Department of Computer Science
Czech Technical University in Prague

Applications of Learning Multiple Dynamical Systems: Joint Problems and Causal modellings

Xiaoyu He

Submitted in partial fulfilment of the requirements for the thesis proposals merged with state doctoral exam at Czech Technical University in Prague, September 25, 2025

Preface

This thesis proposal represents the accumulation of my research during my past two years PhD studies under the supervision of Jakub Mareček at Czech Technical University in Prague. It comprises three manuscripts, each addressing a real-world challenge in causal inference: temporality, confounding and joint clustering. While the chapters are based on published or submitting work, some differences exist between the thesis proposal content and the final papers; these are detailed in Chapter Contributions and Structure.

Acknowledgments

I am deeply grateful to my supervisor, Jakub Mareček, and to Petr Ryšavý and Pavel Rytíř for their patient guidance, insightful discussions, and continuous support during my PhD. I also thank my co-authors Georgios Korpas, Mengjia Niu, and Quan Zhou for the enjoyable collaborations, and my officemates for their friendship and encouragement throughout this journey. I owe profound gratitude to my family for their unconditional love, and especially to my father for his constant support throughout my studies. Finally, I thank George, my habibi, for always standing by me and for being both a true friend and a wonderful partner.

Abstract

This thesis proposal investigates causal discovery in complex biological systems through three interrelated projects, each addressing a distinct challenge in modelling dynamic data. Despite their diversity, all projects share a focus on learning linear dynamical systems and joint relationships, as well as handling real problems.

Chapter 1 introduces the main motivation for the research: the difficulty of causal discovery in dynamic biological systems. Using the Krebs cycle as a canonical benchmark, we illustrate both the scientific importance and methodological challenges of modeling cyclical structures, feedback interactions, and latent biochemical factors. This chapter also provides a brief overview of the datasets and problem settings considered in the subsequent chapters.

Chapter 2 focuses on the challenge of latent confounders in causal inference. We introduce ExMAG, a score-based branch-and-cut algorithm for learning Maximally Ancestral Graphs (MAGs), which extend directed acyclic graphs by modelling hidden variables and bidirected edges. ExMAG provides computational efficiency, global guarantees, and improved accuracy for confounder-aware causal discovery, which build a solid foundation for future studying of causal inference with the presence of unobserved variables.

Chapter 3 addresses the problem of learning causal structures across multiple interacting dynamical systems. We extend the framework to joint learning of linear dynamical systems, combining the NCPOP framework with an Expectation-Maximization heuristic. This approach enables joint time-series clustering, identification of shared latent structures, and modeling of cross-system dependencies, demonstrating the ability to generalize causal discovery across multiple related systems.

Taken together, my research leverages the Krebs cycle as a motivating benchmark, introduces ExMAG as an efficient algorithm for learning confounded causal structures, and extends the framework to joint problems for multiple dynamical systems. Ultimately, this work contributes novel theory and methodology that bridge abstract causal inference with real-world dynamical modeling, enabling robust and interpretable analysis of temporal processes in biology, medicine, and intervention-driven domains.

Contributions and Structure

The publications related to this thesis proposal are listed below, following the sequence of chapter appearances in the proposal:

- [**CausalKrebs**]. Xiaoyu He*, Petr Ryavý*, and Jakub Mareek proposed methods for causal learning in biomedical applications, using the Krebs cycle as a benchmark. This work was currently accepted and will be published in the *F1000Research Journal* shortly.
- [**ExMAG**]. Petr Ryavý, Pavel Rytí, Xiaoyu He, Georgios Korpas, and Jakub Mareek developed *ExMAG*, a method for learning maximally ancestral graphs. This work are submitted to *ICRL*.
- [**JointDynamical**]. Mengjia Niu, Xiaoyu He, Petr Ryavý, Quan Zhou, and Jakub Mareek studied joint problems in learning multiple dynamical systems. This work was accepted and had been presented at the *Allerton Conference*.
- [**ApplicationLDS**]. Xiaoyu He*, Petr Ryavý*, and Jakub Mareek explored applications of learning linear dynamical systems, which is a prototype of this these proposal. This work was presented at the *ECML-PKDD 2024 PhD Forum*.

Notes: The asterisk (*) indicates equal contribution.

Table of Notation

Symbol	Representation
$\mathcal{G}, \hat{\mathcal{G}}$	Graph and estimated graph
$\mathcal{V}$	Set of vertices
v_m, u, v, w	Vertices
$\mathrm{pa}_{\mathcal{G}}(v)$	Parents of vertex v in graph $\mathcal{G}$
$\mathrm{sp}_{\mathcal{G}}(v)$	Spouses of vertex v in graph $\mathcal{G}$
$\mathrm{an}_{\mathcal{G}}(v)$	Ancestors of vertex v in graph $\mathcal{G}$
$\mathrm{dis}_{\mathcal{G}}(v)$	District of vertex v in graph $\mathcal{G}$
$\mathrm{collider}_{\mathcal{G}}(v)$	Collider nodes in graph $\mathcal{G}$
x_t, y_t, z_t	Realistic variable
Φ	function
ϵ	Noise terms
Z	instrument variable
A	Exogenous variable
H	Unobserved term
δ	Threshold for edge weights
θ, γ	Regression parameters
$W, \hat{W}$	Weight matrices and estimated weight matrix
$\mathcal{P}$	Class of distributions P representing perturbations of the original distribution (including confoundings)
$\mathcal{Z}$	Conditioning set
p	Number of data points
i	Index of data points
h	Number of edges
q	Exponent in the cost function ($q = 1$ or $q = 2$)
L_q	Cost function
$Y_{i,m}$	Value of the m-th variable for the i-th data point
$w_{j,m}$	Weight of the edge from variable j to variable m
$e_{j,m}$	Binary variable indicating a directed edge from j to m

Symbol	Representation
$b_{j,m}$	Binary variable indicating a bidirected edge between j and m
$f_{j,m}$	Binary variable indicating no directed edge between j and m
λ	Regularization parameter
E, B	Directed / bidirected edge matrices
F	Direct causal effect matrix (0 indicates no direct effect)
β	The path that contains bidirected edges
$\mathcal{E}$,$\mathcal{U}$	Set of directed and undirected edges
$\mathcal{E}'$	Set of directed edges that participate in the ancestor relationship
D	Distance matrix of E
GT, PR	Edge type in the ground truth graph and predicted graph
t, T	Time index and time length
s, S	Sample index and sample size
k, K	Cluster index and number of clusters, i.e., $k \in \{2, \ldots, K\}$
j, m	Trajectory and variable index
M	Number of variables, nodes and trajectories
n	Hidden state dimension
$\mathbf{Y}$	Observed time-series trajectories / Observations
$Y_t, \hat{Y}_t$	Observed and estimated trajectory and variable at time t
$\mathbf{L}$	LDS system
$\boldsymbol{\varphi}$	System matrix in vector autoregressive / hidden state processes
$\mathbf{G}$	System matrix in observed process
$\mathbf{\Sigma}_H, \mathbf{\Sigma}_O$	Hidden state noise and observation error covariance matrix
X_t	Vector autoregressive / hidden state processes
ω_t	Hidden state noise
υ_t	Observation noise
$\mathbf{Y}^m$	Observation for trajectory m
$l_{m,k}, l_m$	Assignment of trajectory m when $K > 2$ and $K = 2$
C_k	Set of trajectories in cluster k
$Y_t^{l_{m,k}}, \hat{Y}_t^{l_{m,k}}$	Observations and estimations of trajectory m at time t in cluster k
$\mathbf{L}_k$	LDS system for cluster k, $\mathbf{L}_k = (\mathbf{G}_k, \mathbf{F}_k, \mathbf{\Sigma}_H^k, \mathbf{\Sigma}_O^k)$
X_t^k	Hidden state processes of $\mathbf{L}_k$
ω_t^k	Hidden state noises produced by $\mathbf{L}_k$
υ_t^k	Observation error produced by $\mathbf{L}_k$

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Causal Learning in Biomedical Applications: Krebs Cycle as a Benchmark

As a motivating benchmark for evaluating causal modelling in dynamic systems, we employ a time-series dataset derived from the Krebs cycle, a canonical biochemical pathway underlying cellular energy metabolism. Unlike the textbook representation, this time-series version is acyclic, and its non-trivial R^2-sortability enables algorithms to uncover more accurate causal relationships. The benchmark includes multiple scenarios with varying sample sizes and time horizons, together with standardized evaluation across twenty state-of-the-art methods. This provides a realistic and challenging testbed for assessing the effectiveness of our NCPOP-based framework in learning causal structures from complex, real-world temporal data.

1.1 Introduction

Understanding causal models is important in a number of fields, from healthcare to economics, as it allows for precise forecasting and training of reinforcement learning algorithms. Learning causal models involves extracting potential non-linear relationships and dependencies between variables from sampled time series. For example, the modelling of biomarkers of non-communicable disease as a function of diet and action monitoring has shown the potential of being a powerful tool to guide the recommendations for a healthy diet.

The causal learning community agrees that there is a need for better synthetic datasets to test causal learning algorithms[123, 110, 124]. Many synthetic dataset benchmarks suffer from residual information in the data that the R^2-sortability can identify. In the case of real-world datasets, we often cannot be sure what the ground-truth causal relationships are. Often, datasets for causal discovery are too large, and as a result, they are sampled without any standardized sampling approach, thus making different papers using the datasets incomparable.

In this paper, we aim to fill this gap and provide a standardized synthetic dataset that does not suffer from the problems mentioned above. The dataset is based on simulating a set of chemical reactions describing the Krebs cycle, and for that, it uses a publicly available generator at [104]. The randomness in the data is caused by simulating the molecules in a box and providing the molecules with locations and velocities. Whenever

molecules forming the left-hand side of a reaction meet, they are replaced with reactants as given by the equation.

First, we provide a brief review of a variety of methods that can be used in causal learning. Later, we provide a list of requirements that we can expect from such causal learning methods to illustrate their expressiveness. A discussion of which criteria are supported by the existing methods follows. Section 1.4 explains the dataset in detail and shows a possible evaluation of methods on the dataset. We show a comparison on 4 datasets. Next, we compare the presented dataset with other causal learning datasets. In conclusion, we give preference to public repositories where the dataset, as well as the source code for the evaluation of the method, can be found.

1.2 Preliminaries and Related Work

Learning most causal models involves solving NP-hard non-convex optimization problems. Just as there is "one" convex optimization and "many" non-convex optimization problems, there are many causal models and methods for learning them. Perhaps the most elegant approach to causal learning utilises techniques from system identification.

System Identification and Linear Dynamic Systems (LDS) Let n be the hidden state dimension and M be the observational dimension. A single realisation of the LDS of length T, denoted $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_T\} \in \mathbb{R}^{M \times S \times T}$ is defined on *initial conditions* X_0 and *system matrices* $\boldsymbol{\varphi}$ and $\mathbf{G}$ as

$$X_t = \boldsymbol{\varphi} X_{t-1} + \omega_t, \qquad \omega_t \sim N(0, \mathbf{\Sigma_H}) \in \mathbb{R}^{n \times S} \tag{1.1}$$

$$Y_t = \mathbf{G}' X_t + \upsilon_t \qquad \upsilon_t \sim N(0, \mathbf{\Sigma_O}) \in \mathbb{R}^{M \times S}, \tag{1.2}$$

where $X_t \in \mathbb{R}^{n \times S}$ is the vector autoregressive processes with hidden components and $\{\omega_t, \upsilon_t\}_{t \in \{1,2,\ldots,T\}}$ are normally distributed process and observation noises with zero mean and covariance of $\mathbf{\Sigma_H}$ and $\mathbf{\Sigma_O}$ respectively. The transpose of $\mathbf{G}$ is denoted as $\mathbf{G}'$. Vector $Y_t \in \mathbb{R}^{M \times S}$ is the observed output of the system. In non-linear dynamical systems, one replaces the multiplication $\boldsymbol{\varphi} X_{t-1}$ with a function $\Phi(X_{t-1})$. It is well known that there are multiple, equivalent conditions for the identifiability of $\boldsymbol{\varphi}, \mathbf{G}$, given by so-called Hankel matrices, conditions on the transfer function, or frequency-domain conditions, among others[161]. The is also a recent understanding of sample complexity of the problem[150].

Linear Additive Noise Models Throughout causal modelling, one wishes to learn a function Φ, which is known as the structural assignment map and is closely related to the Φ above. Under the assumption that the structural assignments are linear, noises $\epsilon_i, i = 1, \ldots, M$ are independently identically distributed (i.i.d.) and follow the same Gaussian distribution, or alternatively, noises $\epsilon_i, i = 1, \ldots, M$ are jointly independent, non-Gaussian with strictly positive density, one obtains linear additive noise models (ANM). In studying ANM, one may benefit from a long tradition of work on linear system identification. In particular, the identifiability of linear ANM can be reduced to the identifiability of linear dynamical systems (cf. Proposition 7.5 & Theorem 7.6 in [117]).

Bayesian networks Another classic example in causal learning are *Bayesian networks*, first introduced by Pearl in 1985[112]. Bayesian networks are formed by a directed acyclic graph (DAG), where each vertex j represents a variable X_j, with edges going from one variable to another representing causal relationships. It is assumed that each variable X_j is independent of other variables but for its parents, PA_j in the DAG, thus allowing a compressed representation of the joint probability as

$$P(X_1, X_2, \ldots, X_M) = \prod_{j=1}^{M} P(X_j \mid \mathbf{PA}_j). \tag{1.3}$$

The most common approach to exact inference in Bayesian networks is the variable elimination algorithm[113]. Approximate inference algorithms are also often applied. The most common one is the Markov Chain Monte-Carlo (MCMC) algorithm that repeatedly samples from each variable conditioned on the values of its parents. The MCMC algorithm predates Bayesian networks and is often referred to as Gibbs sampling.

In relation to temporal data, the Dynamic Bayesian Networks (DBN) are a well-known extension[43, 103]. DBNs are defined by two Bayesian networks. The first defines the initial state, and the second is the transition model between t and $t+1$, where nodes in layer t are assumed to be independent. The network can then be unrolled into length T so that each of the time slices for $t \geq 1$ is defined by the transition model.

Counterfactual Framework The counterfactual framework can be used to derive causality. This approach focuses on the question of which input variable needs to change in order to change the output of a model. The counterfactual framework is connected with the calculation of interventions, i.e., assessing the change of output variables after a hypothetical change of an input variable. In counterfactuals, we ask which inputs need to change to observe a change in the output, while in intervention, we change the inputs to see the change in the output. The counterfactuals were introduced into Bayesian networks by Pearl [106]. Nowadays, their usage is broad, and they find usage in explainable machine learning models[101].

Granger Causality The goal of the Granger Causality is to detect a causal effect of a time series on another time series[55]. The Granger causality measures correlations between the effect series and shifted cause series, thus detecting a lag that represents the time needed for the cause to take shape. The method uses various statistical tests to detect whether adding a cause into a predictive model significantly improves the prediction capabilities of the model. The original paper used linear regression as the testing predictor[55]. Further modifications of the original paper followed and included nonlinearity[162], learning from multiple time series[30], applications on spectral data, i.e., in the frequency domain[77], model-free modifications [41], and nonstationary[143].

Instrumental Variables Instrumental variables can be used to infer causal effects when we cannot control the experimental setting. Suppose that we want to assess the causal effect of the explanatory variable m on the dependent variable j. Normally, we would try to do statistical tests on whether variable j changes when m changes. However, in many applications in medicine, economics, and others, this is not possible, as both m and j can have a common cause and be, therefore, correlated. This introduces bias in many statistical tests. To overcome the issue, we include a third variable, instrument

variable Z, which we can control and which has influence on j only through m. Then, we observe changes of j on Z. When the applied predictor is linear regression, the predictor is a special case of a linear dynamic system[152]. The existence of a hidden state then allows the removal of the correlations stemming from a common, unobserved cause[152].

Instrumental variables are, however, concepts that can be used well beyond linear regression. Non-linear[107] and non-smooth[24] modifications exist. Sometimes, there is a requirement that instrumental variables might have common cofounders. This multilevel modelling is implemented in the instrumental variable toolkit by [79]. Similarly, [49] allows for a latent (hidden) variable.

Tractable Probabilistic Models The tractable probabilistic models (TPMs) are a large group of methods that can be used to model probabilistic distributions compactly in the spirit of neural networks approximating functions. A prime example of TPMs is sum-product networks (SPNs)[120], which represent the probability distribution as a DAG, where "input" random variables are assigned to leaves. Each non-leaf node corresponds to one of two operations, either sum or product. The weights of the edges are then used to learn the probability distribution. The original paper also proposed an algorithm to learn the structure using backpropagation and expectation maximization[120]. The SPNs are only a subgroup in the broad class of probabilistic circuits[33]. The unified formalism allows using different types of nodes besides the sum and product nodes. Dynamic versions[[98], e.g.] are able to work with temporal data.

See also Table 1.1 in the next section for an overview.

1.3 The Challenge

As suggested in the Introduction, we would like to learn causal models that are more expressive than many traditional models. In our view, the expressivity of the causal model entails:

- **Non-linear** aspects of causality.
- **Hidden states** (latent variables) of an *a priori* unknown dimension.
- At the same time, one would like to preserve as much **explainability** as possible, perhaps through targeted reduction[78].
- **Cycles** in causal relationships.
- **Time-series** aspects, such as nonanticipativity and delays: clearly, causal relationships should be established between the cause in the past and the effect in the future, with some delay between the two.
- **Mixture-model** aspects: clearly, there are variations between the metabolism in various individuals, perhaps due to genomic differences. One should explore joint problems[109], where multiple causal models are learned without the assignment of individuals to subgroups represented by the causal models given *a priori*.

The ability to simulate from the model entails:

- **Quantitative** aspects of causality, in order to simulate from the causal model.

Tool	Quantitative	Non-linear	Hidden st.	Cycles	Temporal	Mixture-models	Multiple trajectories	Structure learning in $\mathcal{P}$	Likelihood calculation in $\mathcal{P}$	Marginalization in $\mathcal{P}$	Simulation from the model
Causal Bayesian networks	✓	✓	✓	✗[1]	✓	✓	✓	✗	✓	✗	✓
Structural Equation Modeling	✓	✓	✓	✓	✓	✓	✓	✗	-	-	-
Counterfactual Framework	✓	✓	✓	✓[2]	✓	✓	✓	✗[3]	-	-	-
Granger Causality	✓	✓	✗	✗	✓	✗	✓	✓[4]	-[5]	-	-
Bayesian Structural Time Series Models	✓	✓	✓	✓	✓	✓	✓	✗[6]	-	-	-
Instrumental Variables	✓	✓	✓	✗	✓	✓	✓	✗	-	-	-
Tractable Probabilistic Models	✓	✗	✗	✗	✓[7]	✓	✓	✗	(✓)	(✓)	(✓)

Table 1.1: Summary of features of selected methods and frameworks.

- **Time** required to simulate from the model scaling modestly (with the number of random variables and numbers of samples).

The ability to learn the model entails:

- **Sample complexity**: number of samples required to build the model. Even simple models such as HMM comprise learning Gaussian mixture models, which are known to have high sample complexity.
- **Time complexity**: time required to learn the model. Again, even HMM are cryptographically hard to learn in the setting where one has access to i.i.d. samples of observation sequences[14, 93].

Let us discuss some of these in more detail.

Cycles Standard Bayesian networks do not normally support cycles between the variables. The causal relationships need to form a directed acyclic graph (DAG). As a result, we are detecting some time lag, that the second variable correlated with the first variable shifted to the future. To obtain a cyclic relationship, we would need a sequence of positive time lags that sum together to zero, which is not possible. Under some circumstances, we can model cyclical relationships with Dynamic Bayesian networks (DBNs). For each variable, we have its realisations for time $t = 1, 2, \ldots, T$. As a result, DBNs can then be used to model situations, such as the one when x_t causes y_{t+1}, y_{t+1} causes z_{t+2}, which in turn causes x_{t+3}. The overall graph is still a DAG, as there cannot be a cycle within a one-time slice, and neither can a variable have an effect on the past.

Hidden state and Mixture-models modelling a hidden state in the model and sampling from the mixture of models are tightly connected, as the second can be reduced to the first. Suppose that we want to model a mixture of two distributions. We can build

two separate models for each of the distributions. Then, we introduce a hidden state that models a binary decision, whether we sample from the first or the second distribution.

Model learning When we are interested in the time complexity of model learning, the time requirements differ based on the techniques used. The Bayesian networks do not generally have exact polynomial-time learning. In Granger causality, the complexity of mining causal relationships depends on the algorithms and methods used. In the simplest scenarios, we can base the causal relationships on the F-test, which can be calculated in linear time, assuming that the cumulative distribution function of the FisherSnedecor distribution (F-distribution) is precomputed.

Non-linear dependencies In many cases, the possibility of having non-linear models is part of extensions of the original methods. A prominent example of such a method is Granger's causality. The original method was developed with linear dependencies between the features. But further extensions were developed to include nonlinearities, for example, [162]. In Bayesian networks, the original version considered only propositional variables[112], but subsequent versions [67, e.g.] considered also continuous variables and non-linear dependencies.

1.4 The Benchmark

In this paper, we present a simulated dataset based on the Krebs cycle. The Krebs cycle, also known as the citric acid cycle, is one of the fundamental pathways of biochemistry. The cycle, as illustrated in Figure 1.1, presents a natural example of time series that can be used to infer causal relationships between concentrations of the reactants.

1.4.1 The Interventions

Causal learning requires interventions. Here, we increase the concentration of one reactant and study how its propagations work through the reaction network. In the Krebs cycle, such intervention can be modeled by increasing the concentration of one of the reactants and studying how it propagates through the reaction network. In the natural setting, it is hard to distinguish correlations from causal effects; however, with one reactant artificially increased, an increase in the second reactant means that it is an effect of the first reactant. As this information propagates further, the information about the intervention slowly vanishes, allowing the system to stabilise. In the presented benchmark, this scenario is targeted at in half of the datasets. In krebs1R, only the concentration of one of the reactants is increased. krebs3R presents a more challenging scenario, with three reactants with increased concentrations. Lastly, krebs3 includes normalisation, which is challenging for some of the linear models, but matches situations where we can measure only relative concentrations.

1.4.2 The Data

Depending on the modelling of the time series, each of the reactions can be represented by one or more causal relationships. Our benchmark is based on a simulator in the GitHub repository at [104]. The simulator creates a virtual box with Krebs cycle particles.

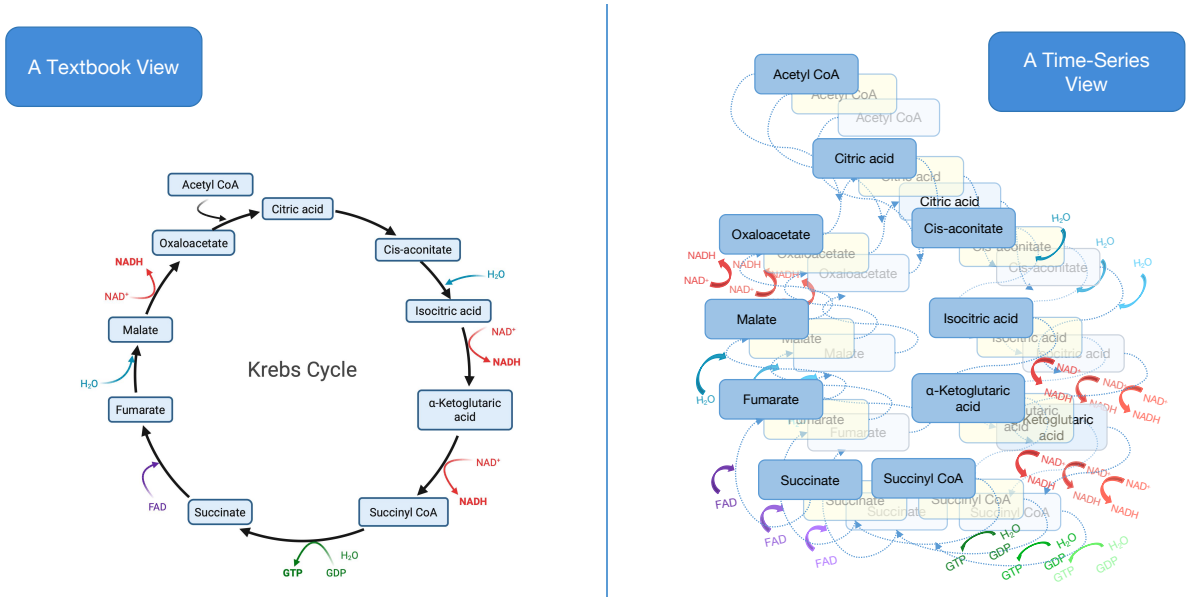


Figure 1.1: **Two views of Krebs cycle.** Left: The textbook view, where the nodes representing reactants form a cycle. Right: A time-expanded graph, where nodes represent concentrations of a reactant at one point in time. Nodes corresponding to one reactant could be seen as a time series, but the graph is acyclic in the time-series view.

The particles move inside the box, following the Boltzmann distribution. Once particles get close to each other, a pre-defined list of reactions is scanned to determine whether a reaction occurs, and if so, reactants are replaced with a product. The simulation continues, and concentrations of the particles are noted as time series. As a result, the time series contains noise (caused by the random location of particles), which is added to the locally linear behaviour of the system.

In this way, we have generated four datasets, consisting of a time series with 5 to 5000 time steps and 16 features for the reactants, including 10 in the main cycle and 6 additional ones (incl. GTP, H2O, FAD, NAD, GDP, NADH). Each of the following datasets is based on simulating approximately 2500 molecules in the bounding box:

KrebsN contains 100 series with normally distributed prior distributions and absolute concentrations.

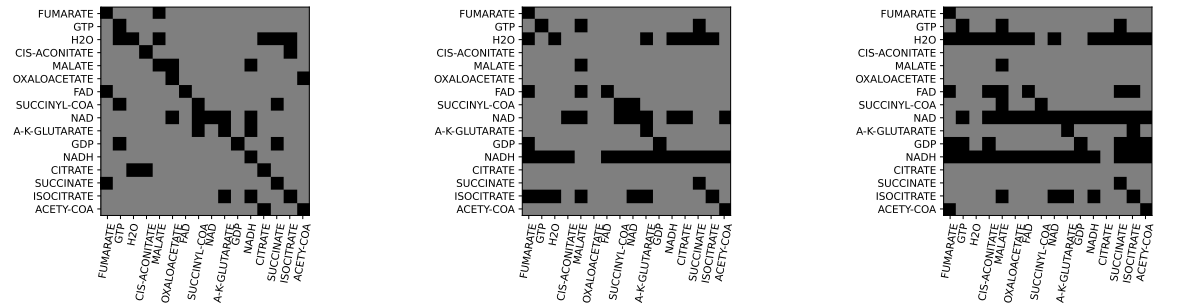
Krebs3 contains 120 series with relative concentrations, where for each triplet of the 10 main cycle reactants, we used uniform priors, and the remaining 7 particles were set to zero. Such a distribution is motivated by allowing the tested approaches to trace how the higher concentration of the three selected compounds move forward in the cycle.

KrebsL focuses on learning from a few long time series. In this case, we have 10 series with 5000 time steps. We use

KrebsS considers 10000 time series with only 5 time steps each, a complementary scenario to *KrebsL*.

Dataset	M.features	Lenght	M.series	Initialisation	Concentrations
KrebsN	16	500	100	Normal distribution	Absolute
Krebs3	16	500	120	Excitation of three	Relative
KrebsL	16	5000	10	Normal distribution	Absolute
KrebsS	16	5	10000	Normal distribution	Absolute

Table 1.2: **Summary of the datasets in the Krebs cycle.**



(a) Ground-truth (b) DyNoTears on *KrebsN* (c) DyNoTears on *Krebs3*

Figure 1.2: **Heatmaps for DyNoTears comparison under different graph settings.** The ground truth matrix representing the set of reactions. Black squares represent 1 an edge in the adjacency matrix, grey 0.

The datasets are summarized in Table 1.2, showing the dimensions of the time series, the number of molecules used in the simulation, as well as other important features of the data.

1.4.3 Evaluation Criteria

For comparison, the dataset includes the ground-truth causal matrix as defined by the equations. The diagonal in Fig. 1.2a indicates that the presence of a substance at time t implies the presence of the same substance at time $t+1$. A single run of an algorithm produces a causal matrix that can be compared to the ground truth one.

We propose that the main measure of the quality of the causal matrix be the Structural Hamming Distance(SHD) and Structural Intervention Distance(SID). SHD measures the number of edges that need to be added and the number of edges that need to be removed to convert the predicted causal graph into the ground-truth causal graph. While the addition or removal of an edge is penalised by 1, change of the orientation is penalised only by 0.5. The Structural Intervention Distance (SID) [116] is a metric used to quantify the discrepancy between two causal graphs $\mathcal{G}_{Groundtruth}$ and $\hat{\mathcal{G}}_{Estimated}$,in terms of their implied interventional distributions. Formally, SID counts the number of pairs of variables (i, j) for which the interventional distributions $\mathbb{P}(X_j \mid do(X_i = x_i))$ differ between $\mathcal{G}$ and $\hat{\mathcal{G}}$.

$$\mathrm{SID}(\mathcal{G}, \hat{\mathcal{G}}) = \#\left\{(i,j) \in \{1,\ldots,d\}^2 \mid \mathbb{P}_{\mathcal{G}}(X_j \mid do(X_i)) \neq \mathbb{P}_{\hat{\mathcal{G}}}(X_j \mid do(X_i))\right\}. \tag{1.4}$$

SID is zero if and only if all pairwise interventional distributions implied by $\hat{\mathcal{G}}$ match those of the true graph $\mathcal{G}$, regardless of the parameterization. Compared to structural

Hamming distance, SID focuses on the correctness of interventional implications, making it a more semantically meaningful metric for evaluating causal discovery methods.

F_1-score, which is the harmonic mean of the precision and recall measures. Let TPR, TNR, FPR, FNR be the true/false positive/negative measures as in a classification task. Then, the False Discovery Rate (FDR), which quantifies the proportion of false positives among all predicted positive instances and F_1-score is defined

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{1.5}$$

$$\text{Precision} = \frac{TPR}{TPR + FPR}, \tag{1.6}$$

$$\text{Recall} = \frac{TPR}{TPR + FNR}, \tag{1.7}$$

$$\text{FDR} = 1 - \text{Precision}. \tag{1.8}$$

The F_1-score can be easily extended to the case where the predicted causal matrix is stochastic. In that case, for example, an edge predicted with weight 0.3 when there is no ground-truth edge, contributes 0.3 to false-positive and 0.7 to true-negative.

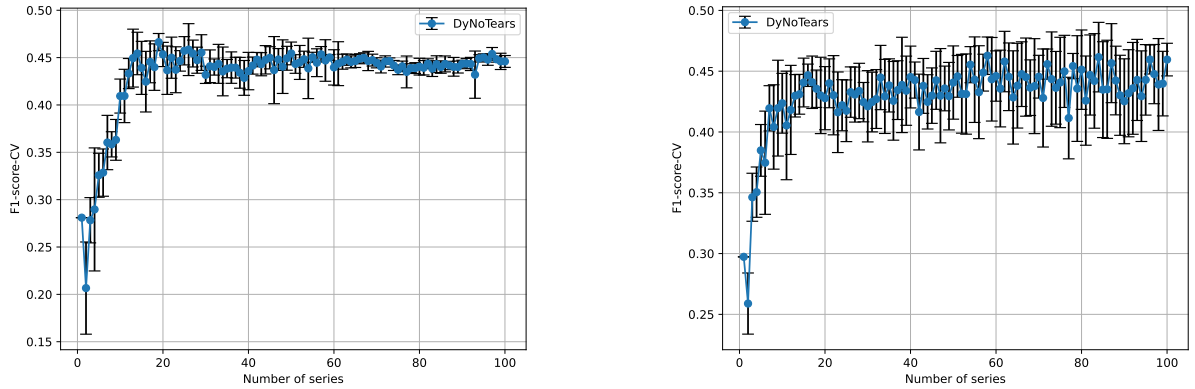
To assess the stability of the method, we recommend to average the results over at least 10 runs of the method, whenever the tested method is randomized. The standard mean should then be calculated. In the case of deterministic methods, the stability of the F_1-score cannot be evaluated by simple repeated evaluations followed by standard deviation calculation. Therefore, we recommend using an approach similar to cross-validation to show the stability of the results. In each evaluation, instead of plain restart, we can keep 10 % of the dataset aside to randomize data instead of the method. As a result, by doing repeated evaluations, it is possible to obtain the results' standard deviations and confidence intervals.

1.4.4 Numerical Comparison

Then, there is a set of two related methods, ExDAG [133], ExDBN [132], and ExMAG [138]. The methods fit a linear structural equation model to the data. The acyclicity constraints are applied in a lazy manner - whenever a cycle is created in the proposed graph, a new constraint is added to the program, until a DAG is found. ExDAG focuses on learning the data with no interslice dependencies, ExDBN extends the model to the Dynamic Bayesian Networks. Those two models were further extended to the ExMAG method [138], where the goal is to learn a Maximally Ancestral Graph.

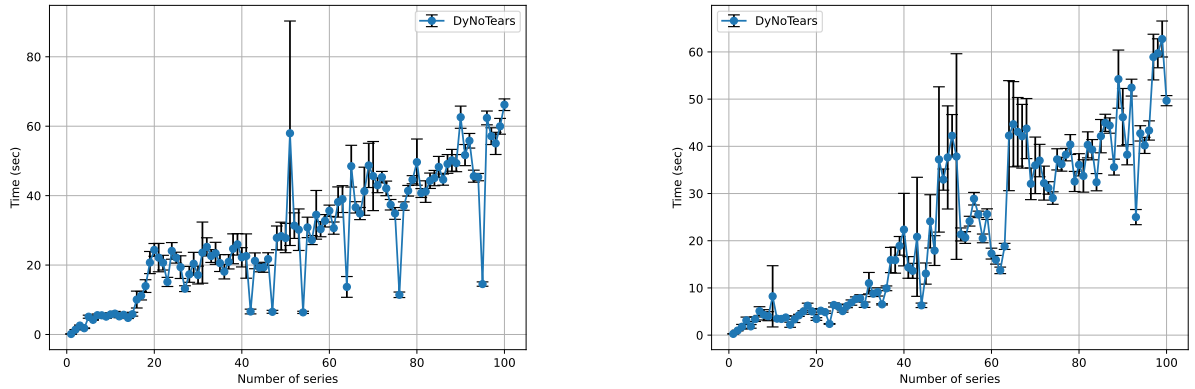
The last method in the comparison is DyNoTears [111], a state-of-the-art method for causal discovery, which was implemented in the CausalNex [7] package. DyNoTears is provided with information that forbids edges within the same time slice, and the regularization parameter λ is selected from the list $10^{-6}, 10^{-5}, \ldots, 10^{6}$, so that the maximum F_1-score is reached. In addition to the F_1 score, we also measured the time needed for structure learning. Figure 1.2 presents the adjacency matrices inferred by DyNoTears, along with the ground truth. We can see that, as the F_1-score is low, both datasets are challenging for causal discovery.

Figure 1.3 illustrates the evolution of the F_1-score as a function of the number of time series used in the evaluation. A consistent improvement in performance is observed with an increasing number of time series, suggesting that more data enhances the reliability of the learned causal structure. Similarly, Figure 1.4 displays the computational time



(a) *KrebsN* (b) *Krebs3*

Figure 1.3: **F_1-score of DyNoTears on the datasets *KrebsN* and *Krebs3*.** Note that the implementation of DyNoTears in CausalNex is deterministic, thus providing the same result each time. To calculate the error bars, randomly selected 10 % of the data were put aside, and then results were averaged over 10 repeats of this procedure.



(a) *KrebsN* (b) *Krebs3*

Figure 1.4: **Comparison of DyNoTears running time across data sizes in the *KrebsN* and *Krebs3* datasets.** The error bars show the standard deviation of the measurements calculated from 10 repetitions.

requirements of the methods across varying dataset sizes. The time complexity increases with the number of time series, but exhibits variability due to the underlying structure and implementation nuances. The error bars indicate standard deviations obtained from 10 repeated runs, providing insights into the stability of each method under varying input conditions. From the results, we can see that the dataset is a major challenge for state-of-the-art identification methods, considering their F_1-score is close to 0.5. Therefore, there is room for methods to improve the results further.

To illustrate the dataset, we additionally include results of several representative causal discovery methods that were implemented in the `GCastle` [170] package. These methods span a diverse set of causal inference paradigms, including:

- **Constraint-based methods**, such as the PC and FCI algorithms [145, 169], which rely on conditional independence testing;
- **Score-based approaches**, including GES and GIES [32, 61], that search for the

best causal graph according to a predefined scoring criterion;

- **Functional causal models**, such as LiNGAM [142], which assume linear non-Gaussian causal mechanisms;
- **Gradient-based and deep learning methods**, including NOTEARS[173], DAG-GNN[168], and GraN-DAG[83], which formulate structure learning as a continuous optimization problem over the space of acyclic graphs.

This variety enables a comprehensive comparison of different causal discovery algorithms across a range of assumptions and data characteristics. To systematically compare the performance of causal discovery algorithms on biochemical pathway data, we evaluated 14 representative methods across four major methodological categories in Figure 1.5. Performance was evaluated on both the original and normalised versions of the *Krebs3* dataset. This comprehensive benchmarking highlights variability in method robustness to data normalisation and facilitates category-level insights into algorithmic behaviour.

For the *Krebs3* dataset, Figure 1.6 illustrates the comparison of different representative causal discovery algorithms using the percentage error across various performance measures.

In order to calculate % error, we calculate % age of absolute difference between the computed value of performance measure and true value to get the percentage error, and is given by the following formula:

$$\text{Percentage Error} = \left| \frac{\text{Computed Value} - \text{True Value}}{\text{True Value}} \right| \times 100. \tag{1.9}$$

For metrics such as Recall, FDR and FPR, the true value is 1. In the case of SHD and SID, the true value is set to 200 and 150, separately. For F_1-score the true values are the respective baseline errors obtained using the ground-truth graph, which are 0.3.

As observed in Figure 1.6, the `PC` and `ExMAG` algorithm consistently outperforms all other methods across all evaluation metrics, achieving the lowest percentage errors. This demonstrates its strong performance in both structural and predictive accuracy on large-scale simulated datasets. In contrast, methods like Notear-Linear and DyNotear yield higher error rates, particularly in SHD and F_1-score, reflecting challenges in accurately learning causal graphs in this biological context. Overall, the results highlight that method performance can vary substantially depending on the evaluation criterion, emphasizing the importance of using a diverse set of metrics when benchmarking causal discovery algorithms.

1.5 Discussion

Once the dataset is presented, we are ready to compare it with other existing possibilities and show how it improves upon the other choices in [111, 56, 26]. We will point out the important advantages that the Krebs dataset has over other datasets.

Does Not Assume Any Ground Truth Structural Model. Instead, our method uses an independent method of simulation from a real-world setting. The dataset is generated by following the chemical reactions in the Krebs cycle. This makes it possible to generate multiple variants (*KrebsN*, *Krebs3*, *KrebsS*, *KrebsS*) consistently. These consist

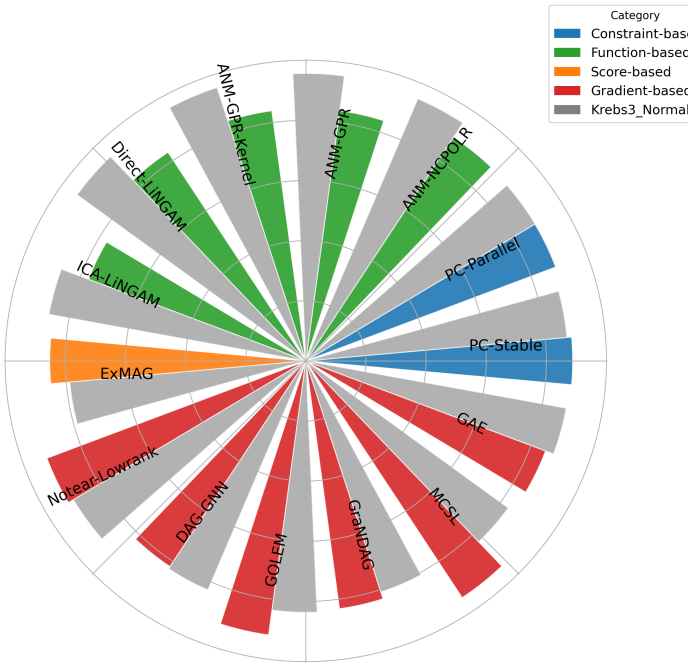


Figure 1.5: **SID scores of various algorithms grouped by category for *Krebs3* and *KrebsN* datasets.** The performance of each method is represented with colour-coded sections in a radial bar plot. The grey overlay indicates the performance on the normalised dataset.

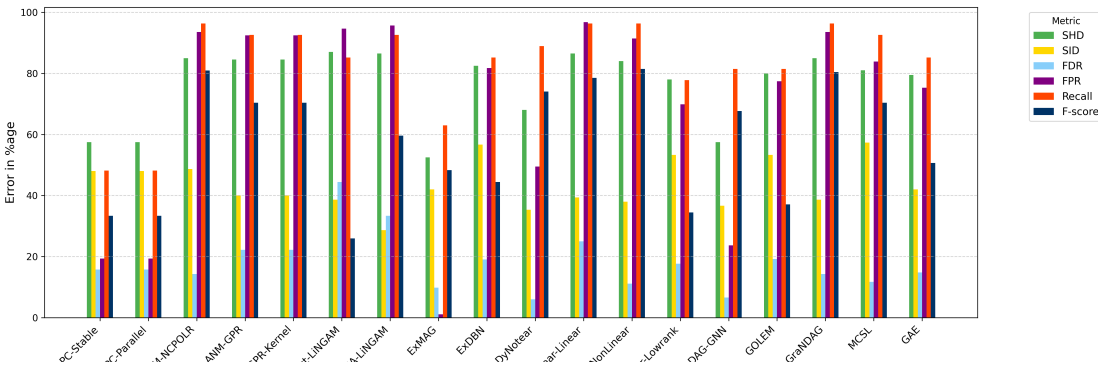


Figure 1.6: **Comparison of varying algorithms by percentage of error in performance metrics for *Krebs3* dataset.**

Dataset	R^2-sortability	Standard variance	Note
krebsN	0.486	0.008	
krebs3	0.501	0.011	
krebsS	0.497	0.035	(please, see caption)
krebsL	0.492	0.005	

Table 1.3: **The evaluation of R^2-sortability for individual time series.** For each of the time series in each dataset, we calculated the R^2-sortability using the `CausalDisco` Python package [123, 124]. To obtain the results, the R^2-sortability values were then averaged over each of the datasets. Please note that for 11 time series in the *krebsS* dataset, the R^2-sortability method did not produce a numeric result. Since this is much less than 1 % of the dataset (and R^2-sortability is bounded by 0 and 1), the average wont be influenced substantially.

of a time series with 5 to 5000 time steps and 16 features for the reactants, including 10 in the main cycle and 6 additional ones (incl. water).

Our Dataset Is Not R^2-Sortable. Our method does not suffer from the R^2-sortability issues other synthetic benchmarks suffer from, as explained by [110] and [124]. Indeed, [110] argue that there are usually patterns left by the simulation from structural models that are easy to exploit. This can be quantified by the R^2-sortability[124]. To illustrate how the Krebs dataset stands compared to the R^2-sortability, we implemented a code evaluating the R^2-sortability for our dataset, the results of which can be seen in Table 1.3. Reference[124] then explains that *0.5 means that ordering the variables by R^2 amounts to a random guess of the causal ordering*, meaning that our dataset is not R^2-sortable. Thus, the fact that we do not assume any underlying framework makes our dataset more universal.

The Ground Truth Causal Relationships Are Known. At the same time, our method comes with widely accepted ground-truth data. The advantage can be seen when compared to datasets such as S&P100 (stock returns for 100 top US companies), used in DyNoTears paper[111]. S&P100 is a real-world dataset that suffers from an unclear ground truth causal matrix. Moreover, the authors had to ensure that the data were stationary, as concept drift is likely to happen in stock trading.

A similar situation is connected with the SACHS dataset[26]. This dataset contains single-cell measurements of levels of 11 proteins in immune cells. With 853 samples, the dataset is of a similar size to ours. However, we cannot be sure what the true causal relationships between the variables representing individual genes are in the case of expression data.

Prospect of Perfect Reconstruction At the same time, our method allows for the prospect of perfect reconstruction. Our dataset is much smaller than another commonly used causality dataset, the DREAM dataset[56]. This is desirable in connection with the fact that most of the problems in causal learning are NP-hard. Because of that, perfect

recovery with many variables is computationally infeasible. *Causal discovery algorithms should be tested on smaller, easy-to-explain datasets first* before proceeding to larger and more complex datasets. The use of larger datasets also brings another reproducibility problem sampling, often done in an ad hoc, paper-specific fashion which is not needed with our data.

1.6 Conclusion

We introduced a synthetic benchmark dataset for causal discovery based on the Krebs cycle. The dataset avoids structural artifacts common in existing benchmarks and includes known ground-truth graphs for evaluation. It supports varied scenarios, including interventions and different time series lengths. Our results show that the dataset poses a meaningful challenge to existing methods and provides a reliable basis for comparing causal inference algorithms.

We publish all source files used to generate the data and the figures in this paper in the GitHub repositories [62, 135]. The repositories also contain numeric results that were generated as input to the plots. The generator of the data can be found in another GitHub repository [136], including a description of how to generate the benchmarking data. The generator is based on a simulator at Nagro [104].

Author Contributions

- Xiaoyu He: Software, Formal Analysis, Validation, Visualisation, Writing – Review & Editing.
- Petr Ryšavý: Methodology, Data Curation, Software, Writing – Original Draft.
- Jakub Mareček: Methodology, Supervision.

All authors meet the authorship criteria as defined by Taylor & Francis and have approved the final manuscript.

Data Availability Statement

The dataset, including values required to replicate all figures and findings in the article, is hosted on [**Hugging Face**]. [*Krebs Benchmark Dataset*][134]. This project contains the following underlying data:

- `KrebsN` [100 series × 500 steps]. (Standard benchmark with normal initialization and absolute concentrations).
- `Krebs3` [120 series × 500 steps]. (Features excitation of three components with relative concentrations).
- `KrebsL` [10 series × 5000 steps]. (Long-series data for analyzing extended dynamics).
- `KrebsS` [10,000 series × 5 steps]. (High-volume short-series data for statistical/ML applications).

Competing Interests

The authors declare no competing interests.

Grant statement

This work was supported by the CoDiet project, which is co-funded by the European Union under Horizon Europe [grant number 101084642]. It was also supported by UK Research and Innovation (UKRI) under the UK governments Horizon Europe funding guarantee [grant number 101084642].

Acknowledgements

The authors would like to thank the CoDiet project consortium for their support and collaboration during the research process. A preprint version of this manuscript is posted on arXiv as [137].

Chapter 2

ExMAG: Learning of Maximally Ancestral Graphs

Building on dynamical system learning and the Krebs cycle benchmark, we extend causal discovery to complex scenarios involving hidden confounders. Maximally ancestral graphs (MAGs) generalize directed acyclic graphs (DAGs) by explicitly representing both causal effects and latent confounding. To address this setting, we developed a score-based branch-and-cut algorithm for learning MAGs, which achieves higher accuracy and improved computational efficiency compared to approaches that do not account for hidden variables. This line of work lays the foundation for tackling increasingly complex problems in causal modelling and for integrating dynamical system learning with causal inference in future research.

2.1 Introduction

As one transitions from statistical to causal learning [141], one is seeking the most appropriate causal model. Dynamic Bayesian networks (DBN) [44, 102] are a popular model, where a weighted directed acyclic graph represents the causal relationships. Stochastic processes are represented by their vertices, and weighted, oriented edges suggest the strength of the causal relationships. The key challenge in learning DBNs is confounding.

To illustrate the challenge of confounding, let us consider Simpson's paradox. Simpson's paradox shows that without considering confounding factors in statistical analysis [96], the direction of causality can be mis-estimated completely. An textbook example [96] comes from the Berkeley graduate admissions [13]. The data show that women find it harder to get admitted to Berkeley graduate schools. Nevertheless, this is because women tend to apply to departments that have lower admission rates. In this example, the choice of the graduate school is the confounder, impacting the probability of admission. Confounding is prevalent throughout high-dimensional statistics [88, 53], such as in biomedical sciences.

Specifically, in biomedical sciences, confounders such as socio-economic status, age, or lifestyle factors can distort the true causal relationship between treatments and outcomes [175]. Techniques such as instrumental variables [122, 74], propensity score matching [129], and double machine learning [31] have been widely used to mitigate the effects of confounding in clinical trials and observational studies. In bioinformatics, particularly in genome-wide association studies (GWAS), confounders, including population stratification and environmental exposures, must be controlled to avoid biased estimates in genetic

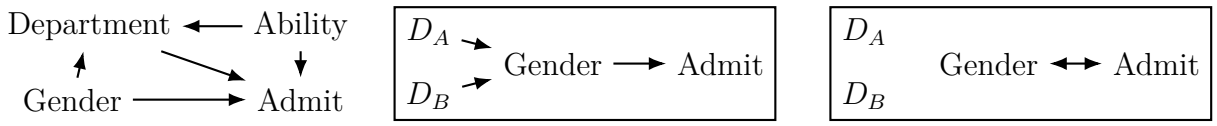
association studies [25, 58]. To mitigate such biases, statistical models that explicitly account for hidden confounders, such as spectral methods and latent variable models, are often employed [58]. Furthermore, meta-analysis and sensitivity analysis are often employed to evaluate the robustness of findings in the presence of potential confounders, especially when combining results from multiple studies [18, 94]. These methodologies ensure that the conclusions drawn are reliable and actionable, improving the credibility of statistical models across disciplines.

In statistical theory, work [20] studies confounding in detail, and many subsequent works develop further methods. [84] shows that leveraging the dominant eigenstructure of time series may improve performance of estimation. Anchor regression, for instance, bridges the gap between causality and robustness by addressing heterogeneity in data [130]. Other significant contributions include spectral deconfounding, a technique designed to mitigate the effects of hidden confounders in high-dimensional settings [19].This approach provides a framework for robust predictions in the presence of shifts in data distributions. Similarly, the invariance principle has emerged as a cornerstone of causal inference, linking causal structure to robust statistical models [18]. Furthermore, the concept of doubly robust inference offers an alternative framework for addressing hidden confounding factors, combining model robustness with efficiency in high-dimensional scenarios [58]. Together, these developments represent a significant step forward in understanding and addressing the challenges posed by complex causal systems with missing or latent variables [156, 125]. While these methods have shown promise, they often rely on simplifying assumptions, such as stationarity or full observability [12].

While there is a long history of the study of confounding, as suggested above, the extensions of DBNs to allow for confounding are rather more recent. Instead of estimating a Directed Acyclic Graph (DAG), one could estimate a Maximal Ancestral Graph (MAG), cf. [127]. MAGs allow for both direct and indirect relationships among variables modelled as directed and undirected edges, even in the presence of confounding factors. In particular, MAGs can represent feedback loops and bidirectional relationships that DAG-based models, such as DBNs, cannot. This makes MAGs a more powerful tool for capturing the complex dynamics of real-world causal systems.

There are only a few studies of MAG estimation [28, 121, 34, 69, 72, 71, 38]. [126, 151] are applicable to both discrete and nonparametric cases, which extend DAG to MAG or ADMG diagrams, see more definitions in Sect. 2.2. Factorization in MAG is not directly decomposable into individual variables and their parent sets, as in DAGs, but must instead consider components connected by bidirected paths (termed *districts* or *c-components*), cf. [126], although [34] proposed to use Markov equivalence classes (MEC) instead. In 2021, [28] introduced a first mixed-integer programming (MIP) formulation, but the number of variables scales with the number of c-components, i. e., exponentially with the number of vertices in the worst case. Such formulations are also known as extended formulations [35]. The same year, [121] explored a constraint-based approach for MAG discovery, leveraging conditional independence testing. Additionally, [174] addresses exogenous covariates in causal formulation that helps explain the heterogeneity in both sampling and causal mechanisms. Dissertation [69] presented an extension of the imsets of [146] from directed acyclic graphs (DAGs) to towards MAGs [72], which allows for the use of the methods of Studený, and a score-based heuristic [71]. More recently, paper [38] enhanced the scalability of methods of [28] by utilising linear programming (LP) relaxations instead of solving the MIP.

Our approach proposes a formulation of MAG estimation within Mixed-Integer Non-



(a) Cofounding graph (b) Result of ExDBN [132] (c) Result of ExMAG

Figure 2.1: **Ground truth with the confounder of Department on the Berkeley graduate admission example.** (left, 2.1a), a dynamic Bayesian network trained on the data (center, 2.1b), and ExMAG output (right, 2.1c). While the dynamic Bayesian network suggests a causal relationship between gender and admission, ExMAG correctly identifies the confounding. See the supplementary material for details.

linear Programming (MINLP) in a dimension polynomial in the number of vertices, in contrast with the so-called extended formulations of [28, 38], where the dimension is exponential in the number of vertices. While both the extended formulation of [28] and ours ensure that confounding factors are properly accounted for and the true underlying data-generating process is better represented by the model, our implementation scales further, from 4-5 stochastic processes in the extended formulation of [28] to 25 or more stochastic processes with the proposed compact formulation.

2.1.1 Motivating Example

Let us revisit the Berkeley graduate admission paradox example from the first page. As in most paradoxes, there is no violation of logic in Simpson's paradox, just a violation of of intuition. The poor intuition being violated in this case is that a positive association in the entire population should also hold within each department. Overall, females in these data did have a harder time getting admitted to graduate school. But that arose, because female applicants chose the departments that were the most difficult to gain admission to for anyone, male or female. In this example, gender influences the choice of department, and the department influences the chance of admission. Controlling for department reveals a more plausible direct causal influence of gender, as illustrated in the DAG in Fig. 2.1a. Our method, ExMAG, is able to reveal the confounders in this Berkeley graduate admission example, as illustrated in the notebook in supplementary materials and Figure 2.1.

2.2 Graphs and Properties

A DAG is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ such that there are no directed cycles. That is, there is no sequence of distinct vertices $v_1, v_2, \ldots, v_M \in \mathcal{V}$ such that $(v_t i, v_{ti+1}) \in \mathcal{E}$ for all $1 \leq m \leq M-1$ and $(v_M, v_1) \in \mathcal{E}$. Maximal ancestral graphs (MAGs), first introduced by [127], provide a framework for modelling distributions through conditional independence (CI) relations. Compared with directed acyclic graphs (DAGs), MAGs allow for latent confounders, accommodating data that arise from distributions with more complex independence structures and revealing hidden states in the graphs. While DAGs allow for the efficient computation of maximum likelihood estimates (MLEs) and scoring (e.g., via BIC), these properties are challenging to extend to MAG due to their structural and computational complexity [70].

ADMG Mixed graphs feature two types of edges: directed ($\to$) and bidirected ($\leftrightarrow$). Mixed graph $\mathcal{G}$ thus consists of a vertex set $\mathcal{V}$, a set of directed edges $\mathcal{E}$ and undirected edges $\mathcal{U}$, where $\mathcal{E}$ are (ordered) pairs of distinct vertices, while $\mathcal{U}$ are (unordered) 2-element subsets of vertices. For a directed edge in $\mathcal{E}$ connecting two vertices($v, w \in \mathcal{V}$), we say these two vertices are the *endpoints* of the edge and the two vertices are *adjacent* (if there is no edge between v and w, they are *non-adjacent*). For a vertex $v \in \mathcal{V}$, we define the *parents*, *spouses*, *ancestors*, and *district* of v, respectively as:

$$\mathrm{pa}_{\mathcal{G}}(v) = \{w : w \to v \text{ in } \mathcal{G}\}, \qquad \mathrm{an}_{\mathcal{G}}(v) = \{w : w \to \cdots \to v \text{ in } \mathcal{G} \text{ or } w = v\},$$
$$\mathrm{sp}_{\mathcal{G}}(v) = \{w : w \leftrightarrow v \text{ in } \mathcal{G}\}, \qquad \mathrm{dis}_{\mathcal{G}}(v) = \{w : w \leftrightarrow \cdots \leftrightarrow v \text{ in } \mathcal{G} \text{ or } w = v\}.$$

Given a directed mixed graph $\mathcal{G}$, the *districts* define a set of equivalence classes of nodes in $\mathcal{G}$. The district for node v is defined as the connected component of v in the subgraph of $\mathcal{G}$ induced by all bidirected edges. As in a DAG, a mixed graph $\mathcal{G}$ is acyclic if it contains no directed cycles in $\mathcal{E}$, i.e., an acyclic directed mixed graph (ADMG) [69].

A directed mixed graph $\mathcal{G}$ is called an ancestral ADMG if the following condition holds for all pairs of nodes v and w in $\mathcal{G}$:

$$\text{If } v \neq w \text{ and } v \in \mathrm{an}_{\mathcal{G}}(w) \cup \mathrm{sp}_{\mathcal{G}}(w), \text{ then } w \notin \mathrm{an}_{\mathcal{G}}(v),$$

which is written as, $\mathcal{G}$ is an ancestral ADMG if it contains no directed cycles ($v \to u \to \ldots \to w \to v$) or almost directed cycles [28, 69]. In an ADMG, an almost directed cycle is of the form $v \to u \to \ldots \to w \leftrightarrow v$; in other words, $\{v, w\}$ is a bidirected edge, and $v \in \mathrm{an}_{\mathcal{G}}(w)$ [28].

Inducing Paths An inducing path from variable X to variable Y in a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a path $P = (X = v_0, v_1, \ldots, v_M = Y)$ such that for all intermediate nodes v_i (where $1 \leq i \leq M-1$), $v_i \in \mathcal{Z}$, where $\mathcal{Z}$ is the conditioning set. If the path is blocked by conditioning on $\mathcal{Z}$, then it is an inducing path.

A node v on a non-overlapping path is called a collider if contains a non-overlapping subpath (w, v, u) with two arrowheads into v. In mathematical form, a collider is represented as

$$\mathrm{collider}_{\mathcal{G}}(v) = \left\{ v : \exists w, u : \; w \to v \leftarrow u \vee w \leftrightarrow v \leftarrow u \vee w \to v \leftrightarrow u \vee w \leftrightarrow v \leftrightarrow u \right\}.$$

m-separation Graphs encode conditional independence via separation criteria. For acyclic directed mixed graphs (ADMGs), *m-separation* generalises d-separation to handle bidirected edges. A path between nodes u and v is *m-connecting* given a conditioning set $\mathcal{Z} \subseteq \mathcal{V}$ if: (i) u and v are the endpoints; (ii) all non-colliders are not in $\mathcal{Z}$; and (iii) all colliders are in $\mathrm{an}_G(\mathcal{Z})$. Nodes u and v are *m-separated* given $\mathcal{Z}$ if no such path exists.

Maximal Ancestral Graph An ADMG $\mathcal{G}$ is called a maximal ancestral graph (MAG) if:

(i) For every pair of nonadjacent vertices u and v, there exists some set $\mathcal{Z}$ such that u, v are *m-separated* given $\mathcal{Z}$ in $\mathcal{G}$ (*Maximality*);

(ii) For every $v \in \mathcal{V}$, $\mathrm{sib}_{\mathcal{G}}(v) \cap \mathrm{anc}_{\mathcal{G}}(v) = \emptyset$ (*Ancestrality*).

where $\mathrm{sib}_{\mathcal{G}}(v) = \{u \in \mathcal{V} \mid \exists w \in \mathcal{V} : w \to v,\ w \to u,\ u \not\to v,\ v \not\to u\}$. We refer to [69] for multiple examples.

2.3 Formulation of the Mixed Integer Quadratic Program

Recent works on high-dimensional confounding or deconfounding clarify the connections between distributional robustness, replicability, and causal inference [130, 58]. Distributional robustness differs significantly from traditional robust statistical methods [73, 60], which typically handle outliers in the training data, while our work focuses on evaluating the existence of a confounding factor.

In this section, we inherit from distributional robustness and present the formulation of the Mixed Integer Quadratic Program (MIQP) to infer the causal structure. Since we cannot observe all relevant variables, we must deal with the situation of hidden confounding. The problem is formalised in the following form corresponding to a structural equation model (SEM) [15, 114]:

$$Y \leftarrow X\hat{W} + g(H, A) + \epsilon_Y,$$

where:

- ϵ_Y is the noise term, independent of all variables that appear "upstream" from Y.
- A is an exogenous variable, though not considered in the following sections.
- H is the unobserved confounding variable vector.

If non-zero components of the vector $\hat{W}$ are correlated with certain components of X, these components are defined as causal X-variables for Y. This means:

$$w_{m,j} \neq 0 \iff X\text{'s } j\text{-th component is a causal variable.} \tag{2.1}$$

For the scenario that there are no exogenous variable perturbations, we describe this with an additive noise model with hidden states as follows:

$$X \leftarrow H\gamma + \epsilon_X, \tag{2.2}$$

$$Y \leftarrow X\hat{W} + H\theta + \epsilon_Y, \tag{2.3}$$

where ϵ_X, ϵ_Y and H are mutually independent. X is the observed covariate vector. H is the unobserved confounding variable vector. Our goal is to infer the confounding-free regression parameter $\hat{W}$ and stabilise the prediction of the relationship between Y and X.

2.3.1 Connecting to Causality

The causal parameter $\hat{W}$ in the Equation 2.3 can be seen as minimizing the worst-case risk:

$$\hat{W} = \arg\min_{w} \max_{P \in \mathcal{P}} \mathbb{E}_P \left[(Y - XW)^2 \right],$$

where $\mathcal{P}$ is a class of distributions containing perturbations of the original distribution, including confoundings. This modelling highlights the inherent connection between causality and distributional robustness [42, 115, 128].

In this perspective, we present the formulation of the Mixed Integer Quadratic Program (MIQP) used to infer the causal structure, with a new binary matrix $B = [b_{j,m}] \in$

$\{0,1\}^{M\times M}$ introduced to account for relationships explained by confounding factors, alongside the weight matrix $W \in \mathbb{R}^{M\times M}$ representing the model weights and binary adjacency matrix E adopted from the ExDAG model [133]. Whenever an entry in $W_{j,m}$ is non-zero, the respective value in either $E_{j,m}$ (directed edge) or $B_{j,m}$ is nonzero (bidirected edge). At the same time, we extend the existing formulation by introducing an additional binary input parameter $F_{j,m}$ for each pair of variables (j,m), where $j \neq m$, indicating that there is no direct causal relationship between variables j and m, but j and m might have a common cofactor. This follows from the meaning of the edges in a MAG $\rightarrow$ edge implies a direct causal relationship but does not rule out a possible latent confounding, $\leftrightarrow$ means no direct causal relationship.

The *Directed edge matrix* E is

$$e_{j,m} = \begin{cases} 1, & \text{if } j \rightarrow m, \\ 0, & \text{otherwise,} \end{cases} \tag{2.4}$$

and the *Bidirected Edge Matrix* B is represented as

$$b_{j,m} = \begin{cases} 1, & \text{if } j \leftrightarrow m, \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

The new *Input Matrix* F, is by the definition,

$$f_{j,m} = \begin{cases} 1, & \text{if } j \not\rightarrow m \quad \text{and} \quad m \not\rightarrow j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

where matrix F is by definition symmetric. Based on these assumptions, one can substitute (2.2) into (2.3), and the work of ExDAG [133] is extended as explained in the next section.

2.3.2 MIQP Formulation

The cost function for the Mixed Integer Quadratic Program(MIQP) of ExMAG is the following L_q norm:

$$L_q = \sum_{i=1}^{p} \sum_{m=1}^{M} \left| Y_{i,m} - \sum_{j=0; j\neq m}^{M} Y_{i,j} w_{j,m} \right|^q + \lambda \sum_{j,m=0, j\neq m}^{M} (e_{j,m} + b_{j,m}), \tag{2.7}$$

where:

- $Y_{i,m}$ represents the value of the m-th variable for the i-th data point;
- $w_{j,m}$ represents the weight of the edge from variable m to variable j;
- $e_{j,m}$ is the binary decision variable indicating the presence of a directed edge from j to m;
- $b_{j,m}$ is the binary decision variable indicating a bidirected edge between j and m;

- $\lambda \in \mathbb{R}^+$ is a regularization parameter controlling the model fit and the edge penalty trade-off.
- The exponent $q \in \mathbb{N}$ can take values $q = 1$ or $q = 2$.

Optimization criterion in (2.7) implies that the dependencies between the variables are linear. The first part of the criterion encodes for the actual cost as an error of the prediction, the second part encodes for regularization, penalising more edges with a larger λ.

As in the ExDAG [133] model, the weights are bounded by introducing a large constant δ. The bounding avoids bilinear terms in the cost function in (2.7) and takes the following form:

$$-\delta \cdot (E + B) \leq W \leq \delta \cdot (E + B) \qquad \text{(Weight Constraint)}$$
$$E + B \leq \mathbf{1} \qquad \text{(Edge Constraint)}$$

The Edge Constraint means that there cannot be a directed as well as a bidirected edge between the same two vertices. Additionally, we enforce that the bidirected matrix is symmetric by (2.8). If $f_{j,m} = 1$, then we know there is no direct causal relationship between j and m, and therefore, $e_{j,m} = 0$. This is formally enforced by (2.9) Inversely, $f_{j,m} = 0$ implies a directed edge rather than a bidirected edge between j and m in (2.10).

$$B = B^T, \tag{2.8}$$
$$F + E \leq \mathbf{1}, \tag{2.9}$$
$$B \leq F. \tag{2.10}$$

Lastly, we must enforce conditions for directed or almost directed cycles and inducing paths. Those conditions are enforced lazily using a separation routine explained later. Directed cycles are enforced in a way adopted from [133]. Therefore, they are left out of this paper. An almost directed cycle formed by edges in set $\mathcal{E}'$ and a bidirected edge (u, v) is forbidden by the constraint

$$b_{u,v} + \sum_{(j,m)\in\mathcal{E}'} e_{j,m} \leq |\mathcal{E}'|. \qquad \text{(Acyclic Constraint)}$$

Similarly, if there is an inducing path formed by path β that contains bidirected edges, and set $\mathcal{E}'$ contains all directed edges that participate in the ancestor relationship (including multiple paths) between the inner points of the path and the terminals of β, this inducing path is forbidden by

$$\sum_{(j,m)\in\beta} b_{j,m} + \sum_{(j,m)\in\mathcal{E}'} e_{j,m} \leq |\mathcal{E}'| + |\beta| - 1. \qquad \text{(Inducing-Paths Constraint)}$$

Note that the second condition does not necessarily eliminate the inducing path, as the optimizer might forbid one of the edges in $\mathcal{E}'$ without influencing the ancestor relationship. This results in path β being found in the next iteration, with a smaller set of directed edges, and the process is repeated.

By enforcing these constraints, we ensure that the MIQP correctly models the causal relationships between the variables while respecting the independence structure defined by $f_{j,m}$ and the potential confounding relationships captured by $b_{j,m}$.

2.4 Separation Routine for the Maximal Ancestral Graphs

The main contribution of this section is the separation routine that identifies whenever a graph is an instance of a maximal ancestral graph. To do so, we need to identify directed cycles, almost directed cycles, and inducing paths. The presence of directed cycles can be detected in $\mathcal{O}(M^2)$ using depth-first-search (DFS); such an approach can be found in [133]. For both inducing paths and almost directed cycles, we will use the distance matrix D constructed on the graph of directed edges E. This distance matrix can be obtained, for example, using the Floyd-Warshall algorithm [52].

Having the distance matrix, to check for almost directed cycles, we can iterate over all bidirected edges and test whether the distance between the endpoints using E is finite, i.e., we have a directed path connected by a bidirected edge. See Algorithm 1 for details.

Algorithm 1 Function that identifies almost directed cycles.

Input: directed edges E, bidirected edges B

```
function ALMOST-DIRECTED-CYCLES(E, B)
    D ← DISTANCE-MATRIX(E)
    for all (j, m) ∈ {1, 2, ..., M} × {1, 2, ..., M} do
        if j ≠ m & b_{j,m} == 1 & D_{j,m} < ∞ then
            E' = TRACE-DISTANCE-MATRIX(D,E,j,m)   ▷ Finds all edges on any j to m path, see Supl.
            Found cycle formed by edges E' and j ↔ m
        end if
    end for
end function
```

In the case of inducing paths, we use a DFS starting from each vertex. Once started from vertex m, the DFS routine checks for all possible inducing paths that terminate in m. For efficiency, a set of all possible endpoints of the path is held. Once this set is empty, the DFS search is terminated, and no further exploration is performed. The set is updated using the distance matrix calculated on the directed edges. If we consider a vertex v, m must be either its ancestor (meaning that possible endpoints for v remain unchanged) or the second inducing path endpoint is among the points that are reachable from v (meaning that the possible endpoints for v are replaced with their intersection with the set of all points reachable from v). See Algorithm 2 for details.

Once having the bidirected edges in the inducing paths and almost directed cycles, we need to trace back the Floyd-Warshall distance matrix to find all directed edges that form the cycle or the ancestor relationship. This is done using calls to the function TRACE-DISTANCE-MATRIX, which can be found in the Supplementary materials.

If directed cycles, almost directed cycles, and inducing paths are found, the algorithm applies lazy constraints in Acyclic Constraint and in Inducing-Paths Constraint. Note that removing one directed edge between two vertices where multiple paths exist is not a necessary condition for the graph to become MAG; however, this procedure can be repeated iteratively. If no inducing paths or almost directed cycles are found, we know that the program converged to the optimum, and we have a maximal ancestral graph, which minimizes (2.7).

Algorithm 2 Function that identifies inducing paths.

Input: directed edges E, bidirected edges B

```
function INDUCING-PATHS(E, B)
    D ← DISTANCE-MATRIX(E)
    for all m ∈ 1, 2, ..., M do
        INDUCING-PATHS-DFS(D, E, B, m, m, {1, 2, ... M}, [m])
    end for
end function

function INDUCING-PATHS-DFS(D, E, B, m, u, possible endpoints, path)
    if possible endpoints == ∅ then
        return {}
    end if
    if LEN(path) > 2 & u ∈ possible endpoints then
        FOUND-INDUCING-PATH(D,E, path)
                ▷ Edges participating in ancestor relation are recovered, see Suppl. mat.
    end if
    for all v ∈ 1, 2, ..., M such that e_{u,v} = 1 do
        if D_{v,m} < ∞ then
            v-endpoints ← possible endpoints
        else if D_{v,m} == ∞ then
            v-endpoints ← possible endpoints ∩{x | D_{v,x} < ∞}
        end if
        INDUCING-PATHS-DFS(D,E,B,m,v,v-endpoints,path+v)
    end for
end function
```

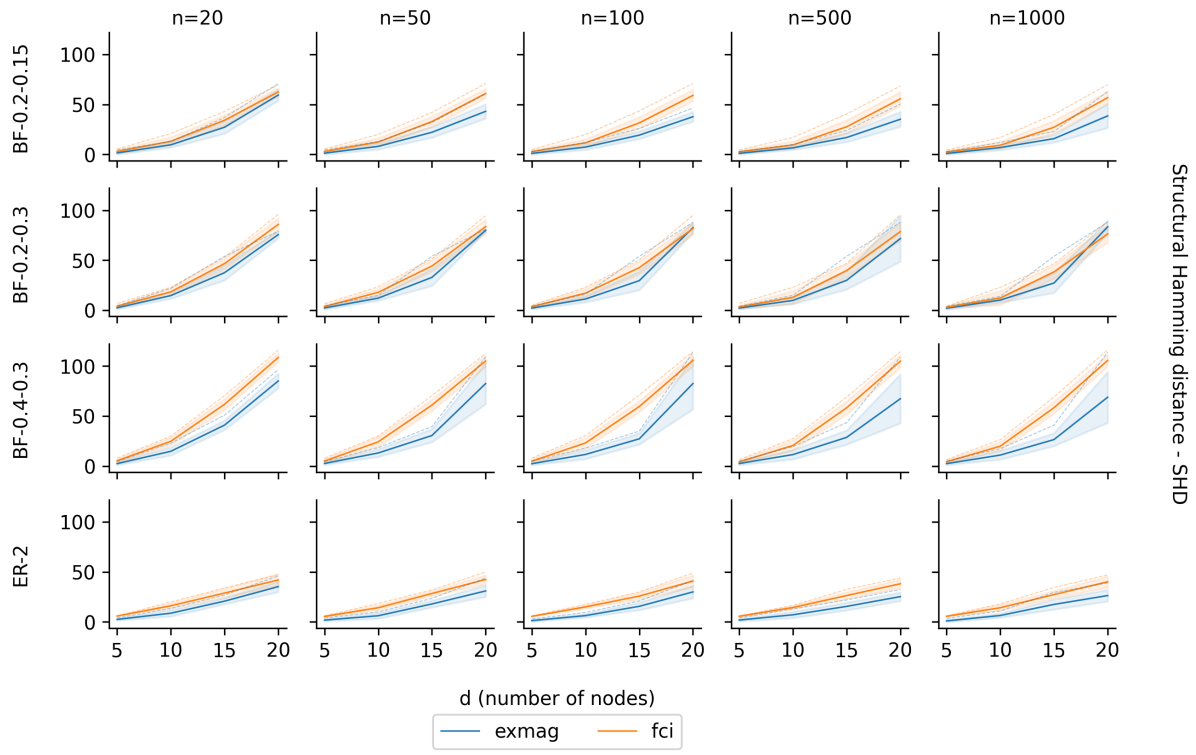


Figure 2.2: **Comparisons between ExMAG and FCI algorithm.** SHD values (in the vertical axis) for different settings of d (in the horizontal axis) and n (horizontal choice of the graph). The plots in the vertical dimension differ according to the dataset used. Standard deviations are depicted as the blurred regions, and dashed lines are the maximum values. See supplementary materials for results on more datasets and error information.

2.5 Experimental Evaluation

Datasets We tested the ExMAG algorithm on both synthetic and real-world datasets. The first synthetic data set is based on the *Erdös-Rényi model* (ER) [50], in which the ground truth graph is randomly selected from all graphs with M vertices and h edges (parameter of the experiment, for example, dataset ER-2 contains 2 edges per variable, that is, $h = 2 \cdot M$). The weights of the graph are randomly sampled from the set $(-2.0, -0.5) \cup (0.5, 2.0)$.

Once the ground truth model is created, the training data are generated using the structural model equation (2.3), (2.2) (H_m set to 0). Then, 20 % of variables are treated as latent variables and hidden from the training data. The respective columns and rows from the ground truth weight matrix W have also been removed. Finally, 20 % of edges between variables not connected by an edge in the ground truth data are marked in F.

The second dataset uses randomly generated *bow-free* (BF) graphs, a subset of all possible MAGs. A bow-free graph is a graph such that for no pair of vertices m, j, $m \to j$ and at the same time $m \leftrightarrow j$. The BF graph generation process has two parameters: the probability of a directed edge and the probability of a bidirected edge. The generation process is as follows. First, a bow-free graph with the given edge probabilities is generated randomly. Then the weights of the sampled graphs are randomly sampled from the set $(-2.0, -0.5) \cup (0.5, 2.0)$.

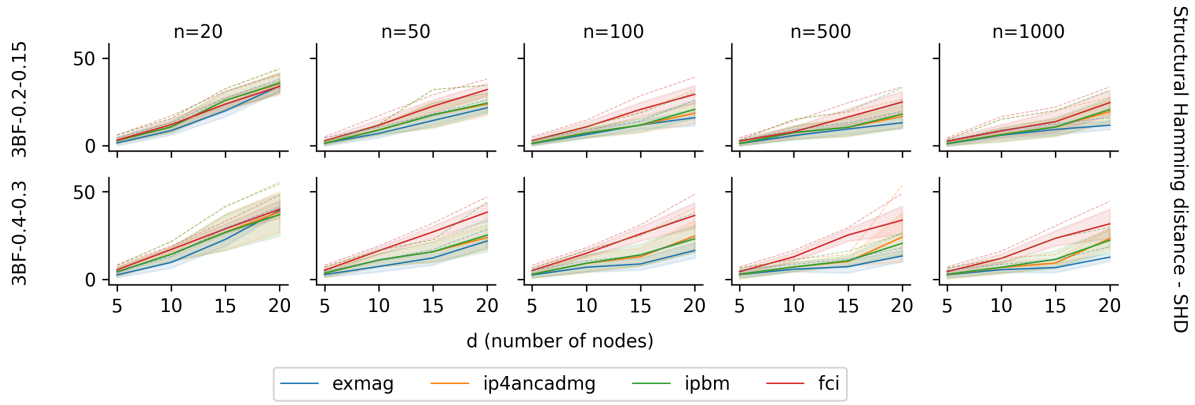


Figure 2.3: **Comparison of SHD values on 3BF datasets.**

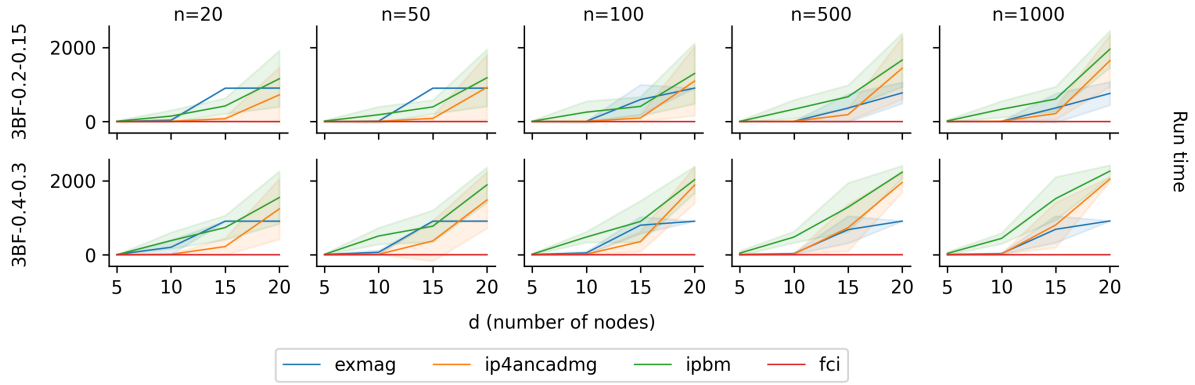


Figure 2.4: **Runtime comparisons between ExMAG, IP4AncADMG [28], IPBMs [38] and FCI algorithm [144] on 3BF datasets in seconds.**

The third synthetic dataset, 3BF, is a modified version of BF. We generate the ground truth graph in the same way as in the BF dataset and then modify it. Specifically, we identify each vertex with a degree greater than three, and randomly remove edges until the vertex has a degree of at most three.

The adjacency matrix of the directed edges defines the weights of the structural equation model. Then the data samples are generated using the structural equation, where the noise is sampled from a multivariate Gaussian distribution with a covariance matrix equal to the adjacency matrix of bidirected edges generated in the previous step.

The fourth dataset uses real-world data from the *financial* sector. Paper [5] works with systemic credit risk, one of the most important concerns within the financial system, using dynamic Bayesian networks. The data show that transport and manufacturing companies are likely to transfer risk to other sectors, while banks and the energy sector are likely to be influenced by the risks from other sectors. The data from [5] contains a 10-time series capturing the spreads of 10 European credit default swaps (CDS), and further six time series are added from [133].

We set matrix F to encode for no direct causal relationship between any two pairs of companies from different sectors. Banks sector includes 48DGFE, 05ABBF, 8B69AP, 06DABK, EFAGG9, 2H6677, FH49GG, and 8D8575. Insurance sector includes GG6EBT, DD359M, and FF667M. And lastly, transportation sector and manufacturing includes 0H99B7, 2H66B7, 8A87AG, NN2A8G, and 6A516F.

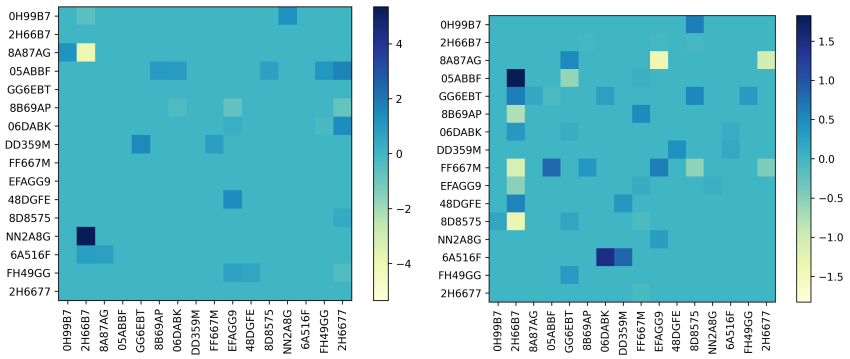


Figure 2.5: **Heatmap of weight matrix W (left) and bidirectional weight matrix B (right) on the financial dataset.**

Evaluation Criteria Suppose that a tested algorithm produced weight matrix $\hat{W}$. Such a matrix can contain nearly zero weights. For such reasons, thresholding is done, keeping only edges with weight greater than or equal to δ. In cases when the ground-truth weight matrix W is known, the best solution (in terms of structural Hamming distance, see below) is kept over those defined by different threshold δ values. In the evaluation, we use the *structural Hamming distance (SHD)*. This distance is the sum of contributions over all pairs of variables in the graph. For two variables m, j, let $GT \in \{\rightarrow, \leftarrow, \leftrightarrow, \emptyset\}$ be the edge type in the ground truth graph and $PR \in \{\rightarrow, \leftarrow, \leftrightarrow, \emptyset\}$ be edge type in the predicted graph. Then the contribution of m, j pair to SHD is 0 if $GT = PR$, 0.5 if $GT \neq PR \wedge GT \neq \emptyset \wedge PR \neq \emptyset$, and 1 otherwise. Other measured criteria include *runtime* and F_1 *score*, i.e., the harmonic mean of precision and recall.

Experiment Setting In the experiments, we show the results of ExMAG. In the case of synthetic datasets, we generated random graphs with the number of vertices M(d in the Figures 2.3 and 2.4) $\in \{5, 10, 15, 20, 25\}$. The number of samples was S(n in the Figures 2.3 and 2.4) $\in \{20, 50, 100, 500, 1000\}$, and for the ground-truth graph, the edge-to-vertex ratio was in $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20\}$. All tested algorithms were run 10 times, each time on synthetic data generated using a random generator initialised with a different seed. The results were then averaged. We compared our method with the FCI algorithm [144], IP4AncADMG [28], and IPBM [38]. We set regularization coefficient λ to 1.0. We ran experiments on a computing cluster with AMD EPYC 7543 cpus and each job had allocated two cores and 64GB RAM. Time limit was 900 seconds for ExMAG and 1800 seconds for other methods. The total cpu time needed for experiments in this paper was around one month.

Experimental Results The SHD results are shown in Figures 2.2 and 2.3 and in the supplementary materials for additional datasets. The plots show a comparison of SHD values for ExMAG on the synthetic datasets. As can be seen, the structural Hamming distance grows with the number of variables and decreases with the number of samples.

As we can see on Figures 2.2 and 2.3, ExMAG performs better than FCI on all scenarios. Since both IPBM and IP4AncADMG have a preprocessing step that depends exponentially on the maximum in-degree of the underlying ground truth graph, we tested these two algorithms only on the 3BF datasets, where the in-degree is bounded by three. We can see in Figure 2.3, that ExMAG also performs better than IPBM and IP4AncADMG.

The run times of the evaluated algorithms are shown in Figure 2.4. For additional results (incl. the F_1-score), please see the supplementary materials.

The results on the real-world dataset can be seen in Figure 2.5. Contrary to the original expectations, the highest risk importer is company 2H66B7, which stands for Lufthansa. The second highest risk importer is 2H6677, i.e., the Deutsche Bank, which is an expected result.

2.6 Conclusion and Limitations

Learning of dynamic Bayesian networks has received considerable attention as a means of causal learning. With a few exceptions, the research has not considered confounding explicitly. Our method, ExMAG, estimates a maximally ancestral graph, capturing confounding and causal relationships using bidirected and directed edges in a mixed graph. The method provides state-of-the-art statistical performance.

As with many other methods for causal learning, the scalability of the method may leave space for improvement. Although the branch-and-cut algorithm runs in time that is exponential in the number of time series in the worst case, Figure 2.4 illustrates that our run time is lower than those of IP4AncADMG [28] and IPBM [38], while improving the SHD of both recent competitors at the same time (cf. Figure 2.3). One could improve upon the run-time further by introducing additional cutting planes and more elaborate data structures for the separation of Acyclic Constraint and Inducing-Paths Constraint, perhaps drawing inspiration from solvers [36, e.g.] for the travelling salesman problem.

In terms of future work, exploring the predictive power of forecasting using variants of dynamical Bayesian networks with confounding considerations seems prominent. Although it seems clear that marginalisation is hard even in dynamical Bayesian networks, and thus the computational complexity may be high, but statistical performance is likely to improve, when confounding is considered.

Chapter 3

Joint Problems in Learning Multiple Dynamical Systems

Beyond single-system analysis, many real-world datasets involve multiple interacting dynamical systems that share latent structures. To capture this complexity, I investigate joint problems *in learning multiple linear dynamical systems (LDS). Specifically, I extend the Non-Commutative Polynomial Optimization (NCPOP) framework with an Expectation-Maximization (EM) heuristic to enable simultaneous clustering and parameter estimation across multiple trajectories. This extension not only improves statistical efficiency but also allows the detection of common latent dynamics across systems. By addressing joint problems, the methodology advances causal inference in dynamic environments and provides a scalable algorithmic framework for modeling interconnected temporal processes in domains such as biomedicine and neuroscience.*

3.1 Introduction

The task of clustering similar time series based on their dynamic patterns has attracted significant attention due to its applications ranging from studying mobility patterns [99] to improving Apple Maps [27], through quantitative, personalised models of metabolism obtained from metabolite concentrations, all the way to state discrimination problems in quantum information theory [92].

We consider a variant, where given a set of trajectories and a number of parts, we jointly partition the set of trajectories and learn the autonomous discrete-time Linear Dynamical System (LDS) [159] models, for each cluster, where X_t are the hidden states and x_t are the observations. The cluster-specific LDSs may exhibit similar behaviours in terms of system matrices $\mathbf{G}, \boldsymbol{\varphi}$, or not. $\mathbf{G}'$ denotes the transpose of $\mathbf{G}$. The observations convey information about the cluster-specific LDSs. Clustering time-series data entails dividing sequences into groups that exhibit similar dynamic behaviours, as exemplified in Figure 3.1.

The main contributions of this paper are the following.

- We propose a novel problem in the clustering time-series considering Linear Dynamic System [159] models for each cluster. The linearity assumption comes without a loss of generality as any non-linear system can be modeled as an LDS [11], in a sufficiently higher dimension.

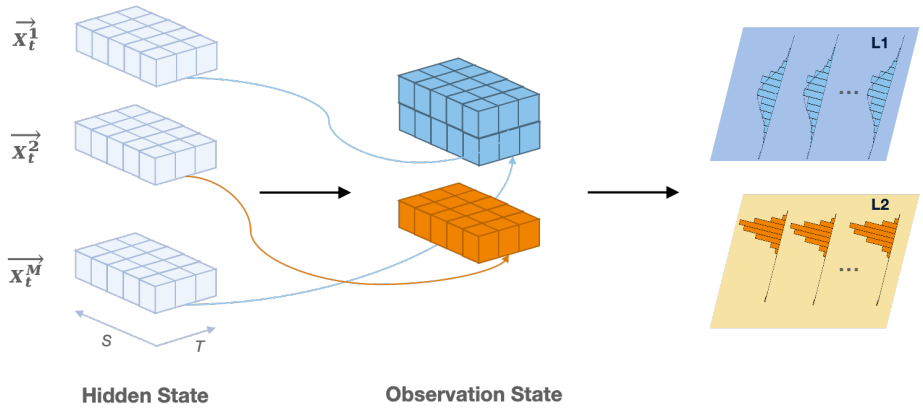


Figure 3.1: **Diagram of clustering time-series** considering linear dynamic system models for each cluster.

- We provide an abstract formulation within Non-Commutative Polynomial Optimization (NCPOP). NCPOP [118] is a framework for operator-valued optimization problems, and thus does not require the dimension of the hidden state to be known ahead of time, which had been known [89] to be a major limitation of LDS-based methods. This paper is one of the first applications of NCPOP in machine learning.

- As a complement to the NCPOP formulation, we provide an efficient Expectation-Maximization (EM) procedure [46]. Through iterative measurement of prediction errors and systematic updates to the system matrix, we can effectively identify the per-cluster LDSs and the assignment of time series to clusters.

3.2 Background

This section provides an overview and necessary definitions of the background, problems, and algorithms. The notations used here follow the Chapter Table of Notation.

3.2.1 Linear Dynamic Systems (LDS)

In system identification research [90], a well-established way to represent Linear Dynamical Systems (LDS) is through a quadruple $\mathbf{L} = (\boldsymbol{\varphi}, \mathbf{G}, \boldsymbol{\Sigma}_H, \boldsymbol{\Sigma}_O)$. Here, $\boldsymbol{\varphi}$ is the system matrix and $\boldsymbol{\Sigma}_H$ is the covariance matrix of the *state transition process*, while $\mathbf{G}$ is the system matrix and $\boldsymbol{\Sigma}_O$ is the covariance matrix of the *observation process* [159]. A single realisation of the LDS or a *trajectory* of length T can be denoted by $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_T\} \in \mathbb{R}^{M \times S \times T}$. Let n be the hidden state dimension and M be the observational dimension and the observed outputs of $\mathbf{L}$(i.e., $Y_t \in \mathbf{Y}$) is obtained by

$$\underset{(n\times S)}{X_t} = \underset{(n\times n)}{\boldsymbol{\varphi}}\,\underset{(n\times S)}{X_{t-1}} + \underset{(n\times S)}{\omega_t}, \tag{3.1}$$

$$\underset{(M\times S)}{Y_t} = \underset{(M\times n)}{\mathbf{G}'}\,\underset{(n\times S)}{X_t} + \underset{(M\times S)}{v_t}, \tag{3.2}$$

where X_t is the vector autoregressive(VAR) processes with hidden components and *initial conditions* X_0. $\{\omega_t, \upsilon_t\}_{t\in\{1,2,\dots,T\}}$ are normally distributed process with zero mean and covariance of $\mathbf{\Sigma}_H \in \mathbb{R}^{n\times S}$ and $\mathbf{\Sigma}_O \in \mathbb{R}^{M\times S}$ respectively, i.e., $\omega_t \sim \mathcal{N}(0, \mathbf{\Sigma}_H)$ and $\upsilon_t \sim \mathcal{N}(0, \mathbf{\Sigma}_O)$. The transpose of $\mathbf{G}$ is denoted as $\mathbf{G}'$. Recently, Zhou and Marecek [176] proposed to find the global optimum of the objective function subject to the feasibility constraints arising from (3.1) and (3.2):

$$\min_{\hat{Y}_t, X_t, \boldsymbol{\varphi}, \mathbf{G}, \omega_t, \upsilon_t} \sum_{t\in\{1,2,\dots,T\}} \|Y_t - \hat{Y}_t\|_2^2 + \|\omega_t\|_2^2 + \|\upsilon_t\|_2^2, \tag{3.3}$$

for a L_2-norm $\|\cdot\|_2$. In the joint problem we are given M trajectories $Y^m \in \mathbb{R}^{S\times T}$. A natural problem to solve is to find the parameters of the LDS that generated the trajectories. In other words, we are interested in finding the optimal objective values, as well as system matrices $\boldsymbol{\varphi}, \mathbf{G}$, and the noise vectors υ_t, ω_t that belong to each LDS.

One should like to remark that learning the LDS is an NP-Hard problem. This is easy to see when one realises [131] that Gaussian mixture models (GMM), autoregressive models, and hidden Markov models are all special cases of LDS, and whose learning is all NP-Hard, even in very special cases such as spherical Gaussians [148] in a GMM. Furthermore, there are also inapproximability results [148] suggesting that there exists an approximation ratio, at which no polynomial-time algorithm is possible unless P = NP.

3.2.2 Clustering with LDS Assumptions

The problem of clustering of time series is relevant in many fields, including [1, 157] applications in Bioinformatics, Multimedia, Robotics, Climate, and Finance. There are a variety of existing methods, including those based on (Fast) Discrete Fourier Transforms (FFTs), Wavelet Transforms, Discrete Cosine Transformations, Singular Value Decomposition, Levenshtein distance, and Dynamic Time Warping (DTW). We compare our method with the FFT- and DTW-based methods.

Many methods combine the ideas of system identification and clustering, sometimes providing tools for clustering time series generated by LDSs, similar to our paper. With three related papers at ICML 2023, this could be seen as a hot topic. We stress that neither of the papers has formulated the problem as either a mixed-integer program or an NCPOP, or attempted to solve the joint problem without decomposition into multiple steps, which necessarily restricts both the quality of the solutions one can obtain in practice, as well as the strength of the guarantees that can be obtained in theory.

To our knowledge, the first mention of clustering with LDS assumptions is in the paper of [87], who introduced ComplexFit, a novel EM algorithm to learn the parameters for complex-valued LDSs and utilised it in clustering. In [89], regularization has been used in learning linear dynamical systems for multivariate time series analysis. In a little-known but excellent paper at AISTATS 2021, Hsu et al. [68] consider clustering with LDS assumptions, but argue for clustering on the eigenspectrum of the state-transition matrix ($\boldsymbol{\varphi}$ in our notation), which can be identified for unknown linear dynamics without full system identification. The main technical contribution is bidirectional perturbation bounds to prove that two LDSs have similar eigenvalues if and only if their output time series have similar parameters within Autoregressive-Moving-Average (ARMA) models. Standard consistent estimators for the autoregressive model parameters of the ARMA models are then used to estimate the LDS eigenvalues, allowing for linear-time algorithms.

We stress that the eigenvalues may not be interpretable as features; one has to provide the dimension of the hidden state as an input.

At ICML 2023, Bakshi et al. [4] presented an algorithm to learn a mixture of linear dynamic systems. Each trajectory is generated so that an LDS is selected on the basis of the weights of the mixture, and then a trajectory is drawn from the LDS. Their approach is, unlike ours, based on moments and the Ho-Kalman algorithm and tensor decomposition, which is generalised to work with the mixture. Empirically, [4] outperforms the previous work of Chen et al. [29] of the previous year, which works in fully observable settings. In the first step, the latter algorithm [4] finds subspaces that separate the trajectories. In the second step, a similarity matrix is calculated, which is then used in clustering and consequently can be used to estimate the model parameters. The paper [4] also discusses the possibility of classification of new trajectories and provides guarantees on the error of the final clustering.

There are also first applications of the joint problems in the domain-specific literature. Similarly to the previous paper, a fully observable setting of vector autoregressive models is considered in [21], with applications in Psychology, namely on depression-related symptoms of young women. Similarly to our method, the least-squares objective is minimized to provide clustering in a manner similar to the EM-heuristic. See also [51] for further applications in Psychology. One can easily envision a number of further uses across Psychology and Neuroscience, especially when the use of mixed-integer programming solvers simplifies the time-consuming implementation of EM algorithms.

3.3 Problem Formulation

Suppose we have M trajectories, denoted by $Y^m \in \mathbb{R}^{S\times T}$ for $m = 1, 2, \ldots, M$. We assume that these trajectories are drawn from K clusters, $\mathbf{C}_0, \mathbf{C}_1, \ldots, \mathbf{C}_{K-1}$, where the trajectories in cluster $\mathbf{C}_k$ are generated by a Linear Dynamical System (LDS) $\mathbf{L}_k$. We aim to jointly partition the given set of trajectories into K clusters and recover the parameters of the LDSs systems of all K clusters. To solve these joint optimization problems, we can introduce an *indicator function* to determine how the M trajectories are assigned to K clusters:

$$l_{m,k} = \begin{cases} 0, & \text{if } m \in \mathbf{C_k}, \\ 1, & \text{if } m \notin \mathbf{C_k}, \end{cases} \tag{3.4}$$

for $m \in \{1, 2, \ldots, M\}$ and $k \in \{0, 1, \ldots, K-1\}$. Simply put, we consider partitioning the given set of trajectories into $K = 2$ clusters, $\mathbf{C}_0$ and $\mathbf{C}_1$. The trajectories in cluster $\mathbf{C}_0$ (resp. $\mathbf{C}_1$) are assumed to be generated by an LDS $\mathbf{L}_0$ (resp. $\mathbf{L}_1$). In this scenario, we introduce a simplified *indicator function* to specify how the M trajectories are assigned to the two clusters:

$$l_m = \begin{cases} 0, & \text{if } m \in \mathbf{C_0}, \\ 1, & \text{if } m \in \mathbf{C_1}, \end{cases} \tag{3.5}$$

for $m \in \{1, 2, \ldots, M\}$.

3.3.1 Least-Squares Objective Function

We define the optimization problem with a least-squares objective that minimizes the difference of measurement estimates $Y_t^m \in \mathbb{R}^S$, $\hat{Y}_t^m \in \mathbb{R}^S$, $\upsilon_t^k \in \mathbb{R}^{M_K\times S}$, $\omega_t^k \in \mathbb{R}^{n\times S}$

and the corresponding measurements. Other variables include noise vectors that come with the estimates; indicator function l_i that assigns the trajectories to the clusters, and parameters of systems $\mathbf{L_k}$(also known as system matrices). The objective function is:

$$\min \sum_{m=1}^{M}\sum_{t=1}^{T}\left\|Y_t^m - \hat{Y}_t^m\right\|_2^2 + \lambda\sum_{k=1}^{K}\sum_{t=1}^{T}\left[\left\|\upsilon_t^k\right\|_2^2 + \left\|\omega_t^k\right\|_2^2\right],$$
$$\omega_t^k \sim \mathcal{N}\left(0, \mathbf{\Sigma_H^k}\right), \upsilon_t^k \sim \mathcal{N}\left(0, \mathbf{\Sigma_O^k}\right), k \in \{0, 1, \ldots, K-1\}, \quad (3.6)$$

where ω_t^k, υ_t^k are defined above and $\|\cdot\|_2$ denotes the L_2 norm. In the first part of the optimization criterion (3.6), we have a sum of squares of the difference between trajectory estimate $\hat{Y}_t^{l_{m,k}}$ and observations of the trajectories assigned to the cluster C_k. It was replaced by multiplication with the indicator function, i.e.,

$$\|Y_t^m - \hat{Y}_t^m\|_2^2 = \sum_{k=1}^{K}\left[\|Y_t^m - \hat{Y}_t^k\|_2^2 \cdot l_{m,k}\right]. \quad (3.7)$$

The second part of the optimization criterion (3.6) provides a form of regularization. Note that the indicator index $l_{m,k}$ defined as (3.4) in (3.6) can be simplified as (3.5) in the case of K=2. Therefore, the objective function leads to

$$\min_{l_m \in \{0,1\}} \sum_{m=1}^{M}\sum_{t=1}^{T}\left[\|Y_t^m - \hat{Y}_t^0\|_2^2 \cdot (1 - l_m) + \|Y_t^m - \hat{Y}_t^1\|_2^2 \cdot l_m\right] + \lambda\sum_{k=0}^{1}\sum_{t=1}^{T}\left[\left\|\upsilon_t^k\right\|_2^2 + \left\|\omega_t^k\right\|_2^2\right],$$
$$\omega_t^k \sim \mathcal{N}\left(0, \mathbf{\Sigma_H^k}\right), \upsilon_t^k \sim \mathcal{N}\left(0, \mathbf{\Sigma_O^k}\right), \forall k \in \{0, 1\}, \quad (3.8)$$

Proof.

$$\|Y_t^m - \hat{Y}_t^m\|_2^2 = \sum_{k=1}^{2}\left\{\|Y_t^m - \hat{Y}_t^k\|_2^2 \cdot l_{m,k}\right\} \quad = \|Y_t^m - \hat{Y}_t^0\|_2^2 \cdot l_{m,0} + \|Y_t^m - \hat{Y}_t^1\|_2^2 \cdot l_{m,1} \quad (3.9)$$

In the two-cluster setting, instead of using a two-dimensional indicator $\{l_{m,0}, l_{m,1}\}$ with $l_{m,0}, l_{m,1} \in \{0, 1\}$ and $l_{m,0} + l_{m,1} = 1$, we introduced a one-dimensional cluster label l_m represented as (3.5), where $l_m = 1$ indicates that trajectory m belongs to cluster $\mathbf{C}_1$ and $l_m = 0$ indicates that trajectory m belongs to cluster $\mathbf{C}_0$. Then the one-dimensional label can be converted from the original two-dimensional by

$$\begin{bmatrix} l_{m,0} & l_{m,1} \end{bmatrix} = \begin{bmatrix} 1 - l_m & l_m \end{bmatrix}.$$

Accordingly, the equation (3.9) is simplified as

$$\|Y_t^m - \hat{Y}_t^0\|_2^2 \cdot l_{m,0} + \|Y_t^m - \hat{Y}_t^1\|_2^2 \cdot l_{m,1} = \|Y_t^m - \hat{Y}_t^0\|_2^2 \cdot (1 - l_m) + \|Y_t^m - \hat{Y}_t^1\|_2^2 \cdot l_m. \quad (3.10)$$

□

3.3.2 Feasible Set in State Space

When clusters K, the feasible set given by constraints is as follows:

$$X_t^k = \boldsymbol{\varphi}_k X_{t-1}^k + \omega_t^k, \tag{3.11}$$

$$\hat{Y}_t^k = \mathbf{G}_k' X_t^k + \upsilon_t^k, \tag{3.12}$$

$$\sum_{m=1}^{M} l_{m,k} = 1, \quad l_{m,k} \in \{0,1\}, \tag{3.13}$$

for $\forall t = 1, \ldots, T$, and $\forall k = 0, 1, \ldots, K-1$. When clusters $k \in \{0,1\}$, the constraint (3.13) equals to the following:

$$l_m^2 = l_m, \tag{3.14}$$

for $\forall m = 1, \ldots, M$. The first two equations in the set of constraints, (3.11) and (3.12), encode that the system is an LDS with system matrices $\mathbf{G}$ and $\boldsymbol{\varphi}$. In the simplest case, both equations are indexed by the cluster index k when $K = 2$, which can be rewritten as twice as many equations, one with $\hat{Y}_t^0, \mathbf{G}_0, \boldsymbol{\varphi}_0, \upsilon_t^0, \omega_t^0$, and X_t^0, the second one with $\hat{Y}_t^1, \mathbf{G}_1, \boldsymbol{\varphi}_1, \upsilon_t^1, \omega_t^1$, and X_t^1. The third equation (3.14) encodes that the indicator function is 0 or 1 for each trajectory.

A weighted combination of the redundant constraints in the spirit of Gomory. While these strengthen the relaxations, the higher-degree polynomials involved come at a considerable cost. Still, even when the dimension of the hidden state n is unknown, we can solve the corresponding operator-valued problem:

Theorem 3.1. *There exists a series of convex relaxations, whose optima asymptotically converge to the true global optimizer of the problem Equation* (3.8) *subject to (3.11–3.14).*

Proof. Let m, n be positive integers, $x \in \mathbb{R}^n$ be a tuple of real-valued variables, and p, q_i, $i = 1, \ldots, m$ be some polynomials in the variable x. Polynomial optimization consider $\min_{x \in \mathbb{R}^n} p(x)$ subject to $q_i(x) \geq 0$ for $i = 1, \ldots, m$. Under the Archimedeam assumption, such feasible region is a compact semi-algebraic set. Note that the formulation is equivalent to finding the maximum number α that makes the polynomial $p(x) - \alpha$ nonnegative on the compact semi-algebraic set defined by $q_i(x)$, $i = 1, \ldots, m$. Then, according to the Putinar's certificate of positive polynomials (i.e., Putinar's positivstellensatz), if a polynomial p is strictly positive on a compact semi-algebraic set, there exists a sequence of sum-of-square polynomials g_i, $i = 0, \ldots, m$ such that $p = g_0 + \sum_{i=1}^{m} q_i g_i$, where verifying sum-of-square polynomials is by solving SDP problems. Considering this, Lasserre's hierarchy of SDP relaxations provides global convergence for Polynomial optimisation [85, 86], following Putinar's positivstellensatz and Curto-Fialkows theorem.

NCPOP is the extension of polynomial optimization to consider the variables $X = \{X_1, \ldots, X_n\}$ which are not simply real numbers but non-commutative variables, for which, in general, $X_i X_j \neq X_j X_i$. The polynomials e.g., $p(x)$, are defined by substituting the variables x by the tuple of operators X in the expression $p(X)$. The global convergence was provided in Navascués-Pironio-Acín (NPA) hierarchy of SDP relaxations [119, 105], following Helton and McCullough's certificate of non-commutative positive polynomials [63]. To lower the computational burden of NCPOP, the sparsity-exploiting variants were provided [154, 80]. □

Despite the existence of the relaxations, we can show that the soft-clustering version of the problem is NP-hard, as the problem can be transformed to finding a clustering of a mixture of Gaussian distributions, a related and well-studied problem known to be NP-hard even for spherical clusters [149].

Theorem 3.2. *Finding a soft clustering of a mixture of LDS trajectories with a log-likelihood within an additive factor of the optimal log-likelihood is NP-hard even when* $K = 2$.

3.3.3 Variants and Guarantees

There are several variants of the formulation above. Notably, one can:

- consider a fixed, finite dimension of the hidden state X to be known and to solve a finite-dimensional (MINLP).
- consider side constraints on the system matrices $\mathbf{G}, \boldsymbol{\varphi}$, as in Ahmadi and El Khadir [2] – or not. At least requiring the norm of $\boldsymbol{\varphi}$ to be 1 is without loss of generality.
- bound the magnitude of the process noise ω^c and observation noise υ^c, or other shape constraints thereupon.
- bound the cardinality of the clusters – or not.

Throughout, one obtains asymptotic guarantees on the convergence of the NCPOP, or guarantees of finite convergence in the case of the MINLP.

3.4 EM Heuristic

In addition to tackling the optimization problem in Section 3.3.1, we provide an alternative solver using the Expectation-Maximization (EM) heuristic [46]. The main idea of the algorithm is to avoid the direct optimization of the criterion in (3.6). Instead, the indicator function is treated separately. In the expectation step, the parameters of the LDSs are fixed, and the assignment of the trajectories to the clusters (i.e., the indicator function) is calculated. Then, in the maximization step, the criterion is optimized, and the LDS parameters are calculated with the indicator function fixed. See Algorithm (3) for the pseudocode. First, the algorithm randomly partitions the trajectories into the clusters. Then, for each cluster, the parameters of the LDSs are found, and with the parameters known, the optimization criterion is recalculated, and each trajectory is put to the cluster, which lowers the error in (3.6).

The advantage is that the problem of finding the parameters and an optimal trajectory for a set of trajectories is easier than clustering the trajectories. Another advantage of the EM heuristic is that it can be easily generalised to an arbitrary number of clusters K, generally for any $K > 1$.

In the supplementary materials, we prove the following theorem that shows that the problem of clustering a mixture of LDSs is no more difficult than clustering a mixture of Gaussian distributions as below.

Theorem 3.3. *There exists a polynomial reduction that reduces the problem of clustering a mixture of autonomous LDSs with hidden states to the clustering of a mixture of Gaussian distributions.*

Algorithm 3 The EM heuristic.

function EM-CLUSTERING(M trajectories $\mathbf{Y}^m \in \mathbb{R}^{S\times T}$, K)

▷ Generate a random clustering into K clusters.

$l_m \leftarrow \text{RANDOMINT}(\{0, 1, \ldots, K-1\})$

▷ Iterate until convergence.

while l_m changes for any $m \in \{1, 2, \ldots, M\}$ **do**

▷ For each cluster find cluster parameters

for $k \in \{1, 2, \ldots, K\}$ **do**

Find the cluster C_k parameters by solving (3.6)

$$\min_{\hat{Y}_t^k, \mathbf{G}_k, \boldsymbol{\varphi}_k, \upsilon_t^k, \omega_t^k, X_k^0} \left[\sum_{m=1}^{M} \sum_{t=1}^{T} \mathbb{1}[l_m = k] \cdot \|Y_t^m - \hat{Y}_t^k\|_2^2 \right] + \|\upsilon_t^k\|_2^2 + \|\omega_t^k\|_2^2$$

end for

▷ Reassign the trajectories to the clusters.

for $m \in \{1, 2, \ldots M\}$ **do**

$$l_m \leftarrow \underset{k\in\{0,1,\ldots,K-1\}}{\arg\min} \sum_{t=1}^{T} \|Y_t^m - \hat{Y}_t^k\|_2^2$$

end for

end while

end function

The theorem justifies the usage of the EM-algorithm. Unfortunately, applying the previous theorem directly to the joint problem is computationally inefficient, as a quadratic number of parameters needs to be estimated. The advantage is that we can exploit the theoretical guarantees for the mixture of Gaussians problem - for example, the local convergence to a global optimum [172], existence of arbitrarily bad local minima [75], and also a linear bound on the number of samples in the case of spherical clusters [82]. In the case when there are only two clusters, the EM-algorithm-based estimates are guaranteed to converge to one of three cases [166]. See the Supplementary materials for more details.

It would be of considerable interest to analyse the behaviour of the EM heuristic in our setting. Indeed, for many problems, such as the parameter estimation of Gaussian mixture models [48, 158], the properties of EM approaches are well understood [48, 65, 158, 171, 66]. The joint problems are very similar to the clustering of mixture of Gaussian distributions over the system matrices as we have seen in Theorem 3.3 - in our setting of autonomous LDSs with hidden state, one can treat all observations of a trajectory as individual features in a high-dimensional space and the resulting vector will follow a normal distribution with additional constraints applied on its parameters.

As EM-algorithm applied to the mixture of Gaussians is, in many scenarios, computationally inefficient, we propose to use heuristic (1), which can be seen as a parallel to the Lloyd's algorithm [91] for k-means problem. See the Supplementary materials for a formal connection to the k-means problem.

3.5 Experiments

In this section, we present a comprehensive set of experiments to evaluate the effectiveness of the proposed (MINLP) and EM heuristic, without considering any side constraints and without any shape constraints. Our experiments are conducted on Google Colab with two Intel(R) Xeon(R) (2.20GHz) CPUs and Ubuntu 22.04. The source code is included in the Supplementary Material.

3.5.1 Methods and Solvers

MIP-IF Our formulation for MINLP with an indicator function (MIP-IF) uses equations (3.8) subject to (3.11–3.14). We specify the dimension of system matrix $\boldsymbol{\varphi}$, i.e., n as the hyperparameter. For every data set, we perform 50 experiments, in which different random seeds are employed to initialise the indicator function in each iteration. These MIP instances are solved via Bonmin[1] [16] and Gurobi [2] [59].

EM Heuristic Iterated EM Heuristic clustering is presented in Algorithm (3). As in MIP-IF, the upper bound of dimension, n, of the system matrix is required. In each iteration, the cluster partition is randomly initialised and we conduct 50 trials with various random seeds for every dataset.

The discussion of runtime As a subroutine of the EM Heuristic, the LDS identification in equation (3.8) is solved via Bonmin, with runtime presented in the center subplot of Figure 3.4. When the dimensions of system matrices are not assumed, the LDS identification becomes an NCPOP, and the runtime increases exponentially if NPA hierarchy [105, 119] is used to find the global optimal solutions, but stays relatively modest if sparity exploit variants [154, 155, 153] are used.

Baselines We consider the following traditional time series clustering methods as baselines:

- **Dynamic Time Warping (DTW)** is used for measuring similarity between given time series [140, 57]. We utilise K-means on DTW distance with tslearn [147] package.
- **Fast Fourier Transform (FFT)** is utilised to obtain the Fourier coefficients and the distance between time series is evaluated as the L2-norm of the difference in their respective Fourier coefficients. Subsequently, K-means is employed to cluster time series using this distance.

3.5.2 Experiments on Synthetic Data

Data Generation We generate LDSs by specifying the quadruple $(\boldsymbol{\varphi}, \mathbf{G}, \boldsymbol{\Sigma}_H, \boldsymbol{\Sigma}_O)$ and the initial hidden state X_0. $\boldsymbol{\varphi}$ and $\mathbf{G}$ are system matrices of dimensions $n \times n$ and $n \times M$, respectively. For each cluster, we derive $\frac{M}{2}$ trajectories including T observations. n is chosen from $\{2, 3, 4\}$. To make these LDSs close to the center of the respective cluster, we fix the system matrices while only changing the covariance matrices $\boldsymbol{\Sigma}_H, \boldsymbol{\Sigma}_O$

[1] https://www.coin-or.org/Bonmin/
[2] https://www.gurobi.com/

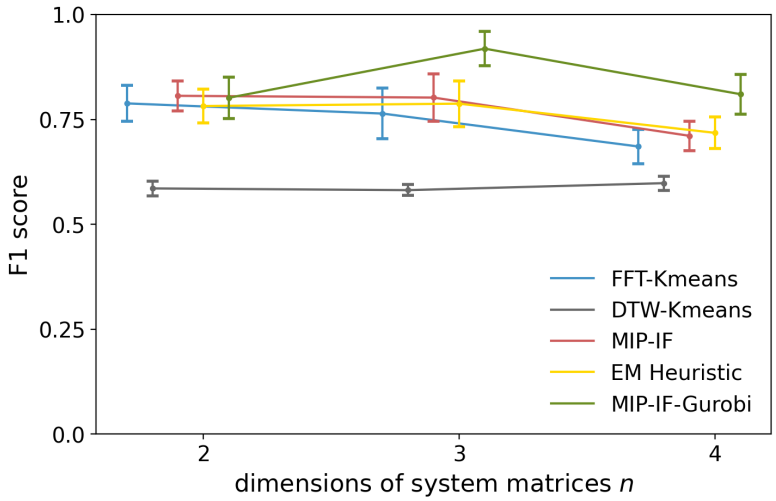


Figure 3.2: $F_1 - scores$ **comparisons between EM Heuristic algorithm and baselines with varying hidden state dimensions setting.** Results are based on data generated by LDSs with $n \in \{2, 3, 4\}$. Vertical error bars denote the 95% confidence intervals over 50 trials. Higher F_1 scores indicate better clustering performance.

from $0.0004, 0.0016, 0.0036$, to 0.0064 respectively. Note that $\mathbf{\Sigma}_H = 0.0004$ here refers to $\mathbf{\Sigma}_H = 0.0004 \times \mathbf{I}_n$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix. We consider all combinations of covariance matrices (16 trajectories) for each cluster.

Results F_1 score is exploited to evaluate models' performance. In our first simulation, we explore the effectiveness of the proposed methods in synthetic datasets. For each choice of $n \in \{2, 3, 4\}$, we run 50 trials. In each trial, the indicator function l_m for MIP-IF and the original clustering partition for EM Heuristic are randomly initialised. Figure 3.2 illustrates the F_1 score of our methods and baselines, with 95% confidence intervals from 50 trials. Different approaches are distinguished by colours. Both solutions proposed yield superior cluster performance considering $n \in \{3, 4\}$. When $n = 2$, our methods can achieve comparable performance to FFT-Kmeans. These experiments thus demonstrate the effectiveness of our approach. See the Supplementary Material for further details.

3.5.3 Experiments on Real-world Data

Next, we conduct experiments on real-world data.

ECG data The test on electrocardiogram (ECG) data gives an inspiring application on guiding cardiologist's diagnosis and treatment [108]. The ECG data ECG5000 [40] is a common dataset for evaluating methods for ECG data, which has also been utilised by other papers [68] on clustering with LDS assumptions. The original data comes from Physionet [3, 54] and contains a 20-hour-long ECG for congestive heart failure. After processing, ECG5000 includes 500 sequences, where there are 292 normal samples and 208 samples of four types of heart failure. Each sequence contains a whole period of heartbeat with 140 time stamps.

Results on ECG We randomly sample two clusters from normal sequences and one type of abnormal sequences respectively. As the entire period of time series data is

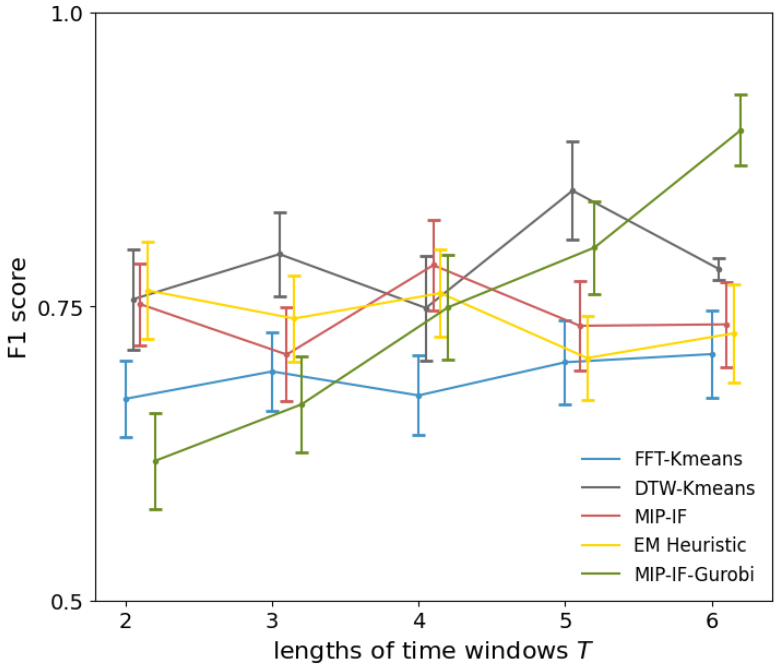


Figure 3.3: $F_1 - scores$ **comparisons between EM Heuristic algorithm and baselines with varying time windows setting.** Results are based on data sampled from ECG5000 with varying time window T. Higher F_1 scores indicate better clustering performance.

not always available, we also extract subsequences with varying time window T to test the clustering performance. In Figure 3.3, with the assumption of the upper bound of hidden state dimension is $n = 5$, we implement all methods for 50 runs at each length of time window. Our methods exhibit competitive performance relative to FFT and DTW when T increases. When the time window decreases, the performance of the baselines significantly deteriorates, while our methods maintain a higher level of robustness.

In the left two subplots of Figure 3.4, we further explore the performance of our methods at varying dimensions of the hidden state ($n \in \{2, 3, 4\}$), because the dimension of the hidden state n of the ECG data is, indeed, unknown. When the length of the time window increases, both methods experience a slight improvement in clustering performance, but this performance remains relatively stable when the dimension n changes. The runtime is presented in the center subplot. Compared to MIP-IF, the EM Heuristic exhibits a modest growth in runtime as the length of the time window increases. Finally,

Methods		EM n=2	EM n=3	EM n=4	NCPOP
F_1 score	T=10	0.728	0.619	0.788	**0.794**
	T=20	0.805	0.897	**0.927**	0.699
	T=30	0.843	0.764	0.842	**0.927**

Table 3.1: F_1 **Scores of the NCPOP-based EM Heuristic**, compared with methods requiring specific hidden state dimensions n, across window sizes $T \in \{10, 20, 30\}$.

when the dimension n of the hidden state is not assumed, the subproblem of the EM

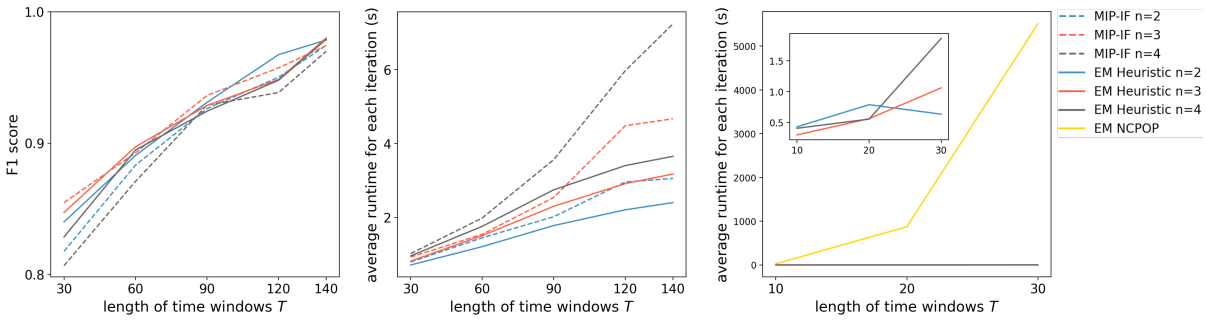


Figure 3.4: **Performance of EM Heuristic. left** and **center**: F_1 scores and runtimes of MIP-IF and EM Heuristic with various time window T chosen from $\{30, 60, 90, 120, 140\}$ respectively. The F_1 score improves as the time window increases. **right**: Runtimes of EM Heuristic using NCPOP compared with the method requiring specific dimension n of hidden state. n is chosen from $\{2, 3, 4\}$.

heuristic becomes an NCPOP. For the implementation with such an assumption, we construct NCPOP using ncpol2sdpa 1.12.2[3] [163]. Subsequently, the relaxation problem is solved by Mosek 10.1[4] [100]. Noted that the execution time of NCPOP escalates rapidly as the trajectory length T grows, we test NCPOP and compare its performance with the aforementioned EM Heuristic method with $T \in \{10, 20, 30\}$. For comparison, we use pyomo[5] [23] to construct the model and solve the problem with Bonmin[6] [16], as above, with the dimension n from $\{2, 3, 4\}$. The overall performance is illustrated in Table 3.1. NCPOP demonstrates the best performance in terms of the F_1 score when $T = 10$ and 30. However, as shown in the right subplots of Figure 3.4, the runtime of NCPOP grows significantly as the length T of the trajectory increases.

3.6 Conclusions and Further Work

We have studied problems in clustering time series, where given a set of trajectories and a number of parts, we jointly partition the set of trajectories and estimate a linear dynamical system (LDS) model for each part, so as to minimize the maximum error across all the models. As discussed in Section 3.3.3, a number of variants of the joint problem remain to be investigated. The computational aspects of the operator-valued problem [176] that consider the dimension of the hidden state to be unknown seem particularly interesting. We present a novel method for causal learning. We publish all source files used to generate the data and the figures in this paper in the GitHub repositories [2] and [3].

[3] https://ncpol2sdpa.readthedocs.io/en/stable/
[4] https://www.mosek.com/
[5] https://www.pyomo.org/
[6] https://www.coin-or.org/Bonmin/
[2] https://github.com/sereneHe/Clustering
[3] https://github.com/nnnnmj/joint-problem

Chapter 4

Appendix

A. More Experimental Results from Chapter 2

A.1 Pseudocode

Algorithm 4 Functions that help in the separation routine.

function TRACE-DISTANCE-MATRIX(D, E, j, k)
  if $D_{j,k} == \infty$ **then**
    return {}
  end if
  visited, stack, edges = $\{(j,k)\}$, (j,k), {}
  while stack $\neq$ empty **do**
    $u, v \leftarrow$ POP(stack)
    for all $w \in 1, 2, \ldots, d$ s t. $D_{u,w} + D_{w,v} < \infty$ **do**
      visited $\leftarrow$ visited $\cup\{(u,w),(w,v)\}$
      if $E_{u,w}$ **then**
        edges $\leftarrow$ edges $\cup\{(u,w)\}$
      else if $E_{w,v}$ **then**
        edges $\leftarrow$ edges $\cup\{(w,v)\}$
      end if
      stack $\leftarrow$ stack $\cup\{(u,w),(w,v)\}$
    end for
  end while
  return edges
end function

function FOUND-INDUCING-PATH(D,E, β)
  $\mathcal{E}' = \{\}$
  for all vertices $j \in \beta$ and $j \notin \{P_0, P_{|P|}\}$ **do**
    $\mathcal{E}' = \mathcal{E}' \cup$ TRACE-DISTANCE-MATRIX(D,E,j,P_0) $\cup$ TRACE-DISTANCE-MATRIX(D,E,j,$P_{|P|}$)
    ▷ Finds all edges on any j to P_0 ($P_{|P|}$) path
  end for
  Found inducing path formed by path P and directed edges $\mathcal{E}'$
end function

A.2 F_1-score Results

In Figure 4.1, the vertical dimension differs depending on the *Erdös-Rényi model* (ER) [50] dataset used. Note that for some of the ER plots, the graphs can be generated only for higher numbers of variables. For example, there exists no ER-5 with $d = 10$, as it would need to contain $10 \times 5 = 50$ edges, while the maximum number of edges for 10 nodes is that of the complete graph K_{10}, which is $\binom{10}{2} = 45$.

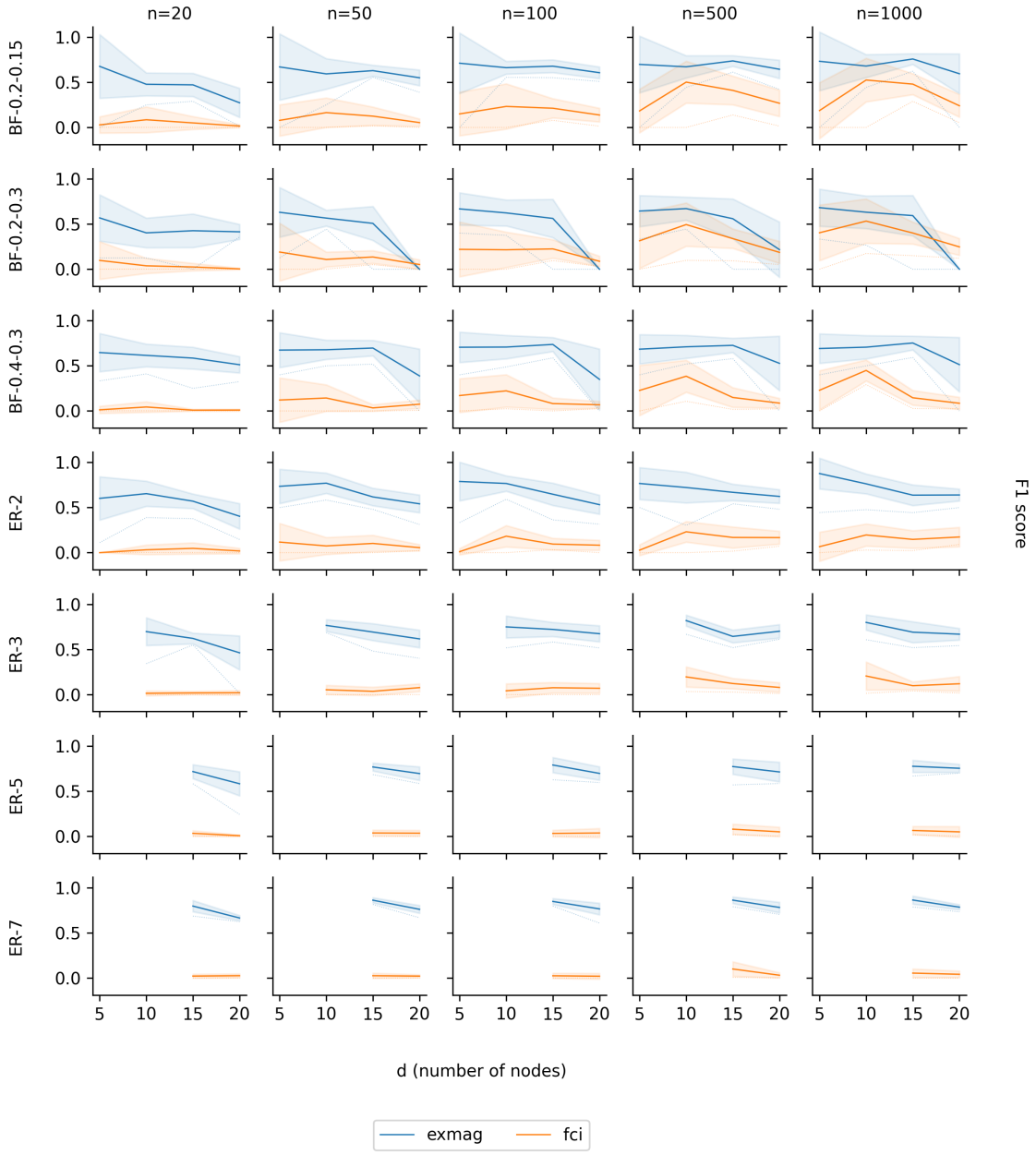


Figure 4.1: **F_1-score comparisons between ExMAG and FCI algorithm for various settings of graphs.** F_1-score (in the vertical axis) for different settings of M (in the horizontal axis) and n (horizontal choice of the graph). Standard deviations are depicted as the blured regions and dashed lines are the minimum values.

A.3 SHD Results

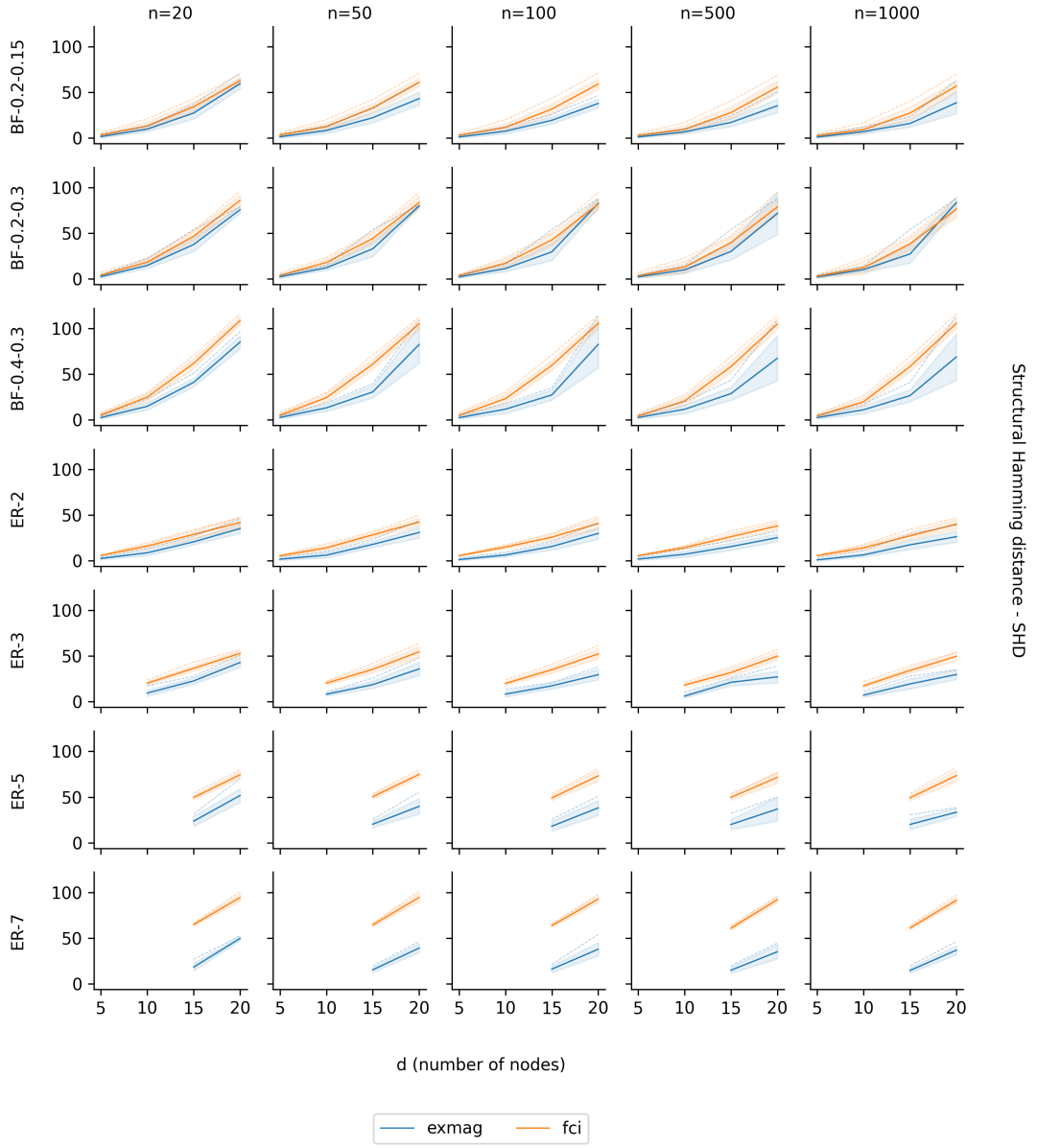


Figure 4.2: **SHD-value comparisons between ExMAG and FCI algorithm for various settings of graphs.** SHD values (in the vertical axis) for different settings of M (in the horizontal axis) and n (horizontal choice of the graph). The plots in the vertical dimension differ according to the dataset used. Standard deviations are depicted as the blured regions and dashed lines are the maximum values.

B. Assumptions and Statistical Significance from Chapter 2

There exists a fundamental premise in structural equation modelling that the residualsrepresenting unexplained variationare asymptotically unbiased, meaning they are independent of both observed and latent variables, and follow a zero-mean distribution. This assumption plays a critical role in ensuring that learned causal relationships are not distorted by hidden confounders or systematic error. The ExMAG framework embraces this principle by design, explicitly modelling residual independence as a safeguard against spurious causal edges. Just as fairness-aware systems aim to isolate structural patterns from social bias [6], ExMAG works to separate signal from statistical noise. The result is a model capable of learning causal mechanisms that are not only mathematically sound but also resilient across different subpopulations, forming a foundation of causal inference.

Causal discovery systems, like decision-making algorithms in high-stakes domains, must operate effectively across structurally diverse populations. This paper uses real-world financial data spanning multiple sectorsbanking, insurance, manufacturing, and transportationeach exhibiting distinct systemic exposures. These domains can be viewed as a *privileged setting* where data availability and quality are high, yet subgroup heterogeneity remains significant. In such contexts, ExMAG successfully identifies dominant risk propagation patterns, even when feature distributions vary across industries. This mirrors broader challenges in fairness: the need to perform robustly across populations with unequal baseline conditions [97]. The models consistent recovery of risk linksillustrated in Figure 2.5not only affirms its structural fidelity but also its capacity to generalise without group-specific tuning.

Understanding the statistical reliability of a models output requires more than average performanceit demands insight into variance. To that end, the authors conduct 10 independent trials for each configuration, reporting both mean and standard deviation for key metrics such as SHD and F_1-score. The inclusion of error bars in Figures 2.2 and 4.1 provides a visual representation of variability, revealing not just how well the model performs, but how consistently. In contrast to baseline methods with large fluctuations, ExMAG demonstrates narrow error margins, underscoring its stability in the face of stochastic elements like data partitioning and initialisation.

C. Introduction to Mixed Integer Quadratic Programming

Let us also provide a short introduction to mixed-integer quadratic programming. An optimization problem is called a mixed-integer quadratically constrained quadratic program (MIQCQP) if it is of the form

$$\min_{x\in\mathbb{R}^n} \quad x^TQx + q^Tx, \tag{4.1}$$

$$\text{s.t. } x^TQ_ix + q_i^Tx \leq a_i, \tag{4.2}$$

$$Ax \leq b, \tag{4.3}$$

$$x \in \mathbb{R}^{n-r} \times \mathbb{Z}^r \tag{4.4}$$

where $Q, Q_i \in \mathbb{R}^{n\times n}$, $q, q_i \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $a \in \mathbb{R}^k$, $b \in \mathbb{R}^m$ and $m, n, k, r \in \mathbb{N}$. (4.1) is called the cost or loss function, (4.2) represents the quadratic constraints, (4.3) are the linear constraints, and (4.4) enforces the integrality constraints for the last r components of the vector of decision variables x.

Mixed-integer quadratic programs have been shown to be NP-hard [45], which often leads to an exhaustive demand for computational resources. The algorithms used to solve MIQP are typically branch-and-bound or cutting plane [37, 17, 160, 81]. Both of these algorithmic treatments are often employed together, often with the addition of a presolving step, the use of heuristics, and parallelism. The aforementioned allows many modern solvers to solve even large problems despite the NP-hardness. Some of these solvers are open source (like SCIP and GLPK), and others are commercial (GUROBI and CPLEX). The powerful infrastructure present in these solvers can be made use of together with additional problem-specific modifications to deliver high-quality solutions.

Due to the exhaustive nature of the algorithms mentioned in the previous paragraph, global convergence is guaranteed [8]. Furthermore, convergence to the global solution may be tracked and the error estimated by computing the dual problem of (4.1–4.4). The dual of the problem is then used to compute the so-called MIP GAP as follows

$$\text{MIP GAP} = \frac{|J(x^*) - J_{\text{dual}}(y^*)|}{|J(x^*)|}, \tag{4.5}$$

where x^* and y^* are the current best solutions of the primal and dual problems, respectively, and J and J^* are the cost functions of the primal and dual problems, respectively. The MIP GAP ensures that we can assess the quality of the minimization during solution time and terminate the computation when the result is good enough (small enough MIP GAP). Furthermore, if the gap reaches 0 at any point, we are sure that the current solution is a global optimum.

D. Proofs for Chapter 3 Claims

In this section, we will show that we can apply the EM-algorithm to the joint problem of clustering trajectories produced by multiple LDS while maintaining the properties of the EM-algorithm when applied to a mixture of Gaussian distributions. The overall idea is that we formalise the assumptions under which an autonomous linear dynamic system produces normally distributed observations at each time step. As the consecutive time steps are connected linearly, we will show that the resulting distribution will be Gaussian if we concatenate all time steps together in a single feature vector.

Through the text, assume that $\mathbf{E}_m$ is an $\mathbb{R}^{m\times m}$ identity matrix.

Assumption 4.1. *The hidden state noise ω_t follows normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_H)$. The observation noise υ_t follows $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_O)$.*

Assumption 4.2. *Hidden state noise ω_t and observation noise υ_t are both independent of the state/observation values and between their samples.*

Assumption 4.3. *The hidden state ϕ_0 is normally distributed, i.e., $\phi_0 \sim \mathcal{N}(\mu_{\phi_0}, \mathbf{\Sigma}_{\phi_0})$.*

D.1 Preliminaries

In the next section, we will need to use some well-known facts about the normal distribution and related consequences. We will formally state those preliminaries in this section.

Lemma 4.1 (Linear transformation theorem of the multivariate normal distribution)**.** *Let*

$$x \sim \mathcal{N}(\mu, \Sigma).$$

Then, any linear transformation of x is also normally distributed

$$\mathbf{A}x + b \sim \mathcal{N}(\mathbf{A}\mu + b, \mathbf{A}\Sigma\mathbf{A}').$$

Lemma 4.2. *Let $x \sim \mathcal{N}(\mu_x, \Sigma_x)$ and $y \sim (\mu_y, \Sigma_y)$ be two independent, normally distributed multivariate normal distributions with n dimensions. Then,*

$$x + y \sim \mathcal{N}(\mu_x + \mu_y, \Sigma_x + \Sigma_y).$$

Proof. Since x and y are independent, then

$$\mathcal{N}\left(\begin{bmatrix}\mu_x\\ \mu_y\end{bmatrix}, \begin{bmatrix}\Sigma_x & 0\\ 0 & \Sigma_y\end{bmatrix}\right)$$

is normally distributed. Using transformation matrix

$$\mathbf{A} = \begin{bmatrix}\mathbf{E}_n & \mathbf{E}_n\end{bmatrix},$$

where $\mathbf{E}_n \in \mathbb{R}^{n\times n}$ is the identity matrix, the lemma is a direct result of Lemma 4.1. □

Lemma 4.3. *Let $x \sim \mathcal{N}(\mu_x, \Sigma_x) \in \mathbb{R}^m$ be a normal distribution, and $y \sim \mathcal{N}(0, \Sigma_y) \in \mathbb{R}^n$ be an independent Gaussian noise. Then, concatenation of x and $\mathbf{A}x + y$ (where $\mathbf{A} \in \mathbb{R}^{n\times m}$) follows the normal distribution, i.e.,*

$$\begin{pmatrix}x\\ \mathbf{A}x + y\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mu_x\\ \mathbf{A}x\end{pmatrix}, \begin{bmatrix}\Sigma_x & \Sigma_x\mathbf{A}'\\ \mathbf{A}\Sigma_x & \mathbf{A}\Sigma_x\mathbf{A}' + \Sigma_y\end{bmatrix}\right). \tag{4.6}$$

Proof. As x and y are independent normal distributions, their concatenation is the following normal distribution

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_x \\ 0 \end{pmatrix}, \begin{bmatrix} \Sigma_x & 0 \\ 0 & \Sigma_y \end{bmatrix}\right). \tag{4.7}$$

Let $\mathbf{E}_m$ ($\mathbf{E}_n$) be the identity matrix from $\mathbb{R}^{m\times m}$ ($\mathbb{R}^{n\times n}$). By Lemma 4.1,

$$\begin{bmatrix} \mathbf{E}_m & 0 \\ \mathbf{A} & \mathbf{E}_n \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{4.8}$$

is a normal distribution with mean

$$\begin{pmatrix} \mu_x \\ \mathbf{A}\mu_x \end{pmatrix} \tag{4.9}$$

and covariance matrix

$$\begin{bmatrix} \mathbf{E}_m & 0 \\ \mathbf{A} & \mathbf{E}_n \end{bmatrix} \begin{bmatrix} \Sigma_x & 0 \\ 0 & \Sigma_y \end{bmatrix} \begin{bmatrix} \mathbf{E}_m & \mathbf{A}' \\ 0 & \mathbf{E}_n \end{bmatrix} = \begin{bmatrix} \Sigma_x & \Sigma_x\mathbf{A}' \\ \mathbf{A}\Sigma_x & \mathbf{A}\Sigma_x\mathbf{A}' + \Sigma_y \end{bmatrix}, \tag{4.10}$$

which finishes the proof. □

D.2 Analysis of the EM-algorithm

First, we will show that the hidden state and observation follow the normal distribution, and we will calculate its parameters.

Lemma 4.4. *For an autonomous LDS $\mathcal{L}$ its hidden state follows the normal distribution*

$$\phi_t \sim \mathcal{N}\left(\mathbf{G}^t\mu_0, \mathbf{G}^t\mathbf{\Sigma}_{\phi_0}(\mathbf{G}')^t + \sum_{i=0}^{t-1}\mathbf{G}^i\mathbf{\Sigma}_H(\mathbf{G}')^i\right), \tag{4.11}$$

and the observations follow the normal distribution

$$x_t \sim \mathcal{N}\left(\mathbf{F}\mathbf{G}^t\mu_0, \mathbf{F}\mathbf{G}^t\mathbf{\Sigma}_{\phi_0}(\mathbf{G}')^t\mathbf{F}' + \left[\sum_{i=0}^{t-1}\mathbf{F}\mathbf{G}^i\mathbf{\Sigma}_H(\mathbf{G}')^i\mathbf{F}'\right] + \mathbf{\Sigma}_O\right). \tag{4.12}$$

Proof. For $t = 0$, our assumption was that

$$\phi_0 \sim \mathcal{N}(\mu_{\phi_0}, \mathbf{\Sigma}_{\phi_0}), \tag{4.13}$$

which proves (4.11) for $t = 0$.

The rest of the proof is done by the mathematical induction. Assume that ϕ_t follows the normal distribution stated in (4.11). Then, according to Lemma 4.1, $\mathbf{G}\phi_t$ follows normal distribution

$$\mathbf{G}\phi_t \sim \mathcal{N}\left(\mathbf{G}\mathbf{G}^t\mu_0, \mathbf{G}\left[\mathbf{G}^t\mathbf{\Sigma}_{\phi_0}(\mathbf{G}')^t + \sum_{i=0}^{t-1}\mathbf{G}^i\mathbf{\Sigma}_H(\mathbf{G}')^i\right]\mathbf{G}'\right) \tag{4.14}$$

$$= \mathcal{N}\left(\mathbf{G}^{t+1}\mu_0, \mathbf{G}^{t+1}\mathbf{\Sigma}_{\phi_0}(\mathbf{G}')^{t+1} + \sum_{i=1}^{t}\mathbf{G}^i\mathbf{\Sigma}_H(\mathbf{G}')^i\right). \tag{4.15}$$

By Lemma 4.2,

$$\phi_{t+1} = \mathbf{G}\phi_t + \omega_{t+1} \sim \mathcal{N}\left(\mathbf{G}^{t+1}\mu_0, \mathbf{G}^{t+1}\mathbf{\Sigma}_{\phi_0}(\mathbf{G}')^{t+1} + \sum_{i=0}^{t}\mathbf{G}^i\mathbf{\Sigma}_H(\mathbf{G}')^i\right), \tag{4.16}$$

which finishes the proof. The proof for observation x_t is analogous. □

As all the observations are normally distributed, we can ask whether their concatenation would be normally distributed as well. In that case, we might use algorithms for clustering a mixture of Gaussian distributions to cluster a mixture of LDS trajectories. We will answer this question in the next paragraphs.

Lemma 4.5. *Vector*

$$\begin{pmatrix}\phi_0\\ \phi_1\\ \vdots\\ \phi_T\end{pmatrix} \tag{4.17}$$

is normally distributed.

Proof. The proof will be done by mathematical induction. Vector ϕ_0 is normally distributed by Assumption 4.3.

Assume that $(\phi_0, \phi_1, \ldots, \phi_t)'$ is normally distributed up to some time t. Then, as the noise is independent of the hidden state and between its samples (see Assumption 4.2), $(\phi_0, \phi_1, \ldots, \phi_t, \omega_{t+1})'$ is normally distributed. By Lemma 4.3,

$$(\phi_0, \phi_1, \ldots, \phi_t, \phi_{t+1})' = (\phi_0, \phi_1, \ldots, \phi_t, \mathbf{G}\phi_t + \omega_{t+1})' \tag{4.18}$$

is normally distributed, where the transformation matrix $\mathbf{A}$ applied to vector $(\phi_0, \phi_1, \ldots, \phi_t, \omega_{t+1})'$ in Lemma 4.3 is equal to

$$\begin{bmatrix}\mathbf{E}_n & 0 & \cdots & 0 & 0\\ 0 & \mathbf{E}_n & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & \mathbf{G} & \mathbf{E}_n\end{bmatrix}. \tag{4.19}$$

The proof is then finished by the standard mathematical induction argument. □

An alternative way to prove Lemma 4.5 would be to use a direct proof, similar to the proof of Lemma 4.3. In that case, we can see that the transformation matrix needed to transform vector $(\phi_0, \omega_1, \ldots, \omega_T)'$ to $(\phi_0, \phi_1, \ldots, \phi_T)'$ is

$$\begin{bmatrix}\mathbf{E}_n & 0 & 0 & \cdots & 0 & \\ \mathbf{G} & \mathbf{E}_n & 0 & 0 & \cdots & 0\\ \mathbf{G}^2 & \mathbf{G} & \mathbf{E}_n & 0 & \cdots & 0\\ \mathbf{G}^3 & \mathbf{G}^2 & \mathbf{G} & \mathbf{E}_n & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots\\ \mathbf{G}^{n-1} & \mathbf{G}^{n-2} & \mathbf{G}^{n-3} & \mathbf{G}^{n-4} & \cdots & \mathbf{E}_n\end{bmatrix}. \tag{4.20}$$

The linear transformation theorem 4.1 can then be used to calculate the exact parameters of the distribution.

Corollary 4.1. *Vector of concatenated observations*

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_T \end{pmatrix} \tag{4.21}$$

is normally distributed.

Proof. By Lemma 4.5, vector $(\phi_0, \phi_1, \ldots, \phi_T)'$ follows the normal distribution. By the linear transformation theorem $\mathbf{F}(\phi_0, \phi_1, \ldots, \phi_T)'$ is normally distributed. Since the observation noise is independent of the state values and between its samples, Lemma 4.2 proves that

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_T \end{pmatrix} = \mathbf{F}' \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_T \end{pmatrix} + \begin{pmatrix} \upsilon_0 \\ \upsilon_1 \\ \vdots \\ \upsilon_T \end{pmatrix} \tag{4.22}$$

follows the normal distribution. □

Corollary 4.1 means that clustering a mixture of multiple LDSs is no more difficult than clustering a mixture of Gaussian distributions. We state this finding formally in the following theorem.

Theorem 4.1. *There exists a polynomial reduction that reduces the problem of clustering a mixture of autonomous LDSs with hidden states to the clustering of a mixture of Gaussian distributions.*

Proof. The reduction comes from the Corollary 4.1. In polynomial time, we can concatenate the vector of observations to a vector, one vector per trajectory. As the resulting concatenations are normally distributed, they can be clustered by any algorithm clustering a mixture of Gaussian distributions. □

Since there exists a reduction from clustering a mixture of autonomous LDS trajectories to clustering a mixture of Gaussian distributions, it is worth formally stating the reduction in the other way despite it being trivial to prove.

Lemma 4.6. *There exists a polynomial reduction from the problem of clustering a mixture of Gaussians to the clustering of a mixture of LDS trajectories.*

Proof. For a point in the Gaussian mixture, consider a trajectory with a length of 1, where we set $n = m$, $\mathbf{F} = \mathbf{E}_n$, and let $\upsilon = 0$ so that the observation is equal to the hidden state. For each point in the Gaussian mixture, we create a single trajectory of length 1 where the initial hidden state ϕ_0 is set to equal the point. The problem of clustering the mixture of Gaussian distributions can then be solved by finding a clustering of a mixture of LDS trajectories, showing that the problem of clustering of LDS trajectories is at least as difficult as clustering a mixture of Gaussian distributions. □

Theorem 4.2. *Finding a soft clustering of a mixture of LDS trajectories with a log-likelihood within an additive factor of the optimal log-likelihood is NP-hard when $k = 2$.*

Proof. The statement is a direct corollary of 4.6. The problem of clustering a mixture of Gaussian distributions is known to be NP-hard, even in the special case of spherical clusters. [149] The initial conditions in proof of Lemma 4.6 are defined so that the initial hidden state is propagated into the observation so that the original Gaussian distribution is clustered directly. Paper [149] assumes that the variances are non-negligible and the Gaussians are spherical, which is a special case covered by the problem of clustering of LDSs. As the problem of clustering of LDSs includes a subset of inputs that can be used to solve an NP-hard problem, soft-clustering of LDSs is NP-hard. □

In the next section, we will focus on the consequences of the property that the concatenation of the observations is normally distributed. It is also worth mentioning that the result from Corollary 4.1 does not apply to LDSs with a control input as, in that case, the distribution cannot be modeled by only a single Gaussian, but a mixture of Gaussian distributions is needed (under similar assumptions). In the case of LDS with control input, Lemma 4.4 does not hold.

D.3 Implications of the Normally Distributed Observations

As we have seen in the last section, finding the clustering of a mixture of autonomous LDSs is, in principle, the same as finding a clustering of a mixture of Gaussian distributions. As finding a clustering for a mixture of Gaussian distributions is a well-studied problem (and with more results than those that apply to the joint problem), we will summarize some of the important results in this section.

- In general, the EM-algorithm is guaranteed to converge to a local minimum, maximum, or saddle point of the likelihood function under the assumption of continuity [164].
- The EM-algorithm is connected to gradient ascent. See paper [167] for details.
- If means of the Gaussians in the mixture are provided, local convergence to a global optimum of the likelihood function is guaranteed [172]. The paper uses upper and lower bounds to prove that the EM algorithm update rule behaves as a contraction in the neighborhood of the global optimum.
- Paper [75] shows that in the case of a mixture of more than two Gaussians, the local minima of the likelihood function can be arbitrarily bad, compared to the global optimum, even if the Gaussians are well-separated. The paper also gives a lower bound on convergence to bad critical points, which emphasizes the influence of the initialisation on the final results.
- Recent paper [82] proves a linear bound on the number of samples needed for EM-algorithm to converge in the case of a mixture of three or more spherical, well-separated Gaussians.

As can be seen, when there are three or more components in the mixture, the statistical guarantees are not favorable in the case of likelihood maximization using the EM-algorithm. Besides those general properties, when a mixture of only two Gaussians is considered, better convergence guarantees have been found in special cases.

- Paper [165] shows that with random initialisation, the EM-algorithm form mixture of two Gaussians converges in $\mathcal{O}(\sqrt{n})$ with a high probability in Euclidean distance for sufficiently large n (linearly growing with dimension up to a logarithmic factor). The result holds generally, even if no separation conditions are met.

- If we consider a mixture of two balanced Gaussians with known covariance matrices, there exist global convergence guarantees - given an infinite number of samples, the EM-algorithm converges geometrically to the correct mean vectors [39].

- Paper [166, 76] proves convergence of the sequence of estimates for population EM when applied to a mixture of two Gaussians. The algorithm gives three possible optima for mean convergence and also provides parameter settings when the means are identified correctly or the algorithm converges to the point when the estimates are both the average of the true mean values. The results are then extended to the sample-based EM, and the probability of convergence is proven.

To contrast the previous paragraphs, even when there are two clusters with spherical Gaussians and shared variance, the soft clustering problem is NP-hard [149]. The NP-hardness is proved for approximation of the log-likelihood within an additive factor. The same paper [149] also shows that the NP-hardness remains for non-negligible variances. The complexity is shown by a reduction to the k-means problem.

Recent analyses focus on many special cases of the clustering of mixture of Gaussian distributions.

- Paper [48] focuses on weakly identifiable models. They analyze the case of mixture of two equal-sized spherical Gaussian distributions that share the covariance matrices. The locations of the Gaussian distributions are then assumed to be symmetric with respect to axes origin. The paper than discusses the univariate case and shows that the statistical estimation error of the EM estimates is of the order of $n^{-\frac{1}{8}}$ and after $n^{\frac{3}{4}}$ steps, the error is in the order of $n^{-\frac{1}{4}}$. In the multivariate case, shared covariances improve the convergence criteria compared to the general case.

- Paper [158] studies a similar case - two symmetrically located spherical Gaussians, however, the mixture in this case is assumed imbalanced with known mixture coefficient. The authors then prove that the population-based EM-algorithm is globally convergent if the initial estimate has non-negative inner product with the mean of the larger component. When initialised to center the axis, error rate is given after a number of iterations inversely proportional to the mixing ratio and the norm of the cluster centers. Bounds for the empirical iteration are given as well.

- Further analyses of the weakly separated case are provided in [65]. The paper shows that the convergence rate is of the order of $n^{-\frac{1}{6}}$ or $n^{-\frac{1}{8}}$. The paper shows that sometimes the EM-algorithm shows high likelihood of the cluster means being equal despite this being false.

- In [171], the authors develop a generalisation of the standard EM-algorithm that can work in distributed setting. The method is consistent and retains the $\mathcal{O}(\sqrt{n})$ consistency under specified conditions. The authors then compare the method with some of the existing approaches, showing its superiority.

- Lastly, paper [66] provides convergence rates for Gaussian mixtures of experts, which is a class of regression models. The authors state the notion of algebraic independence allowing them to establish a connection to partial differential equations, which in turn are used to prove the convergence rate.

D.4 Practical Applicability of the Gaussian Mixture-Based EM-algorithm

Using EM-algorithm directly on concatenated vectors requires fitting $\mathcal{O}(T(m+n)k)$ parameters in the case of mean values, and unfortunately, $\mathcal{O}(T^2(n^2+m^2)k)$ parameters in the covariance matrix. By exploiting the transformation matrix in (4.20), the number of parameters of the covariance matrices can be simplified by removing some degree of freedom from the problem, keeping only free parameters in $\mathbf{\Sigma}_H$, $\mathbf{\Sigma}_O$, $\mathbf{G}$, and $\mathbf{F}$. Thus, we need only $\mathcal{O}((n^2+m^2)k)$ parameters. Adding those constraints can, however, cause loss of the theoretical properties of the EM-algorithm.

A direct approach to solving the joint problem is to use the MLE estimates. In the case of spherical clusters with equal variance and under the negligence of the cost for the initial hidden state, the joint problem reduces to the minimization of

$$\min_{\omega_t,\upsilon_t,\phi_0,\mathbf{G},\mathbf{F},l_i} \sum_{i=1}^{N} \left(\sum_{t=2}^{T} \|\omega_t^i\|_2^2 + \sum_{t=1}^{T} \|\upsilon_t^i\|_2^2 \right), \tag{4.23}$$

We can see in (4.23) that the MLE estimate requires to have a single parameter for each time step and each trajectory, which is the noise value assigned to the trajectory at a particular time. This means $\mathcal{O}(T(m+n)Nk)$ parameters, again too much for practical usability.

For completeness, the formula above leads to the following EM-heuristic formulation.

$$l_i \leftarrow \underset{c\in\{0,1,\dots,K-1\}}{\arg\min} \min_{\omega_t^i,\upsilon_t^i,\phi_0} \left(\sum_{t=2}^{T} \|\omega_t^i\|_2^2 + \sum_{t=1}^{T} \|\upsilon_t^i\|_2^2, \right) \tag{4.24}$$

where each of the minimization problems is subject to

$$\phi_t^i = (\mathbf{G}^c)\phi_{t-1}^i + \omega_t^i, \qquad \forall t \in \{2,3,\dots,T\}, \tag{4.25}$$

$$x_t^i = (\mathbf{F}^c)'\phi_t^i + \upsilon_t^i, \qquad \forall t \in \{1,2,\dots,T\},. \tag{4.26}$$

The algorithm is guaranteed to converge to a local optimum or a saddle point.

D.5 Connection to k-means

In our effort to improve the practical applicability of the algorithm, we can take inspiration from the mixture of the Gaussians approach. For the classical EM-algorithm, there exists a faster heuristic - Lloyd's algorithm [91] for the k-means problem. In this section, we will show the connection of the minimization problem from the main paper body to the k-means problem and the connection of the heuristic to Lloyd's algorithm.

Recall the objective function,

$$\min_{\substack{\hat{Y}_t^0,\mathbf{G}_0,\boldsymbol{\varphi}_0,\upsilon_t^0,\omega_t^0,X_0^0,\\ \hat{Y}_t^1,\mathbf{G}_1,\boldsymbol{\varphi}_1,\upsilon_t^1,\omega_t^1,X_0^1\\ l_t}} \sum_{t=1}^{N}\sum_{t=1}^{T} \|Y_t^t - \hat{Y}_t^{l_i}\|_2^2 + \sum_{c\in\{0,1\}} \sum_{t=1}^{T} \left[\|\upsilon_t^c\|_2^2 + \|\omega_t^c\|_2^2 \right]. \tag{4.27}$$

The first term of the objective function calculates the difference between the cluster means to the observations; the second term then minimizes noise that is induced by the optimal trajectory defined by the cluster means. With $N \to \infty$ the cost function goes to

$$\min_{\substack{\hat{Y}_t^0, \mathbf{G}_0, \boldsymbol{\varphi}_0, \upsilon_t^0, \omega_t^0, X_0^0, \\ \hat{Y}_t^1, \mathbf{G}_1, \boldsymbol{\varphi}_1, \upsilon_t^1, \omega_t^1, X_0^1 \\ l_t}} \sum_{t=1}^{N} \sum_{t=1}^{T} \|Y_t^t - \hat{Y}_t\|_2^2 \tag{4.28}$$

as the other terms do not increase with the number of trajectories. The formula in (4.28) is the standard k-means criterion. Applying the same reasoning to the EM-heuristic in the main paper body, leads to the standard Lloyd's algorithm, as with $N \to \infty$, minimization

$$\min_{\hat{Y}_t^c, \mathbf{G}_c, \boldsymbol{\varphi}_c, \upsilon_t^c, \omega_t^c, X_c^0} \left[\sum_{t=1}^{N} \sum_{t=1}^{T} \mathbb{1}[l_i = c] \cdot \|Y_t^t - \hat{Y}_t^c\|_2^2 \right] + \|\upsilon^c\|_2^2 + \|\omega^c\|_2^2 \tag{4.29}$$

converges to the following minimization problem

$$\min_{\hat{Y}_t^c, \mathbf{G}_c, \boldsymbol{\varphi}_c, \upsilon_t^c, \omega_t^c, X_c^0} \left[\sum_{t=1}^{N} \sum_{t=1}^{T} \mathbb{1}[l = c] \cdot \|Y_t^t - \hat{Y}_t^c\|_2^2 \right], \tag{4.30}$$

which is minimized by f_t^c being the cluster means. To wrap this up, with the increasing number of trajectories $N \to \infty$, the EM-heuristic converges to Lloyd's algorithm [91] for the k-means problem.

D.6 Clustering Performance Metrics

The F_1 score, which has been widely used in classification performance measurements, is defined as

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.31}$$

where precision $= \frac{TP}{TP+FP}$ and recall $= \frac{TP}{TP+FN}$ and TP, FP, and FN are the numbers of true positives, false positives, and false negatives, respectively. We calculate the F_1 score twice for each class, once with one class labelled as positive and once with the other class labelled as positive, and then we select the higher score for each class. This approach is used because there is no predefined positive and negative labels.

E. Introduction to Mixed-Integer Programming (MIP)

To search for global optima, we developed relaxations to bound the optimal objective values in non-convex Mixed-Integer Nonlinear Programs (MINLPs) [9]. Our study is based on the Mixed-Integer Nonlinear Programs of the form:

$$\begin{aligned} \min\ & f(x,y,z) && && \text{(MINLP)}\\ \text{s.t. } & g_i(x,y,z) \le 0, && \forall i \in I, && (4.32)\\ & h_j(x) \le 0, && \text{if } z_j = 1, \forall j \in J, && (4.33)\\ & && x \in \mathbb{R}^n, y \in \mathbb{Z}^m \end{aligned}$$

where functions f, g_i and h_j are assumed to be continuous and twice differentiable. Such problems are non-convex, both in terms of featuring integer variables and in terms of the functions f, g_i.

While MINLP problems may seem too general a model for our joint problem, notice that the NP-hardness and inapproximability results discussed in Section 3.2.1 suggest that this may be the appropriate framework. For bounded variables, standard branch-and-bound-and-cut algorithms run in finite time. Both in theory – albeit under restrictive assumptions, such as in [47] – and in practice, the expected runtime is often polynomial. In the formulation of the next section, f, g_i are trilinear, and various monomial envelopes have been considered and implemented in global optimization solvers such as BARON [139], SCIP [10], and Gurobi. In our approach, we consider a mixed-integer programming formulation of a piecewise polyhedral relaxation of a multilinear term using its convex-hull representation.

F. Non-Commutative Polynomial Optimization (NCPOP)

To extend the search for global optima from a fixed finite-dimensional state to an operator in an unknown dimension, we formulate the problem as a non-commutative polynomial optimization problem (NCPOP), cf. [119, 22]. In contrast to traditional scalar-valued, vector-valued, or matrix-valued optimization techniques, the variables considered in NCPOP are operators, whose dimensions are unknown *a priori*.

Let $X = (X_1, \ldots, X_n)$ be a tuple of bounded operators on a Hilbert space $\mathcal{H}$. Let $[X, X^\dagger]$ denote these $2n$ operators, with the $\dagger$-algebra being conjugate transpose. Let monomials ω, μ be products of powers of variables from $[X, X^\dagger]$. The degree of a monomial, denoted by $|\omega|$, refers to the sum of the exponents of all operators in the monomial ω, e.g., $|X_n^3 X_n^\dagger| = 4$. Let p and q_i, $i = 1, \ldots, m$ be polynomials in these $2n$ variables. Let $\deg(p)$ denote the polynomial degree of p. In the following, we will view these $2n$ variables as the new tuple X. Using the set of monomials generated from the tuple X, polynomials p and q_i, $i = 1, \ldots, m$ can be rewritten as linear combinations of monomials:

$$p(X) = \sum_{|\omega| \leq \deg(p)} p_\omega \omega, \quad q_i(X) = \sum_{|\mu| \leq \deg(q_i)} q_{i,\mu} \mu,$$

for $i = 1, \ldots, m$, and $p_\omega, q_{i,\mu}$, are coefficients of these polynomials. For instance, $p(X) = X_1^3 X_n^\dagger + 5X_n = \omega_1 + 5\omega_2$, where $\omega_1 = X_1^3 X_n^\dagger$ and $\omega_2 = X_n$.

Let $\langle \cdot, \cdot \rangle$ denotes inner product. Suppose there is a normalised vector ψ, i.e., $\langle \psi, \psi \rangle = 1$, also defined on the Hilbert space $\mathcal{H}$. Let $p(X), q_i(X)$ be the Hermitian operators, i.e., $p^\dagger(X) = p(X)$. The formulation considered in NCPOP reads

$$\begin{aligned} \underset{(\mathcal{H}, X, \psi)}{\text{minimize}} \quad & \langle \psi, p(X)\psi \rangle \\ \text{subject to} \quad & q_i(X) \succcurlyeq 0, \, i = 1, \ldots, m, \\ & \langle \psi, \psi \rangle = 1, \end{aligned} \tag{4.34}$$

where the constraint $q_i(X) \succcurlyeq 0$ denotes that the variable $q_i(X)$ is positive semidefinite.

Under the Archimedean assumption, such that the tuple of operators X are bounded, one can utilise the Sums of Squares theorem of [64] and [95] to derive semidefinite programming (SDP) relaxations of the Navascules-Pironio-Acin (NPA) hierarchy [105, 119]. There are also variants [154, 155, 153] that exploit various forms of sparsity.

Bibliography

[1] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering - a decade review. *Information Systems*, 53:16–38, 2015. ISSN 0306-4379. doi: https://doi.org/10.1016/j.is.2015.04.007. URL https://www.sciencedirect.com/science/article/pii/S0306437915000733.

[2] Amir Ali Ahmadi and Bachir El Khadir. Learning dynamical systems with side information. In *Learning for Dynamics and Control*, pages 718–727. PMLR, 2020.

[3] Donald S Baim, Wilson S Colucci, E Scott Monrad, Harton S Smith, Richard F Wright, Alyce Lanoue, Diane F Gauthier, Bernard J Ransil, William Grossman, and Eugene Braunwald. Survival of patients with severe congestive heart failure treated with oral milrinone. *Journal of the American College of Cardiology*, 7(3): 661–670, 1986.

[4] Ainesh Bakshi, Allen Liu, Ankur Moitra, and Morris Yau. Tensor decompositions meet control theory: Learning general mixtures of linear dynamical systems. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 1549–1563. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/bakshi23a.html.

[5] Laura Ballester, Jesúa López, and Jose M. Pavía. European systemic credit risk transmission using Bayesian networks. *Research in International Business and Finance*, 65:101914, 2023. ISSN 0275-5319. doi: https://doi.org/10.1016/j.ribaf.2023.101914. URL https://www.sciencedirect.com/science/article/pii/S0275531923000405.

[6] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. Available at https://fairmlbook.org/.

[7] Paul Beaumont, Ben Horsburgh, Philip Pilgerstorfer, Angel Droth, Richard Oentaryo, Steven Ler, Hiep Nguyen, Gabriel Azevedo Ferreira, Zain Patel, and Wesley Leong. CausalNex, October 2021. URL https://github.com/quantumblacklabs/causalnex.

[8] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1131, 2013. doi: 10.1017/S0962492913000032.

[9] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22: 1–131, 2013.

[10] Ksenia Bestuzheva, Antonia Chmiela, Benjamin Müller, Felipe Serrano, Stefan Vigerske, and Fabian Wegscheider. Global optimization of mixed-integer nonlinear programs with scip 8. *arXiv preprint arXiv:2301.00587*, 2023.

[11] Petar Bevanda, Stefan Sosnowski, and Sandra Hirche. Koopman operator dynamical models: Learning, analysis and control. *Annual Reviews in Control*, 52:197–212, 2021. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2021.09.002.

[12] Rohit Bhattacharya, Tushar Nagarajan, Daniel Malinsky, and Ilya Shpitser. Differentiable causal discovery under unmeasured confounding. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130, pages 2314–2322. PMLR, 2021.

[13] Peter J Bickel, Eugene A Hammel, and J William O'Connell. Sex bias in graduate admissions: Data from berkeley. *Statistics and public policy*, pages 113–130, 1977.

[14] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Annual International Cryptology Conference*, pages 278–291. Springer, 1993.

[15] K.A. Bollen. *Structural Equations With Latent Variables*, volume 210. John Wiley & Sons, 1989.

[16] Pierre Bonami, Lorenz T Biegler, Andrew R Conn, Gérard Cornuéjols, Ignacio E Grossmann, Carl D Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas Sawaya, et al. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete optimization*, 5(2):186–204, 2008.

[17] Pierre Bonami, Mustafa Kilinç, and Jeff Linderoth. Algorithms and software for convex mixed integer nonlinear programs. In *Mixed integer nonlinear programming*, pages 1–39. Springer, 2011.

[18] Peter Bühlmann. Invariance, causality and robustness. *Statistical Science*, 35(3): 404–426, 2020.

[19] Peter Bühlmann and Domagoj Ćevid. Deconfounding and causal regularisation for stability and external validity. *International Statistical Review*, 88:S114–S134, 2020.

[20] Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, Berlin, Heidelberg, 2011. ISBN 978-3-642-20191-2. doi: 10.1007/978-3-642-20192-9.

[21] Kirsten Bulteel, Francis Tuerlinckx, Annette Brose, and Eva Ceulemans. Clustering vector autoregressive models: Capturing qualitative differences in within-person dynamics. *Frontiers in Psychology*, 7, 2016. ISSN 1664-1078. doi: 10.3389/fpsyg.2016.01540. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2016.01540.

[22] Sabine Burgdorf, Igor Klep, and Janez Povh. *Optimization of polynomials in non-commuting variables*. Springer, 2016.

[23] Michael L. Bynum, Gabriel A. Hackebeil, William E. Hart, Carl D. Laird, Bethany L. Nicholson, John D. Siirola, Jean-Paul Watson, and David L. Woodruff. *Pyomo–optimization modeling in python*, volume 67. Springer Science & Business Media, third edition, 2021.

[24] Mehmet Caner and Bruce E Hansen. Instrumental variable estimation of a threshold model. *Econometric theory*, 20(5):813–843, 2004.

[25] Domagoj Cevid, Peter Bhlmann, and Nicolai Meinshausen. Spectral deconfounding for causal inference. *Annals of Statistics*, 46(6B):3313–3340, 2018.

[26] Bertrand Charpentier, Simon Kibler, and Stephan Günnemann. Differentiable DAG sampling. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9wOQOgNe-w.

[27] Chen Chen, Cewu Lu, Qixing Huang, Qiang Yang, Dimitrios Gunopulos, and Leonidas Guibas. City-scale map creation and updating using gps collections. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1465–1474, 2016.

[28] Rui Chen, Sanjeeb Dash, and Tian Gao. Integer programming for causal structure learning in the presence of latent variables. In *International Conference on Machine Learning*, pages 1550–1560. PMLR, 2021.

[29] Yanxi Chen and H. Vincent Poor. Learning mixtures of linear dynamical systems. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 3507–3557. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/chen22t.html.

[30] Yonghong Chen, Govindan Rangarajan, Jianfeng Feng, and Mingzhou Ding. Analyzing multiple nonlinear time series with extended granger causality. *Physics Letters A*, 324(1):26–35, 2004. ISSN 0375-9601. doi: https://doi.org/10.1016/j.physleta.2004.02.032. URL https://www.sciencedirect.com/science/article/pii/S0375960104002403.

[31] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. *Double/debiased machine learning for treatment and structural parameters*. Oxford University Press Oxford, UK, 2018.

[32] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[33] Y Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA. URL: http://starai. cs. ucla. edu/papers/ProbCirc20. pdf*, page 6, 2020.

[34] Tom Claassen and Ioana G. Bucur. Greedy equivalence search in the presence of latent confounders. In *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2022. URL https://arxiv.org/abs/2205.07280.

[35] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Extended formulations in combinatorial optimization. *4OR*, 8(1):1–48, 2010.

[36] William J Cook, David L Applegate, Robert E Bixby, and Vasek Chvátal. *The traveling salesman problem: a computational study*. Princeton university press, 2011.

[37] R. J. Dakin. A tree-search algorithm for mixed integer programming problems. *Comput. J.*, 8:250–255, 1965. URL https://api.semanticscholar.org/CorpusID:62138114.

[38] Sanjeeb Dash, Joao Goncalves, and Tian Gao. Integer programming based methods and heuristics for causal graph learning. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. URL https://openreview.net/forum?id=0FvarE8SMs.

[39] Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. Ten steps of em suffice for mixtures of two gaussians. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 704–710. PMLR, 07–10 Jul 2017. URL https://proceedings.mlr.press/v65/daskalakis17b.html.

[40] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. The ucr time series classification archive, October 2018. URL https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[41] Richard A Davis, Pengfei Zang, and Tian Zheng. Sparse vector autoregressive modeling. *Journal of Computational and Graphical Statistics*, 25(4):1077–1096, 2016.

[42] A. P. Dawid and V. Didelez. Causal inference in graphical models. *Journal of Causal Inference*, 2(1):22–38, 2010. doi: 10.1214/10-JCI260.

[43] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(2):142–150, 1989. doi: https://doi.org/10.1111/j.1467-8640.1989.tb00324.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8640.1989.tb00324.x.

[44] Thomas L. Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5, 1989. URL https://api.semanticscholar.org/CorpusID:57798167.

[45] Alberto Del Pia, Santanu Dey, and Marco Molinaro. Mixed-integer quadratic programming is in np. *Mathematical Programming*, 162, 07 2014. doi: 10.1007/s10107-016-1036-0.

[46] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. URL http://www.jstor.org/stable/2984875.

[47] Santanu S Dey, Yatharth Dubey, and Marco Molinaro. Branch-and-bound solves random binary ips in polytime. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 579–591. SIAM, 2021.

[48] Raaz Dwivedi, Nhat Ho, Koulik Khamaru, Martin Wainwright, Michael Jordan, and Bin Yu. Sharp analysis of expectation-maximization for weakly identifiable models. In *International Conference on Artificial Intelligence and Statistics*, pages 1866–1876. PMLR, 2020.

[49] Peter Ebbes, Michel Wedel, Ulf Böckenholt, and Ton Steerneman. Solving and testing for regressor-error (in) dependence when no instrumental variables are available: With new evidence for the effect of education on income. *Quantitative Marketing and Economics*, 3:365–392, 2005.

[50] Paul Erdős and Alfred Rényi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

[51] Anja Ernst, Marieke Timmerman, Feng Ji, Bertus Jeronimus, and Casper Albers. Mixture multilevel vector-autoregressive modeling. *Psychological Methods*, 29, 08 2023. doi: 10.1037/met0000551.

[52] Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, June 1962. ISSN 0001-0782. doi: 10.1145/367766.368168. URL https://doi.org/10.1145/367766.368168.

[53] Yoav Gilad and Orna Mizrahi-Man. A reanalysis of mouse encode comparative gene expression data. *F1000Research*, 4, 05 2015. doi: 10.12688/f1000research.6536.1.

[54] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[55] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1912791.

[56] Alex Greenfield, Aviv Madar, Harry Ostrer, and Richard Bonneau. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PLoS ONE*, 5, 2010. URL https://api.semanticscholar.org/CorpusID:12755511.

[57] Dimitrios Gunopulos and Gautam Das. Time series similarity measures and time series indexing. In *SIGMOD Conference*, page 624, 2001.

[58] Zijian Guo, Domagoj Ćevid, and Peter Bühlmann. Doubly debiased lasso: High-dimensional inference under hidden confounding. *Annals of statistics*, 50(3):1320, 2022.

[59] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL https://www.gurobi.com.

[60] F. R. Hampel et al. Robust statistics. *Wiley-Interscience*, 18, 1986. doi: 10.1002/jsc.127.

[61] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 450–477, 2012.

[62] Xiaoyu He. Causal learning benchmark [computer software], 2023.

[63] J Helton and Scott McCullough. A positivstellensatz for non-commutative polynomials. *Transactions of the American Mathematical Society*, 356(9):3721–3737, 2004.

[64] J William Helton. "Positive" noncommutative polynomials are sums of squares. *Annals of Mathematics*, 156(2):675–694, 2002.

[65] Nhat Ho, Avi Feller, Evan Greif, Luke Miratrix, and Natesh Pillai. Weak separation in mixture models and implications for principal stratification. In *International Conference on Artificial Intelligence and Statistics*, pages 5416–5458. PMLR, 2022.

[66] Nhat Ho, Chiao-Yu Yang, and Michael I Jordan. Convergence rates for Gaussian mixtures of experts. *The Journal of Machine Learning Research*, 23(1):14523–14603, 2022.

[67] Reimar Hofmann and Volker Tresp. Discovering structure in continuous variables using Bayesian networks. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'95, page 500506, Cambridge, MA, USA, 1995. MIT Press.

[68] Chloe Hsu, Michaela Hardt, and Moritz Hardt. Linear dynamics: Clustering without identification. In *International Conference on Artificial Intelligence and Statistics*, pages 918–929. PMLR, 2020.

[69] Z. Hu. *Causal Discovery with Ancestral Graphs*. DPhil dissertation, University of Oxford, 2023.

[70] Zhenyu Hu and Robin J. Evans. Faster algorithms for markov equivalence. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020. URL https://arxiv.org/abs/2002.03230.

[71] Zhongyi Hu and Robin Evans. A fast score-based search algorithm for maximal ancestral graphs using entropy. *arXiv preprint arXiv:2402.04777*, 2024.

[72] Zhongyi Hu and Robin J Evans. Towards standard imsets for maximal ancestral graphs. *Bernoulli*, 30(3):2026–2051, 2024.

[73] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177707015.

[74] Guido W. Imbens. Instrumental Variables: An Econometricians Perspective. *Statistical Science*, 29(3):323 – 358, 2014. doi: 10.1214/14-STS480. URL https://doi.org/10.1214/14-STS480.

[75] Chi Jin, Yuchen Zhang, Sivaraman Balakrishnan, Martin J Wainwright, and Michael I Jordan. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/3875115bacc48cca24ac51ee4b0e7975-Paper.pdf.

[76] Chi Jin, Yuchen Zhang, Sivaraman Balakrishnan, Martin J Wainwright, and Michael I Jordan. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/3875115bacc48cca24ac51ee4b0e7975-Paper.pdf.

[77] Maciej Kamiński, Mingzhou Ding, Wilson A. Truccolo, and Steven L. Bressler. Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance. *Biological Cybernetics*, 85(2): 145–157, Aug 2001. ISSN 1432-0770. doi: 10.1007/s004220000235. URL https://doi.org/10.1007/s004220000235.

[78] Armin Kekić, Bernhard Schölkopf, and Michel Besserve. Targeted reduction of causal models. *arXiv preprint arXiv:2311.18639*, 2023.

[79] Jee-Seon Kim and Edward W. Frees. Multilevel modeling with correlated effects. *Psychometrika*, 72(4):505–533, Dec 2007. ISSN 1860-0980. doi: 10.1007/s11336-007-9008-1. URL https://doi.org/10.1007/s11336-007-9008-1.

[80] Igor Klep, Victor Magron, and Janez Povh. Sparse noncommutative polynomial optimization. *Mathematical Programming*, pages 1–41, 2021.

[81] Jan Kronqvist, Andreas Lundell, and Tapio Westerlund. The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization*, 64, 06 2015. doi: 10.1007/s10898-015-0322-3.

[82] Jeongyeol Kwon and Constantine Caramanis. The em algorithm gives sample-optimality for learning mixtures of well-separated gaussians. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 2425–2487. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/kwon20a.html.

[83] Sébastien Lachapelle, Céline Brouard, Simon Lacoste-Julien, and Alexandre Drouin. Gradient-based neural DAG learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[84] Clifford Lam and Qiwei Yao. Factor modeling for high-dimensional time series: inference for the number of factors. *The Annals of Statistics*, pages 694–726, 2012.

[85] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[86] Jean Bernard Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009.

[87] Lei Li and B Aditya Prakash. Time series clustering: Complex is simpler! In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 185–192, 2011.

[88] Shin Lin, Yiing Lin, Joseph R. Nery, Mark A. Urich, Alessandra Breschi, Carrie A. Davis, Alexander Dobin, Christopher Zaleski, Michael A. Beer, William C. Chapman, Thomas R. Gingeras, Joseph R. Ecker, and Michael P. Snyder. Comparison of the transcriptional landscapes between human and mouse tissues. *Proceedings of the National Academy of Sciences*, 111(48):17224–17229, 2014. doi: 10.1073/pnas.1413624111. URL https://www.pnas.org/doi/abs/10.1073/pnas.1413624111.

[89] Zitao Liu and Milos Hauskrecht. A regularized linear dynamical system framework for multivariate time series analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29. AAAI Press, 2015.

[90] Lennart Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.

[91] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi: 10.1109/TIT.1982.1056489.

[92] Easwar Magesan, Jay M Gambetta, Antonio D Córcoles, and Jerry M Chow. Machine learning for discriminating quantum measurement trajectories and improving readout. *Physical review letters*, 114(20):200501, 2015.

[93] Gaurav Mahajan, Sham Kakade, Akshay Krishnamurthy, and Cyril Zhang. Learning hidden markov models using conditional samples. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2014–2066. PMLR, 2023.

[94] Maya B. Mathur and Tyler J. VanderWeele. Methods to address confounding and other biases in meta-analyses: Review and recommendations. *Annual Review of Public Health*, 43(Volume 43, 2022):19–35, 2022. ISSN 1545-2093. doi: https://doi.org/10.1146/annurev-publhealth-051920-114020.

[95] Scott McCullough. Factorization of operator-valued polynomials in several noncommuting variables. *Linear Algebra and its Applications*, 326(1-3):193–203, 2001.

[96] Richard McElreath. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC, 2018.

[97] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[98] Mazen Melibari, Pascal Poupart, Prashant Doshi, and George Trimponias. Dynamic sum product networks for tractable inference on sequence data. In *Conference on Probabilistic Graphical Models*, pages 345–355. PMLR, 2016.

[99] Mohamed Mokbel, Mahmoud Sakr, Li Xiong, Andreas Züfle, Jussara Almeida, Walid Aref, Gennady Andrienko, Natalia Andrienko, Yang Cao, Sanjay Chawla, et al. Towards mobility data science (vision paper). *arXiv preprint arXiv:2307.05717*, 2023.

[100] MOSEK, ApS. The MOSEK Optimizer API for Python 9.3, 2023. URL https://docs.mosek.com/9.3/pythonapi/index.html. Accessed: 2025-08-22.

[101] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 607617, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369367. doi: 10.1145/3351095.3372850. URL https://doi.org/10.1145/3351095.3372850.

[102] Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[103] Kevin Patrick Murphy. *Dynamic Bayesian networks: representation, inference and learning*. University of California, Berkeley, 2002.

[104] August Nagro. Chemistry-engine. https://github.com/AugustNagro/Chemistry-Engine, 2015.

[105] Miguel Navascués, Stefano Pironio, and Antonio Acín. Sdp relaxations for noncommutative polynomial optimization. *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 601–634, 2012.

[106] Leland Gerson Neuberg. Causality: Models, reasoning, and inference, by judea pearl, cambridge university press, 2000. *Econometric Theory*, 19(4):675685, 2003. doi: 10.1017/S0266466603004109.

[107] Whitney K. Newey. Efficient instrumental variables estimation of nonlinear models. *Econometrica*, 58(4):809–837, 1990. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/2938351.

[108] Kasra Nezamabadi, Neda Sardaripour, Benyamin Haghi, and Mohamad Forouzanfar. Unsupervised ecg analysis: A review. *IEEE Reviews in Biomedical Engineering*, 16:208–224, 2023. doi: 10.1109/RBME.2022.3154893.

[109] Mengjia Niu, Xiaoyu He, Petr Ryšavý, Quan Zhou, and Jakub Mareček. Joint problems in learning multiple dynamical systems. In *Proceedings of the 61st Allerton Conference on Communication, Control, and Computing*, 2025.

[110] Weronika Ormaniec, Scott Sussex, Lars Lorch, Bernhard Schölkopf, and Andreas Krause. Standardizing structural causal models, 2024. URL https://arxiv.org/abs/2406.11601.

[111] Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pages 1595–1605. Pmlr, 2020.

[112] Judea Pearl. Bayesian netwcrks: A model cf self-activated memory for evidential reasoning. In *Proceedings of the 7th conference of the Cognitive Science Society, University of California, Irvine, CA, USA*, pages 15–17, 1985.

[113] Judea Pearl. Chapter 2 - Bayesian inference. In Judea Pearl, editor, *Probabilistic Reasoning in Intelligent Systems*, pages 29–75. Morgan Kaufmann, San Francisco (CA), 1988. ISBN 978-0-08-051489-5. doi: https://doi.org/10.1016/B978-0-08-051489-5.50008-4. URL https://www.sciencedirect.com/science/article/pii/B9780080514895500084.

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

[115] J. Peters et al. *Elements of Causal Inference*. MIT Press, 2016. doi: 10.7551/mitpress/9780262034442.001.0001.

[116] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Structural intervention distance (sid) for evaluating causal graphs. *Neural Computation*, 27(3):771–799, 2015.

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

[118] S. Pironio, M. Navascués, and A. Acín. Convergent relaxations of polynomial optimization problems with noncommuting variables. *SIAM Journal on Optimization*, 20(5):2157–2180, 2010. doi: 10.1137/090760155. URL https://doi.org/10.1137/090760155.

[119] Stefano Pironio, Miguel Navascués, and Antonio Acin. Convergent relaxations of polynomial optimization problems with noncommuting variables. *SIAM Journal on Optimization*, 20(5):2157–2180, 2010.

[120] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690. IEEE, 2011.

[121] Kari Rantanen, Antti Hyttinen, and Matti Järvisalo. Maximal ancestral graph structure learning via exact search. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1237–1247. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/rantanen21a.html.

[122] Olav Reiersøl. *Confluence analysis by means of instrumental sets of variables*. PhD thesis, Almqvist & Wiksell, 1945.

[123] Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27772–27784. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/e987eff4a7c7b7e580d659feb6f60c1a-Paper.pdf.

[124] Alexander G. Reisach, Myriam Tami, Christof Seiler, Antoine Chambaz, and Sebastian Weichwald. A scale-invariant sorting criterion to find a causal order in additive noise models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

[125] Christian Reiser. Causal discovery for time series with latent confounders. In *Proceedings of the Conference on Causal Learning and Reasoning (CLeaR)*. University of Stuttgart, 2020. URL https://arxiv.org/abs/2011.12942. Presented at a workshop or preprint.

[126] Thomas S. Richardson. A factorization criterion for acyclic directed mixed graphs. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 462–469, 2009. URL https://arxiv.org/abs/1406.6764.

[127] Thomas S. Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002. doi: 10.1214/aos/1031689016.

[128] M. Rojas-Carulla et al. Causal inference and distributional robustness. *Statistical Science*, 33(3):432–445, 2018. doi: 10.1214/18-STS635.

[129] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[130] Dominik Rothenhäusler, Nicolai Meinshausen, Peter Bühlmann, and Jonas Peters. Anchor regression: heterogeneous data meets causality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):1101–1122, 2018.

[131] Sam Roweis and Zoubin Ghahramani. A unifying review of linear Gaussian models. *Neural computation*, 11(2):305–345, 1999.

[132] Pavel Rytíř, Aleš Wodecki, Georgios Korpas, and Jakub Mareček. ExDBN: Exact learning of dynamic Bayesian networks, 2024. URL https://arxiv.org/abs/2410.16100.

[133] Pavel Rytíř, Aleš Wodecki, and Jakub Mareček. ExDAG: Exact learning of DAGs. *arXiv preprint arXiv:2406.15229*, 2024.

[134] Petr Ryšavý. Krebs benchmark dataset [data set], 2023.

[135] Petr Ryšavý. krebsdynotears [computer software], 2023.

[136] Petr Ryšavý. Krebs cycles generator, 2023.

[137] Petr Ryšavý, Xiaoyu He, and Jakub Mareček. Causal learning in biomedical applications: A benchmark. *arXiv preprint arXiv:2406.15189*, 2024. URL https://arxiv.org/abs/2406.15189.

[138] Petr Ryšavý, Pavel Rytíř, Xiaoyu He, Georgios Korpas, and Jakub Mareček. ExMAG: Learning of maximally ancestral graphs, 2025. URL https://arxiv.org/abs/2503.08245.

[139] Nikolaos V Sahinidis. Baron: A general purpose global optimization software package. *Journal of global optimization*, 8:201–205, 1996.

[140] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[141] Bernhard Schölkopf and Julius von Kügelgen. From statistical to causal learning. In *Proceedings of the International Congress of Mathematicians*, volume 1, 2022.

[142] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Aapo Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006.

[143] Ali Shojaie and George Michailidis. Discovering graphical Granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 09 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq377. URL https://doi.org/10.1093/bioinformatics/btq377.

[144] Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 499–506, San Francisco, CA, USA, August 1995. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-385-1.

[145] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT press, 2000.

[146] M. Studeny. *Probabilistic Conditional Independence Structures*. Information Science and Statistics. Springer London, 2006. ISBN 9781846280832. URL https://books.google.cz/books?id=NJ4iwCMoznIC.

[147] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. Tslearn, a machine learning toolkit for time series data. *Journal of Machine Learning Research*, 21(118):1–6, 2020. URL http://jmlr.org/papers/v21/20-091.html.

[148] Christopher Tosh and Sanjoy Dasgupta. Maximum likelihood estimation for mixtures of spherical Gaussians is NP-hard. *J. Mach. Learn. Res.*, 18:175–1, 2017.

[149] Christopher Tosh and Sanjoy Dasgupta. Maximum likelihood estimation for mixtures of spherical gaussians is np-hard. *Journal of Machine Learning Research*, 18 (175):1–11, 2018. URL http://jmlr.org/papers/v18/16-657.html.

[150] Anastasios Tsiamis, Ingvar Ziemann, Nikolai Matni, and George J Pappas. Statistical learning theory for control: A finite-sample perspective. *IEEE Control Systems Magazine*, 43(6):67–97, 2023.

[151] Thijs van Ommen. Efficiently deciding algebraic equivalence of bow-free acyclic path diagrams. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024. URL https://openreview.net/forum?id=0s7uKfEfua.

[152] Arun Venkatraman, Wen Sun, Martial Hebert, J. Bagnell, and Byron Boots. Online instrumental variable regression with applications to online linear system identification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016. doi: 10.1609/aaai.v30i1.10215. URL https://ojs.aaai.org/index.php/AAAI/article/view/10215.

[153] Jie Wang, Martina Maggio, and Victor Magron. Sparsejsr: A fast algorithm to compute joint spectral radius via sparse sos decompositions. In *2021 American Control Conference (ACC)*, pages 2254–2259. IEEE, 2021.

[154] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. Tssos: A moment-sos hierarchy that exploits term sparsity. *SIAM Journal on Optimization*, 31(1):30–58, 2021.

[155] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. Chordal-tssos: a moment-sos hierarchy that exploits term sparsity with chordal extension. *SIAM Journal on Optimization*, 31(1):114–141, 2021.

[156] Xu Wang and Ali Shojaie. Causal discovery in high-dimensional point process networks with hidden nodes. *Entropy*, 23(12):1622, 2021.

[157] T. Warren Liao. Clustering of time series data – a survey. *Pattern Recognition*, 38(11):1857–1874, 2005. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2005.01.025. URL https://www.sciencedirect.com/science/article/pii/S0031320305001305.

[158] Nir Weinberger and Guy Bresler. The em algorithm is adaptively-optimal for unbalanced symmetric Gaussian mixtures. *The Journal of Machine Learning Research*, 23(1):4424–4502, 2022.

[159] Mike West and Jeff Harrison. *Bayesian forecasting and dynamic models*. Springer Science & Business Media, 2006.

[160] Tapio Westerlund and Frank Pettersson. An extended cutting plane method for solving convex minlp problems. *Computers & Chemical Engineering*, 19:131–136, 1995. ISSN 0098-1354. doi: https://doi.org/10.1016/0098-1354(95)87027-X. URL https://www.sciencedirect.com/science/article/pii/009813549587027X. European Symposium on Computer Aided Process Engineering.

[161] Jan C Willems, Paolo Rapisarda, Ivan Markovsky, and Bart LM De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.

[162] Axel Wismüller, Adora M. Dsouza, M. Ali Vosoughi, and Anas Abidin. Large-scale nonlinear granger causality for inferring directed dependence from short multivariate time-series data. *Scientific Reports*, 11(1):7817, Apr 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-87316-6. URL https://doi.org/10.1038/s41598-021-87316-6.

[163] Peter Wittek. Algorithm 950: Ncpol2sdpasparse semidefinite programming relaxations for polynomial optimization problems of noncommuting variables. *ACM Transactions on Mathematical Software (TOMS)*, 41(3):1–12, 2015.

[164] C. F. Jeff Wu. On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95 – 103, 1983. doi: 10.1214/aos/1176346060. URL https://doi.org/10.1214/aos/1176346060.

[165] Yihong Wu and Harrison H Zhou. Randomly initialized em algorithm for two-component gaussian mixture achieves near optimality in $o(\sqrt{\{n\}})$ iterations. *Mathematical Statistics and Learning*, 4(3), 2021.

[166] Ji Xu, Daniel J Hsu, and Arian Maleki. Global analysis of expectation maximization for mixtures of two gaussians. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/792c7b5aae4a79e78aaeda80516ae2ac-Paper.pdf.

[167] Lei Xu and Michael I. Jordan. On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*, 8(1):129–151, 01 1996. ISSN 0899-7667. doi: 10.1162/neco.1996.8.1.129. URL https://doi.org/10.1162/neco.1996.8.1.129.

[168] Yue Yu, Kun Zheng, Animashree Anandkumar, and Yizhou Yue. DAG-GNN: DAG structure learning with graph neural networks. In *International Conference on Machine Learning (ICML)*, pages 7154–7163, 2019.

[169] Jiji Zhang. Causal inference and reasoning in causally insufficient systems. *Philosophy of Science*, 75(5):748–760, 2008.

[170] Keli Zhang, Shengyu Zhu, Marcus Kalander, Ignavier Ng, Junjian Ye, Zhitang Chen, and Lujia Pan. gcastle: A python toolbox for causal discovery. *arXiv preprint arXiv:2111.15155*, 2021.

[171] Qiong Zhang and Jiahua Chen. Distributed learning of finite Gaussian mixtures. *The Journal of Machine Learning Research*, 23(1):4265–4304, 2022.

[172] Ruofei Zhao, Yuanzhi Li, and Yuekai Sun. Statistical convergence of the EM algorithm on Gaussian mixture models. *Electronic Journal of Statistics*, 14(1):632 – 660, 2020. doi: 10.1214/19-EJS1660. URL https://doi.org/10.1214/19-EJS1660.

[173] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. DAGs with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

[174] Fangting Zhou, Kejun He, and Yang Ni. Causal discovery with heterogeneous observational data, 2022. URL https://arxiv.org/abs/2201.12392.

[175] Fangting Zhou, Kejun He, and Yang Ni. Causal discovery with heterogeneous observational data. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022. URL https://openreview.net/forum?id=SfMArLi9e9.

[176] Quan Zhou and Jakub Mareček. Learning of linear dynamical systems as a non-commutative polynomial optimization problem. *IEEE Transactions on Automatic Control*, 2023.