have developed an adjustment criterion for identifying and recovering causal effects of a *do*-operator from missing data on an m-graph. The definition is given below.

Definition 4.2.1 (M-Adjustment Formula [Saadati and Tian, 2019])**.** *Given an m-*

graph G over $\boldsymbol{R}$ and $\boldsymbol{B}$, a set $\boldsymbol{Z} \subset \boldsymbol{R}$ is called an m-adjustment (adjustment under missing data) set for estimating the causal effect of X on Y, if, for every model compatible with G, and $\boldsymbol{W} = \boldsymbol{R}^M \cap (X \cup Y \cup \boldsymbol{Z})$, then

$$P(y|do(x)) = \sum_{\boldsymbol{z}} P(y|x, \boldsymbol{z}, \boldsymbol{B_W} = 0) P(\boldsymbol{z}|\boldsymbol{B_W} = 0). \tag{4.2.1}$$

Definition 4.2.2. *A set of variables $\boldsymbol{Z}$ is a m-adjustment set for estimating the causal effect of $\boldsymbol{X}$ on $\boldsymbol{Y}$) if, letting $\boldsymbol{W} = \boldsymbol{V}_m \cap (\boldsymbol{X} \cup \boldsymbol{Y} \cup \boldsymbol{Z})$,*

1. *no element of $\boldsymbol{Z}$ is a descendant in $G_{\overline{X}}$ of any $W \in \boldsymbol{X}$ which lies on a proper causal path from $\boldsymbol{X}$ to $\boldsymbol{Y}$,*
2. *all non-causal paths between $\boldsymbol{X}$ and $\boldsymbol{Y}$ in G are blocked by $\boldsymbol{Z}$,*
3. *$\boldsymbol{B_W}$ is d-separated from $\boldsymbol{Y}$ given $\boldsymbol{X}, \boldsymbol{Z}$, i.e. $Y \perp\!\!\!\perp \boldsymbol{B_W} | X, \boldsymbol{Z}$,*
4. *$\boldsymbol{Z}$ is d-separated from $\boldsymbol{B_W}$, i.e.,$\boldsymbol{Z} \perp\!\!\!\perp \boldsymbol{B_W}$.*

Definition 4.2.2 provides the sufficient conditions for a set $\boldsymbol{Z}$ to be an m-adjustment set. Note that the adjustment criterion generalises the back-door criterion to the complete graph. As pointed out by Shpitser et al. [2012], if $\boldsymbol{Z}$ satisfies the back-door criterion with respect to (X, Y) in G, then $\boldsymbol{Z}$ satisfies the adjustment criterion with respect to (X, Y) in G.

4.2.2 Recoverability of probabilities from the M-CEG

Recall that the d-event $b_{\mathcal{F}(v),0}$ refers to the missing information represented by the floret $\mathcal{F}(v)$. Let $\mathcal{F}(x)$ denote the set of florets so that one of the edge in the floret $\mathcal{F} \in \mathcal{F}(x)$ is labelled by the d-event x. Let $\tilde{\pi}(\cdot)$ denote the path related probability on the M-CEG.

Let $W(\boldsymbol{b}_{\mathcal{F}(x),0})$ denote the set of positions on the M-CEG which are the receiving nodes of the edges labelled by $\boldsymbol{b}_{\mathcal{F}(x),0}$, *i.e.* $\mathcal{F}(x)$ is not missing. Then, for $w \in W(\boldsymbol{b}_{\mathcal{F}(x),0})$, the edges emanating from w, denoted by $E(w)$, contain one edge labelled by x.

We next give a general case for defining the recoverability of causal query. Let x_o and z_o denote the two d-events which are always observed, x_m and z_m denote the unobservable events. We firstly find the florets associated with x_m and z_m. Denote it by $\mathcal{F}_{x_m \cup z_m}$, and

$$\mathcal{F}_{x_m \cup z_m} = \{\mathcal{F} : \mathcal{F} \in \mathcal{F}(x_m) \cup \mathcal{F}(z_m) \text{ and } \mathcal{F} \in \mathcal{F}^M\}. \tag{4.2.2}$$

The associated missing floret indicators are $\boldsymbol{B}_{\mathcal{F}_{x_m \cup z_m}} = \{B_{\mathcal{F}}\}_{\mathcal{F} \in \mathcal{F}_{x_m \cup z_m}}$. Let the collection of paths associated with observing $\mathcal{F}(x_m)$ and $\mathcal{F}(z_m)$ be denoted by

$$\Lambda_{\boldsymbol{b}_{\mathcal{F}_{x_m \cup z_m},0}} = \Lambda_{b_{\mathcal{F}(x_m),0}} \cap \Lambda_{b_{\mathcal{F}(z_m),0}} = \Lambda(E(\boldsymbol{b}_{\mathcal{F}(x_m),0})) \cap \Lambda(E(\boldsymbol{b}_{\mathcal{F}(z_m),0})), \tag{4.2.3}$$

and let

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_m \cup z_m},0})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(x_m),0}) \cap \Lambda(W(\boldsymbol{b}_{\mathcal{F}(z_m),0}). \tag{4.2.4}$$

The M-CEG enables us to visualise the unfolded events including the missingness information. So it is straightforward to find the collection of paths $\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_m \cup z_m},0}))$.

Definition 4.2.3. *The set of **manifest paths** for $x_m, z_m \in \mathbb{X}^M_{\mathcal{C}^M}$ is the largest set of the root-to-sink paths on the M-CEG $\mathcal{C}^M$ traversing the florets $\mathcal{F}(x_m)$ and $\mathcal{F}(z_m)$, i.e. $\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_m \cup z_m},0}))$.*

If we estimate $\pi(\Lambda_{x_o}, \Lambda_{x_m} | \Lambda_{z_o}, \Lambda_{z_m})$ on the CEG, it is nontrivial to identify the proper set of the root-to-sink paths associated with these d-events when x_m or z_m is missing, while the semantics of the M-CEG can solve this problem. The M-CEGs allows us to estimate $\pi(\Lambda_{x_o}, \Lambda_{x_m} | \Lambda_{z_o}, \Lambda_{z_m})$ by conditioning on the corresponding manifest paths. In this case, we say that the probability $\pi(\Lambda_{x_o}, \Lambda_{x_m} | \Lambda_{z_o}, \Lambda_{z_m})$ can be recovered [Mohan et al., 2013; Mohan and Pearl, 2014; Saadati and Tian, 2019] on the M-CEG. We formalise this in the following lemma.

Lemma 4.2.4. *The probability $\pi(\Lambda_{x_o}, \Lambda_{x_m} | \Lambda_{z_o}, \Lambda_{z_m})$ on the CEG is **recoverable** from the partially observed data if we can estimate $\tilde{\pi}(\Lambda_{x_o}, \Lambda_{x_m} | \Lambda_{z_o}, \Lambda_{z_m}, \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_m \cup z_m,0}})))$ on the M-CEG.*

We next show that when data are corrupted by missing values, the M-CEG is a simple and intuitive vehicle for identifying the effects of a singular manipulation or a stochastic manipulation or a composite of manipulations.

4.2.3 Recover causal queries on the M-CEG

Singular manipulations. For a singular manipulation on the d-event x, if we are interested in exploring its effects on the d-event y, as defined in Chapter 2, we need to estimate $\pi(\Lambda_y || \Lambda_x)$ on the CEG. Given a M-CEG, we aim to recover this probability. Note that y and x might always be observed or unobserved. On a CEG, we find appropriate events $\boldsymbol{z}$ so that $\Lambda_{\boldsymbol{z}}$ partitions $\Lambda_{\mathcal{C}}$. Here we also need to define such a partition in order to identify the effects on y via the back-door theorem. The restriction we impose here is that $z \in \boldsymbol{z}$ is not a missingness indicator. Then for $z \in \boldsymbol{z}$, $z \in \mathbb{X}^M_{\mathcal{C}^M}$ or $z \in \mathbb{X}^O_{\mathcal{C}^M}$.

We first define the manifest paths for the controlled d-event x and the effect d-event y on the M-CEG. According to Lemma 4.2.4, we recover a probability by conditioning on the corresponding events being observed. Therefore to recover the effect of x on y we also condition on them being observed. Accordingly, the back-door partition is required to partition the manifest paths of x and y. Let

$$\mathcal{F}_{x\cup y} = \{\mathcal{F} : \mathcal{F} \in \mathcal{F}(x) \cup \mathcal{F}(y) \text{ and } \mathcal{F} \notin \mathcal{F}^{MI}\}. \tag{4.2.5}$$

When $x \in \mathbb{X}^M_{\mathcal{C}^M}$, $y \in \mathbb{X}^O_{\mathcal{C}^M}$, since x is unobservable, the manifest paths for x and y are

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(x),0})). \tag{4.2.6}$$

When $x \in \mathbb{X}^O_{\mathcal{C}^M}$, $y \in \mathbb{X}^M_{\mathcal{C}^M}$, the manifest paths for x and y are

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y),0})). \tag{4.2.7}$$

When $x, y \in \mathbb{X}^M_{\mathcal{C}^M}$, then the manifest paths for x and y are

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(x),0})) \cap \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y),0})). \tag{4.2.8}$$

When $x, y \in \mathbb{X}^O_{\mathcal{C}^M}$, then the manifest paths are the whole collection of the root-to-sink paths on the M-CEG, $\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})) = \Lambda_{\mathcal{C}^M}$.

We next construct a sub-M-CEG by the manifest paths defined on the M-CEG $\mathcal{C}^M$. We call this the **manifest M-CEG** and denote its topology by $\mathcal{C}^{M*}$. The root-to-sink paths of the manifest M-CEG are then the manifest paths $\Lambda_{\mathcal{C}^{M*}} = \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))$. In fact, $\Lambda_{\boldsymbol{z}}$ is only required to partition $\Lambda_{\mathcal{C}^{M*}}$.

We next pick the subset of root-to-sink paths of the manifest M-CEG which pass along the edges representing the controlled event x. These paths are called the **manipulated paths**. Let $\hat{\Lambda}$ denote the manipulated paths so that

$$\hat{\Lambda} = \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})) \cap \Lambda_x. \tag{4.2.9}$$

The manipulated M-CEG is constructed from the manipulated paths. We denote the manipulated M-CEG by $\hat{\mathcal{C}}^M$, then $\Lambda_{\hat{\mathcal{C}}^M} = \hat{\Lambda}$.

By Lemma 4.2.4, we then recover $\pi(\Lambda_y||\Lambda_x)$ from the manifest M-CEG. Denote this causal query by

$$\tilde{\pi}^{\hat{\Lambda}}(\Lambda_y) = \tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y||\Lambda_x). \tag{4.2.10}$$

Theorem 4.2.5 (The m-back-door theorem for singular manipulation)**.** *When there*

are missing values in datasets, the effect of a singular manipulation on x on y is identifiable on the M-CEG if we can find a partition $\Lambda_{\boldsymbol{z}}$ of $\Lambda_{\mathcal{C}^{M*}}$, $z \in \mathbb{X}_{\mathcal{C}}^{M}$ or $z \in \mathbb{X}_{\mathcal{C}}^{O}$ for any $z \in \boldsymbol{z}$, so that

$$\tilde{\pi}^{\hat{\Lambda}}(\Lambda_y) = \sum_{z} \tilde{\pi}(\Lambda_y|\Lambda_x, \Lambda_z, \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})))\tilde{\pi}(\Lambda_z|\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) \tag{4.2.11}$$

can be estimated uniquely from the observable events.

We next specify the criteria for $\{\Lambda_{\boldsymbol{z}}\}$ so that the m-back-door theorem defined above is valid. This is analogous to the criteria given in Theorem 2.2.2 which was proved by Thwaites [2013]. For $w' \in pa(W(x))$ such that $e_{w',w''} \in E(x)$, we can simply validate the following given a singular manipulation on Λ_x.

$$\begin{aligned}\tilde{\pi}(\Lambda_y|\Lambda_x, \Lambda_z, \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) &= \tilde{\pi}(\Lambda_y|\Lambda(w'), \Lambda_x, \Lambda_z, \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) \\ &= \tilde{\pi}(\Lambda_y|\Lambda(w'), \bigcup_{e\in E(x)} \Lambda(e), \Lambda_z, \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) \\ &= \tilde{\pi}(\Lambda_y|\Lambda(e_{w',w''}), \Lambda_z, \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})))\end{aligned} \tag{4.2.12}$$

When there exists a BN equivalent to this M-CEG, then equation (4.2.12) can be translated as $Y \perp\!\!\!\perp pa(X)|X, \boldsymbol{Z}, \boldsymbol{B_W}$, where $\boldsymbol{W} = X \cup Y$.

Note that if $\Lambda_x \cap \Lambda_y \neq \emptyset$, then $\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})) \subset \Lambda(E(y))$. We also have $\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})) \subset \Lambda(E(x))$. Then interestingly the following always holds on the M-CEG.

$$\begin{aligned}&\tilde{\pi}(\Lambda_y|\Lambda_x, \Lambda_z, \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) \\ &= \tilde{\pi}(\bigcup_{e_{w_y,w'_y}\in E(y)} \Lambda(e_{w_y,w'_y})| \bigcup_{e_{w_x,w'_x}\in E(x)} \Lambda(e_{w_x,w'_x}), \Lambda_z, \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) \\ &= \tilde{\pi}(\bigcup_{e_{w_y,w'_y}\in E(y)} \Lambda(e_{w_y,w'_y})| \bigcup_{e_{w_x,w'_x}\in E(x)} \Lambda(e_{w_x,w'_x}), \Lambda_z) \\ &= \tilde{\pi}(\Lambda_y|\Lambda_x, \Lambda_z)\end{aligned} \tag{4.2.13}$$

When there exists a BN equivalent to this M-CEG, then this equation can be translated as $Y \perp\!\!\!\perp \boldsymbol{B_W}|X, \boldsymbol{Z}$. This coincides with the third condition given in the definition of the m-adjustment set, see Definition 4.2.2.

$$\begin{cases} Y \perp\!\!\!\perp pa(X)|X, \boldsymbol{Z}, \boldsymbol{B_W} \\ Y \perp\!\!\!\perp \boldsymbol{B_W}|X, \boldsymbol{Z} \end{cases} \implies Y \perp\!\!\!\perp (pa(X), \boldsymbol{B_W})|X, \boldsymbol{Z} \implies Y \perp\!\!\!\perp pa(X)|X, \boldsymbol{Z}.$$

Note that the last statement is the criterion for the back-door partition when working on the BN or the CEG without missingness indicators.

For $w' \in pa(W(x))$ such that $e_{w',w''} \in E(x)$, we also need $\{\Lambda_z\}$ to satisfy

$$\tilde{\pi}(\Lambda_z|\Lambda(w'), \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) = \tilde{\pi}(\Lambda_z|\Lambda(e_{w',w''}), \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))). \tag{4.2.14}$$

When there exists a BN equivalent to this M-CEG, this is analogous to the statement $\boldsymbol{Z} \perp\!\!\!\perp X|pa(X), \boldsymbol{B_W}$ on the BN.

Theorem 4.2.6. *The m-back-door partition $\{\Lambda_z\}$ for recovering the singular manipulation must satisfy equation (4.2.12) and equation (4.2.14).*

Equation (4.2.12) and equation (4.2.14) are sufficient conditions for the manifest recovery of $\pi(\Lambda_y||\Lambda_x)$ and are called the m-back-door criteria. We can now prove Theorem 4.2.5 in the same way as the proof of the back-door theorem for the singular manipulation on CEGs [Thwaites, 2013].

Proof of Theorem 4.2.5.

$$\begin{aligned}
\tilde{\pi}^{\hat{\Lambda}}(\Lambda_y) &= \sum_{w'\in pa(W(x))} \tilde{\pi}^{\hat{\Lambda}}(\Lambda(w'))\tilde{\pi}^{\hat{\Lambda}}(\Lambda_y|\Lambda(w')) \\
&= \sum_{w'\in pa(W(x)),e_{w',w''}\in E(x)} \tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda(w'))\tilde{\pi}^{\hat{\Lambda}}(\Lambda_y|\Lambda(w'),\Lambda(w'')) \\
&= \sum_{w'\in pa(W(x)),e_{w',w''}\in E(x)} \tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda(w'))\tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y|\Lambda(w'')) \\
&= \sum_{w'\in pa(W(x)),e_{w',w''}\in E(x)} \tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda(w'))\tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y|\Lambda(e_{w',w''}),\Lambda(w'')) \\
&= \sum_{w'\in pa(W(x)),E_{w',w''}\in e(x)} \tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda(w'))\tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y|\Lambda(e_{w',w''})) \\
&= \sum_{w'\in pa(W(x)),e_{w',w''}\in E(x)} \tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda(w'))\sum_z \tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_z,\Lambda_y|\Lambda(e_{w',w''})) \\
&= \sum_{w'\in pa(W(x)),e_{w',w''}\in E(x)} \tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda(w'))\sum_z \tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y|\Lambda_z,\Lambda(e_{w',w''}))\tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_z|\Lambda(e_{w',w''})) \\
&= \sum_{w'\in pa(W(x)),e_{w',w''}\in E(x)} \tilde{\pi}(\Lambda(w')|\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})))\sum_z \tilde{\pi}(\Lambda_y|\Lambda_z,\Lambda(e_{w',w''}),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) \\
&\qquad \times \tilde{\pi}(\Lambda_z|\Lambda(e_{w',w''}),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})))
\end{aligned} \tag{4.2.15}$$

Applying equation (4.2.12) and equation (4.2.14) gives

$$\begin{aligned}\tilde{\pi}^{\hat{\Lambda}}(\Lambda_y) = &\sum_{w'\in pa(W(x)),E_{w',w''}\in E(x)} \tilde{\pi}(\Lambda(w')|\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) \sum_z \tilde{\pi}(\Lambda_y|\Lambda_x,\Lambda_z,\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) \\ &\times \tilde{\pi}(\Lambda_z|\Lambda(w'),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))) \\ = &\sum_z \tilde{\pi}(\Lambda_y|\Lambda_x,\Lambda_z,\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}})))\tilde{\pi}(\Lambda_z|\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x\cup y,0}}))).\end{aligned} \tag{4.2.16}$$

□

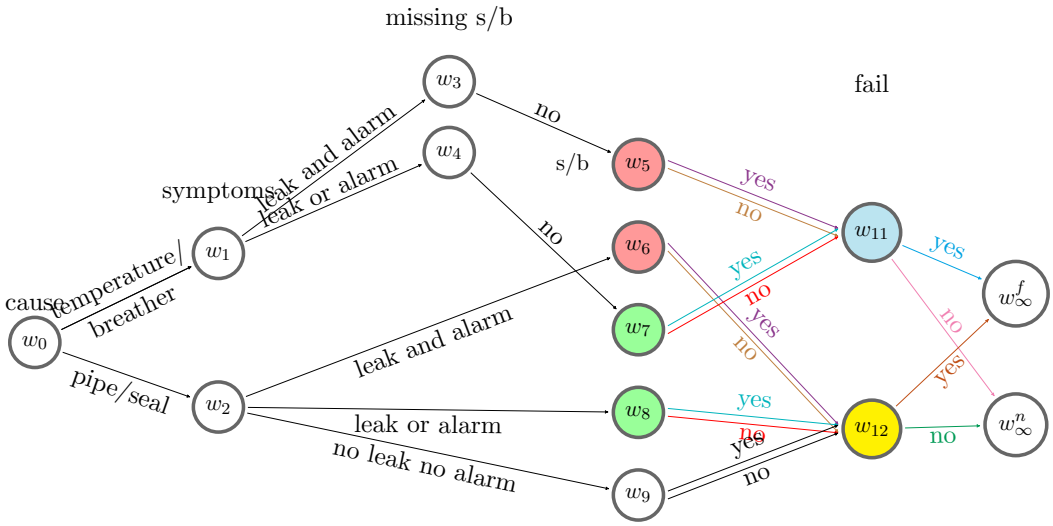


Figure 4.3: The manifest M-CEG for Example 13.

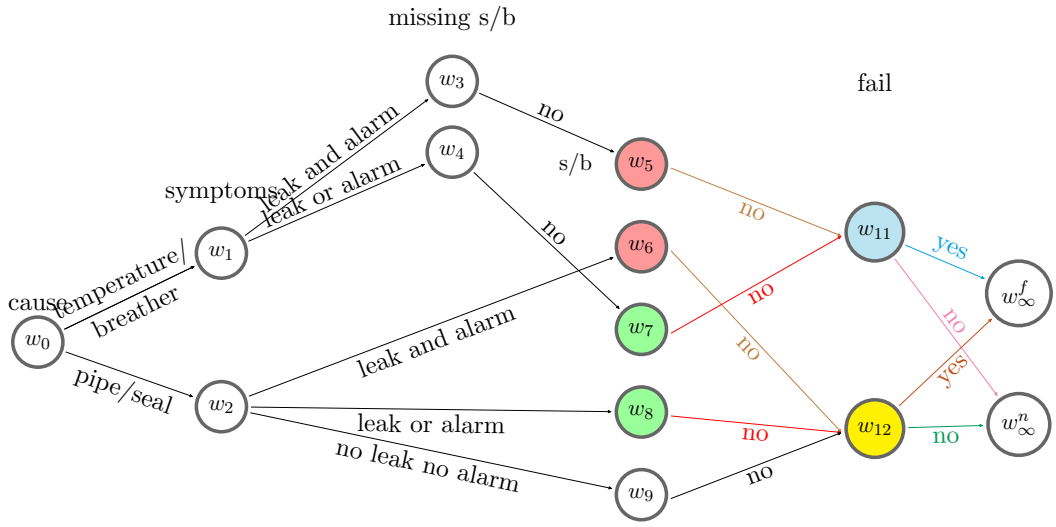


Figure 4.4: The manipulated M-CEG for Example 13.

Example 13. *We continue with Example 12. The M-CEG is shown in Figure 4.2. Suppose we are interested in the effect on system failure from forcing $x_{s/b,0}$, i.e. there is no sight glass or buchholz defect. The edges associated with the controlled d-event are $E(x_{s/b,0}) = \{e^2_{w_5,w_{11}}, e^2_{w_6,w_{12}}, e^2_{w_7,w_{11}}, e^2_{w_8,w_{12}}, e^2_{w_9,w_{12}}\}$. Note that $x_{s/b,0}$ is unobservable. We have $W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b}},0}) = \{w_5, w_6, w_7, w_8, w_9\}$. The effect event $x_{f,1}$ is always observed. Then the manifest paths are $\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0})) = \bigcup_{w\in\{w_5,\cdots,w_9\}} \Lambda(w)$. We can construct the corresponding manifest M-CEG by the manifest paths, see Figure 4.3. By forcing $e \in E(x_{s/b,0})$ to be passed with probability 1, from the manifest M-CEG we can further construct the manipulated M-CEG, see Figure 4.4.*

Here we let the partition be $\Lambda_{z^1} = \Lambda(e_{w_1,w_3}) \cup \Lambda(e_{w_1,w_4})$ and $\Lambda_{z^2} = \Lambda(e_{w_2,w_6}) \cup \Lambda(e_{w_2,w_8}) \cup \Lambda(e_{w_2,w_9})$. So that z^1 refers to the symptoms conditional on temperature change/breather defect, while z^2 refers to the symptoms conditional on pipe/seal defect. Both z^1 and z^2 are d-events which are always observed.

We can check whether the criteria in equation (4.2.12) and equation (4.2.14) are satisfied by the chosen partition set. For example, considering Λ_{z^1} and $\Lambda(e^2_{w_5,w_{11}}) \subset \Lambda_{x_{s/b,0}}$,

$$\begin{aligned}
&\tilde{\pi}(\Lambda_{x_{f,1}} | \Lambda(w_5), \Lambda_{x_{s/b,0}}, \Lambda_{z^1}, \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0}))) \\
&= \tilde{\pi}(\Lambda(e_{w_{11},w_\infty^f}) | \Lambda(w_5), \Lambda(e^2_{w_5,w_{11}}), \Lambda(e_{w_1,w_3}) \cup \Lambda(e_{w_1,w_4}), \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0}))) \\
&= \tilde{\pi}(\Lambda(e_{w_{11},w_\infty^f}) | \Lambda(e^2_{w_5,w_{11}}), \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0}))).
\end{aligned} \tag{4.2.17}$$

So equation (4.2.12) is satisfied.

$$\begin{aligned}\tilde{\pi}(\Lambda_{z^1}|\Lambda(w_5),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0}))) &= \tilde{\pi}(\Lambda(e_{w_1,w_3})\cup\Lambda(e_{w_1,w_4})|\Lambda(w_5),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0})))\\ &= \tilde{\pi}(\Lambda(e_{w_1,w_3})|\Lambda(w_5),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0})))\\ &= 1.\end{aligned} \tag{4.2.18}$$

$$\tilde{\pi}(\Lambda_{z^1}|\Lambda(e^2_{w_5,w_{11}}),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0}))) = 1 \tag{4.2.19}$$

Therefore,

$$\tilde{\pi}(\Lambda_{z^1}|\Lambda(w_5),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0}))) = \tilde{\pi}(\Lambda_{z^1}|\Lambda(e^2_{w_5,w_{11}}),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x_{s/b,0}\cup x_{f,1}},0}))). \tag{4.2.20}$$

So equation (4.2.14) is satisfied.

Therefore, there exists $\{\Lambda_z\}$ *satisfying the m-back-door criterion so that we can recover the effect of intervening* $x_{s/b,0}$ *on* $x_{f,1}$ *from the M-CEG.*

Stochastic manipulations. The recoverability of the effects of stochastic manipulations that are imported to the idle system by the remedia intervention is an extension of the recoverability of the effects of the singular manipulation. Here, we employ the same notations as we defined in Section 2.3 for the remedial intervention. Assume the probabilities $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$ which are assigned after the intervention are known. Then for every controlled event $x \in x(E(\boldsymbol{w}^*))$, let $\hat{\tilde{\pi}}(\Lambda_x) = \tilde{\pi}(\Lambda_x||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}) = \hat{\tilde{\pi}}^{\Lambda(\boldsymbol{w}^*)}(\Lambda_x)$. This probability can be computed in the same way as we demonstrated in Chapter 2. Let

$$\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y} = \{\mathcal{F} : \mathcal{F} \in \mathcal{F}(\boldsymbol{w}^*) \cup \mathcal{F}(y) \text{ and } \mathcal{F} \notin \mathcal{F}^{MI}\}. \tag{4.2.21}$$

When $x(E(\boldsymbol{w}^*)) \in \mathbb{X}^M_{\mathcal{C}^M}$ and $y \in \mathbb{X}^O_{\mathcal{C}^M}$, the manifest paths for $x(E(\boldsymbol{w}^*))$ and y are

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(\boldsymbol{w}^*),0})) = \Lambda(\boldsymbol{w}^*). \tag{4.2.22}$$

When $x(E(\boldsymbol{w}^*)) \in \mathbb{X}^O_{\mathcal{C}^M}$ and $y \in \mathbb{X}^M_{\mathcal{C}^M}$, the manifest paths for $x(E(\boldsymbol{w}^*))$ and y are

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y)),0})). \tag{4.2.23}$$

When $x(E(\boldsymbol{w}^*)) \in \mathbb{X}^M_{\mathcal{C}^M}$ and $y \in \mathbb{X}^M_{\mathcal{C}^M}$, the manifest paths for $x(E(\boldsymbol{w}^*))$ and y are

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(\boldsymbol{w}^*),0})) \cap \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y)),0})) = \Lambda(\boldsymbol{w}^*) \cap \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y)),0})). \tag{4.2.24}$$

Then we construct the manifest M-CEG $\mathcal{C}^{M*}$ by these manifest paths so that $\Lambda_{\mathcal{C}^{M*}} = \Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0}))$.

For the CEG without informed missingness, we have demonstrated that for the remedial intervention, the manipulated paths are $\Lambda(\boldsymbol{w}^*)$. For the M-CEG, analogous to how we define the manipulated paths for the singular manipulation, here the manipulated paths are $\Lambda(\boldsymbol{w}^*) \cap \Lambda_{\mathcal{C}^{M*}}$. This is a subset of the manifest paths and can be found on the manifest M-CEG.

To recover the effect of the stochastic manipulation on $\mathcal{F}(\boldsymbol{w}^*)$, we estimate $\tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*})$:

$$\tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}) = \sum_{x\in x(E(\boldsymbol{w}^*))}\sum_{z}\tilde{\pi}(\Lambda_y|\Lambda_x,\Lambda_z,\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0})))\tilde{\pi}(\Lambda_z|\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0})))$$
$$\times\,\hat{\tilde{\pi}}(\Lambda_x|\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0}))). \tag{4.2.25}$$

The m-back-door partition $\{\Lambda_z\}$ needs to satisfy the following criteria. For every $x \in x(E(\boldsymbol{w}^*))$, for $w' \in pa(W(x))$ such that $e_{w',w''} \in E(x)$,

$$\tilde{\pi}(\Lambda_y|\Lambda(w'),\Lambda_x,\Lambda_z,\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0}))) = \tilde{\pi}(\Lambda_y|\Lambda(e_{w',w''}),\Lambda_z,\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0}))), \tag{4.2.26}$$

$$\tilde{\pi}(\Lambda_z|\Lambda(w'),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0}))) = \tilde{\pi}(\Lambda_z|\Lambda(e_{w',w''}),\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0}))). \tag{4.2.27}$$

These are analogous to the criteria specified in equation (4.2.12) and equation (4.2.14) for the singular manipulation on the M-CEG.

Example 14. *Here we continue with the bushing example given in Chapter 2. Suppose the endogenous root causes are likely to be missing. We plot a hypothesised M-CEG for this system in Figure 4.5. Note that when the endogenous causes are missing, the observed symptoms could be oil leak, no leak, loss of oil, mix of oil, thermal runaway and electrical discharge. Interestingly, in this case, we can deduce the missing root cause from the observed symptoms. For example, if we observe loss of oil, then the cause of the failure comes from the insulator; if we observe thermal runaway, then the root cause is not gasket, porcelain or insulator. However, if we observe oil leak, then we cannot distinguish whether the missing cause is gasket or porcelain because they can yield the same set of symptoms.*

When there is a stochastic manipulation over the endogenous root causes, $\boldsymbol{w}^* = \{w_2\}$. If we are interested in the failure of the system, then the effect d-event should always be observed. Therefore, the manifest paths are $\Lambda(\boldsymbol{w}^*) = \Lambda(w_2)$ and the underlying manifest M-CEG is plotted in Figure 4.6. The manipulated paths are

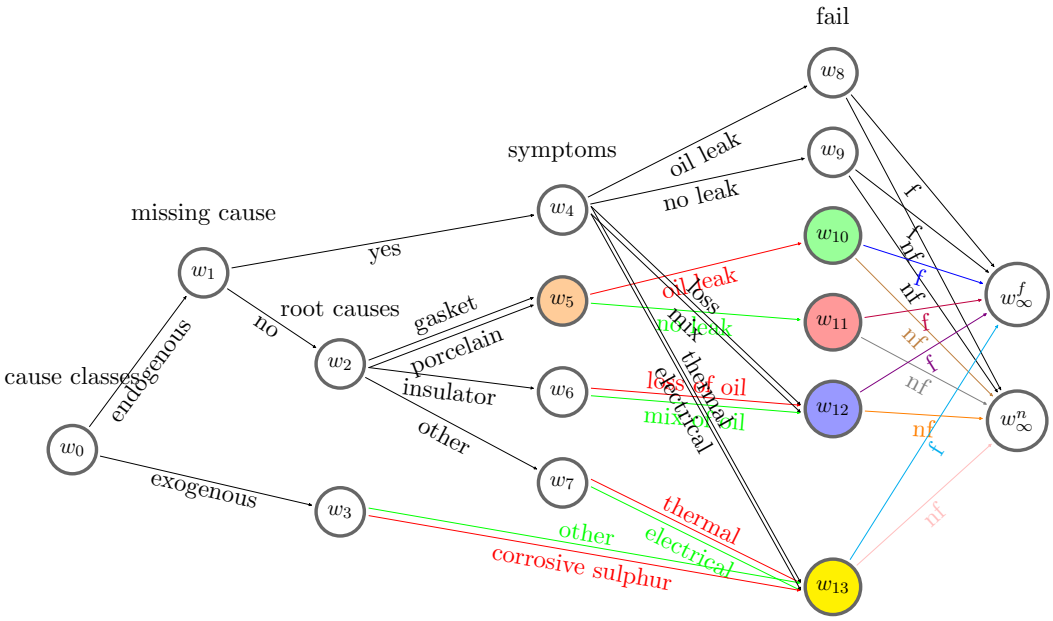


Figure 4.5: The M-CEG constructed for the bushing system. Some of the labelled d-events are simplified to fit the figure.

also $\Lambda(\boldsymbol{w}^*)$*. So the topology of the manipulated M-CEG is the same as the topology of the manifest M-CEG in this example.*

Let the partition of $\Lambda(\boldsymbol{w}^*)$ *be* $\{\Lambda_{z_1}, \Lambda_{z_2}\}$ *so that* $E(z_1) = \{e_{w_5,w_{10}}, e^1_{w_6,w_{12}}, e^1_{w_7,w_{13}}\}$ *and* $E(z_2) = \{e_{w_5,w_{11}}, e^2_{w_6,w_{12}}, e^2_{w_7,w_{13}}\}$*. The events* z_1 *and* z_2 *represent symptoms which are always observable. The criteria stated in equation (4.2.26) and equation (4.2.27) are obviously satisfied by* $\{\Lambda_{z_1}, \Lambda_{z_2}\}$*. This is because by replacing* $\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{x(E(\boldsymbol{w}^*))\cup y},0}))$ *by* $\Lambda(\boldsymbol{w}^*)$*, these are just the criteria we specified for the stochastic manipulation in Chapter 2.*

Routine intervention. In Section 2.4, we explained that the routine intervention can lead to a singular manipulation, a stochastic manipulation, composite singular manipulations, and composite singular and stochastic manipulations. We have shown the causal identifiability on the M-CEG for the former two scenarios. Here we simply extend the theorem to the last two scenarios.

When there are composite singular manipulations on $\boldsymbol{x}$, we estimate the effects from the M-CEG by recovering $\pi(\Lambda_y||\Lambda_{\boldsymbol{x}})$. Let

$$\mathcal{F}_{\boldsymbol{x}\cup y} = \{\mathcal{F} : \mathcal{F} \in (\cup_{x_i\in\boldsymbol{x}}\mathcal{F}(x_i)) \cup \mathcal{F}(y) \text{ and } \mathcal{F} \notin \mathcal{F}^{MI}\}. \tag{4.2.28}$$

Let $\boldsymbol{x}^*$ be a subset of $\boldsymbol{x}$ so that $\boldsymbol{x}^* \subset \mathbb{X}^M_{\mathcal{C}^M}$ are unobservable. If the set $\boldsymbol{x}^*$ is not

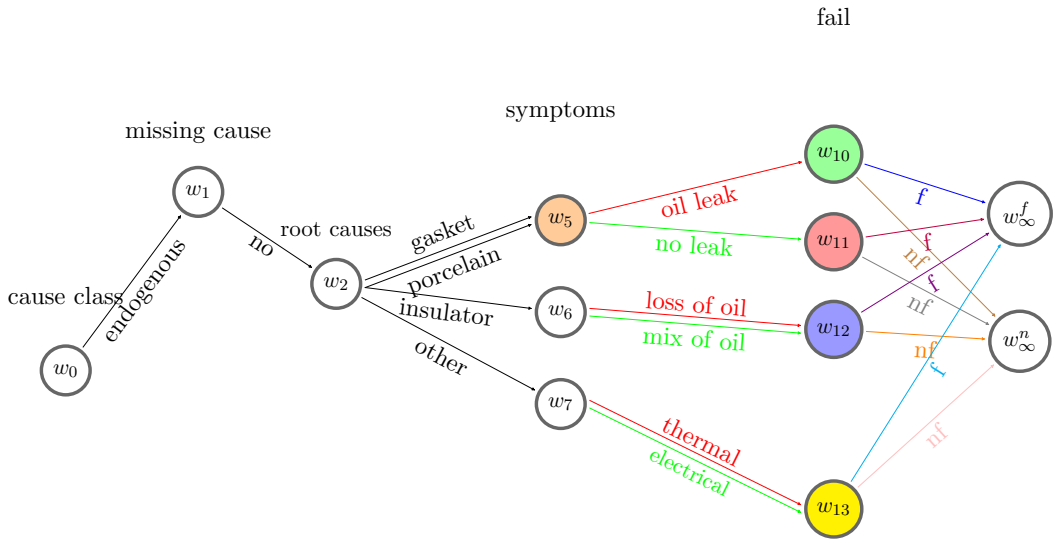


Figure 4.6: The manifest M-CEG for Example 14. Some of the labelled d-events are simplified to fit the figure.

empty, and $y \in \mathbb{X}^O_{\mathcal{C}^M}$, then the manifest paths for $\boldsymbol{x}$ and y are

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) = \cap_{x_i\in\boldsymbol{x}^*}\Lambda(W(\boldsymbol{b}_{\mathcal{F}(x_i),0})). \tag{4.2.29}$$

If $y \in \mathbb{X}^M_{\mathcal{C}^M}$, then

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) = (\cap_{x_i\in\boldsymbol{x}^*}\Lambda(W(\boldsymbol{b}_{\mathcal{F}(x_i),0}))) \cap \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y),0})). \tag{4.2.30}$$

When $\boldsymbol{x}^*$ is empty, if $y \in \mathbb{X}^O_{\mathcal{C}^M}$ the manifest paths are $\Lambda(W(\boldsymbol{b}_{\mathcal{F}(y),0}))$. Otherwise, the manifest paths are $\Lambda_{\mathcal{C}^M}$.

In equation (2.4.9), we have shown how to use the back-door partition $\Lambda_{\boldsymbol{z}}$ to identify such causal effects on the CEG. Here, given the M-CEG, we extend this expression to

$$\tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y||\Lambda_{\boldsymbol{x}}) = \sum_z \tilde{\pi}(\Lambda_y|\Lambda_{\boldsymbol{x}},\Lambda_z,\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})))\tilde{\pi}(\Lambda_z|\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0}))). \tag{4.2.31}$$

In Section 2.4, we demonstrated the back-door criteria for a composite of two singular manipulations on x_{r1} and x_{r2}. Here we still demonstrate the criteria for the m-back-door partition $\{\Lambda_z\}$ on these two singular manipulations. For any $w_1 \in pa(W(x_{r1}))$, $w_2 \in pa(W(x_{r2}))$, let $e_{w_1,w_1^*} \in E(x_{r1})$ and $e_{w_2,w_2^*} \in E(x_{r2})$.

Then the criteria are

$$\pi(\Lambda_y|\Lambda(w_1),\Lambda(w_2),\Lambda_{\boldsymbol{x}},\Lambda_z,\Lambda(w(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0}))) = \pi(\Lambda_y|\Lambda(e_{w_1,w_1^*}),\Lambda(e_{w_2,w_2^*}),\Lambda_z,\Lambda(w(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0}))), \tag{4.2.32}$$

$$\pi(\Lambda_z|\Lambda(w_1),\Lambda(w_2),\Lambda(w(\boldsymbol{b}_{F_{\boldsymbol{x}\cup y},0}))) = \pi(\Lambda_z|\Lambda(e_{w_1,w_1^*}),\Lambda(e_{w_2,w_2^*})),\Lambda(w(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0}))). \tag{4.2.33}$$

When there is a singular manipulation on $x^\dagger$ and a stochastic manipulation on $\boldsymbol{\theta}_{\boldsymbol{w}^*}$ so that the new probabilities are $\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}$. Let $\boldsymbol{x} = \{x^\dagger, x(E(\boldsymbol{w}^*))\}$ and

$$\mathcal{F}_{\boldsymbol{x}\cup y} = \{\mathcal{F} : \mathcal{F} \in \mathcal{F}(x^\dagger) \cup \mathcal{F}(\boldsymbol{w}^*) \cup \mathcal{F}(y) \text{ and } \mathcal{F} \notin \mathcal{F}^{MI}\}. \tag{4.2.34}$$

When $x^\dagger \in \mathbb{X}^M_{\mathcal{C}^M}$ and $x(E(\boldsymbol{w}^*)) \subset \mathbb{X}^O_{\mathcal{C}^M}$, if $y \in \mathbb{X}^O_{\mathcal{C}^M}$ then the manifest paths for $\boldsymbol{x}$ and y are:

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(x^\dagger),0})). \tag{4.2.35}$$

If $y \in \mathbb{X}^M_{\mathcal{C}^M}$ then the manifest paths for $\boldsymbol{x}$ and y are:

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(x^\dagger),0})) \cap \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y),0})). \tag{4.2.36}$$

When $x^\dagger \in \mathbb{X}^O_{\mathcal{C}^M}$ and $x(E(\boldsymbol{w}^*)) \subset \mathbb{X}^M_{\mathcal{C}^M}$, if $y \in \mathbb{X}^O_{\mathcal{C}^M}$, then the manifest paths for $\boldsymbol{x}$ and y are:

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) = \Lambda(\boldsymbol{w}^*). \tag{4.2.37}$$

If $y \in \mathbb{X}^M_{\mathcal{C}^M}$ then the manifest paths for $\boldsymbol{x}$ and y are:

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) = \Lambda(\boldsymbol{w}^*) \cap \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y),0})). \tag{4.2.38}$$

When $x^\dagger \in \mathbb{X}^M_{\mathcal{C}^M}$ and $x(E(\boldsymbol{w}^*)) \subset \mathbb{X}^M_{\mathcal{C}^M}$, if $y \in \mathbb{X}^O_{\mathcal{C}^M}$, then the manifest paths for $\boldsymbol{x}$ and y are:

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(x^\dagger),0})) \cap \Lambda(\boldsymbol{w}^*). \tag{4.2.39}$$

If $y \in \mathbb{X}^M_{\mathcal{C}^M}$ then the manifest paths for $\boldsymbol{x}$ and y are:

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(x^\dagger),0})) \cap \Lambda(\boldsymbol{w}^*) \cap \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y),0})). \tag{4.2.40}$$

When $x^\dagger \in \mathbb{X}^O_{\mathcal{C}^M}$ and $x(E(\boldsymbol{w}^*)) \subset \mathbb{X}^O_{\mathcal{C}^M}$, if $y \in \mathbb{X}^O_{\mathcal{C}^M}$, then the manifest paths for $\boldsymbol{x}$ and y are $\Lambda_{\mathcal{C}^M}$. If $y \in \mathbb{X}^M_{\mathcal{C}^M}$ then the manifest paths are

$$\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) = \Lambda(W(\boldsymbol{b}_{\mathcal{F}(y),0})). \tag{4.2.41}$$

The manipulated paths are $\Lambda(W(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})) \cap \Lambda_{x^\dagger} \cap \Lambda(\boldsymbol{w}^*)$.

Then we estimate $\tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}, \Lambda_{x^\dagger})$ from the manifest M-CEG by finding the m-back-door partition $\{\Lambda_z\}$ so that

$$\tilde{\pi}^{\Lambda_{\mathcal{C}^{M*}}}(\Lambda_y||\hat{\boldsymbol{\theta}}_{\boldsymbol{w}^*}, \Lambda_{x^\dagger}) = \sum_{x\in\boldsymbol{x}}\sum_{z}\tilde{\pi}(\Lambda_y|\Lambda_x, \Lambda_z, \Lambda(w(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y},0})))\tilde{\pi}(\Lambda_z|\Lambda_x, \Lambda(w(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y}})))$$
$$\times\, \hat{\tilde{\pi}}(\Lambda_x|\Lambda(w(\boldsymbol{b}_{\mathcal{F}_{\boldsymbol{x}\cup y\cup},0}))). \tag{4.2.42}$$

Then for any $w_1 \in pa(W(x^\dagger))$, $w_2 \in \boldsymbol{w}^*$, when $e_{w_1,w_1^*} \in E(x^\dagger)$ and $e_{w_2,w_2^*} \in E(\boldsymbol{w}^*)$, the criteria in equation (4.2.32) and equation (4.2.33) will be satisfied by $\{\Lambda_z\}$.

4.3 Event-dependent missingness

In this section, we discuss another type of missingness which is represented on the Global Net (GN). Suppose given a consistent assignment $\mathcal{H}$, the floret variable takes value $Y^{\mathcal{H}}(w_i) = y_{w_i,w_j} \neq 0$ so that there is a transition on the M-CEG along e_{w_i,w_j}. Let $\boldsymbol{L}_{i,j} = \{L_{1,ij}, ..., L_{n,ij}\}$, $n > 0$, denote the sub-community of core event variables associated with y_{w_i,w_j}. In the previous section, we focused on the scenario when the value of $Y(w_i)$ was missing. Here we assume the value of the floret variable can still be inferred when the core event variables in the sub-community associated with the value of the floret variable are not completely missing. In other words, we observe $\boldsymbol{L}_{i,j}$ partially.

We do not assume that every core event variable is unobservable. Instead we assume there exist some core event variables that are always observed. Let $\boldsymbol{L}^{\ddagger}_{i,j} = \{L_{i,j,k1}, \cdots, L_{i,j,km}\} \subseteq \boldsymbol{L}_{i,j}$ denote the subset of core event variables associated with y_{w_i,w_j} which are unobservable. Then we define the missingness indicators for these unobservable core event variables as follows.

Definition 4.3.1. *For every core event variable $L_{i,j,k} \in \boldsymbol{L}^{\ddagger}_{i,j}$, we define a* ***missing event indicator*** *$B_{i,j,k}$ so that*

$$B_{i,j,k} = \begin{cases} 1, & \text{if } L_{i,j,k} \text{ is missing,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.3.1}$$

The set of missing event indicators associated with $\boldsymbol{L}^{\ddagger}_{i,j}$ is

$$\boldsymbol{B}^{\ddagger}_{i,j} = \{B_{i,j,k1}, \cdots, B_{i,j,km}\}. \tag{4.3.2}$$

We call this type of missingness the **event-dependent missingness**. We

next represent these missing event indicators on the GN by adding the corresponding vertices and edges. First of all, we make assumptions about the relationships between the missing event indicators and other variables.

ASSUMPTION 4.3.2. *Given a consistent assignment $\mathcal{H}$,*

1. *the parent variables of a missing event indicator $B_{i,j,k}$ on the GN consist of the N core event variables preceding $L_{i,j,k}$, where $N > 0$. We denote this parent set by $pa^{\mathcal{H}}(B_{i,j,k})$. We call this missingness the N-event-dependent missingness;*

2. *the core event variable $L_{i,j,k}$ depends on $B_{i,j,k}$;*

3. *the missing event indicator $B_{i,j,k}$ shares the same direct superior as $L_{i,j,k}$.*

Under Assumption 4.3.2, the set of parent variables of $B_{i,j,k}$ that lie on the GN is

$$pa^{\mathcal{H}}(B_{i,j,k}) = \{L^{\mathcal{H}}_{i,j,k-1}, ..., L^{\mathcal{H}}_{i,j,k-N}\}. \tag{4.3.3}$$

The direct superior of $B_{i,j,k}$ is

$$Dsup^{\mathcal{H}}(B_{i,j,k}) = Y^{\mathcal{H}}(w_i). \tag{4.3.4}$$

The flattened parents are

$$pa^{\mathcal{H}^{\downarrow}}(B_{k,ij}) = \{Y^{\mathcal{H}}(w_i), L^{\mathcal{H}}_{i,j,k-1}, ..., L^{\mathcal{H}}_{i,j,k-N}\}. \tag{4.3.5}$$

By the conditional independence assumption made in statement (3.3.8), we have

$$B_{i,j,k} \perp\!\!\!\perp nd^{\mathcal{H}}(B_{i,j,k}) | Y^{\mathcal{H}}(w_i), pa^{\mathcal{H}}(B_{i,j,k}). \tag{4.3.6}$$

By adding vertices corresponding to the missing event indicators and adding edges to the GN based on Assumption 4.3.2, we have a new graph lying at the surface level of the hierarchical model. We call this the **missingness GN** (M-GN). In Figure 4.7 we give an example of the M-GN for the 1-event-dependent missingness.

We call the hierarchical model that embeds the event-dependent missingness and the floret-dependent missingness the **missingness GN-CEG** (M-GN-CEG) model. In order to learn a root-to-sink path for each document, we must specify the conditional probabilities $p(b_{i,j,k}|pa^{\mathcal{H}^{\downarrow}}(B_{i,j,k}))$ and $p(l_{i,j,k}|pa^{\mathcal{H}^{\downarrow}}(L_{i,j,k}), b_{i,j,k})$. Given these conditional probabilities, the joint distribution in equation (3.3.11) can be

revised for the M-GN-CEG model as follows.

$$p^{\mathcal{H}}(\boldsymbol{l},\boldsymbol{b},\boldsymbol{y},\boldsymbol{i}) = \prod_{l_j\in\boldsymbol{l}} \Big(p(b_j|pa^{\mathcal{H}^{\downarrow}}(b_j))p(l_j|b_j,pa^{\mathcal{H}}(l_j),Dsup^{\mathcal{H}}(l_j))\Big)^{\mathbb{I}_{l_j\in\boldsymbol{\iota}^{\ddagger}}} \times p(l_j|pa^{\mathcal{H}}(l_j),Dsup^{\mathcal{H}}(l_j))^{\mathbb{I}_{l_j\notin\boldsymbol{\iota}^{\ddagger}}} \times \prod_{w\in\lambda^{\mathcal{H}}} \pi^{\mathcal{H}}(w'|w). \tag{4.3.7}$$

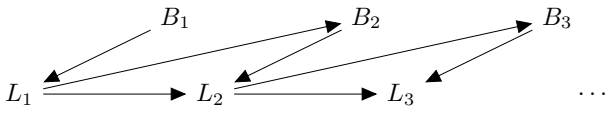


Figure 4.7: An example of the M-GN.

In summary, this chapter has explored the possibility to model missingness in the GN-CEG model. We have focused on the informed missingness, specifically MNAR, by extending the framework proposed in Chapter 2 and Chapter 3. This allows us to have a more realistic model for analysing the real-world data. Future work can make effort to extend the current model to fit other types of missingness that may appear in reliability data.

Chapter 5

Generative Process and Experiments

In this chapter, we will design experiments to support the methods we have developed and demonstrated in previous chapters. This includes the causal algebras for the remedial intervention and the routine intervention, see Chapter 2, and the GN-CEG model for embedding the causal dependencies, see Chapter 3.

We have explained how the effects of these domain-specific interventions can be incorporated into the conjugate learning algorithm of CEGs in Section 2.5. In Section 5.1, we will show with examples how the causal structure and parameter estimation can be improved by learning with causal algebras.

This is followed by a detailed demonstration of the general process for learning the parameters in the GN-CEG model in Section 5.2. This is accompanied by a proposal of a simple Gibbs sampler for estimating the posterior distributions.

In Section 5.3, we will run the proposed algorithm for the GN-CEG model on a synthetic dataset for a bushing system and a real dataset for a selected conservator system. We will also briefly discuss the evaluation of the text processing algorithms proposed in Section 3.2 for the selected real dataset. Sensitivity analyses and the evaluation of the Gibbs algorithm are then performed for these examples to show that this algorithm provides an effective way to estimate the parameters.

5.1 Learning a CEG with bespoke causal algebras

5.1.1 Learning effects from a remedial intervention

In previous chapters, we implicitly assumed a causal order of events modelled on the tree. Such order arranges the failure indicator as the last variable. With this

order, we interpret the conditional relationships causally and examine the effects of different interventions on machine's failure. When the events are ordered temporally along the root-to-leaf paths on the tree and this order can be interpreted causally, the event tree can be assumed to be a causal tree and the CEG derived from it is a causal CEG.

In the domain of reliability engineering, there are various types of data. Heterogeneity of data may affect the way we construct the event tree and the CEG. In terms of the maintenance logs of a system, the engineers wrote down the texts when a failure had happened or been observed. So the recorded events are conditioned on a failure. Therefore, if we aim to learn the best structure of a CEG that is most consistent with the data extracted from the maintenance logs, then the events modelled on the tree should start with the failure indicator and then followed by root causes and symptoms. We call such a tree the **learning tree** and the best scoring CEG derived from the learning tree the **learning CEG**.

(a) (b) (c) (d)

Figure 5.1: Some candidate BNs associated with the CEGs to be searched across.

Note that when using the MAP algorithm to select the best scoring CEG which introduces the failure indicator as the first component, we search over a class of models that is not equivalent to the class of models which introduces the failure indicator as the last component. We give a simple example to explain this point. Suppose we search over the CEGs associated with BNs with three variables X, Y, Z. Let X be the failure indicator. The BNs in Figures 5.1a-5.1c are in the equivalence class [Smith, 2010] and the corresponding staged trees are statistically equivalent [Görgen and Smith, 2018]. Figure 5.1d has a collider structure and belongs to a different class from the first three BNs. In this case, X and Y are not arranged as the last component on the associated CEGs. So if we fix the order so that the failure indicator X is the last component, then the CEGs associated with Figure 5.1d will not be considered. However, this class of models will be considered as candidate models if X is assumed to be the first component. Therefore, the MAP algorithm searches across two different classes of models for the two different orderings, although the two classes of models share many equivalent models. The class of CEGs that provide the most interpretable descriptions of what might happen are the ones where the failure event comes last. The best scoring learning CEG is not chosen from the optimal class of models for causal interpretation but as explained above our data is conditioned on a failure has been observed. Therefore, in our

case the CEGs starting with failure indicators are the easiest class of CEGs to learn about and score without considering the missingness mentioned in Chapter 4. We can estimate the best CEG within this class simply in a conjugate way.

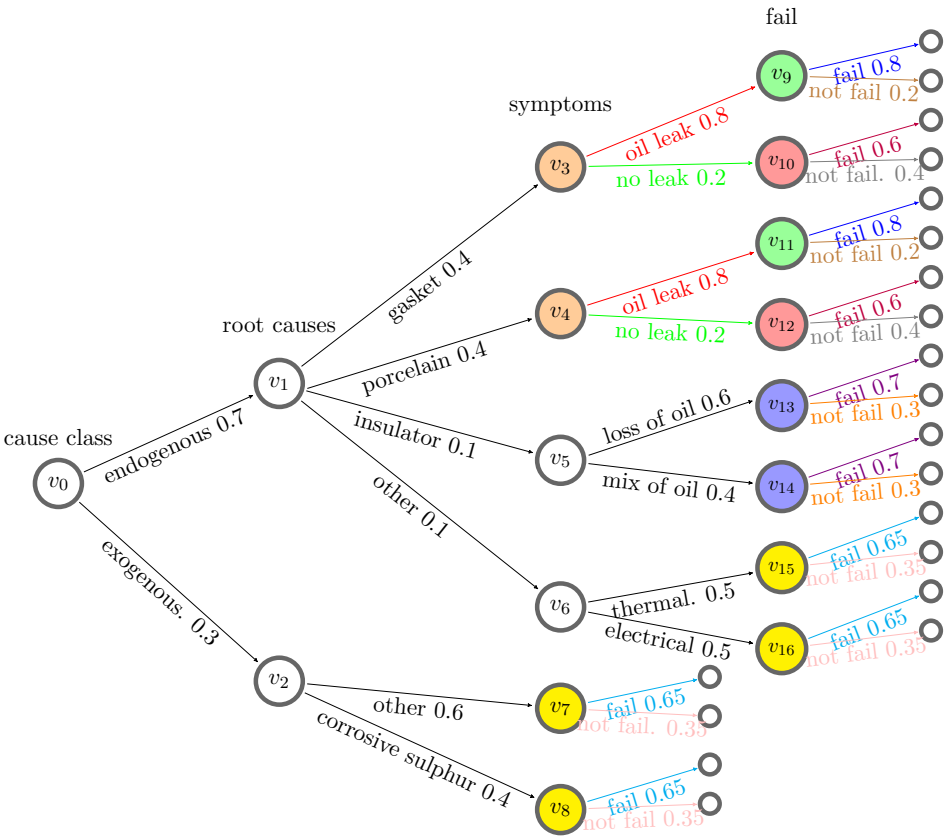


Figure 5.2: The causal staged tree for a bushing system.

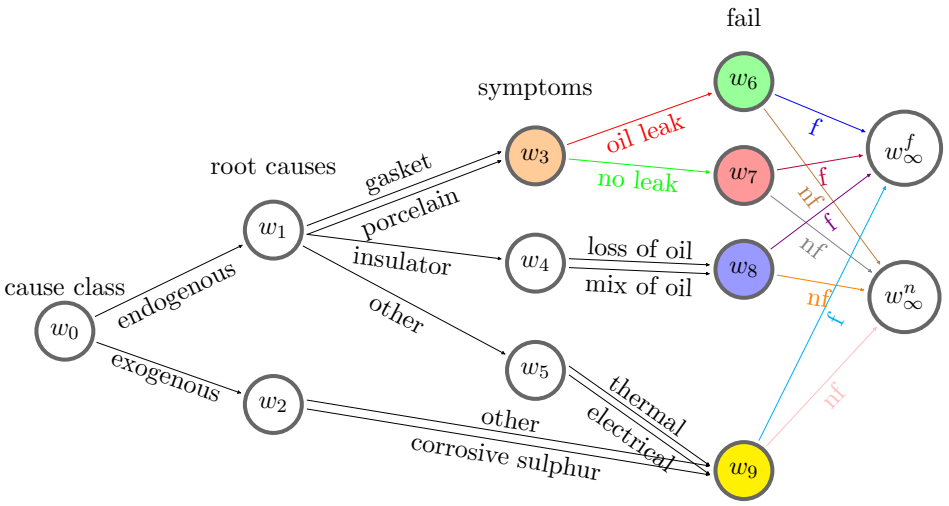


Figure 5.3: The causal CEG elicited from the staged tree in Figure 5.2. The label "f" refers to fail, "nf" refers to not fail.

We use a bushing example to demonstrate the difference between a learning tree and a causal tree. Suppose we have a staged tree for a bushing system in Figure 5.2 whose corresponding CEG is depicted in Figure 5.3. This CEG is similar to the one we plotted in Example 4 which was used to explain the stochastic manipulations on root causes in response to the remedial intervention. The events are arranged on the tree following the temporal order. We assume that Figure 5.2 is the causal staged tree for the selected bushing system, and Figure 5.3 is the idle CEG for causal inference. In order to design experiment for simulating synthetic data, we make further assumptions about the ground truth transition probabilities and label them on the corresponding edges in Figure 5.2. We can simply transform these probabilities onto the learning tree. The learning tree of this causal tree is plotted in Figure 5.4, where we move the failure indicator from the leaves of the tree to the root of the tree. The order of other events remains unchanged. The conditional probability associated with each edge in the learning tree can be evaluated from the underlying causal tree by simply applying the Bayes rule. For example, for edge e_{v_1,v_3} in the learning tree, we need to compute

$$\begin{aligned} p(\text{endogenous cause}|\text{fail}) &= \frac{p(\text{endogenous cause, fail})}{p(\text{fail})} \\ &= \frac{p(\text{fail}|\text{endogenous cause}) \times p(\text{endogenous cause})}{p(\text{fail})}, \end{aligned} \tag{5.1.1}$$

where $p(\text{fail}|\text{endogenous cause})$ and $p(\text{endogenous cause})$ can be directly read from

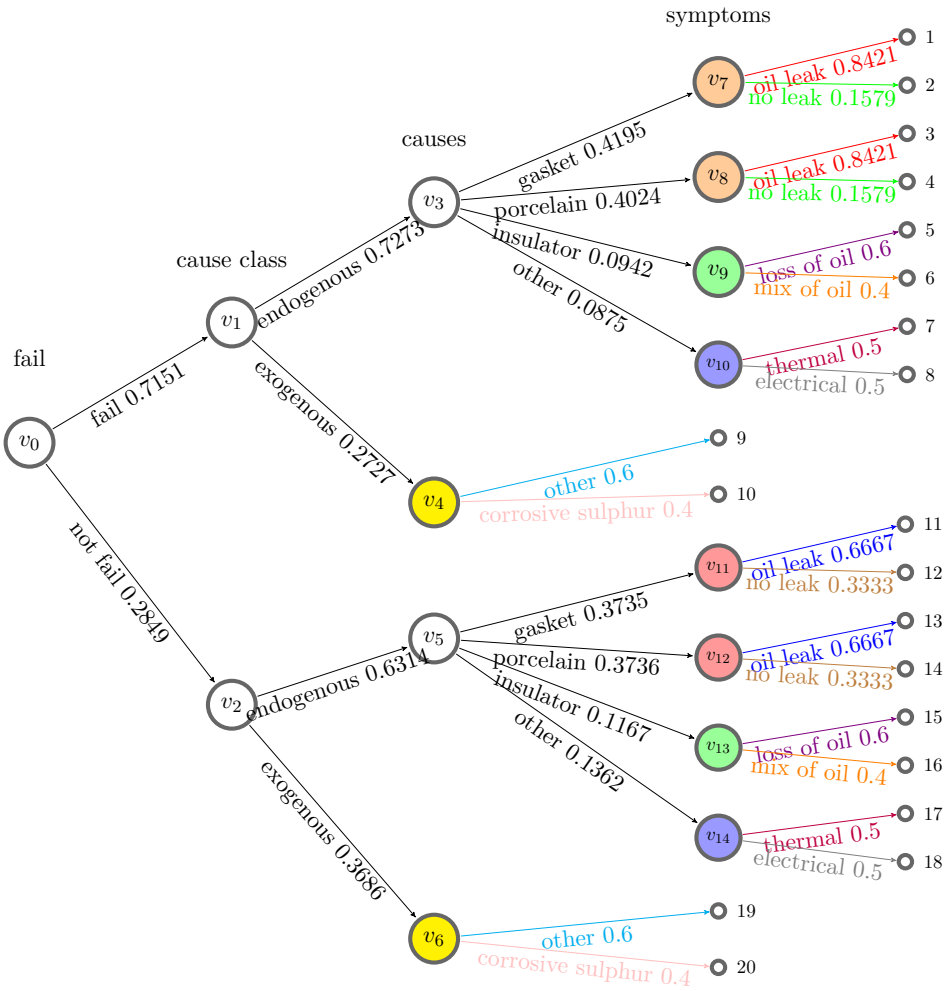


Figure 5.4: The staged tree transformed from Figure 5.2 for learning purpose.

the causal tree. Having all these conditional probabilities, we then check the stages in the learning tree. Note that the colours of the stages are reassigned in the learning tree and the colouring is not inherited by stages of the causal tree since stages will often be defined conditioning on variables taken in a different order. To keep the same paintbox we use the same sets of colours to distinguish the different stages within each of the models. For example, v_7, v_8, v_{15}, v_{16} in the causal staged tree in Figure 5.2 are coloured in yellow, whose florets represent conditional failure indicators. However, v_4, v_6 in the learning staged tree in Figure 5.4, which are also coloured in yellow, represent conditional symptoms caused by exogenous reasons. The learning CEG elicited from Figure 5.4 is depicted in Figure 5.5.

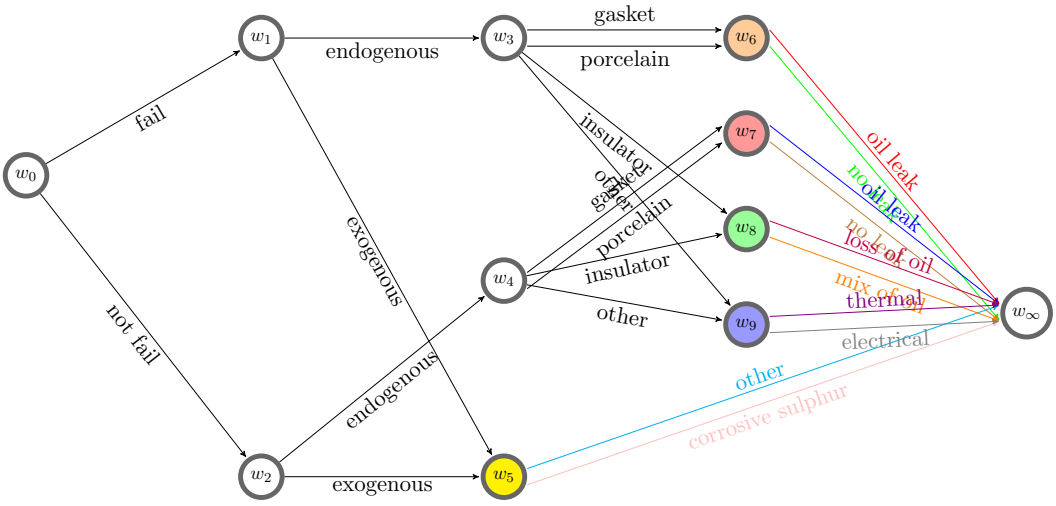


Figure 5.5: The CEG for the learning staged tree in Figure 5.4.

We next design experiments to show that when there is a remedial intervention, we can integrate its causal effects into the learning algorithm of the CEG and this can improve the parameter estimation. Suppose there is a stochastic manipulation on the endogenous root causes when the faulty gasket is replaced by a new one. We then reweigh the transitions along $e_{v_1,v_3}, e_{v_1,v_4}, e_{v_1,v_5}, e_{v_1,v_6}$ in the causal tree so that each edge is expected to be traversed with a new probability. Let the post-intervened conditional probabilities be 0.2, $\frac{8}{15}$, $\frac{2}{15}$, $\frac{2}{15}$ respectively. We can then compute the corresponding conditional probabilities for florets $\mathcal{F}(v_3)$ and $\mathcal{F}(v_5)$ for the ground truth learning tree.

We simulated a synthetic dataset D_0 from the ground truth tree in Figure 5.4 with 5000 cases. This dataset is composed of the processes conditional on failures. It can therefore be used to emulate the information extracted from the failure reports, and the processes conditional on the system being operational, which can be used to emulate the information collected from other sources. By selecting the cases which are conditioned on failures from D_0, we constructed a new dataset D_1. It consists of 3587 cases.

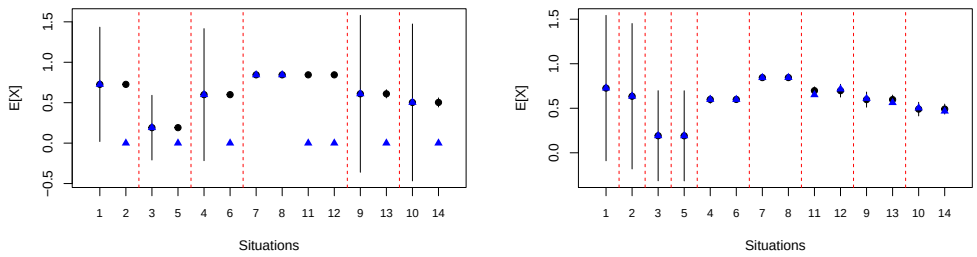
Following an established method first proposed by Heckerman et al. [1995] for learning BNs, Collazo et al. [2018] suggested treating each Dirichlet hyperparamter α_{uj} as the number of phantom units, which is believed to arrive at j^{th} child of stage u. Let the number of phantom units entering the root vertex be $\overline{\alpha} = 1$. We usually weigh the edges emanating from the same node equally likely. For example, there are 0.5 phantom units entering v_1 and 0.5 phantom units entering v_2. Let $\boldsymbol{\alpha}_0$ denote the set of Dirichlet hyperparameters for this system, which is unmanipulated. For

the intervened system, we manipulated the prior hyperparameters as suggested in Section 2.3 so that the weights assigned to the endogenous root causes are $3:8:2:2$. Let $\boldsymbol{\alpha}_1$ denote the set of Dirichlet hyperparaemters for the intervened system.

We ran a model search algorithm with respect to D_1 for $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_1$ respectively. The best scoring CEG learned without considering the manipulations, denoted by $\mathcal{C}_1$, is scored -7123.106. The best scoring CEG learned with the intervened priors, denoted by $\mathcal{C}_2$, is scored -7118.292. The latter is scored higher which means the data are better learned by the corresponding structure and the parameters are better estimated. To further compare $\mathcal{C}_1$ and $\mathcal{C}_2$, we computed the total situational error $\Xi(\mathcal{T})$:

$$\Xi(\mathcal{T}) = \sum_{v \in V_{\mathcal{T}}} ||\boldsymbol{\theta}_v^* - \tilde{\boldsymbol{\theta}}_v||_2. \tag{5.1.2}$$

This is a sum of the Euclidean distance between the true conditional probabilities $\boldsymbol{\theta}_v^*$ and the mean posterior probabilities $\tilde{\boldsymbol{\theta}}_v$ estimated on the best scoring model for all stages. The total situational error for the model learned from the intervened prior is 1.8×10^{-5} lower than the total situational error for the model learned from $\boldsymbol{\alpha}_0$. So the parameter learning is improved, but to a small extent in this example.



(a) For D_1.

(b) For D_0.

Figure 5.6: Leave-one-out stage monitor.The blue points are observed means of the situation labelled on the x-axis. The black points are the posterior means of the corresponding stages. The black lines are the two standard deviations from the posterior means when leaving the situation out. Each red dashed line split the situations by stages.

We further examined the selected model $\mathcal{C}_2$ by performing the leave-one-out diagnosis on stages [Wilkerson and Smith, 2019]. Figure 5.6a plots the leave-one-out diagnostic results. This plot checks whether the observed mean of the left out situation lies within the standard deviations of the posterior mean of the stage after leaving it out of this stage. The red dashed vertical lines split the situations into different stages according to the model selection result so that situations in the same

block are in the same stage. We can see that the empirical mean for situations v_2, $v_5, v_6, v_{11}, v_{12}, v_{13}, v_{14}$ fall out of the range bounded by standard deviations. And the stages for v_2, v_5, v_{11}, v_{12} are misspecified. This means the staging of these situations are not well supported by the data. This is not surprising because D_1 only provides data conditional on failures so there is no observed path passing through v_2. We can instead run the algorithm with the intervened prior for the full dataset D_0. The diagnosis from leave-one-out stage monitor is shown in Figure 5.6b. The stages are correctly learned and no blue vertex fall out of the black lines. The total situational error for this scenario is 0.175 and the score of the best model is -13198.56. If we perform the learning algorithm with the unmanipulated prior, then the total situation error is 0.244 and score is -13201.92 for the best scoring model. This reveals the advantage of integrating the manipulations for predictive inference. Having the estimated mean posterior probabilities on the learning tree, we can then transform these conditional probabilities back onto the causal tree for a causal analysis.

Furthermore, we can check whether incorporating the effect of the intervention on time-to-failure gives a more accurate prediction of the residual lifetime of the maintained equipment. Suppose the equipment had been used for $\tau = 10$ weeks before the maintenance and the maintenance prolonged the lifetime of the equipment. In particular, let $\xi = 0.5$ so that the virtual age of the equipment after the maintenance was $\xi\tau = 5$ weeks.

We first simulated the "real" residual lifetime for each case in D_1. Assume $Weibull(0.7, 3)$ is the idle lifetime sampling distribution, *i.e.* the lifetime for a new system. After carrying out the maintenance described above, the residual lifetime should then be sampled from the left-truncated $Weibull(0.7, 3)$ so that the sampled lifetime is greater than 5 weeks.

Using conjugate inference with respect to the lifetime for the idle system so that:

$$H_\lambda \sim \text{Weibull}(0.7, \eta_\lambda) \text{ and } \eta_\lambda \sim \text{InverseGamma}(2, 3). \tag{5.1.3}$$

For the intervened system, the likelihood is the conditional Weibull as specified in equation (2.6.10).

The log-likelihood score associated with lifetime using equation (5.1.3) is -14288.02 while the score associated with the left-truncated lifetime is -12856.21 for this dataset. The proposed conditional Weibull distribution better captures the feature of the residual lifetime of the equipment after maintenance. We can measure

the prediction error of the total path time by

$$\iota(\mathcal{T}) = \sqrt{\sum_{\lambda \in \Lambda_{\mathcal{T}}} (\hat{H}_\lambda - H^*_\lambda)^2}, \tag{5.1.4}$$

where $\hat{H}_\lambda$ is the empirical mean of the lifetime of equipment whose associated path is λ while H^*_λ denotes the estimated mean of the lifetime using the posterior mean probabilities. The prediction error $\iota(\mathcal{T})$ is 17.5 for using the lifetime distribution in equation (2.6.10), while the error is larger when using the idle lifetime distribution, which is 21.14. Thus, considering the effects of the intervention on the path time can improve the estimation of the posterior distribution of lifetime of the system.

5.1.2 Learning with effects from a routine intervention

In our recent work [Yu and Smith, 2021*b*], we designed a comparative study to examine how integrating the stochastic manipulation induced by the routine intervention can improve the prediction of failure. Here we summarise the result of this comparative study. We used an example of a conservator system. This was similar to Example 2, except that the symptom variable here was split into two indicator variables for oil leak and alarm respectively.

Suppose there was a routine intervention which cleaned oil leak, checked oil level and toppled up the oil but still did not fully prevent the oil leak. We made the following assumptions for this example. Assume that

1. the tree in Figure 5.7 is the ground truth staged tree with informed missingness
2. Dirichlet priors are independent,see Section 4.1.2,
3. all pieces of equipment in the dataset were intervened by the same routine maintenance,
4. a complete and unique root-to-sink path on the tree can be identified for each case in the synthetic dataset,
5. we were given the estimated posteriors from the past failure data before maintenance, these were then used as priors to generate the data that would be observed after the routine maintenance, where the florets $\mathcal{F}(w_2), \mathcal{F}(w_3), \mathcal{F}(w_4)$ are stochastically manipulated in response to the routine maintenance.

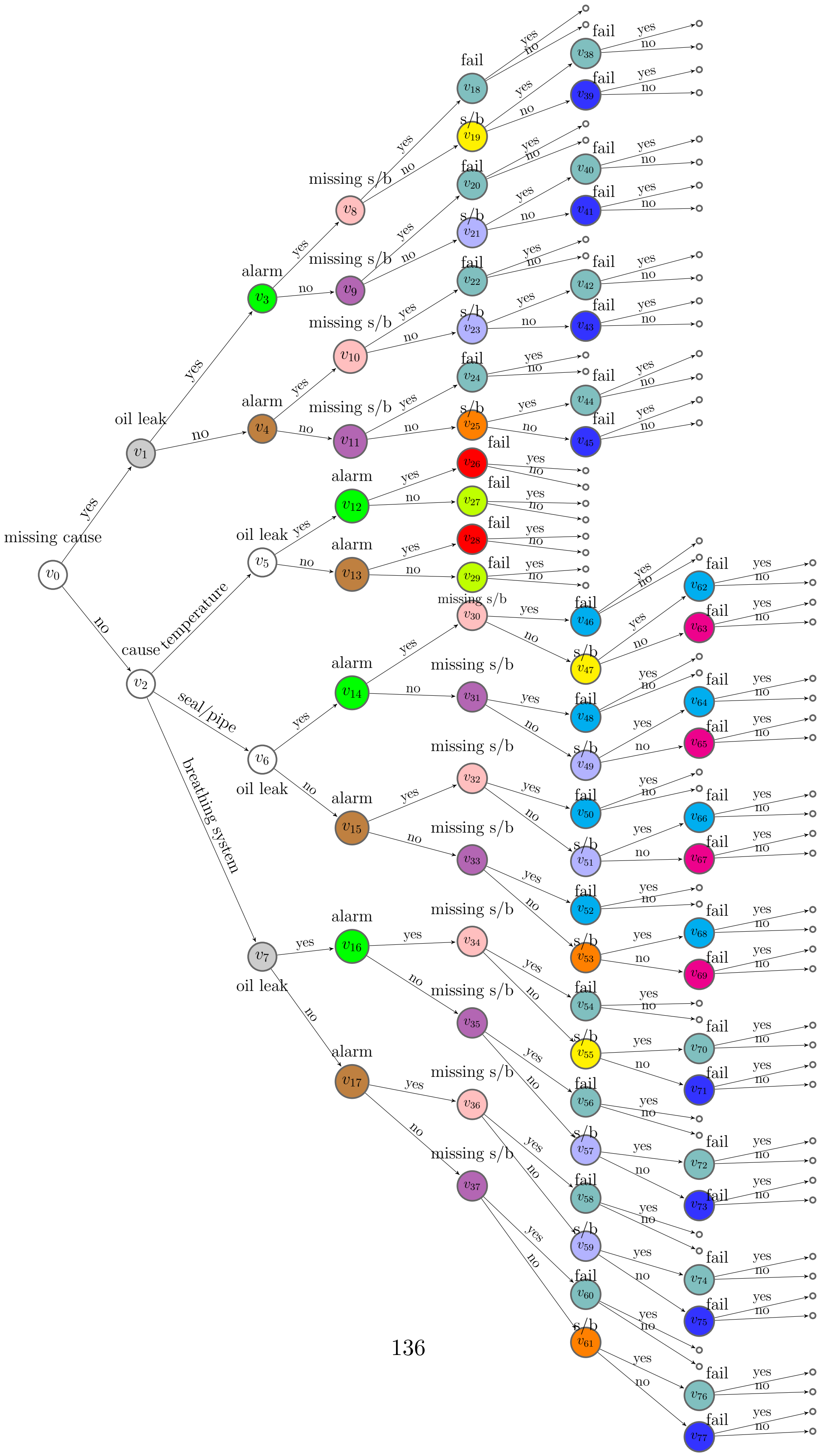
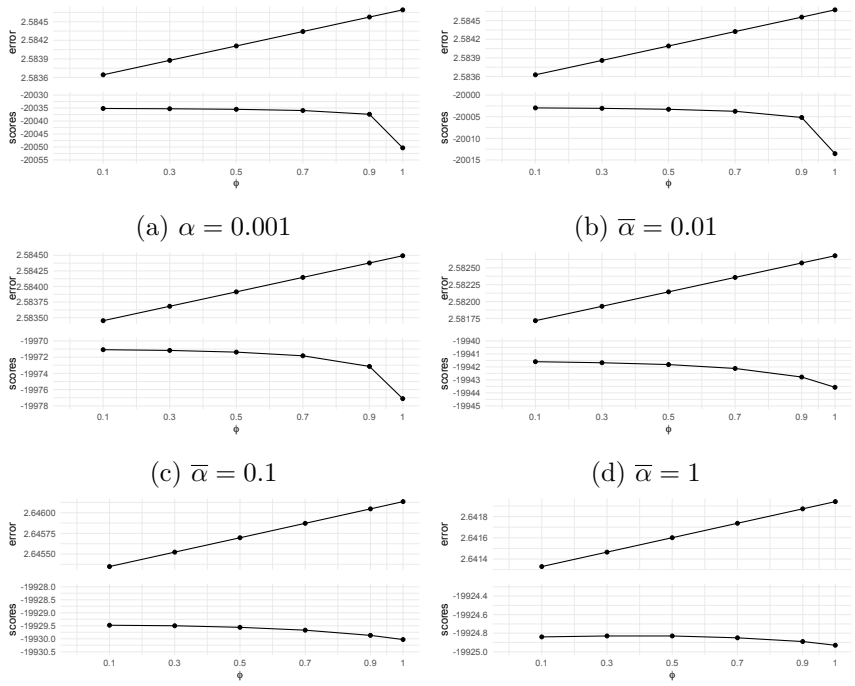


Figure 5.7: A missingness staged tree.



(a) $\alpha = 0.001$ (b) $\overline{\alpha} = 0.01$

(c) $\overline{\alpha} = 0.1$ (d) $\overline{\alpha} = 1$

(e) $\overline{\alpha} = 3$ (f) $\overline{\alpha} = 5$

Figure 5.8: Comparing situational errors and MAP scores for the best scoring models selected to fit D_2 [Yu and Smith, 2021*a*]. The x-axis of each plot is labelled by different values of ϕ, where $\phi = 1$ refers to the case when no manipulation is imported to the prior. Each plot displays results for a specified total phantom number α.

Based on these assumptions we simulated 5000 cases to construct another dataset for the intervened system, denoted by D_2. For a sensitivity analysis, we set different sizes of the total phantom units which enters the root vertex v_0: $\overline{\alpha} = 0.001, \overline{\alpha} = 0.01, \overline{\alpha} = 0.1, \overline{\alpha} = 1, \overline{\alpha} = 3, \overline{\alpha} = 5$.

In Section 2.4, we defined ϕ to add uncertainties to the intervened floret distributions in response to the routine intervention. We set different values of ϕ to compare the estimates on the best scoring M-CEG: $\phi = 0.1$, $\phi = 0.3$, $\phi = 0.5$, $\phi = 0.7$, $\phi = 0.9$, and $\phi = 1$. When $\phi = 1$, the priors were unmanipulated.

We ran the MAP structural learning algorithm for each scenario and assessed the selected structures and the estimated parameters by comparing MAP model scores and total situational errors $\Xi(\mathcal{T})$.

The diagnostic results were given in Figure 5.8. The total situational errors were displayed in the upper panel of each plot. The MAP scores for the best scoring models for different values of ϕ were displayed in the lower panel of each plot. When $\phi = 0.1$, the total situational error was the smallest compared to other values of

ϕ for all $\overline{\alpha}$ we chose. By contrast, $\phi = 1$ gave the largest total situational error. So incorporating the effects of the routine intervention into the learning algorithm can improve the estimation of transition probabilities which can then be used for prediction. The results of model scores were consistent with this conclusion, where $\phi = 0.1$ gave the highest model score while $\phi = 1$ gave the lowest.

5.2 A model demonstration of learning the GN-CEG

Given the framework of the GN-CEG model defined in Section 3.3, in this section, we give an example of the algorithm for learning the latent paths and the parameters. This algorithm needs to be applied only after having extracted the core events and constructed the core event variables on the GN. So in this sense we only focus on learning the latent paths for the observed ordered core event variables lying on the observation layer, *i.e.* the GN.

In this case, the observations are the core event variables and the time-to-failures. Recall that in Chapter 3, we let $\boldsymbol{l}_d = \{l_{d_1}, \cdots, l_{d_m}\}$ denote the observed values of the m core event variables for the d^{th} document and $h_d > 0$ denote the observed lifetime for the process recorded in this document. So the observations are denoted by $O_D = \{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}$ for D documents.

Next we list the hidden variables or parameters in the hierarchical model and specify the corresponding prior distributions. Firstly, the values of the floret variables are latent, *i.e.* the positions on the tree are latent states. Let $\boldsymbol{w}_d = (w_{d_j})_{w_{d_j}\in W_{\lambda_d}}$ denote the sequence of positions traversed by the latent path $\lambda_d \in \Lambda_{\mathcal{C}}$ for the d^{th} document. There could be two edges emanating from the same position and received by the same position. So we can also use the edges to represent the latent states. Let $\boldsymbol{e}_d = (e_{w,w'})_{e_{w,w'}\in E_{\lambda_d}}$ denote the sequence of edges along the latent path λ_d. Then for the whole dataset, $\{\boldsymbol{w}_d\}_{d\in\{1,\cdots,D\}}$ and $\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}$ are latent.

The transition probability along each edge on the tree is also unknown. We can let all documents share the same set of transition probabilities, denoted by $\{\boldsymbol{\theta}_w\}_{w\in W}$. An alternative hypothesis is to assume that each document is associated with a unique system so that each system has its own set of transition probabilities, denoted by $\{\boldsymbol{\theta}_w^d\}_{w\in W, d\in\{1,\cdots,D\}}$. By assuming Dirichlet prior independence, we can define a Dirichlet prior for each $\boldsymbol{\theta}_w^d$ with hyperparameters $\boldsymbol{\alpha}_w$ so that

$$\boldsymbol{\theta}_w^d \sim \text{Dirichlet}(\boldsymbol{\alpha}_w). \tag{5.2.1}$$

Given the ordered core event variables $\boldsymbol{L}_d = \{L_{d_1}, \cdots, L_{d_m}\}$ for the d^{th} document, unless the size of set of core event variables is not big and we have

other prior information knowledge, we may not know which of them are in the same community. Then we introduce a vector $\boldsymbol{a}_d = \{a_{L_{d_1}}, \cdots, a_{L_{d_m}}\}$ for each $L_{d_j} \in \boldsymbol{L}_d$ as the boundary indicators. Then $a_{L_{d_j}} = 1$ means $w_{d_j} \neq w_{d_{j-1}}$ and there is a transition from $w_{d_{j-1}}$ to w_{d_j} along the edge $e_{w_{d_{j-1}}, w_{d_j}} \in E_{\mathcal{C}}$ on the tree. Otherwise, when $a_{L_{d_j}} = 0$, $w_{d_j} = w_{d_{j-1}}$ so that l_{d_j} lies in the same sub-community as $l_{d_{j-1}}$. So each $a_{L_{d_j}}$ is a binary variable. When the value of this indicator is unknown, we assume that it is drawn from a Bernoulli distribution:

$$a_{L_{d_j}} \sim \text{Bernoulli}(\gamma_{L_{d_j}}). \tag{5.2.2}$$

This gives a new set of parameters $\boldsymbol{\gamma} = \{\gamma_L\}_{L \in \boldsymbol{L}_D}$, where $\boldsymbol{L}_D$ is the whole set of core event variables on the GN. Since Bernoulli is conjugate to Beta distribution, we simply assign a Beta prior to each γ_L by

$$\gamma_L \sim \text{Beta}(r_1, r_2), \tag{5.2.3}$$

where the hyperparameters are $r_1, r_2 > 0$.

The emission probability $p(l|pa^{\mathcal{H}}(l), Dsup^{\mathcal{H}}(l))$ also needs to be specified. Recall that $pa^{\mathcal{H}\downarrow}(L) = \{pa^{\mathcal{H}}(L), Dsup^{\mathcal{H}}(L)\}$ is the flattened parent of L. The direct superior $Dsup^{\mathcal{H}}(L)$, *i.e.* the latent state of the observed core event, depends on the assignment $\mathcal{H}$. Given different $\mathcal{H}$, the flattened parent may differ. If L takes values in $\mathbb{L} = \{l_1, \cdots, l_n\}$, and the flattened parent is $pa^{\mathcal{H}\downarrow}(L)$, then let $\boldsymbol{\phi}_L^{pa^{\mathcal{H}\downarrow}(L)} = \{\phi_{l_1}^{pa^{\mathcal{H}\downarrow}(L)}, \cdots, \phi_{l_n}^{pa^{\mathcal{H}\downarrow}(L)}\}$ so that

$$p(L = l_i | pa^{\mathcal{H}\downarrow}(L), Dsup^{\mathcal{H}\downarrow}(L)) = \phi_{l_i}^{pa^{\mathcal{H}\downarrow}(L)}. \tag{5.2.4}$$

Let $\sum_{i=1}^{n} \phi_{l_i}^{pa^{\mathcal{H}\downarrow}(L)} = 1$ and $\phi_{l_i}^{pa^{\mathcal{H}\downarrow}(L)} > 0$. If the emission probabilities are unknown, we then assume they are drawn from a Multinomial distribution with a Dirichlet prior. Thus,

$$l_i | pa^{\mathcal{H}\downarrow}(L) \sim \text{Multinomial}(\boldsymbol{\phi}_L^{pa^{\mathcal{H}\downarrow}(L)}), \tag{5.2.5}$$

and

$$\boldsymbol{\phi}_L^{pa^{\mathcal{H}\downarrow}(L)} \sim \text{Dirichlet}(\boldsymbol{\nu}_L^{pa^{\mathcal{H}\downarrow}(L)}), \tag{5.2.6}$$

where the hyperparameters are $\boldsymbol{\nu}_L^{pa^{\mathcal{H}\downarrow}(L)} = \{\nu_{l_1}^{pa^{\mathcal{H}\downarrow}(L)}, \cdots, \nu_{l_n}^{pa^{\mathcal{H}\downarrow}(L)}\}$. Let $\boldsymbol{\phi} = \{\boldsymbol{\phi}_L^{pa^{\mathcal{H}\downarrow}(L)}\}_{L \in \boldsymbol{L}_D, \mathcal{H}}$ denote the set of emission probabilities for all core event variables and the corresponding flattened parents. Let $\boldsymbol{\nu} = \{\boldsymbol{\nu}_L^{pa^{\mathcal{H}\downarrow}(L)}\}_{L \in \boldsymbol{L}_D, \mathcal{H}}$ denote the set of corresponding Dirichlet hyperparameters which need to be specified.

Here for modelling the lifetime, we use the Weibull path time as an example.

This can be replaced by the gamma distributed path time as discussed in Section 2.6. For the d^{th} document with latent path λ_d, let

$$h_d \sim \text{Weibull}(\beta_{\lambda_d}, \eta_{\lambda_d}), \tag{5.2.7}$$

where $\beta_{\lambda_d} > 0$ is the shape parameter which needs to be specified for each path and $\eta_{\lambda_d} > 0$ is the scale parameter with an inverse-Gamma prior. Let

$$\eta_{\lambda_d} \sim \text{InverseGamma}(\zeta_{\lambda_d}, \mu_{\lambda_d}), \tag{5.2.8}$$

where μ_{λ_d} is the scale parameter and ζ_{λ_d} is the shape parameter.

To summarise, we have

- a set of observations:
$$O_D = \{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}; \tag{5.2.9}$$
- a set of hidden variables or parameters:
$$Z = \{\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{\theta}_w^d\}_{w\in W, d\in\{1,\cdots,D\}}, \{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}, \boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\eta}\}; \tag{5.2.10}$$
- a collection of hyperparameters:
$$\Omega = \{\boldsymbol{\alpha}, r_1, r_2, \boldsymbol{\nu}, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta}\}. \tag{5.2.11}$$

5.2.1 The general process

Given the parameters and distributions specified above, we now set out a general process for learning a latent path on the CEG.

1. For every core event variable $L \in \boldsymbol{L}_D$,
   (a) sample the parameter for the boundary indicator by
   $$\gamma_L \sim \text{Beta}(r_1, r_2), \tag{5.2.12}$$
   (b) for every possible flattened parent of L, $pa^{\mathcal{H}\downarrow}(L)$, sample the emission probabilities by
   $$\boldsymbol{\phi}_L^{pa^{\mathcal{H}\downarrow}(L)} \sim \text{Dirichlet}(\boldsymbol{\nu}_L^{pa^{\mathcal{H}\downarrow}(L)}). \tag{5.2.13}$$
2. For every path $\lambda \in \Lambda_{\mathcal{C}}$, sample
$$\eta_\lambda \sim \text{InverseGamma}(\zeta_\lambda, \mu_\lambda). \tag{5.2.14}$$

3. For every document $d \in \{1, \cdots, D\}$,

   (a) for each position $w \in W$, sample the transition probabilities by

$$\boldsymbol{\theta}_w^d \sim \text{Dirichlet}(\boldsymbol{\alpha}_w). \tag{5.2.15}$$

   (b) for each observed core event variable in this document $l_{d_j} \in \boldsymbol{l}_d$,

      i. sample the boundary indicator from

$$a_{L_{d_j}} \sim \text{Bernoulli}(\gamma_{L_{d_j}}), \tag{5.2.16}$$

      if $a_{L_{d_j}} = 0$, then no transition is made, thus

$$w_{d_j} = w_{d_{j-1}}, e_{d_j} = e_{d_{j-1}}; \tag{5.2.17}$$

      otherwise, $a_{L_{d_j}} = 1$ and sample a new transition by

$$e_{d_j} \sim \text{Multinomial}(\boldsymbol{\theta}_{w_{d_j-1}}), \tag{5.2.18}$$

      where the receiving node of e_{d_j} is w_{d_j};

      ii. given $e_{d_j}, a_{L_{d_j}}$, sample the observed value of the core event variable from

$$l_{d_j} \sim \text{Multinomial}(\boldsymbol{\phi}_{L_{d_j}}^{pa^{\mathcal{H}_\downarrow}(L_{d_j})}). \tag{5.2.19}$$

   (c) given the sampled edges $\boldsymbol{e}_d$, let $\tilde{\boldsymbol{e}}_d$ be the unique edges in $\boldsymbol{e}_d$, the path λ_d for this document satisfies $E_{\lambda_d} = \tilde{\boldsymbol{e}}_d$. Draw the total time from

$$h_d \sim \text{Weibull}(\beta_{\lambda_d}, \eta_{\lambda_d}). \tag{5.2.20}$$

5.2.2 The parameter learning algorithm

For the d^{th} document, the complete data likelihood is

$$\begin{aligned}
&p(\boldsymbol{l}_d, h_d, \boldsymbol{e}_d, \boldsymbol{\theta}^d, \boldsymbol{a}_d, \boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\eta} | \boldsymbol{\alpha}, r_1, r_2, \boldsymbol{\nu}, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta}) \\
&= p(\boldsymbol{l}_d, h_d | \boldsymbol{e}_d, \boldsymbol{\theta}^d, \boldsymbol{a}_d, \boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\beta}) \times p(\boldsymbol{e}_d, \boldsymbol{\theta}^d, \boldsymbol{a}_d, \boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\eta} | \boldsymbol{\alpha}, r_1, r_2, \boldsymbol{\nu}, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta}) \\
&= p(h_d | \boldsymbol{e}_d, \boldsymbol{\eta}, \boldsymbol{\beta}) \times p(\boldsymbol{l}_d | \boldsymbol{e}_d, \boldsymbol{\phi}) \times p(\boldsymbol{e}_d | \boldsymbol{\theta}^d, \boldsymbol{a}_d) \times p(\boldsymbol{\theta}^d | \boldsymbol{\alpha}) \times p(\boldsymbol{a}_d | \boldsymbol{\gamma}) \times \\
&\quad p(\boldsymbol{\gamma} | r_1, r_2) \times p(\boldsymbol{\phi} | \boldsymbol{\nu}) \times p(\boldsymbol{\eta} | \boldsymbol{\zeta}, \boldsymbol{\mu}) \\
&= p(h_d | \boldsymbol{e}_d, \boldsymbol{\eta}, \boldsymbol{\beta}) \times p(\boldsymbol{\eta} | \boldsymbol{\zeta}, \boldsymbol{\mu}) \times p(\boldsymbol{\phi} | \boldsymbol{\nu}) \times p(\boldsymbol{\gamma} | r_1, r_2) \times p(\boldsymbol{\theta}^d | \boldsymbol{\alpha}) \times \\
&\prod_{j=1}^{m_d} \{ p(l_{d_j} | \boldsymbol{\phi}_{L_{d_j}}^{pa^{\mathcal{H}_\downarrow}(L_{d_j})}) \times p(e_{w_{d_{j-1}}, w_{d_j}} | \boldsymbol{\theta}_{w_{d_{j-1}}}^d, a_{d_j}) \times p(a_{L_{d_j}} | \gamma_{L_{d_j}}) \}.
\end{aligned} \tag{5.2.21}$$

Marginalising out $\boldsymbol{e}_d$ and $\boldsymbol{a}_d$, then we have

$$\begin{aligned}
&p(\boldsymbol{l}_d, h_d|\boldsymbol{\theta}^d, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \\
=&\sum_{\lambda_d} p(\boldsymbol{l}_d, h_d, \lambda_d|\boldsymbol{\theta}^d, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \\
=&\sum_{\lambda_d} p(h_d|\eta_{\lambda_d}, \beta_{\lambda_d}) \times p(\boldsymbol{l}_d, \lambda_d|\boldsymbol{\theta}^d, \boldsymbol{\phi}, \boldsymbol{\gamma}) \\
=&\sum_{\lambda_d} p(h_d|\eta_{\lambda_d}, \beta_{\lambda_d}) \times \prod_{j=1}^{m_d} \sum_{a_{L_{d_j}} \in \{0,1\}} p(l_{d_j}|\boldsymbol{\phi}_{L_{d_j}}^{pa^{\mathcal{H}\downarrow}(L_{d_j})}) \times p(e_{w_{d_{j-1}}, w_{d_j}}|\boldsymbol{\theta}^d_{w_{d_{j-1}}}, a_{L_{d_j}}) \times p(a_{L_{d_j}}|\gamma_{L_{d_j}}).
\end{aligned} \tag{5.2.22}$$

The joint distribution has the following expression.

$$\begin{aligned}
&p(\{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\alpha}, r_1, r_2, \boldsymbol{\nu}, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta}) \\
=&\underbrace{p(\{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \boldsymbol{\nu}, r_1, r_2, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta})}_{(1)} \times \underbrace{p(\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\alpha}, r_1, r_2)}_{(2)}.
\end{aligned} \tag{5.2.23}$$

The component labelled by (1) in this expression can be written as:

$$\begin{aligned}
&p(\{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \boldsymbol{\nu}, r_1, r_2, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta}) \\
=&\int p(\{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\beta}) p(\boldsymbol{\phi}|\boldsymbol{\nu}) p(\boldsymbol{\eta}|\boldsymbol{\zeta}, \boldsymbol{\mu})\, d\boldsymbol{\phi} d\boldsymbol{\eta} \\
=&\underbrace{\int p(\{\boldsymbol{l}_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \boldsymbol{\phi}) p(\boldsymbol{\phi}|\boldsymbol{\nu})\, d\boldsymbol{\phi}}_{(I)} \underbrace{\int p(\{h_d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\eta}, \boldsymbol{\beta}) p(\boldsymbol{\eta}|\boldsymbol{\zeta}, \boldsymbol{\mu})\, d\boldsymbol{\eta}}_{(II)}.
\end{aligned} \tag{5.2.24}$$

We next compute the components (I) and (II) separately. Beginning with the probability $p(\{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\beta})$.

$$
\begin{aligned}
& p(\{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\beta}) \\
&= \prod_{d=1}^{D} p(\boldsymbol{l}_d, h_d|\boldsymbol{e}_d, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\beta}) \\
&= \prod_{d=1}^{D} p(\boldsymbol{l}_d|\boldsymbol{e}_d, \boldsymbol{\phi}) p(h_d|\boldsymbol{e}_d, \boldsymbol{\eta}, \boldsymbol{\beta}) \\
&= \prod_{d=1}^{D} \left(\prod_{j=1}^{m} p(l_{d_j}|\boldsymbol{\phi}_{L_{d_j}}^{pa^{\mathcal{H}\downarrow}(L_{d_j})}) \right) \times p(h_d|\eta_{\lambda_{\tilde{\boldsymbol{e}}_d}}, \beta_{\lambda_{\tilde{\boldsymbol{e}}_d}}) \\
&= \prod_{d=1}^{D} \left(\prod_{j=1}^{m_d} \phi_{l_{d_j}}^{pa^{\mathcal{H}\downarrow}(L_{d_j})} \right) \times \frac{\beta_{\lambda_d}}{\eta_{\lambda_d}} h_d^{\beta_{\lambda_d}-1} exp(-\frac{h_d^{\beta_{\lambda_d}}}{\eta_{\lambda_d}}),
\end{aligned}
\tag{5.2.25}
$$

where $\tilde{\boldsymbol{e}}_d$ are the unique edges in $\boldsymbol{e}_d$, $\lambda_{\tilde{\boldsymbol{e}}_d}$ denotes the root-to-sink path which is composed of $\tilde{\boldsymbol{e}}_d$, and $\lambda_d = \lambda_{\tilde{\boldsymbol{e}}_d}$. Let $n_{l_{d_j}}^{pa^{\mathcal{H}\downarrow}(L_{d_j})}$ denote the number of times l_{d_j} is observed given the flattened parent $pa^{\mathcal{H}\downarrow}(L_{d_j})$. Let n_{λ_d} denote the number of documents whose latent paths are λ_d. Then the above probability can be re-expressed as

$$
\begin{aligned}
& p(\{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\beta}) \\
&= \underbrace{\left(\prod_{l_j\in\mathbb{L}_D} \prod_{pa_j\in pa^{\mathcal{H}\downarrow}(L_j)} (\phi_{l_j}^{pa_j})^{n_{l_j}^{pa_j}} \right)}_{p(\{\boldsymbol{l}_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}},\boldsymbol{\phi})} \times \underbrace{\prod_{\lambda\in\Lambda_{\mathcal{C}}} \left(\left(\frac{\beta_\lambda}{\eta_\lambda}\right)^{n_\lambda} \prod_{d:\lambda_d=\lambda} h_d^{\beta_\lambda-1} exp(-\frac{h_d^{\beta_\lambda}}{\eta_\lambda}) \right)}_{p(\{h_d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\eta},\boldsymbol{\beta})}.
\end{aligned}
\tag{5.2.26}
$$

Thus, the probability (I) can be written as:

$$
\begin{aligned}
(I) &= \int \left(\prod_{l_j\in\mathbb{L}_D} \prod_{pa_j\in pa^{\mathcal{H}\downarrow}(L_j)} (\phi_{l_j}^{pa_j})^{n_{l_j}^{pa_j}} \right) \times \left(\prod_{pa_j\in pa^{\mathcal{H}\downarrow}(L_j), L_j\in\boldsymbol{L}_D} \frac{1}{B(\boldsymbol{\nu}_{L_j}^{pa_j})} \prod_{l_j\in\mathbb{L}_j} (\phi_{l_j}^{pa_j})^{\nu_{l_j}^{pa_j}-1} \right) d\boldsymbol{\phi} \\
&= \int \prod_{pa_j\in pa^{\mathcal{H}\downarrow}(L_j), L_j\in\boldsymbol{L}_D} \frac{1}{B(\boldsymbol{\nu}_{L_j}^{pa_j})} \prod_{l_j\in\mathbb{L}_j} (\phi_{l_j}^{pa_j})^{n_{l_j}^{pa_j}+\nu_{l_j}^{pa_j}-1} d\boldsymbol{\phi} \\
&= \prod_{pa_j\in pa^{\mathcal{H}\downarrow}(L_j), L_j\in\boldsymbol{L}_D} \frac{B(\boldsymbol{n}_{L_j}^{pa_j} + \boldsymbol{\nu}_{L_j}^{pa_j})}{B(\boldsymbol{\nu}_{L_j}^{pa_j})},
\end{aligned}
\tag{5.2.27}
$$

where $\boldsymbol{n}_{L_j}^{pa_j} = \{n_{l_j}^{pa_j}\}_{l_j \in \mathbb{L}_j}$ with state space $\mathbb{L}_j$ for the core event variable L_j, and $\mathbb{L}_D = \{\mathbb{L}_d\}_{d \in \{1,\cdots,D\}}$. Here $B(\boldsymbol{\nu}_{L_j}^{pa_j}) = \frac{\prod_{l_j \in \mathbb{L}_j} \Gamma(\nu_{l_j}^{pa_j})}{\Gamma(\sum_{l_j \in \mathbb{L}_j} \nu_{l_j}^{pa_j})}$ is the beta function.

The probability (II) can be written as:

$$
\begin{aligned}
(II) &= \int \prod_{\lambda \in \Lambda_{\mathcal{C}}} \left(\left(\frac{\beta_\lambda}{\eta_\lambda} \right)^{n_\lambda} \prod_{d:\lambda_d = \lambda} h_d^{\beta_\lambda - 1} exp(-\frac{h_d^{\beta_\lambda}}{\eta_\lambda}) \right) \prod_{\lambda \in \Lambda_{\mathcal{C}}} \frac{\mu_\lambda^{\zeta_\lambda}}{\Gamma(\zeta_\lambda)\eta_\lambda^{\zeta_\lambda+1}} exp(-\frac{\mu_\lambda}{\eta_\lambda})\, d\boldsymbol{\eta} \\
&= \int \left(\prod_{\lambda \in \Lambda_{\mathcal{C}}} \frac{\beta_\lambda^{n_\lambda} \mu_\lambda^{\zeta_\lambda}}{\Gamma(\zeta_\lambda)} \prod_{d:\lambda_d = \lambda} h_d^{\beta_\lambda - 1} \right) \left(\prod_{\lambda \in \Lambda_{\mathcal{C}}} \frac{1}{\eta_\lambda^{\zeta_\lambda + n_\lambda + 1}} exp(-\frac{\mu_\lambda + \sum_{d:\lambda_d=\lambda} h_d^{\beta_\lambda}}{\eta_\lambda}) \right) d\boldsymbol{\eta} \\
&= \prod_{\lambda \in \Lambda_{\mathcal{C}}} \frac{\beta_\lambda^{n_\lambda} \mu_\lambda^{\zeta_\lambda} \Gamma(\zeta_\lambda + n_\lambda)}{\Gamma(\zeta_\lambda)(\mu_\lambda + \sum_{d:\lambda_d=\lambda} h_d^{\beta_\lambda})^{\zeta_\lambda + n_\lambda}} \prod_{d:\lambda_d=\lambda} h_d^{\beta_\lambda - 1} \times \\
&\quad \int \prod_{\lambda \in \Lambda_{\mathcal{C}}} \frac{(\mu_\lambda + \sum_{d:\lambda_d=\lambda} h_d^{\beta_\lambda})^{\zeta_\lambda + n_\lambda}}{\Gamma(\zeta_\lambda + n_\lambda)\eta_\lambda^{\zeta_\lambda + n_\lambda + 1}} exp(-\frac{\mu_\lambda + \sum_{d:\lambda_d=\lambda} h_d^{\beta_\lambda}}{\eta_\lambda})\, d\boldsymbol{\eta} \\
&= \prod_{\lambda \in \Lambda_{\mathcal{C}}} \frac{\beta_\lambda^{n_\lambda} \mu_\lambda^{\zeta_\lambda}}{(\mu_\lambda + \sum_{d:\lambda_d=\lambda} h_d^{\beta_\lambda})^{\zeta_\lambda + n_\lambda}} \frac{\Gamma(\zeta_\lambda + n_\lambda)}{\Gamma(\zeta_\lambda)} \prod_{d:\lambda_d=\lambda} h_d^{\beta_\lambda - 1}.
\end{aligned}
\tag{5.2.28}
$$

Combining this expression and equation (5.2.27), the probability in equation (5.2.24) can therefore be evaluated by:

$$
\begin{aligned}
&p(\{\boldsymbol{l}_d, h_d\}_{d \in \{1,\cdots,D\}} | \{\boldsymbol{e}_d\}_{d \in \{1,\cdots,D\}}, \boldsymbol{\nu}, r_1, r_2, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta}) \\
&= \prod_{pa_j \in pa^{\mathcal{H}\downarrow}(L_j), L_j \in \boldsymbol{L}_D} \frac{B(\boldsymbol{n}_{L_j}^{pa_j} + \boldsymbol{\nu}_{L_j}^{pa_j})}{B(\boldsymbol{\nu}_{L_j}^{pa_j})} \prod_{\lambda \in \Lambda_{\mathcal{C}}} \frac{\beta_\lambda^{n_\lambda} \mu_\lambda^{\zeta_\lambda}}{(\mu_\lambda + \sum_{d:\lambda_d=\lambda} h_d^{\beta_\lambda})^{\zeta_\lambda + n_\lambda}} \frac{\Gamma(\zeta_\lambda + n_\lambda)}{\Gamma(\zeta_\lambda)} \prod_{d:\lambda_d=\lambda} h_d^{\beta_\lambda - 1}.
\end{aligned}
\tag{5.2.29}
$$

So far we have shown how to compute the first component in equation (5.2.23). Now we expand the expression of component (2).

$$
\begin{aligned}
& p(\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\alpha}, r_1, r_2) \\
=& \int p(\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{\theta}^d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\alpha}, r_1, r_2)\, d\boldsymbol{\theta} \\
=& \int p(\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{\theta}^d\}_{d\in\{1,\cdots,D\}}) p(\{\boldsymbol{\theta}^d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\alpha})\times \\
& p(\{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\gamma}) p(\boldsymbol{\gamma}|r_1, r_2)\, d\boldsymbol{\theta} d\boldsymbol{\gamma} \\
=& \int p(\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}|\{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{\theta}^d\}_{d\in\{1,\cdots,D\}}) p(\{\boldsymbol{\theta}^d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\alpha})\, d\boldsymbol{\theta}\times \\
& \int p(\{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\gamma}) p(\boldsymbol{\gamma}|r_1, r_2)\, d\boldsymbol{\gamma} \\
=& \int \prod_{d=1}^{D} \prod_{w\in W} \frac{1}{B(\boldsymbol{\alpha}_w)} \prod_{e\in E(w)} (\theta_e^d)^{\alpha_e + \mathbb{I}_{e\in E_{\lambda_d}} - 1}\, d\boldsymbol{\theta}\times \\
& \int \prod_{L_j\in \boldsymbol{L}_d} \frac{1}{B(r_1, r_2)} \gamma_{L_j}^{n_{L_j,1}+r_1-1} (1-\gamma_{L_j})^{N_{L_j}-n_{L_j,1}+r_2-1}\, d\boldsymbol{\gamma},
\end{aligned}
\tag{5.2.30}
$$

where N_{L_j} denotes the total number of observations for the core event variable L_j in the dataset, $n_{L_j,1}$ denotes the number of observations for the core event variable L_j satisfying $a_{L_j} = 1$. Then $n_{L_j,0} = N_{L_j} - n_{L_j,1}$ is the number of observations for the core event variable L_j satisfying $a_{L_j} = 0$. For the d^{th} document, let $\boldsymbol{n}_{d,E(w)} = \{n_{d,e_{w,w'}}\}_{e_{w,w'}\in E(w)}$, where $n_{d,e_{w,w'}}$ denotes the number of times the edge $e_{w,w'} \in E(w)$ is traversed by λ_d. In our case, $n_{d,e_{w,w'}} = 0$ or 1.

The above expression can be further simplified as follows.

$$
\begin{aligned}
& p(\{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\alpha}, r_1, r_2) \\
=& \prod_{d=1}^{D} \prod_{w\in W} \frac{B(\boldsymbol{\alpha}_w + \boldsymbol{n}_{d,E(w)})}{B(\boldsymbol{\alpha}_w)} \times \prod_{L_j\in \boldsymbol{L}_D} \frac{B(n_{L_j,1}+r_1, n_{L_j,0}+r_2)}{B(r_1, r_2)}.
\end{aligned}
\tag{5.2.31}
$$

Combining the expressions in equation (5.2.29) and equation (5.2.31), the joint

distribution in equation (5.2.23) can now be expanded as:

$$
\begin{aligned}
&p(\{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{e}_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}|\boldsymbol{\alpha}, r_1, r_2, \boldsymbol{\nu}, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta}) \\
&= \left(\prod_{pa_j\in pa^{\mathcal{H}\downarrow}(L_j), L_j\in\boldsymbol{L}_D} \frac{B(\boldsymbol{n}_{L_j}^{pa_j} + \boldsymbol{\nu}_{L_j}^{pa_j})}{B(\boldsymbol{\nu}_{L_j}^{pa_j})}\right) \times \left(\prod_{\lambda\in\Lambda_{\mathcal{C}}} \frac{\beta_\lambda^{n_\lambda}\mu_\lambda^{\zeta_\lambda}}{(\mu_\lambda + \sum_{d:\lambda_d=\lambda} h_d^{\beta_\lambda})^{\zeta_\lambda+n_\lambda}} \frac{\Gamma(\zeta_\lambda + n_\lambda)}{\Gamma(\zeta_\lambda)} \prod_{d:\lambda_d=\lambda} h_d^{\beta_\lambda-1}\right) \\
&\quad \times \left(\prod_{d=1}^{D}\prod_{w\in W} \frac{B(\boldsymbol{\alpha}_w + \boldsymbol{n}_{d,E(w)})}{B(\boldsymbol{\alpha}_w)}\right) \times \left(\prod_{L_j\in\boldsymbol{L}_d} \frac{B(n_{L_j,1} + r_1, n_{L_j,0} + r_2)}{B(r_1, r_2)}\right).
\end{aligned}
\tag{5.2.32}
$$

We next apply blocked Gibbs sampler by grouping $\{\boldsymbol{e}_d, \boldsymbol{a}_d\}$ and sampling them conditioned on all other variables. Equivalently, we sample the block $\{\lambda_d, \boldsymbol{a}_d\}$. Let $\boldsymbol{e}_{-d}$ denote the edges sampled for documents except the d^{th} document, and $\boldsymbol{a}_{-d}$ denote the boundary indicators sampled for documents except the d^{th} document.

$$
\begin{aligned}
&p(\boldsymbol{e}_d, \boldsymbol{a}_d|\boldsymbol{e}_{-d}, \boldsymbol{a}_{-d}, \{\boldsymbol{l}_n, h_n\}_{n\in\{1,\cdots,D\}}) \\
=&\frac{p(\{\boldsymbol{e}_n\}_{n\in\{1,\cdots,D\}}, \{\boldsymbol{a}_n\}_{n\in\{1,\cdots,D\}}, \{\boldsymbol{l}_n, h_n\}_{n\in\{1,\cdots,D\}})}{p(\boldsymbol{e}_{-d}, \boldsymbol{a}_{-d}, \{\boldsymbol{l}_n, h_n\}_{n\in\{1,\cdots,D\}})} \\
=&\frac{p(\{\boldsymbol{l}_n, h_n\}_{n\in\{1,\cdots,D\}}|\{\boldsymbol{e}_n\}_{n\in\{1,\cdots,D\}}, \{\boldsymbol{a}_n\}_{n\in\{1,\cdots,D\}})}{p(\{\boldsymbol{l}_n, h_n\}_{n\in\{1,\cdots,D\}\backslash d}|\boldsymbol{e}_{-d}, \boldsymbol{a}_{-d})p(\boldsymbol{l}_d, h_d)} \times \frac{p(\{\boldsymbol{e}_n\}_{n\in\{1,\cdots,D\}}, \{\boldsymbol{a}_n\}_{n\in\{1,\cdots,D\}})}{p(\boldsymbol{e}_{-d}, \boldsymbol{a}_{-d})}
\end{aligned}
\tag{5.2.33}
$$

Let $n_{l_j,^-d}^{pa_j}$ denotes the number of the observed core event l_j whose flattened parent is pa_j excluding the observations from the d^{th} document. Let $\boldsymbol{n}_{L_j,^-d}^{pa_j} = \{n_{l_j,^-d}^{pa_j}\}_{l_j\in\mathbb{L}_j}$. Let $n_{L_j,1}^{-d}$ denote the number of the observations for the core vent variable L_j satisfying $a_{L_j} = 1$ excluding the observations from the d^{th} document.

$$
\begin{aligned}
&p(\boldsymbol{e}_d, \boldsymbol{a}_d|\boldsymbol{e}_{-d}, \boldsymbol{a}_{-d}, \{\boldsymbol{l}_n, h_n\}_{n\in\{1,\cdots,D\}}) \\
&= \left(\prod_{pa_j\in pa^{\mathcal{H}\downarrow}(L_j), L_j\in\boldsymbol{L}_D} \frac{B(\boldsymbol{n}_{L_j}^{pa_j} + \boldsymbol{\nu}_{L_j}^{pa_j})}{B(\boldsymbol{n}_{L_j,^-d}^{pa_j} + \boldsymbol{\nu}_{L_j}^{pa_j})}\right) \times \left(\frac{\beta_{\lambda_d}\mu_{\lambda_d}^{\zeta_{\lambda_d}}}{(\mu_{\lambda_d} + \sum_{n:\lambda_n=\lambda_d} h_n^{\beta_{\lambda_d}})} \frac{\zeta_{\lambda_d} + n_{\lambda_d} - 1}{\zeta_{\lambda d} - 1} \times h_d^{\beta_{\lambda_d}-1}\right) \times \\
&\quad \left(\prod_{w\in W_{\lambda_d}} \frac{B(\boldsymbol{\alpha}_w + \boldsymbol{n}_{d,E(w)})}{B(\boldsymbol{\alpha}_w)}\right) \times \left(\prod_{L_j\in\boldsymbol{L}_D} \frac{B(n_{L_j,1} + r_1, n_{L_j,0} + r_2)}{B(n_{L_j,1}^{-d} + r_1, n_{L_j,0}^{-d} + r_2)}\right).
\end{aligned}
\tag{5.2.34}
$$

Given the counts and the hyperparameters, we can estimate the parameters as

follows. The estimated transition probability for edge e and document d is

$$\hat{\theta}_e^d = \frac{n_{d,e} + \alpha_e}{\sum_{e \in E(w)} n_{d,e} + \alpha_e}. \tag{5.2.35}$$

Let $\hat{\boldsymbol{\theta}}^d = \{\hat{\theta}_e\}_{e \in E_{\mathcal{C}}}$. The estimated emission probability for l_j and the flattened parent pa_j is

$$\hat{\phi}_{l_j}^{pa_j} = \frac{n_{l_j}^{pa_j} + \nu_{l_j}^{pa_j}}{\sum_{l_{ji} \in \mathbb{L}_j} n_{l_{ji}}^{pa_j} + \nu_{l_{ji}}^{pa_j}}. \tag{5.2.36}$$

Let $\hat{\boldsymbol{\phi}}_{L_j}^{pa^{\mathcal{H}\downarrow}(L_j)} = \{\hat{\phi}_{l_{ji}}^{pa^{\mathcal{H}\downarrow}(L_j)}\}_{l_{ji} \in \mathbb{L}_j}$ and $\hat{\boldsymbol{\phi}} = \{\hat{\phi}_{L_j}^{pa^{\mathcal{H}\downarrow}(L_j)}\}_{L_j \in \boldsymbol{L}_D, \mathcal{H}}$. The probability of a transition along an edge when the core event variable is L_j is estimated to be

$$\hat{\gamma}_{L_j} = \frac{n_{L_j,1} + r_1}{n_{L_j,1} + r_1 + n_{L_j,0} + r_2}. \tag{5.2.37}$$

Let $\hat{\boldsymbol{\gamma}} = \{\hat{\gamma}_{L_j}\}_{L_j \in \boldsymbol{L}_D}$. For each path λ, the hyperparamters for the path time are updated by

$$\hat{\zeta}_\lambda = \zeta_\lambda + n_\lambda \tag{5.2.38}$$

$$\hat{\mu}_\lambda = \mu_\lambda + \sum_{d:\lambda_d = \lambda} h_d^{\beta_\lambda}. \tag{5.2.39}$$

Let $\hat{\boldsymbol{\zeta}} = \{\zeta_\lambda\}_{\lambda \in \Lambda_{\mathcal{C}}}$ and $\hat{\boldsymbol{\mu}} = \{\mu_\lambda\}_{\lambda \in \Lambda_{\mathcal{C}}}$.

Based on the general process described in Section 5.2.1 and the probability expressions given in this section, we now devise a blocked Gibbs sampler for estimating the parameters and learning hidden variables. We call this the **HcaGibbs** Algorithm, see Algorithm 7. The inputs of the algorithm are $O_D = \{\boldsymbol{l}_d, h_d\}_{d \in \{1,\cdots,D\}}$ and $\Omega = \{\boldsymbol{\alpha}, r_1, r_2, \boldsymbol{\nu}, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta}\}$, and the outputs are $\{\boldsymbol{e}_d\}_{d \in \{1,\cdots,D\}}$, or equivalently $\{\lambda_d\}_{d \in \{1,\cdots,D\}}$, $\{\boldsymbol{\theta}_w^d\}_{w \in W, d \in \{1,\cdots,D\}}$, $\boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\phi}$.

Algorithm 7 HcaGibbs

Input: $\{\boldsymbol{l}_d, h_d\}_{d\in\{1,\cdots,D\}}, \boldsymbol{\alpha}, r_1, r_2, \boldsymbol{\nu}, \boldsymbol{\zeta}, \boldsymbol{\mu}, \boldsymbol{\beta}$

Output: $\{\lambda_d\}_{d\in\{1,\cdots,D\}}, \{\boldsymbol{\theta}_w^d\}_{w\in W, d\in\{1,\cdots,D\}}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\phi}$

set the count statistics $\{n_{l_j}^{pa_j}\}, n_{d,e}, n_{L_j,1}, n_{L_j,0}, n_\lambda$ to 0 // Initialisation
for document $d \in \{1, \cdots, D\}$ **do**
  $a_{d_1} = 1$, sample $e_{d_1} \sim \text{Multinomial}(\frac{1}{N_{E(w_0)}})$ // $N_{E(w_0)}$: number of edges emanating from w_0
  update $n_{d,e_{d_1}} \mathrel{+}= 1, n_{l_{d_1}}^{pa^{\mathcal{H}\downarrow}(L_{d_1})} \mathrel{+}= 1, n_{l_{d_1},1} \mathrel{+}= 1$
  for $j \in \{2, \cdots, m_d\}$ **do** // m_d: number of observed events for the d^{th} document
    sample $a_{L_{d_j}} \sim \text{Bernoulli}(\frac{1}{2})$
    if $a_{L_{d_j}} = 0$ **then**
      $e_{d_j} = e_{d_{j-1}}$
      update $n_{l_{d_j}}^{pa^{\mathcal{H}\downarrow}(L_{d_j})} \mathrel{+}= 1, n_{L_{d_j},0} \mathrel{+}= 1$
    else
      sample $e_{d_j} \sim \text{Multinomial}(\frac{1}{N_{E(w_{d_{j-1}})}})$
      update $n_{d,e_{d_j}} \mathrel{+}= 1, n_{l_{d_j}}^{pa^{\mathcal{H}\downarrow}(L_{d_j})} \mathrel{+}= 1, n_{L_{d_j},1} \mathrel{+}= 1$
    end if
  end for
end for
$\lambda_d = \lambda_{\tilde{\boldsymbol{e}}_d}$, update $n_{\lambda_d} \mathrel{+}= 1$
while run **do** // blocked Gibbs sampling
  for document $d \in \{1, \cdots, D\}$ **do**
    $n_{\lambda_{\boldsymbol{e}_d}} \mathrel{-}= 1$ // decrement counts
    for $j \in \{1, \cdots, m_d\}$ **do**
      $n_{l_{d_j}}^{pa^{\mathcal{H}\downarrow}(L_{d_j})} \mathrel{-}= 1, n_{L_{d_j},a_{L_{d_j}}} \mathrel{-}= 1$
      for $e \in \tilde{\boldsymbol{e}}_d$ **do**
        $n_{d,e} \mathrel{-}= 1$
      end for
    end for
    sample $\boldsymbol{e}_d, \boldsymbol{a}_d \sim p(\boldsymbol{e}_d, \boldsymbol{a}_d | \boldsymbol{e}_{-d}, \boldsymbol{a}_{-d}, \{\boldsymbol{l}_n, h_n\}_{n\in\{1,\cdots,D\}})$ // resample
    $\lambda_d = \lambda_{\tilde{\boldsymbol{e}}_d}$, update $n_{\lambda_d} \mathrel{+}= 1$ // increment counts
    for $j \in \{1, \cdots, m_d\}$ **do**
      $n_{l_{d_j}}^{pa^{\mathcal{H}\downarrow}(L_{d_j})} \mathrel{+}= 1, n_{l_{d_j},a_{L_{d_j}}} \mathrel{+}= 1$
      for $e \in \tilde{\boldsymbol{e}}_d$ **do**
        $n_{d,e} \mathrel{+}= 1$
      end for
    end for
  end for
end while
Estimate parameters from the count statistics and the hyperparameters using equations (5.2.35) to (5.2.39).

5.3 Experiments for the GN-CEG model

Having specified the learning algorithm, we next design experiments to examine how it performs. In this section, we first evaluate the algorithm on synthetic data generated from a GN-CEG model of known structure and parameters, and then apply it on a real dataset for the conservator of a transformer.

5.3.1 Analysis using synthetic data

Here we still use the bushing system mentioned in Section 5.1 as an example. Assume the ground truth causal CEG lying at the bottom level of the hierarchical model has the structure plotted in Figure 5.3. The learning staged tree of this causal CEG is shown in Figure 5.4. For learning purpose, this tree or equivalently the corresponding CEG in Figure 5.5 lies at the deeper level of the hierarchical model instead of the causal tree since we learn from failure reports. For simplicity we here assume Figure 3.8 is the GN for this example.

$p(l_{1,1}\|x_{c,1})$	0.6	$p(l_{1,2}\|x_{c,1})$	0.4
$p(l_{2,1}\|x_{c,2})$	0.5	$p(l_{2,2}\|x_{c,2})$	0.5
$p(l_{3,1}\|x_{c,3})$	0.4	$p(l_{3,2}\|x_{c,3})$	0.6
$p(l_{5,1}\|x_{c,4})$	0.8	$p(l_{5,2}\|x_{c,4})$	0.2
$p(l_{6,1}\|l_{5,1},x_{c,4})$	0.5	$p(l_{6,2}\|l_{5,1},x_{c,4})$	0.5
$p(l_{6,1}\|l_{5,2},x_{c,4})$	0.5	$p(l_{6,2}\|l_{5,2},x_{c,4})$	0.5
$p(l_{8,1}\|x_{c,6})$	0.7	$p(l_{8,2}\|x_{c,6})$	0.3
$p(l_{9,1}\|x_{c,5})$	0.4	$p(l_{9,2}\|x_{c,5})$	0.6
$p(l_{10,1}\|l_{9,1},x_{c,5})$	0.5	$p(l_{10,2}\|l_{9,1},x_{c,5})$	0.5
$p(l_{10,1}\|l_{9,2},x_{c,5})$	1/6	$p(l_{10,2}\|l_{9,2},x_{c,5})$	5/6
$p(l_{4,1}\|x_{s,1},l_1,l_2,l_3)$	0.5	$p(l_{4,2}\|x_{s,1},l_1,l_2,l_3)$	0.5
$p(l_{4,3}\|x_{s,2},l_1,l_2,l_3)$	1	$p(l_{4,1}\|x_{s,3},l_1,l_2,l_3)$	0.1
$p(l_{4,4}\|x_{s,3},l_1,l_2,l_3)$	0.9	$p(l_{4,5}\|x_{s,4},l_1,l_2,l_3)$	1
$p(l_{7,1}\|x_{s,5})$	1	$p(l_{7,2}\|x_{s,6})$	1

Table 5.1: The ground truth emission probabilities. The core event variables take values $l_{1,1}$ = {failed gasket}, $l_{1,2}$ = {aging gasket},$l_{2,1}$ = {seal crack}, $l_{2,2}$ = {axial crack},$l_{3,1}$ = {crack}, $l_{3,2}$ = {no crack},$l_{5,1}$ = {yes}, $l_{5,2}$ = {no},$l_{6,1}$ = {oxidant contact}, $l_{6,2}$ = {contact resistance},$l_{8,1}$ = {lightening}, $l_{8,2}$ = {weather},$l_{9,1}$ = {temperature}, $l_{9,2}$ = {nitrogen blanket},$l_{10,1}$ = {oil corrosion}, $l_{10,2}$ = {sulphur corrosion},$l_{4,1}$ = {oil level low}, $l_{4,2}$ = {leak}, $l_{4,3}$ = {normal oil level}, $l_{4,4}$ = {loss of oil}, $l_{4,5}$ = {transformer oil and bushing oil},$l_{7,1}$ = {thermal runaway}, $l_{7,2}$ = {electrical discharge}.

Firstly, we simulated synthetic data from the deeper level CEG to the GN to emulate what we could extract from maintenance logs. We have assumed the ground truth transition probabilities on the CEG in Section 5.1 so that we could

generate path for each document. We still used D_1 with 3587 cases as the set of synthetic paths. For each path, we simulated observed core event variables using the hypothesised ground truth emission probabilities in Table 5.1. In this table, we condition the probability on the d-event, which can be replaced by the edge labelled by it. In this way we have constructed the set of observed core event variables for each document. Let F_1 denote the 3587 sets of the synthetic observations associated with D_1.

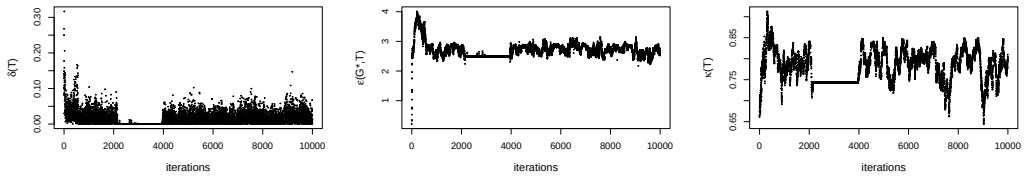
As for the analyses in the previous sections in this chapter, we ran the algorithm for different values of the phantom number $\overline{\alpha}$. Specifically, we ran the HcaGibbs algorithm for 10000 iterations for $\overline{\alpha} = 0.01, 0.1, 1, 5, 10$ respectively. Let $J_1, \cdots, J_5$ denote the output of the algorithm for each experiment. There are only have 10 core event variables and the boundaries of the potential communities are obvious. So we simply assumed $\{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}$ were known for these experiments. Moreover, the hyperparameters for the emission probabilities were fixed to be the same in these five experiment. In particular, for each L_j and the corresponding flattened parent $pa^{\mathcal{H}\downarrow}(L_j)$, let $\nu_{l_{ji}}^{pa^{\mathcal{H}\downarrow}(L_j)} = \nu_{l_{jk}}^{pa^{\mathcal{H}\downarrow}(L_j)} = \nu_0$ for any $l_{ji}, l_{jk} \in \mathbb{L}_j$ unless we have prior knowledge about the weights of the possible values of L_j. Here we have chosen to let $\nu_0 = 1$ to check the performance of the algorithm for different values of $\overline{\alpha}$. We will check the performance of the algorithm for other values of ν_0 in the later experiments.

We assessed the performance of the algorithm in each experiment from two perspectives: (1) visualising the samples; (2) examining the precision of parameter estimations.

Visualising the samples. We first checked the traceplots of some statistics to ensure that the Gibbs samples were "mixed well" and not stuck anywhere. Specifically, we examined how the following three statistics behaved.

1. The **situational difference** $\delta(\mathcal{T})$ measures the distance between the transitional probabilities estimated from the samples of the t^{th} iteration and the $(t+1)^{th}$ iteration. We employ the idea of the total situational error, see equation (5.1.2). Let $\boldsymbol{\theta}_v(t)$ denote the transition probabilities estimated from the t^{th} iteration for situation $v \in S_{\mathcal{T}}$. Then, the situational difference is measured by
$$\delta(\mathcal{T}) = \sum_{v\in S_{\mathcal{T}}} ||\boldsymbol{\theta}_v(t+1) - \boldsymbol{\theta}_v(t)||_2. \tag{5.3.1}$$

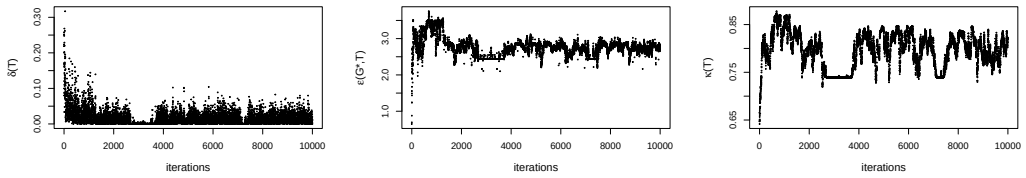
2. The **emission difference** $\epsilon(G^*, \mathcal{T})$ measures the distance between the emission probabilities estimated from the samples of the t^{th} iteration and the $(t+1)^{th}$ iteration. As for the definition of the situational difference, we define



(a) $\overline{\alpha} = 0.01, \delta(\mathcal{T})$

(b) $\overline{\alpha} = 0.01, \epsilon(G^*, \mathcal{T})$

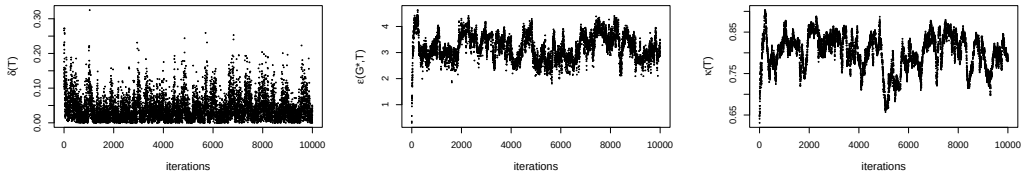
(c) $\overline{\alpha} = 0.01, \kappa(\mathcal{T})$



(d) $\overline{\alpha} = 0.1, \delta(\mathcal{T})$

(e) $\overline{\alpha} = 0.1, \epsilon(G^*, \mathcal{T})$

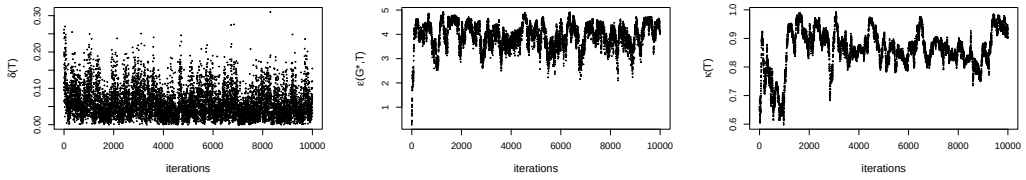
(f) $\overline{\alpha} = 0.1, \kappa(\mathcal{T})$



(g) $\overline{\alpha} = 1, \delta(\mathcal{T})$

(h) $\overline{\alpha} = 1, \epsilon(G^*, \mathcal{T})$

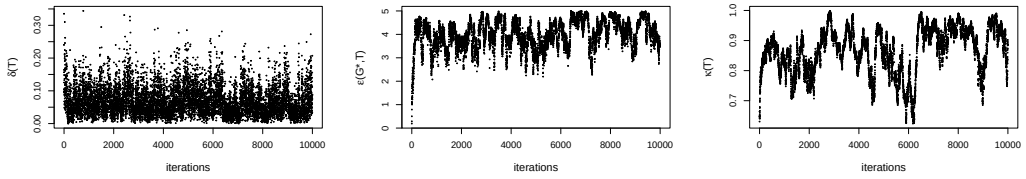
(i) $\overline{\alpha} = 1, \kappa(\mathcal{T})$



(j) $\overline{\alpha} = 5, \delta(\mathcal{T})$

(k) $\overline{\alpha} = 5, \epsilon(G^*, \mathcal{T})$

(l) $\overline{\alpha} = 5, \kappa(\mathcal{T})$



(m) $\overline{\alpha} = 10, \delta(\mathcal{T})$

(n) $\overline{\alpha} = 10, \epsilon(G^*, \mathcal{T})$

(o) $\overline{\alpha} = 10, \kappa(\mathcal{T})$

Figure 5.9: Traceplots for the experiment with synthetic data for different values of $\overline{\alpha}$.

this as follows. Let $\boldsymbol{\phi}_{L_j}^{pa^{\mathcal{H}\downarrow}(L_j)}(t)$ denote the emission probabilities estimated from the t^{th} iteration for L_j with flattened parent $pa^{\mathcal{H}\downarrow}(L_j)$. Then,

$$\epsilon(G^*, \mathcal{T}) = \sum_{L_j \in \boldsymbol{L}_D, pa_j \in pa^{\mathcal{H}\downarrow}(L_j)} ||\boldsymbol{\phi}_{L_j}^{pa_j}(t+1) - \boldsymbol{\phi}_{L_j}^{pa_j}(t)||_2. \tag{5.3.2}$$

3. The **paths matching precision** $\kappa(\mathcal{T})$: this is the proportion of the accurately estimated latent paths for the whole dataset for each iteration t. Let $\lambda_d(t)$ denote the path index for the d^{th} document sampled at the t^{th} iteration. Then,

$$\kappa(\mathcal{T}) = \frac{\sum_{d=1}^{D} \mathbb{I}_{\lambda_d(t)=\lambda_d^*}}{D}. \tag{5.3.3}$$

Traceplots provide simple and transparent visualisations of the performance of the sampler which allow us to check the performance of the samples straightforwardly. Figure 5.9 displays the traceplots for these three statistics for $\overline{\alpha} = 0.01, 0.1, 1, 5, 10$ respectively. From Figure 5.9a to Figure 5.9c, we can see that the sampler does not perform well between 2000 and 4000 iterations. For $\overline{\alpha} = 0.1$, see Figure 5.9d to Figure 5.9f, between 2500 and 3500 iterations, the traceplots fluctuate within a narrow range but are not stuck within this interval. For $\overline{\alpha} = 1, 5, 10$, see Figure 5.9g to Figure 5.9o, the chains of the estimated statistics behave well.

Precision of parameter estimations. Since we have assumed the ground truth transition probabilities, emission probabilities and latent paths for all the documents in the dataset, we can examine how well these parameters are estimated from the algorithm. Specifically, we evaluated the following four types of errors.

1. The **mean path matching precision**. This is the average of the paths matching precision $\kappa(\mathcal{T})$ for all the iterations.

$$\overline{\kappa}(\mathcal{T}) = \frac{\sum_{t=1}^{N} \sum_{d=1}^{D} \mathbb{I}_{\lambda_d(t)=\lambda_d^*}}{D \times N}. \tag{5.3.4}$$

2. The total situational error, see $\Xi(\mathcal{T})$ equation (5.1.2).

3. The **total emission error** is the sum of the Euclidean distance between the true emission probabilities $\boldsymbol{\phi}^*$ and the mean posterior probabilities $\tilde{\boldsymbol{\phi}}$ estimated from the samples.

$$\varpi(G^*, \mathcal{T}) = \sum_{L_j \in \boldsymbol{L}_D, pa_j \in pa^{\mathcal{H}\downarrow}(L_j)} ||\boldsymbol{\phi}_{L_j}^{*pa_j} - \tilde{\boldsymbol{\phi}}_{L_j}^{pa_j}||_2. \tag{5.3.5}$$

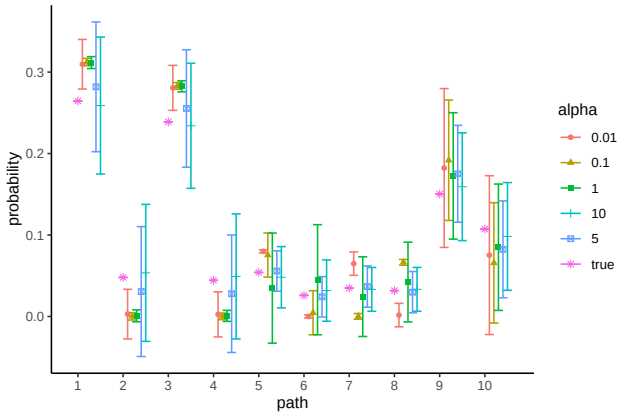


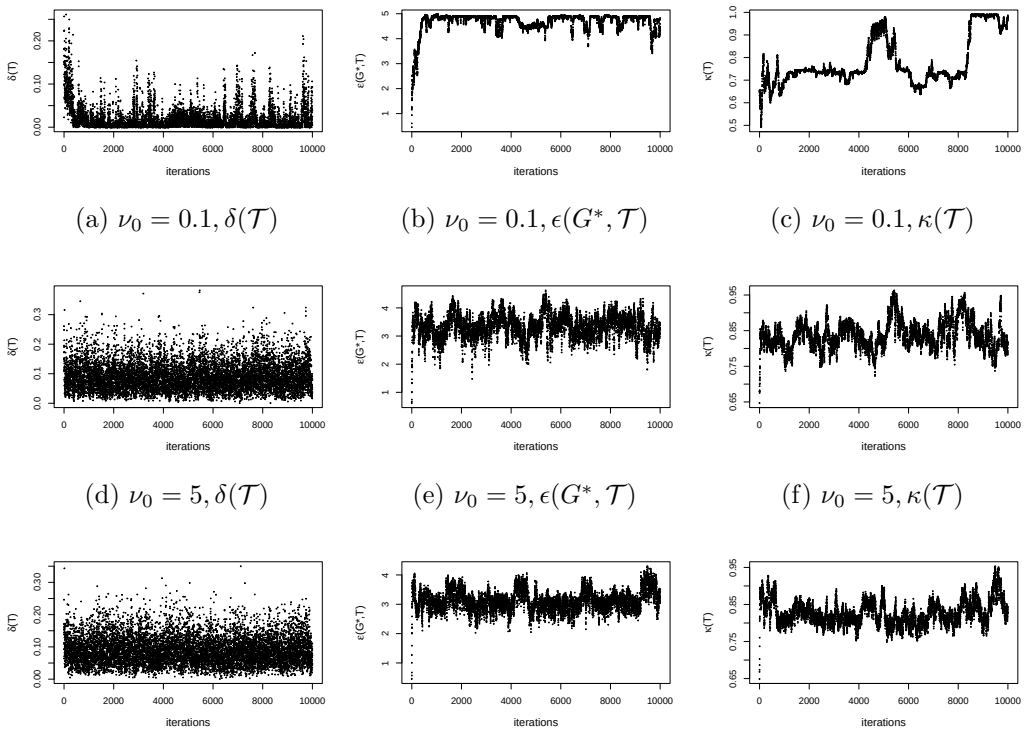
Figure 5.10: Comparing the estimated path probabilities: "alpha" in the figure refers to $\overline{\alpha}$.

From the sampled results for different values of $\overline{\alpha}$, the mean path matching precision $\overline{\kappa}(\mathcal{T})$ is 0.781 when $\overline{\alpha} = 0.01$, 0.797 when $\overline{\alpha} = 0.1$, 0.801 when $\overline{\alpha} = 1$, 0.861 when $\overline{\alpha} = 5$ and 0.867 when $\overline{\alpha} = 10$. The highest precision appears when $\overline{\alpha} = 10$. We can further check the mean posterior path probability and the standard errors for each sampled result. Figure 5.10 compares the estimates with the ground truth path probabilities. As mentioned before, only the failure paths are likely to be the latent paths for the failure reports. There are 10 candidate paths here. The indices of the paths have been labelled in Figure 5.4. Then we estimated the path probability for each of the 10 paths by computing the mean posterior path probability from the 10000 samples for the 3587 synthetic documents. In the figure, different colours are assigned for the results of different experiments with different phantom numbers $\overline{\alpha}$. The dots represent the mean posterior probability while the bar cross each dot represents the interval of the mean plus or minus two standard deviations. In this figure, we can see that for each path, the real path probability is inclusive in the bar when $\overline{\alpha} = 5$ or 10. The estimates given by $\overline{\alpha} = 0.01$ and $\overline{\alpha} = 0.1$ are less accurate. For $\overline{\alpha} = 0.01$, only the bars for path 9 and path 10 include the corresponding real path probabilities. This is not surprising because the mean path matching precision estimated from the corresponding two chains are lower than others. We can also check the corresponding traceplots in Figure 5.9c and Figure 5.9f. As we have mentioned earlier, part of the chain does not behave well for both cases. This may cause low precision of path matching.

For $\overline{\alpha} = 0.01, 0.1, 1, 5, 10$, the total situational errors are 2.724, 2.742, 1.868, 1.535, 1.175 respectively. The transition probabilities are most accurately estimated in the experiment with $\overline{\alpha} = 10$. In addition, we computed the total emission errors

for these experiments, which are 3.256, 4.200, 3.057,1.727, 1.967 respectively. The last two samplers give the emission errors lower than 2.

From these experiments, we can see that the mixing of the samples is not bad and the paths can be well estimated. But we may need to carefully determine $\overline{\alpha}$ to avoid setting this value too small in this example otherwise the sampler will not mix properly.



(a) $\nu_0 = 0.1, \delta(\mathcal{T})$ (b) $\nu_0 = 0.1, \epsilon(G^*, \mathcal{T})$ (c) $\nu_0 = 0.1, \kappa(\mathcal{T})$

(d) $\nu_0 = 5, \delta(\mathcal{T})$ (e) $\nu_0 = 5, \epsilon(G^*, \mathcal{T})$ (f) $\nu_0 = 5, \kappa(\mathcal{T})$

(g) $\nu_0 = 10, \delta(\mathcal{T})$ (h) $\nu_0 = 10, \epsilon(G^*, \mathcal{T})$ (i) $\nu_0 = 10, \kappa(\mathcal{T})$

Figure 5.11: Traceplots for the experiment with synthetic data for different ν_0.

Next we fixed $\overline{\alpha} = 10$ and ran the algorithm for different values of ν_0. We already had the output for $\nu_0 = 1$. Here, let $\nu_0 = 0.1, 5, 10$ and the outputs of the HcaGibbs algorithm for these scenarios be denoted by J_6, J_7, J_8 respectively. The corresponding traceplots are shown in Figure 5.11. The chains for $\nu_0 = 0.1$ behave not as well as the others. It is likely only a small proportion of latent paths are updated at each iteration in intervals 1000-4000, 6000-8000, 8500-10000.

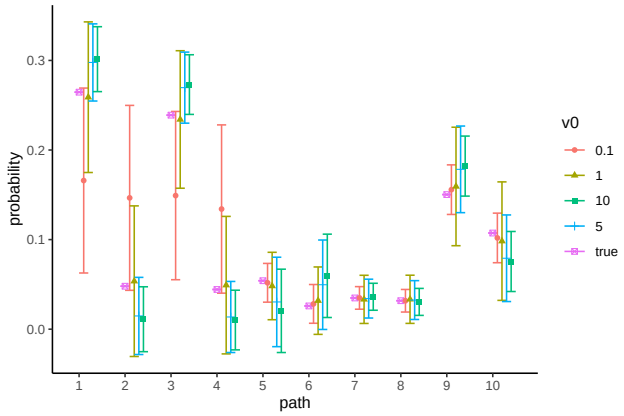
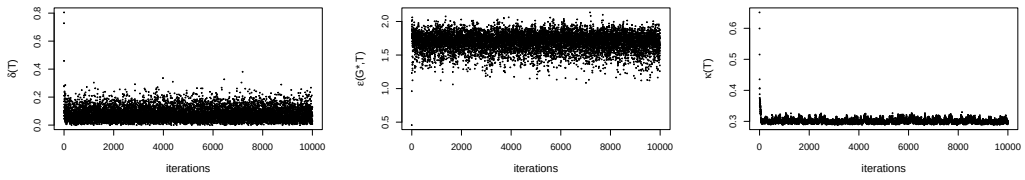


Figure 5.12: Comparing the estimated path probabilities for different ν_0.

The path probability estimates for the first four paths are badly estimated by J_6 when $\nu_0 = 0.1$, see the red bars in Figure 5.12. Though the estimates from J_7 and J_8, see the blue and the green bars, are also worse than the estimates from J_5, the mean path matching precision evaluated from these samples indicates that around 83.7% and 82.7% latent paths for the 3587 documents are correctly estimated by these two experiments respectively. This is close to 86.7%, the mean path matching precision of J_5 when $\nu_0 = 1$. Only 77.9% of the documents have their latent paths correctly estimated by J_6. The total situational errors computed from J_6, J_7, J_8 are 1.674,1.740,1.996 respectively. The difference between these errors is small. There is also no big difference between the total emission errors for these three sets of samples. The total emission errors computed from J_6, J_7, J_8 are 2.008,2.172,2.162 respectively.



(a) $\delta(\mathcal{T})$ (b) $\epsilon(G^*, \mathcal{T})$ (c) $\kappa(\mathcal{T})$

Figure 5.13: Traceplots for the experiment with synthetic failure time.

Assume we have observed the failure time for each process. Then we can use the designed algorithm for learning the model parameters with total path time. Let the shape parameter of the InverseGamma prior be 2 for every path, and the scale parameter be 5 for every path. Let the shape parameter for the Weibull path time

be 2 for every path. Taking $\overline{\alpha} = 10, \nu_0 = 1$ and running the algorithm for 10000 iterations, the traceplots of the three statistics specified above are shown in Figure 5.13. The traceplots show that although the chains mix well, the precision of path matching is very low. The situational error is 4.214 and the emission error is 3.347, both of which are high. This is likely to be caused by the imprecise path matching. We can also evaluate the prediction error of the total path time, see equation (5.1.4) and it is 5.440. There are 10 failure paths in this example, so the mean prediction error of the total path time is 0.544.

All the above experiments assume a ground truth CEG. When we do not know the stages, and so the positions, but only know the ground truth event tree, then we may need to redesign the algorithm so that the stages and the parameters are learned simultaneously. However, due to the length of this thesis and the complexity of learning the CEG structure and the GN-CEG parameters at the same time, we will not propose a new algorithm for this. Instead, here we simply try to perform the paths matching task by Algorithm 7 and then apply the MAP algorithm to find the best CEG topology from the learned paths. This is obvious not ideal but here we only show an insight of learning from the GN on the tree graph and the development of such algorithms would constitute a thesis on its own.

Assume that we have a ground truth event tree for the selected bushing system whose structure is shown in Figure 5.14. Here for simplicity we ignored the failure time. Now we ran the algorithm for $\overline{\alpha} = 1, 5, 10$ when $\nu_0 = 1$ and $\nu_0 = 5, 10$ when $\overline{\alpha} = 10$. Each experiment was implemented with 10000 iterations. The MAP scores and the situational errors estimated from the best scoring structures are shown in Table 5.2.

For the best models selected for J_9 and J_{10}, v_1 and v_2 are in the same stage, while in the ground truth staged tree they are in different stages. The situations v_{11} and v_{12} are in different stages in the selected model, despite them being in the same stage in the ground truth staged tree. For the best models selected for J_{11}, J_{12} and J_{13}, v_{11} is misclassified into the same stage as v_7 and v_8. Apart from this misassignment, all other stages are correctly learned.

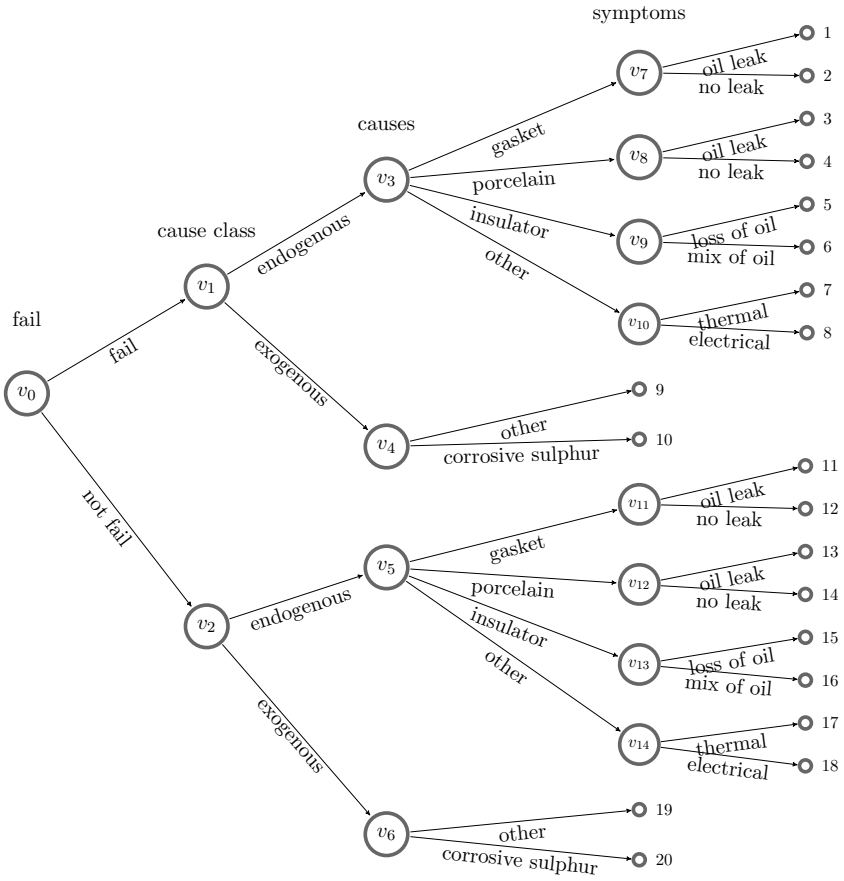


Figure 5.14: The learning event tree for the selected bushing system.

ν_0	$\overline{\alpha}$	outputs	MAP scores	situational errors
1	1	J_9	-6206.506	1.550
1	5	J_{10}	-6214.010	1.547
1	10	J_{11}	-7163.873	1.110365
5	10	J_{12}	-6122.471	2.211
10	10	J_{13}	-6122.471	2.211

Table 5.2: The MAP scores and the total situational errors of the best scoring models.

We next considered the synthetic failure time with the synthetic documents, and took $\overline{\alpha} = 10, \nu_0 = 1$. We ran the algorithm again for 10000 iterations and checked the selected best structure of the CEG. We have shown above that when learning the parameters for the failure time distribution, the mean path matching precision is low. So compared to the model learned from J_{11}, which are the output from the algorithm when setting $\overline{\alpha} = 10$ and $\nu_0 = 1$, there are more stages misspecified. Specifically, in the selected model, v_4 and v_6 are not in the same stage, v_{11}

and v_{12} are not in the same stage, v_9 and v_{13} are not in the same stage, v_{10} and v_{14} are not in the same stage. These pairs are in the same stage in the ground truth staged tree. Due to the imprecise staging result, it is not unexpected that the total situational error is high, which is 4.769 in this experiment.

5.3.2 Conservator system data

Core event extraction assessment

Since devising the NLP algorithms is not our essential task in this thesis, and we only propose a method that combines existing ideas, we here only evaluate the extracted core events using one of the popular assessment in NLP. To measure the accuracy of the proposed algorithms 1-6, we compute the F-score from the precision and the recall [Björne et al., 2010]. In information retrieval, two popular performance metrics are [Powers, 2020]:

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}, \tag{5.3.6}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}. \tag{5.3.7}$$

Given the precision and the recall, one can compute the F-measure, *i.e.* the F-score, as the harmonic mean of the precision and the recall:

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{5.3.8}$$

Here we follow the steps in Algorithm 1 to Algorithm 6 to extract the partially ordered core events and then apply these metrics to measure the accuracy of the extracted cause-effect pairs of core events. If our dataset is not large, we can read through the texts to annotate causally related events. We call these the annotated causal pairs. Now let

$$\text{precision} = \frac{|\{\text{extracted causal pairs}\} \cap \{\text{annotated causal pairs}\}|}{|\{\text{extracted causal pairs}|} \tag{5.3.9}$$

and

$$\text{recall} = \frac{|\{\text{extracted causal pairs}\} \cap \{\text{annotated causal pairs}\}|}{|\{\text{annotated causal pairs}|}. \tag{5.3.10}$$

In this case, $\{\text{extracted causal pairs}\} \cap \{\text{annotated causal pairs}\}$ gives the true positives.

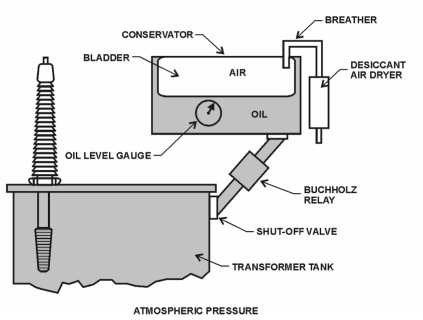


Figure 5.15: The conservator system in a transformer [Reclamation, 2005].

We can now pick all the documents whose entries for "component" are "SGT", and entries for "subcomponent" are "Conservator", so that these records are related to defects of a conservator system in a transformer. Figure 5.15 plots the subcomponents in this system. Running algorithms 1-6, we have 1023 pairs of the extracted core events with repetitions. There are 496 unique cause-effect pairs in this extracted dataset. We picked the unique pairs of these core events and compared them with a set of 296 cause-effect pairs which were assumed to be true given the documents and the background knowledge about the transformer [Reclamation, 2005]. We were then able to compute the scores using the formulas given above.

There are 195 correctly selected cause-effect pairs, *i.e.*

$$|\{\text{extracted causal pairs}\} \cap \{\text{annotated causal pairs}\}| = 195. \tag{5.3.11}$$

The precision is 0.39 and the recall is 0.66. The precision is low because there are many non-causal patterns picked up by our algorithm. These non-causal pairs may include the pairs of an indirect cause and the effect event, the pairs whose causal order is wrongly extracted, and the pairs whose order is non-causal but only temporal. From the value of the recall, we conclude that approximately 66% of the real causal relations are extracted by the proposed algorithms. This percentage is acceptable since many contemporary cutting-edge event extraction algorithms have the recall score around 60% [Björne et al., 2010]. Note that since the "true" causal pairs were not annotated by domain experts, the precision and the recall can only be used as a reference for evaluating the algorithms. Due to the low precision, when we balance the precision and the recall, the f-score is not high, 0.49. However, this is not very problematic because after this step we cluster the core events to construct the core event variables before embedding them on the CEG.

Learning within the GN-CEG model

We constructed 13 core event variables $L_1, \cdots, L_{13}$ from the extracted core events. Table 5.3 displays the core event variables and the corresponding state spaces. The extracted GN is shown in Figure 5.16.

variable	state space
L_1	$l_{1,1} = \{\text{low temperatures}\}, l_{1,2} = \{\text{high humidity}\}, l_{1,3} = \{\text{no change}\}$
L_2	$l_{2,1} = \{\text{gauge defect}\}, l_{2,2} = \{\text{glass dirty}\}, l_{2,3} = \{\text{sight glass}\}, l_{2,4} = \{\text{indicator ok}\}$
L_3	$l_{3,1} = \{\text{damaged component}\}, l_{3,2} = \{\text{seal integrity defect}\}, l_{3,3} = \{\text{crack}\}, l_{3,4} = \{\text{seal deterioration}\}, l_{3,5} = \{\text{loose fixing}\}, l_{3,6} = \{\text{gasket}\}, l_{3,7} = \{\text{contact fault}\}, l_{3,8} = \{\text{ferrule}\}, l_{3,9} = \{\text{terminal cover}\}, l_{3,10} = \{\text{contact ok}\}$
L_4	$l_{4,1} = \{\text{float chamber mechanism defect}\}, l_{4,2} = \{\text{mercury switch defective}\}, l_{4,3} = \{\text{magnet or reed switch}\}, l_{4,4} = \{\text{mechanism ok}\}$
L_5	$l_{5,1} = \{\text{fuse or mcb}\}, l_{5,2} = \{\text{high resistance connection}\}, l_{5,3} = \{\text{mechanical indicator defect}\}, l_{5,4} = \{\text{power supply}\}, l_{5,5} = \{\text{calibration}\}, l_{5,6} = \{\text{conservator operated}\}, l_{5,7} = \{\text{defrost defect}\}, l_{5,8} = \{\text{dessicant abnormal}\}, l_{5,9} = \{\text{pipework defect}\}, l_{5,10} = \{\text{component failure}\}, l_{5,11} = \{\text{ir}\}, l_{5,12} = \{\text{out of service}\}, l_{5,13} = \{\text{incorrect air purge}\}, l_{5,14} = \{\text{tank defect}\}, l_{5,15} = \{\text{no fault}\}$
L_6	$l_{6,1} = \{\text{oil leak}\}, l_{6,2} = \{\text{no leak}\}$
L_7	$l_{7,1} = \{\text{oil level low}\}, l_{7,2} = \{\text{normal}\}$
L_8	$l_{8,1} = \{\text{oil level incorrect}\}, l_{8,2} = \{\text{normal}\}$
L_9	$l_{9,1} = \{\text{buchholz}\}, l_{9,2} = \{\text{buchholz trip}\}, l_{9,3} = \{\text{relay defect}\}, l_{9,4} = \{\text{buchholz ok}\}$
L_{10}	$l_{10,1} = \{\text{drycol control unit defect}\}, l_{10,2} = \{\text{control box}\}, l_{10,3} = \{\text{control ok}\}$
L_{11}	$l_{11,1} = \{\text{drycol breather}\}, l_{11,2} = \{\text{breather blocked}\}, l_{11,3} = \{\text{bag defect}\}, l_{11,4} = \{\text{breather ok}\}$
L_{12}	$l_{12,1} = \{\text{low oil alarm}\}, l_{12,2} = \{\text{alarm}\}, l_{12,3} = \{\text{drycol alarm}\}, l_{12,4} = \{\text{no alarm}\}$
L_{13}	$l_{13,1} = \{\text{transformer abnormal or reflash}\}, l_{13,2} = \{\text{transformer other}\}, l_{13,3} = \{\text{transformer ok}\}$

Table 5.3: The core event variables.

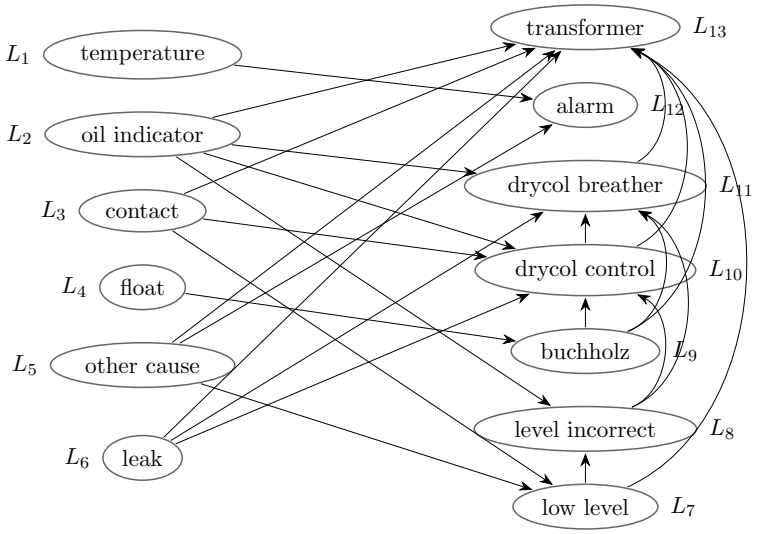


Figure 5.16: The extracted GN for the conservator system.

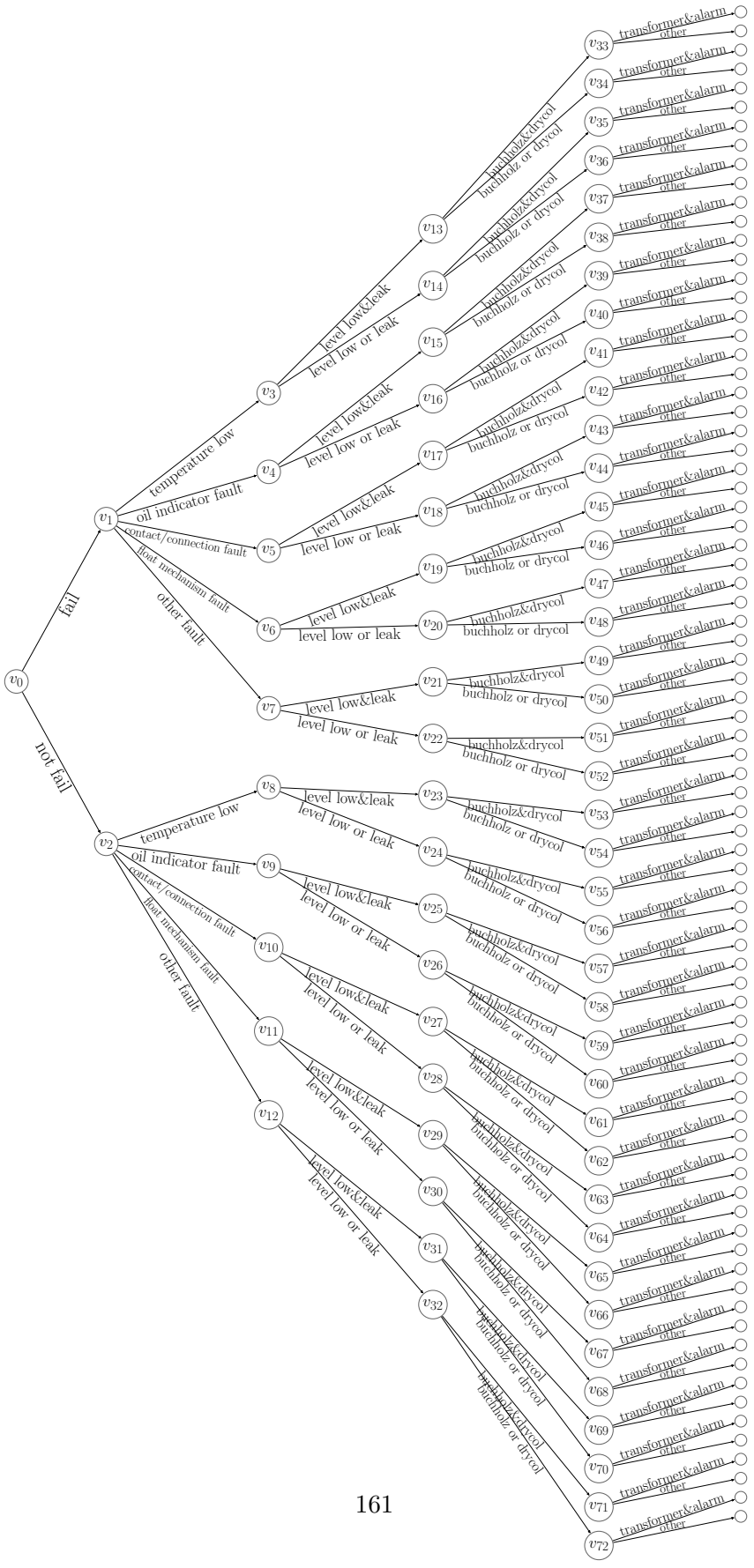
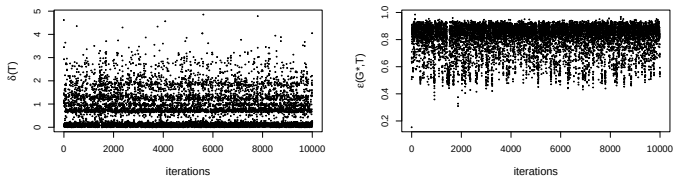


Figure 5.17: The learning event tree for the conservator system.

Reclamation [2005] briefly introduced the conservator system and the other systems in the transformer. On the basis of this information, we constructed a ground truth learning event tree, see Figure 5.17. It starts with the failure indicator, followed by the root causes: temperature low, contact or connection fault, oil indicator fault, float mechanism fault, other fault. The following d-events describe the oil condition which has two states: oil level low and leak, oil level low or leak or oil condition ok. These are followed by the condition of the buchholz, the drycol dontrol unit and the drycol breather. It has two states: buchholz fault and drycol fault, buchholz fault or drycol fault or both ok. The last component modelled by the tree represents the condition of the transformer and the alarm. It has two states: transformer defect and active alarm, otherwise.

The boundaries of communities can easily be found in this example, so we do not sample $\{\boldsymbol{a}_d\}_{d\in\{1,\cdots,D\}}$. We also do not consider the failure time here since the data we collected may contain multiple documents for the same machine and these documents correspond to the consecutive failures that happened to this machine. In the future we plan to investigate a dynamic version of the GN-CEG model with the adapted algorithm for this dataset. If we only pick the initial failures for different machines, then the size of the data is too small for the analysis. Therefore, we do not predict the failure time here.

We ran the HcaGibbs for 10000 iterations and then found the best scoring CEG structure. It is not easy to assess the Gibbs results in a rigorous way in this experiment because the ground truth is unknown. Therefore, we only checked the mixture of the samples by computing the situational difference $\delta(\mathcal{T})$ and the emission difference $\epsilon(G^*, \mathcal{T})$ at each iteration. Figure 5.18 shows that the chains of these two statistics behaved well and did not become stuck anywhere.



(a) $\delta(\mathcal{T})$ (b) $\epsilon(G^*, \mathcal{T})$

Figure 5.18: Traceplots for the conservator experiment.

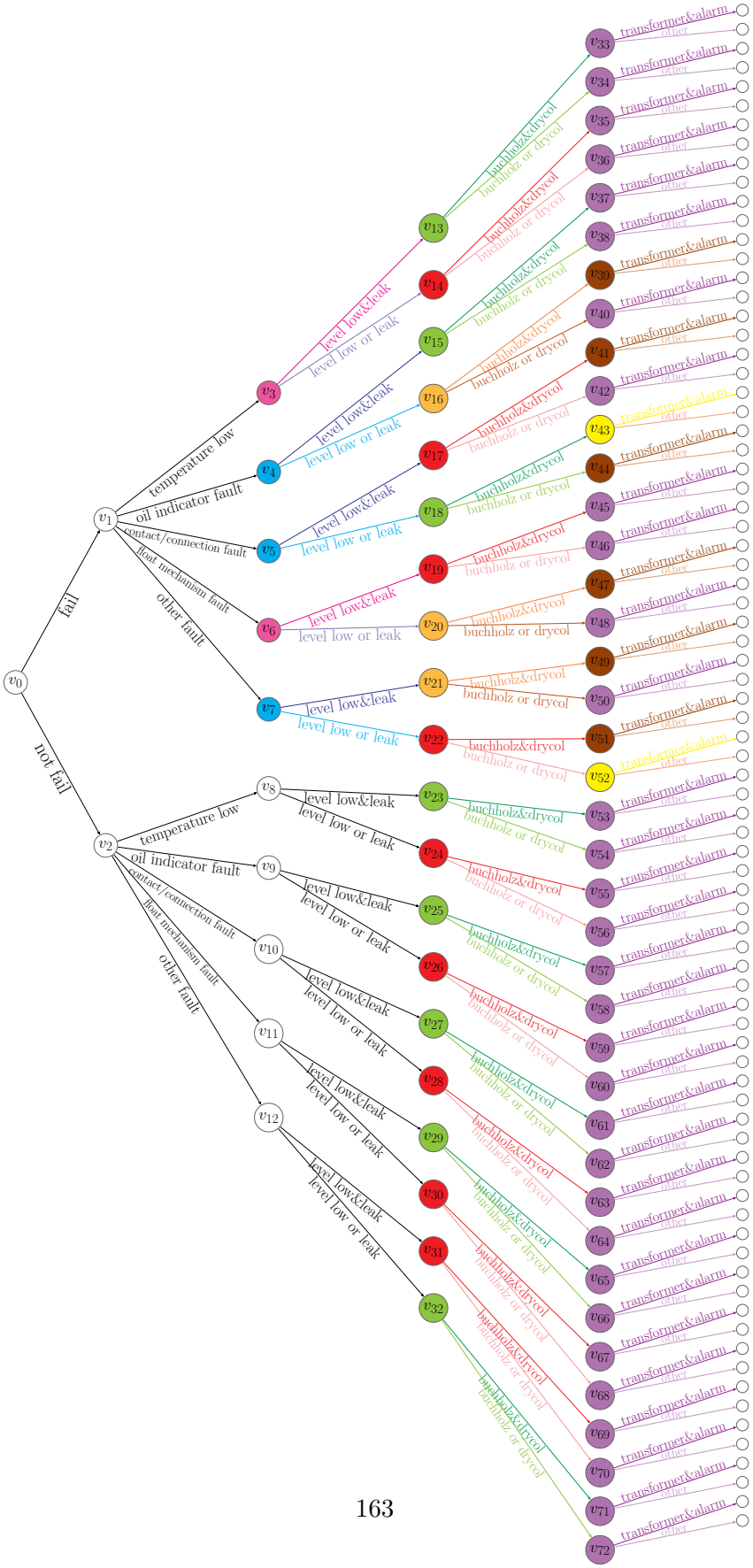


Figure 5.19: The learning event tree for the conservator system.

The estimated staged tree that best explains the observed core events is plotted in Figure 5.19 and the corresponding CEG is plotted in Figure 5.20. Conditioned on failure, oil level low and oil leak are more likely to happen as a result of the low temperature or high humidity or the float mechanism fault than other root causes that can lead to system failure. The estimated conditional probability for the former is 0.0825, which is approximately 0.072 higher that the probability of oil level low and leak conditioned on any of other root causes. There are three stages for the situations whose emanating edges representing the condition of the buchholz and the drycol unit, coloured in green, red and orange respectively in the figures. There is no evidence showing that the buchholz fault and the drycol fault depend on the condition of oil given a failed system. The mean posterior probability of having both components faulted given position w_{13} is 0.990. This is higher than the estimation of this probability conditional on any of w_{14}, w_{15}, w_{16}, which is 0.588. It is also higher than that conditional on either w_{17} or w_{18}, which is 0.355. There are three stages for the situations whose emanating edges represent the condition of the transformer and the alarm. There are three positions associated with these stages, w_{19}, w_{20}, w_{21} respectively. The estimated mean posterior probability of transformer fault and active alarm is $0.574, 0.065, 0.001$ conditional on these three stages respectively.

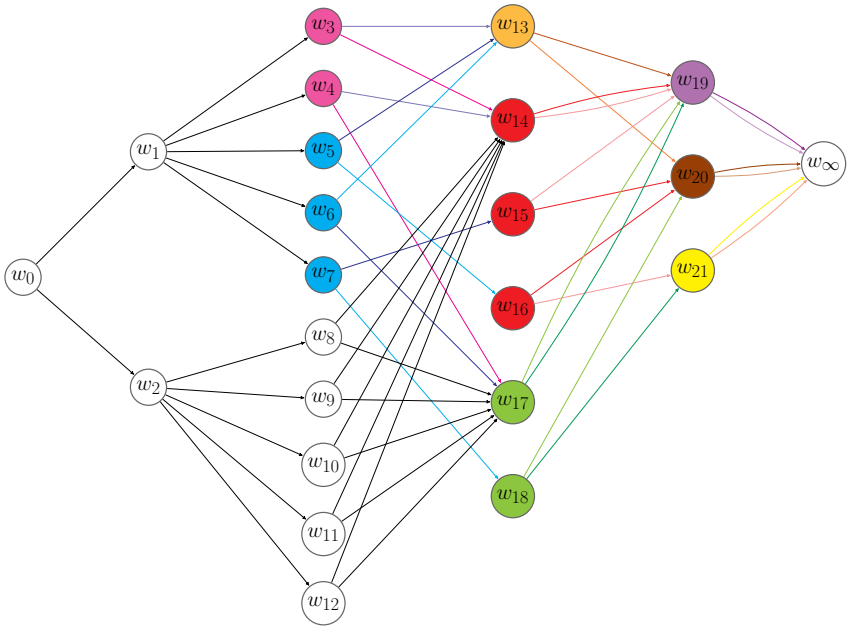


Figure 5.20: The CEG for the conservator system.

Conditioned on a system failure, the condition of the buchholz and the drycol

unit does not affect the probability of having transformer defect and active alarm when the failure is caused by temperature low or oil indicator fault with low oil level and leak. By learning from the conservator dataset, we also find that "other fault" is mostly likely to contribute to the conservator system failure, with transition probability estimate 0.586. This makes sense since we summarised 14 different types of causes for this category. Apart from this category, the contact or connection fault should be paid attention by the engineers, with estimated transition probability 0.293. More importantly, from our results, we find that the damaged component is commonly a contact or connection fault. This is followed by the seal deterioration and the seal integrity defect. The emission probability of observing a damaged component or observing a seal fault when having a contact or connection fault is estimated to be 0.692 and 0.158 respectively. Furthermore, the oil indicator fault is most likely to be led by the gauge defect, with probability 0.88; the float mechanism fault is most likely to be led by the float chamber defect, with probability 0.423.

Chapter 6

Discussion

In this chapter, I will first summarise the contributions of this thesis in Section 6.1. This will be followed by a brief discussion of the extension of the methodology introduced in this thesis. Specifically, Section 6.2 considers a type of missingness which is not covered in Chapter 4 – missing subpaths on the tree. In this case, the ground truth tree we assumed for the selected system is not realistic. In Section 6.3, we will discuss the potential of developing a dynamic extension of the GN-CEG framework which can better capture the features of longitudinal data. In Section 6.4, we list the other potential extensions of the current work.

6.1 Summary

The main contributions of this thesis are threefold. First of all, in Chapter 2, I have introduced a general approach to customise a CEG for a selected system in the domain of system engineering. This is a novel application of CEGs. On the basis of the context-specific CEG which is assumed to be causal, I have devised two domain-specific interventions: the remedial intervention and the routine intervention. The causal algebras for the former have been informed by exploring the features of the remedial work recorded in the maintenance logs by the field engineers. The causal algebras for the latter have been informed by background knowledge, especially the features of scheduled maintenance for preventive purpose. These are the central development in this thesis. I have demonstrated how to import the direct effects of these two types of interventions onto CEGs and shown that the semantics of CEGs are expressive in representing various types of asymmetric manipulations imposed by these interventions. The identifiability of the causal effects through the remedial or routine intervention has been proved by adapting the back-door theorem.

The second contribution of this thesis is the hierarchical model proposed for embedding causal relationships from texts. This is innovative as an application of CEG and as a causality embedding method. In Chapter 3, I have proposed a sequence of algorithms to extract the partially ordered core events from maintenance logs based on naive linguistic patterns. The shallow causal dependencies of these core events are then embedded into the GN. For a probabilistic analysis, these causal relations are further mapped to the semantics of a causal CEG.

In Chapter 4, I have given a concise discussion of missingness that can be captured in the hierarchical model. I have explicitly demonstrated this new missingness technology on the CEG and the GN respectively. I have defined the floret-dependent missingness and shown how to transform the original event tree to the m-tree with this type of missingness so that the M-CEG can then be derived from it. The back-door theorems for identifying effects of the remedial intervention and the routine intervention have been extended to the M-CEG. Another type of missingness that has been specified is event-dependent missingness. The M-GN is constructed from the underlying GN by accommodating this type of missingness.

Based on these innovations, examples have been given in Chapter 5 to illustrate the extent by which predictive inference can be improved when incorporating the customised causal algebras to the learning algorithm. In addition, an example of modelling the general process for the GN-CEG model and a simple version of the Gibbs algorithm for inferring the latent states and estimating parameters from this process have been given. This has helped to validate the applicability of the proposed model for causality embedding.

6.2 Missing subpaths

When constructing the event tree for the real-world data, a problem could emerge: there exist unknown failure processes or deteriorating processes to the domain experts. Then the event tree we construct is not "faithful" to reflect all the possible processes. In this case, we may have multiple or single full/partial root-to-leaf path missing.

If there exist core event variables taking values $\boldsymbol{l}_d \in \mathbb{L}$ for some document d, such that there does not exist a root-to-sink path on the CEG associated with $\boldsymbol{l}_d$, then the mapping $\chi : (\boldsymbol{l}_d, \Omega_{NC}) \mapsto \lambda$ is not well-defined any more. Let $\boldsymbol{l}_d^{k_1:k_d} = \{l_{d,k_1}, \cdots, l_{d,k_d}\} \subseteq \boldsymbol{l}_d$ denote the set of values taken by some core event variables that do not have the associated d-events labelled on any edge on the event tree or the CEG. This means the d-event space is also not complete. New d-events need to

be defined for $\boldsymbol{l}_d^{k_1:k_d}$. Denote this set of missing d-events as $\boldsymbol{x}_{[\boldsymbol{l}_d^{k_1:k_d}]}$. Note that this set of missing d-events are likely to be unidentified yet. Then the space of d-events is extended to $\mathbb{X}_{\mathcal{T}} \cup \boldsymbol{x}_{[\boldsymbol{l}_d^{k_1:k_d}]}$. Accordingly, on the event tree, the new set of root-to-leaf paths associated with the d-events $\boldsymbol{x}_{[\boldsymbol{l}_d^{k_1:k_d}]}$ are $\Lambda_{\boldsymbol{x}_{[\boldsymbol{l}_d^{k_1:k_d}]}}$. The primitive probabilities still need to satisfy $\sum_{v' \in ch(v)} \theta_{v,v'} = 1$ and $\theta_{v,v'} > 0$ after adding the new paths.

If $\boldsymbol{l}_d^{k_1:k_d}$ are associated with symptoms or sequence of faults, it is easy to determine the corresponding missing d-events when we know the root cause or observe the root cause from $\boldsymbol{l}_d \setminus \boldsymbol{l}_d^{k_1:k_d}$. Suppose $x_{rc}^{\star}$ is the root cause of these missing symptoms. Then $\boldsymbol{x}_{[\boldsymbol{l}_d^{k_1:k_d}]}$ should be added to the set of d-events associated with the symptoms conditioned on $x_{rc}^{\star}$. On the tree, new paths $\Lambda_{\boldsymbol{x}_{[\boldsymbol{l}_d^{k_1:k_d}]}}$ also need to be added to $\Lambda_{x_{rc}^{\star}}$. Specifically, new subpaths rooted at $W(x_{rc}^{\star})$ and terminating in the leaves of the tree, denoted by $\mu(W(x_{rc}^{\star}))$, are added to the tree.

If the root cause of the observations $\boldsymbol{l}_d^{k_1:k_d}$ is unknown, then various tests are required to be performed to diagnose this new category of failure. Alternatively, random maintenance can be conducted so that we deduce the cause from the status of the system after maintenance according to the discussion of different types of remedies we defined in Chapter 2. Let $\tilde{x}_{rc}$ denote the d-event associated with the new root cause. Then the edges associated with this d-event, denoted by $E(\tilde{x}_{rc})$, must be added to the florets representing root causes and $E(\boldsymbol{x}_{[\boldsymbol{l}_d^{k_1:k_d}]})$ should be added to the florets following $E(\tilde{x}_{rc})$.

If $\boldsymbol{l}_d^{k_1:k_d}$ are associated with root causes, then the new d-event $x_{\boldsymbol{l}_d^{k_1:k_d}}$ represents a root cause. The set of edges $E(x_{\boldsymbol{l}_d^{k_1:k_d}})$ are added to the florets representing root causes and the florets associated with symptoms are attached to each of $E(x_{\boldsymbol{l}_d^{k_1:k_d}})$. Then new subpaths are created which end in failure indicators.

The difficulty lies in automating this process given a dataset. Future work could develop algorithms for the paths matching from the GN to the event tree whilst adding vertices and edges when detecting the core event variables that are not associated with any existing d-event.

6.3 Dynamic processes

We can also extend the CEG to apply our methodology within a dynamic analysis to model the recovery process and the post-intervention deteriorating or failure process. This would provide statistical analyses not dissimilar to those of the reduced dynamic CEG (RDCEG) proposed by Shenvi and Smith [2018] but now applied and adjusted to this new domain.

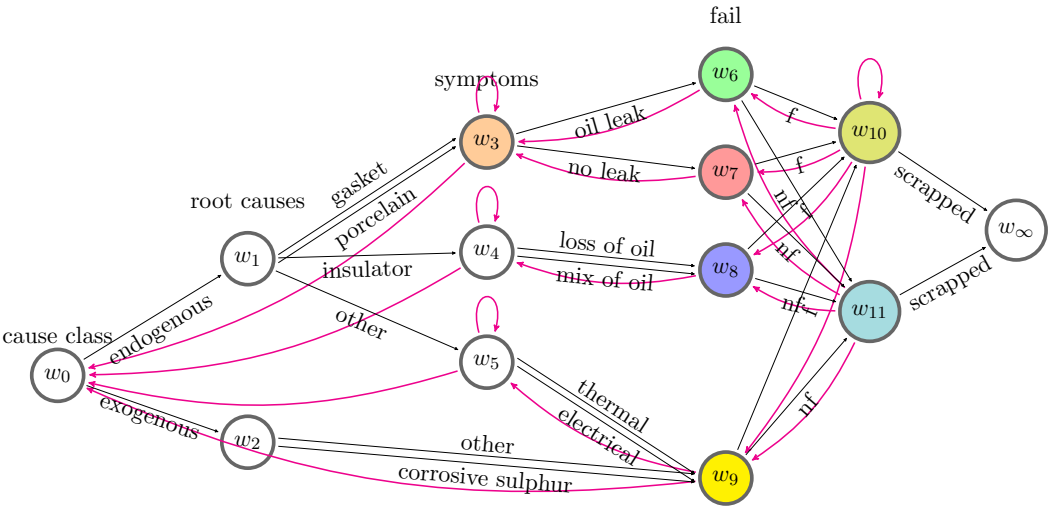


Figure 6.1: The RDCEG constructed for a bushing system.

We use the bushing system example to explain how this dynamic process can be modelled. The original CEG is plotted in Figure 5.3. The RDCEG devised for this system is shown in Figure 6.1. First, we add an **absorbing node** to represent that the machine has dropped out from the analysis. In this context this would correspond to the machine being scrapped. This information is provided by the engineers' report, so it is observable. This absorbing node is now the sink node and the failure and working sink nodes, w_∞^f and w_∞^n, are relabelled as positions w_{10} and w_{11}. Here we simply assume that whether the machine is scrapped depends only on whether the machine has failed.

We next add edges to represent the transitions of the status of the machine after maintenance. We call such edges the **retroactive edges** (retro-edges). To distinguish the original edges and the retro edges, we change the colour of all the original edges to black and colour the retro-edges red. The vertices w_1, w_2 whose emanating edges representing root causes represent the AGAN status. The root floret classifies endogenous and exogenous root causes, so we assume the status at w_0 is also AGAN. This is because, when maintenance is carried out at w_3, and if the gasket/porcelain fault is fixed, then it make more sense to return to w_0 instead of w_1 because what happens next could be an exogenous event.

Here we assume that there is no maintenance scheduled for completely new systems. In other words, the maintenance only takes place when the system is worn-out or failed. Therefore, there is no intervention at w_0, w_1, w_2. On the other hand, if w_i, w_j do not correspond to the AGAN status, and there is an edge e_{w_i,w_j} in the original CEG, then we add the retro-edge e_{w_j,w_i} to the tree structure. If w_j

is associated with the AGAN status and $w_i \neq w_0$, then we add e_{w_j,w_0}. There could also be self-loops for some positions. If the maintenance does not improve or worsen the current status, then a self-loop is added. For example, if the insulator is faulty, then we are at position w_4. If there is preventive maintenance scheduled here, which fails to renew the status of the insulator, then the status cannot transit to w_1. If there is also no loss of oil or mix of oil after the maintenance, then it cannot transit to w_8. In this case, it stays at w_4 after maintenance, so we add an edge emanating from w_4 and received by w_4. The self-loop can happen at w_3, w_4, w_5, w_{10}. There is no self loop at any of w_6, w_7, w_8, w_9 because their emanating edges correspond to failure indicators, which already represent status change. If the status is not reversed by the maintenance, then there is a transition to w_{11}. So there is no need to add more edges for these positions.

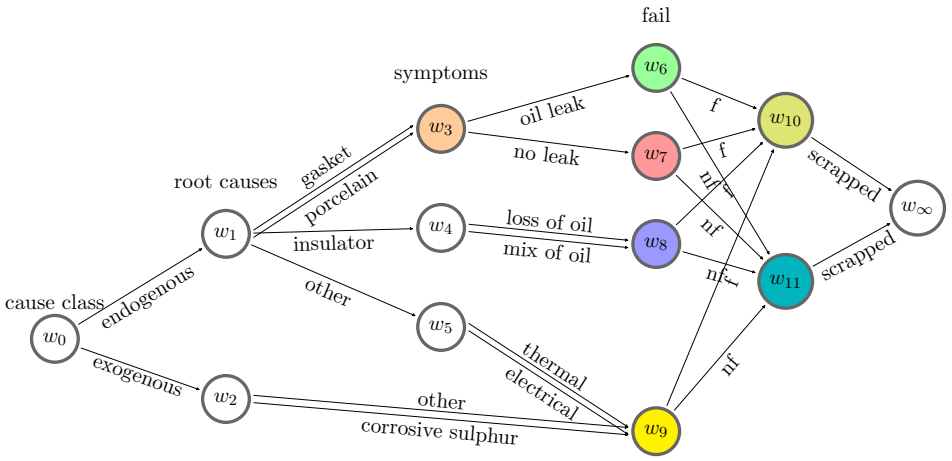


Figure 6.2: The RDCEG constructed for a non-intervened bushing system.

Most importantly, the transitions along retro-edges are only triggered after any intervention because they correspond to the immediate status change caused by the maintenance before the deterioration starts again. When the system is deteriorating between two consecutive interventions, the transition probability assigned to any retro-edge e is 0. In particular, deteriorations are analysed on the tree in Figure 6.2. However, when an intervention is imported to the system, a retro-edge is traversed as an immediate effect of the intervention. We depict this recovery process on Figure 6.1. In this case, the transition probabilities of all the edges on this tree should be redefined for this revised topology so that $\sum_{v' \in ch(v)} \tilde{\theta}_{v,v'} = 1$ and $\tilde{\theta}_{v,v'} > 0$ are still satisfied. For example, if a system is deteriorating and has problems with the insulator so that its status arrives at w_4, then the status can only transit from w_4 to w_8 either along $e^1_{w_4,w_8}$ or $e^2_{w_4,w_8}$. The corresponding primitive

probabilities are therefore defined on these two edges as $\theta_{e^1_{w_4,w_8}}$ and $\theta_{e^2_{w_4,w_8}}$ so that $\theta_{e^1_{w_4,w_8}} + \theta_{e^2_{w_4,w_8}} = 1$ and $\theta_{e^1_{w_4,w_8}}, \theta_{e^2_{w_4,w_8}} > 0$. If there is an intervention being carried out at w_4, then we turn to Figure 6.1. After the maintenance, the condition of the system may be improved then the unit transits along e_{w_0,w_4}; or the condition stays the same then the unit transits along e_{w_4,w_4}; or the condition gets worse if the maintenance did not stop the deteriorating process, then the unit transits along $e^1_{w_4,w_8}$ or $e^2_{w_4,w_8}$. Then

$$\tilde{E}(w_4) = \{e_{w_0,w_4}, e_{w_4,w_4}, e^1_{w_4,w_8}, e^2_{w_4,w_8}\}. \tag{6.3.1}$$

Correspondingly, the primitive probabilities $\tilde{\theta}_{e_{w_0,w_4}}, \tilde{\theta}_{e_{w_4,w_4}}, \tilde{\theta}_{e_{w_4,w_8}}, \tilde{\theta}_{e^1_{w_4,w_8}}$ and $\tilde{\theta}_{e^2_{w_4,w_8}}$ need to be specified so that

$$\tilde{\theta}_{e_{w_0,w_4}} + \tilde{\theta}_{e_{w_4,w_4}} + \tilde{\theta}_{e_{w_4,w_8}} + \tilde{\theta}_{e^1_{w_4,w_8}} + \tilde{\theta}_{e^2_{w_4,w_8}} = 1 \tag{6.3.2}$$

and $\tilde{\theta}_e > 0$ for $e \in \tilde{E}(w_4)$.

We can also extend the causal algebras customised for the remedial intervention and the routine intervention on the RDCEG. Given a failure, the machine arrived at the position w_{10}. A perfect remedial intervention will brings back the status to w_0 whatever the type of failure is. However, which retro-subpath $\mu(w_{10}, w_0)$ is traversed depends on the observed failure process. For example, if the failure process passes through w_1, w_4, w_8, w_{10}, then the retro-subpath traverses $e_{w_{10},w_8}, e_{w_8,w_4}, e_{w_4,w_0}$. The manipulation of the distribution over $\mathcal{F}(w_1)$ imposed by the remedial intervention are then imported into the idle system in the same way as we discussed in Section 2.3. Then the transition probabilities $\boldsymbol{\theta}_{w_1}$ are updated.

For a routine intervention, one might intervene at any position in $\{w_3, w_4, \cdots, w_{11}\}$. For example, if the scheduled maintenance fixed the oil leak at position w_6 to some extent, then the status maybe returned to w_3 by transitioning along e_{w_6,w_3} or returned to w_0 by transitioning e_{w_6,w_3} and e_{w_3,w_0}. There can also be a stochastic manipulation on $\mathcal{F}(w_3)$ depending on the extent to which the oil leak is fixed. Note that only transition probabilities along $E(w_3) = \{e_{w_3,w_6}, e_{w_3,w_7}\}$ are manipulated. No retro-edges are involved. This is because the effect on the intervened florets for predictive inference instead of the immediate effect on the status.

On the surface layer, what could happen is more complex. The definition of the topology of the GN as a DAG is not necessary anymore. To model the dynamic process with intervention, the structure could be cyclic. We can then add a maintenance variable R to the GN. There are edges pointing from other core event variables, representing root causes or faults or failure, to R because if the

maintenance is remedial then these events could be the cause of R. At the same time, the maintenance may affect the events related to cause or faults or failure. When this is the case there are edges pointing from R to the affected faults. This, therefore, may make the directed graph cyclic. The dynamic GN is also likely to evolve with time so that each time interval may have a unique GN.

6.4 Other potential future work

Apart from the two aspects of future research given above, we next briefly discuss a few other perspectives of extensions of the methodology demonstrated in this thesis.

Firstly, since designing NLP algorithms was not the primary development within this thesis, we only adopted and adapted existing software to process the texts. And we only evaluated the algorithms in the simplest way by assessing its accuracy. However, although this is only the preprocessing stage, the accuracy of text processing is actually not unimportant especially because we register the output on the GN for further analysis. It is obviously desirable if faster and more accurate NLP algorithms can be developed to extract the causally ordered events. Specifically, our methods involve much human work, for example, designing rules to classify the core events into core event variables. If the advanced methods can be proposed to reduce the human involvement, tuning of parameters and to automate the process, then the preprocessing step can be simplified and accelerated. The evaluation of the preprocessing step can also be improved to check how robust the proposed algorithms are.

Secondly, as we mentioned in Chapter 3 when defining the GN, we allow flexibility in choosing this causal network. So future work can explore alternative causal graphs. One possibility is to use a causal CEG. We have highlighted the advantages of a tree graph throughout this thesis. If the core events can be registered on the edges of the tree, then we can construct an event tree which is highly likely to be asymmetric lying at the surface layer of the model. However, as we mentioned in Chapter 3, it could be more complicated than using the causal DAG as we proposed in the thesis. Below we list the main challenges which maybe encountered if the GN is a tree. Firstly, how to automate the process of constructing the tree and then deriving the CEG? We definitely hope to avoid much human involvement here. Secondly, if the event tree is not too large, we maybe able to use the existing model search algorithms for CEGs [Collazo and P.G., 2017] and assert the best scoring model causal. If we have large size of the extracted core events and large size of the underlying partial orderings, then it could be challenging to score all of the

massive number of alternative models in a feasible time. Thirdly, the missingness of the events may not be that easy to be embraced. Fourthly, translating the causal relations from one tree to another tree is a completely novel aspect of research. It is also worth working on advanced algorithms to directly map the core events and their partial orderings to the deeper layer CEG, then the surface layer GN is not required.

In the experiment chapter, we use a Gibbs sampler, which is not easy to converge and is not fast, as is the case for any Markov chain Monte Carlo (MCMC) method. Especially, if we have a real world example, the tree topology is large and the dataset has large size, then the drawbacks of Gibbs sampler become more obvious. Alternative algorithms can be designed for learning and predicting for the GN-CEG model. Either the Expectation-Maximisation (EM) algorithm or the variational inference [Blei et al., 2003] is a popular method for approximation. So future work can focus on developing algorithms to elaborate the Gibbs algorithm proposed in this thesis for faster and more accurate estimation. In addition to this, we also mentioned in Chapter 5 that an algorithm to learn the latent paths and the structure of the CEG still needs to be designed.

The fourth aspect of extension is to elaborate the model by considering covariates such as the amount of oil topped-up each time, the expense of maintenance, the engineers who carry out the maintenance and so on. These can all be taken into account when inferring the causal effects.

A numeric perspective of the future research is an exploration of the missingness in the model. Consider the case when in the presence of missing values, for a floret, only a subset of the events labelled on the edges in this floret are likely to be missing. Then the events which are always observed represented by this florets are correlated. This may induce correlations between different florets too. Then the Dirichlet prior independence assumption made on the tree is not valid anymore. This is a very important area when dealing with real word data where the missing events can also be correlated. Moreover, there are existing methods for dealing with missing data, such as multiple imputation and bootstrapping [Efron, 1994; Schomaker and Heumann, 2018]. If we use these methods to re-estimate the primitive probabilities, we may keep the CEG without reconstructing it. Experiments can be designed to compare it with the M-CEG.

Lastly, although the framework proposed in this thesis is customised for system reliability, many of the concepts and algorithms can be transformed to other domains. For example, in the domain of medication, the electronic health records (EHR) provides the trajectory of patients condition and treatments. These function

as maintenance logs. We can extract the partially ordered core events from these records and project them on a causal CEG. The effects of medical interventions can be analysed by paralleling the intervention calculus introduced in this thesis on the CEG.

Although there is therefore considerable amount of methodological development to complete this rich research programme, I hope I have demonstrated the promise of using CEG based methods to explore latent causal mechanisms embedded in reliability systems.

Bibliography

Al Abri, T., Lal, M., Al Balushi, I. and Al Zedjali, M. [2017], 'Bushing failure-investigation process & findings', *Procedia engineering* **202**, 88–108.

Attwell, D. and Smith, J. [1991], 'A bayesian forecasting model for sequential bidding', *Journal of Forecasting* **10**(6), 565–577.

Barclay, L. M., Collazo, R. A., Smith, J. Q., Thwaites, P. A., Nicholson, A. E. et al. [2015], 'The dynamic chain event graph', *Electronic Journal of Statistics* **9**(2), 2130–2169.

Barclay, L. M., Hutton, J. L. and Smith, J. Q. [2013], 'Refining a bayesian network using a chain event graph', *International Journal of Approximate Reasoning* **54**(9), 1300–1309.

Barclay, L. M., Hutton, J. L., Smith, J. Q. et al. [2014], 'Chain event graphs for informed missingness', *Bayesian Analysis* **9**(1), 53–76.

Barlow, R. E. and Proschan, F. [1996], *Mathematical theory of reliability*, SIAM.

Bedford, T., Cooke, R. et al. [2001], *Probabilistic risk analysis: foundations and methods*, Cambridge University Press.

Bicen, Y. [2015], Monitoring of critical substation equipment, *in* '2015 3rd International Istanbul Smart Grid Congress and Fair (ICSG)', IEEE, pp. 1–4.

Bird, S. [2006], Nltk: the natural language toolkit, *in* 'Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions', pp. 69–72.

Björne, J., Ginter, F., Pyysalo, S., Tsujii, J. and Salakoski, T. [2010], 'Complex event extraction at pubmed scale', *Bioinformatics* **26**(12), i382–i390.

Blei, D. M. and Lafferty, J. D. [2006], Dynamic topic models, *in* 'Proceedings of the 23rd international conference on Machine learning', pp. 113–120.

Blei, D. M. and Lafferty, J. D. [2007], 'A correlated topic model of science', *The annals of applied statistics* **1**(1), 17–35.

Blei, D. M., Ng, A. Y. and Jordan, M. I. [2003], 'Latent dirichlet allocation', *the Journal of machine Learning research* **3**, 993–1022.

Borgia, O., De Carlo, F., Peccianti, M. and Tucci, M. [2009], 'The use of dynamic object oriented bayesian networks in reliability assessment: a case study', *Recent advances in maintenance and infrastructure management. London: Springer-Verlag* pp. 153–170.

Bryant, R. E. [1995], Binary decision diagrams and beyond: Enabling technologies for formal verification, *in* 'Proceedings of IEEE International Conference on Computer Aided Design (ICCAD)', IEEE, pp. 236–243.

Cai, B., Kong, X., Liu, Y., Lin, J., Yuan, X., Xu, H. and Ji, R. [2018], 'Application of bayesian networks in reliability evaluation', *IEEE Transactions on Industrial Informatics* **15**(4), 2146–2157.

Casini, L., Illari, P. M., Russo, F. and Williamson, J. [2011], 'Models for prediction, explanation and control: recursive bayesian networks', *THEORIA. Revista de Teoría, Historia y Fundamentos de la Ciencia* **26**(1), 5–33.

Chambers, N., Cassidy, T., McDowell, B. and Bethard, S. [2014], 'Dense event ordering with a multi-pass architecture', *Transactions of the Association for Computational Linguistics* **2**, 273–284.

Chen, D., Cao, Y. and Luo, P. [2020], Pairwise causality structure: Towards nested causality mining on financial statements, *in* 'CCF International Conference on Natural Language Processing and Chinese Computing', Springer, pp. 725–737.

Collazo, R. A., Görgen, C. and Smith, J. Q. [2018], *Chain event graphs*, CRC Press.

Collazo, R. and P.G., T. [2017], *ceg: Chain Event Graph.* R package version 0.1.0.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P. [2011], 'Natural language processing (almost) from scratch', *Journal of machine learning research* **12**(ARTICLE), 2493–2537.

Cooper, G. F. and Yoo, C. [2013], 'Causal discovery from a mixture of experimental and observational data', *arXiv preprint arXiv:1301.6686* .

Cowell, R. G., Smith, J. Q. et al. [2014], 'Causal discovery through map selection of stratified chain event graphs', *Electronic Journal of Statistics* **8**(1), 965–997.

Dasgupta, T., Saha, R., Dey, L. and Naskar, A. [2018], Automatic extraction of causal relations from text using linguistically informed deep neural networks, *in* 'Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue', pp. 306–316.

Dawid, A. P. [2000], 'Causal inference without counterfactuals', *Journal of the American statistical Association* **95**(450), 407–424.

Dawid, A. P. [2002], 'Influence diagrams for causal modelling and inference', *International Statistical Review* **70**(2), 161–189.

Dawid, A. P. and Didelez, V. [2005], Identifying the consequences of dynamic treatment strategies, Technical report, Citeseer.

Didelez, V., Dawid, P. and Geneletti, S. [2012], 'Direct and indirect effects of sequential treatments', *arXiv preprint arXiv:1206.6840* .

Doyen, L. and Gaudoin, O. [2004], 'Classes of imperfect repair models based on reduction of failure intensity or virtual age', *Reliability Engineering & System Safety* **84**(1), 45–56.

Dudhabaware, R. S. and Madankar, M. S. [2014], Review on natural language processing tasks for text documents, *in* '2014 IEEE International Conference on Computational Intelligence and Computing Research', IEEE, pp. 1–5.

Eaton, D. and Murphy, K. [2007], Exact bayesian structure learning from uncertain interventions, *in* 'Artificial intelligence and statistics', PMLR, pp. 107–114.

Eddy, S. R. [2004], 'What is a hidden markov model?', *Nature biotechnology* **22**(10), 1315–1316.

Efron, B. [1994], 'Missing data, imputation, and the bootstrap', *Journal of the American Statistical Association* **89**(426), 463–475.

Feinerer, I., Hornik, K. and Meyer, D. [2008], 'Text mining infrastructure in r', *Journal of statistical software* **25**, 1–54.

Fenton, N. and Neil, M. [2018], *Risk assessment and decision analysis with Bayesian networks*, Crc Press.

Freeman, G. [2010], Learning and predicting with chain event graphs, PhD thesis, University of Warwick.

Freeman, G. and Smith, J. Q. [2011*a*], 'Bayesian map model selection of chain event graphs', *Journal of Multivariate Analysis* **102**(7), 1152–1165.

Freeman, G. and Smith, J. Q. [2011*b*], 'Dynamic staged trees for discrete multivariate time series: forecasting, model selection and causal analysis', *Bayesian Analysis* **6**(2), 279–305.

Gautam [2021], 'Preventive maintenance checklist for transformer', `https://learnelectrician.com/preventive-maintenance-checklist-for-transformer/`.

Gillies, D. [2018], *Causality, probability, and medicine*, Routledge.

Görgen, C. and Smith, J. Q. [2016], A differential approach to causality in staged trees, *in* 'Conference on Probabilistic Graphical Models', PMLR, pp. 207–215.

Görgen, C. and Smith, J. Q. [2018], 'Equivalence classes of staged trees', *Bernoulli* **24**(4A), 2676–2692.

Guessoum, Y. and Aupied, J. [2010], Modelling the impact of preventive maintenance over the lifetime of equipments, *in* 'CIRED Workshop'.

Gupta, G. and Malhotra, S. [2015], 'Text document tokenization for word frequency count using rapid miner (taking resume as an example)', *Int. J. Comput. Appl* **975**, 8887.

He, S., Zhang, T., Bai, X., Wang, X. and Dong, Y. [2009], Incorporating multi-task learning in conditional random fields for chunking in semantic role labeling, *in* '2009 International Conference on Natural Language Processing and Knowledge Engineering', IEEE, pp. 1–5.

Heckerman, D. [2008], 'A tutorial on learning with bayesian networks', *Innovations in Bayesian networks* pp. 33–82.

Heckerman, D., Geiger, D. and Chickering, D. M. [1995], 'Learning bayesian networks: The combination of knowledge and statistical data', *Machine learning* **20**(3), 197–243.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. O., Padó, S., Pennacchiotti, M., Romano, L. and Szpakowicz, S. [2019], 'Semeval-2010 task 8:

Multi-way classification of semantic relations between pairs of nominals', *arXiv preprint arXiv:1911.10422* .

Hirschberg, J. and Manning, C. D. [2015], 'Advances in natural language processing', *Science* **349**(6245), 261–266.

Honnibal, M. and Montani, I. [2017], 'spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing', *To appear* **7**(1), 411–420.

Hull, E., Jackson, K. and Dick, J. [2010], System modelling for requirements engineering, *in* 'Requirements Engineering', Springer, pp. 47–76.

Hund, L. and Schroeder, B. [2020], 'A causal perspective on reliability assessment', *Reliability Engineering & System Safety* **195**, 106678.

Iung, B., Veron, M., Suhner, M.-C. and Muller, A. [2005], 'Integration of maintenance strategies into prognosis process to decision-making aid on system operation', *CIRP annals* **54**(1), 5–8.

Jaeger, M., Nielsen, J. D. and Silander, T. [2006], 'Learning probabilistic decision graphs', *International Journal of Approximate Reasoning* **42**(1-2), 84–100.

Jeude, J. v. L. d., French, S. and Mackay, R. [2015], 'University of warwick – national grid: Data driven asset management defect analysis and dissolved gas analysis'.

Kijima, M. [1989], 'Some results for repairable systems with general repair', *Journal of Applied probability* **26**(1), 89–102.

Kostolny, J., Kvassay, M. and Zaitseva, E. [2014], Analysis of algorithms for computation of direct partial logic derivatives in multiple-valued decision diagrams, *in* '2014 Ninth International Conference on Availability, Reliability and Security', IEEE, pp. 356–361.

Krieger, N. and Davey Smith, G. [2016], 'The tale wagged by the dag: broadening the scope of causal inference and explanation for epidemiology', *International journal of epidemiology* **45**(6), 1787–1808.

Langseth, H. and Portinale, L. [2007], 'Bayesian networks in reliability', *Reliability Engineering & System Safety* **92**(1), 92–108.

Lee, W.-S., Grosh, D. L., Tillman, F. A. and Lie, C. H. [1985], 'Fault tree analysis, methods, and applications a review', *IEEE transactions on reliability* **34**(3), 194–203.

Li, J. and Shi, J. [2007], 'Knowledge discovery from observational data for process control using causal bayesian networks', *IIE transactions* **39**(6), 681–690.

Liddy, E. D. [2001], 'Natural language processing'.

Lienig, J. and Bruemmer, H. [2017], Reliability analysis, *in* 'Fundamentals of electronic systems design', Springer, pp. 45–73.

Lisnianski, A., Frenkel, I. and Karagrigoriou, A. [2017], *Recent advances in multi-state systems reliability: Theory and applications*, Springer.

Loper, E. and Bird, S. [2002], 'Nltk: The natural language toolkit', *arXiv preprint cs/0205028* .

Maintenance Tips for Electrical Transformers [2020], `https://www.dfliq.net/blog/maintenance-tips-electrical-transformers/`.

Mani, S. and Cooper, G. F. [1999], A study in causal discovery from population-based infant birth and death records., *in* 'Proceedings of the AMIA Symposium', American Medical Informatics Association, p. 315.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S. and McClosky, D. [2014], The stanford corenlp natural language processing toolkit, *in* 'Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations', pp. 55–60.

Miller, G. A. [1995], 'Wordnet: a lexical database for english', *Communications of the ACM* **38**(11), 39–41.

Mirza, P. and Tonelli, S. [2016], Catena: Causal and temporal relation extraction from natural language texts, *in* 'Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers', pp. 64–75.

Mohamed, H., Omar, N. and Ab Aziz, M. J. [2011], Statistical malay part-of-speech (pos) tagger using hidden markov approach, *in* '2011 International Conference on Semantic Technology and Information Retrieval', IEEE, pp. 231–236.

Mohan, K. [2017], Graphical models for inference with missing data, PhD thesis, UCLA.

Mohan, K. and Pearl, J. [2014], 'Graphical models for recovering probabilistic and causal queries from missing data', *Advances in Neural Information Processing Systems* **27**.

Mohan, K. and Pearl, J. [2021], 'Graphical models for processing missing data', *Journal of the American Statistical Association* pp. 1–42.

Mohan, K., Pearl, J. and Jin, T. [2013], Missing data as a causal inference problem, *in* 'Proceedings of the neural information processing systems conference (nips)'.

Nadkarni, P. M., Ohno-Machado, L. and Chapman, W. W. [2011], 'Natural language processing: an introduction', *Journal of the American Medical Informatics Association* **18**(5), 544–551.

Natvig, B. [1985], Recent developments in multistate reliability theory, *in* 'Probabilistic methods in the mechanics of solids and structures', Springer, pp. 385–393.

Nichols, L., French, S. and Mackay, R. [2017], 'National grid - university of warwick collaboration 2016-17'.

Ning, Q., Feng, Z., Wu, H. and Roth, D. [2019], 'Joint reasoning for temporal and causal relations', *arXiv preprint arXiv:1906.04941* .

Nyberg, M. [2013], Failure propagation modeling for safety analysis using causal bayesian networks, *in* '2013 Conference on control and fault-tolerant systems (SysTol)', IEEE, pp. 91–97.

Pearl, J. [1993], '[bayesian analysis in expert systems]: Comment: graphical models, causality and intervention', *Statistical Science* **8**(3), 266–269.

Pearl, J. [1995], 'Causal diagrams for empirical research', *Biometrika* **82**(4), 669–688.

Pearl, J. [2009], *Causality*, Cambridge university press.

Pensar, J., Talvitie, T., Hyttinen, A. and Koivisto, M. [2020], A bayesian approach for estimating causal effects from observational data, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 34, pp. 5395–5402.

Perrone, V., Jenkins, P. A., Spano, D. and Teh, Y. W. [2017], 'Poisson random fields for dynamic feature models', *Journal of Machine Learning Research* **18**.

Powers, D. M. [2020], 'Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation', *arXiv preprint arXiv:2010.16061* .

Pustejovsky, J., Knippen, R., Littman, J. and Saurí, R. [2005], 'Temporal and event information in natural language text', *Language resources and evaluation* **39**(2), 123–164.

Reclamation, B. [2005], 'Transformers: Basics, maintenance and diagnostics', *US Department of the Interior Bureau of Reclamation. Denver, Colorado, USA* .

Riccomagno, E. and Smith, J. [2005], 'The causal manipulation and bayesian estimation of'.

Rubin, D. B. [1976], 'Inference and missing data', *Biometrika* **63**(3), 581–592.

Rubin, D. B. [2003], 'Basic concepts of statistical inference for causal effects in experiments and observational studies', *Course material in Quantitative Reasoning* **33**.

Ruiz-Tagle, A., Lopez Droguett, E. and Groth, K. M. [2021], 'Exploiting the capabilities of bayesian networks for engineering risk assessment: Causal reasoning through interventions', *Risk Analysis* .

Saadati, M. and Tian, J. [2019], Adjustment criteria for recovering causal effects from missing data, *in* 'Joint European Conference on Machine Learning and Knowledge Discovery in Databases', Springer, pp. 561–577.

Schmitt, X., Kubler, S., Robert, J., Papadakis, M. and LeTraon, Y. [2019], A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate, *in* '2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)', IEEE, pp. 338–343.

Schomaker, M. and Heumann, C. [2018], 'Bootstrap inference when using multiple imputation', *Statistics in medicine* **37**(14), 2252–2266.

Schuler, K. K. [2005], *VerbNet: A broad-coverage, comprehensive verb lexicon*, University of Pennsylvania.

Scutari, M. [2009], 'Learning bayesian networks with the bnlearn r package', *arXiv preprint arXiv:0908.3817* .

Scutari, M. [2014], 'Bayesian network constraint-based structure learning algorithms: Parallel and optimised implementations in the bnlearn r package', *arXiv preprint arXiv:1406.7648* .

Scutari, M. and Ness, R. [2012], 'bnlearn: Bayesian network structure learning, parameter learning and inference', *R package version* **3**.

Shafer, G. [1996], *The art of causal conjecture*, MIT press.

Shenvi, A. and Smith, J. Q. [2018], 'A bayesian dynamic graphical model for recurrent events in public health', *arXiv preprint arXiv:1811.08872* .

Shenvi, A., Smith, J. Q., Walton, R. and Eldridge, S. [2018], Modelling with non-stratified chain event graphs, *in* 'International Conference on Bayesian Statistics in Action', Springer, pp. 155–163.

Shpitser, I., VanderWeele, T. and Robins, J. M. [2012], 'On the validity of covariate adjustment for estimating causal effects', *arXiv preprint arXiv:1203.3515* .

Smith, J. [1979], 'A generalization of the bayesian steady forecasting model', *Journal of the Royal Statistical Society: Series B (Methodological)* **41**(3), 375–387.

Smith, J. [1990], 'Non-linear state space models with partially specified distributions on states', *Journal of Forecasting* **9**(2), 137–149.

Smith, J. [1992], 'A comparison of the characteristics of some bayesian forecasting models', *International Statistical Review/Revue Internationale de Statistique* pp. 75–87.

Smith, J. Q. [2010], *Bayesian decision analysis: principles and practice*, Cambridge University Press.

Smith, J. Q. and Anderson, P. E. [2008], 'Conditional independence and chain event graphs', *Artificial Intelligence* **172**(1), 42–68.

Sorgente, A., Vettigli, G. and Mele, F. [2013], 'Automatic extraction of cause-effect relations in natural language text.', *DART@ AI* IA* **2013**, 37–48.

Spirtes, P., Glymour, C. N., Scheines, R. and Heckerman, D. [2000], *Causation, prediction, and search*, MIT press.

Tang, B., Wu, Y., Jiang, M., Chen, Y., Denny, J. C. and Xu, H. [2013], 'A hybrid system for temporal information extraction from clinical text', *Journal of the American Medical Informatics Association* **20**(5), 828–835.

Thwaites, P. [2008], Chain event graphs: Theory and application, PhD thesis, University of Warwick.

Thwaites, P. [2013], 'Causal identifiability via chain event graphs', *Artificial Intelligence* **195**, 291–315.

Thwaites, P., Smith, J. Q. and Riccomagno, E. [2010], 'Causal analysis with chain event graphs', *Artificial Intelligence* **174**(12-13), 889–909.

Torres-Toledano, J. G. and Sucar, L. E. [1998], Bayesian networks for reliability analysis of complex systems, *in* 'Ibero-American Conference on Artificial Intelligence', Springer, pp. 195–206.

Types of maintenance: The 9 different strategies explained [n.d.], `https://www.roadtoreliability.com/types-of-maintenance/`.

What is Preventive Maintenance? [n.d.], `https://www.twi-global.com/technical-knowledge/faqs/what-is-preventive-maintenance`.

Wilkerson, R. [2020], Customising Structure of Graphical Models, PhD thesis, University of Warwick.

Wilkerson, R. L. and Smith, J. Q. [2019], 'Bayesian diagnostics for chain event graphs'.

Williamson, J. and Gabbay, D. [2005], 'Recursive causality in bayesian networks and self-fibring networks', *Laws and models in the sciences* pp. 173–221.

Xun, G., Li, Y., Zhao, W. X., Gao, J. and Zhang, A. [2017], A correlated topic model using word embeddings., *in* 'IJCAI', pp. 4207–4213.

Young, T., Hazarika, D., Poria, S. and Cambria, E. [2018], 'Recent trends in deep learning based natural language processing', *ieee Computational intelligenCe magazine* **13**(3), 55–75.

Yu, S.-Z. [2010], 'Hidden semi-markov models', *Artificial intelligence* **174**(2), 215–243.

Yu, X. and Smith, J. Q. [2021*a*], 'Causal algebras on chain event graphs with informed missingness for system failure', *Entropy* **23**(10).

Yu, X. and Smith, J. Q. [2021*b*], 'Causal algebras on chain event graphs with informed missingness for system failure', *Entropy* **23**(10).

Yu, X. and Smith, J. Q. [2021*c*], 'Hierarchical causal analysis of natural languages on a chain event graph', *arXiv preprint arXiv:2110.01129* .

Yu, X., Smith, J. Q. and Nichols, L. [2020], 'Bayesian learning of causal relationships for system reliability', *arXiv preprint arXiv:2002.06084* .

Zhao, S., Wang, Q., Massung, S., Qin, B., Liu, T., Wang, B. and Zhai, C. [2017], Constructing and embedding abstract event causality networks from text snippets,

in 'Proceedings of the Tenth ACM International Conference on Web Search and Data Mining', pp. 335–344.

Zitouni, A., Damankesh, A., Barakati, F., Atari, M., Watfa, M. and Oroumchian, F. [2010], Corpus-based arabic stemming using n-grams, *in* 'Asia Information Retrieval Symposium', Springer, pp. 280–289.