Information Measures and Deep Learning Techniques for Network Inference

Arman Rahimzamani

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Sreeram Kannan, Chair

Payman Arabshahi

Wei Sun

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Information Measures and Deep Learning Techniques for Network Inference

Arman Rahimzamani

Chair of the Supervisory Committee:
Sreeram Kannan
Electrical and Computer Engineering

The pursuit of uncovering causal relationships among random variables through observational data is a longstanding and captivating focus in graphical models. While the formalization of causality has commenced only recently, this field continues to grapple with numerous intricate conceptual challenges.

We study the problem of causal inference in different settings, primarily centered on the applications in genomics, specifically focusing on the challenges and opportunities presented by single-cell transcriptome sequencing experiments (scRNA-seq). In our study:
(1) We introduce *Graph Divergence Measure* (GDM) to measure the compatibility of the independently observed samples to a hypothetical graphical structure. We design novel estimators for GDM which we show to be able to handle various data types and prove them to be consistent. We show GDM's performance superiority to the existing estimators.
(2) We address the problem of inferring the causal relationships in a dynamical system from the time series data by introducing *Restricted Directed Information* (RDI) and show that it recovers the graph correctly in all deterministic or stochastic instances. We then define the *Potential Conditional Mutual Information* (qCMI) as the conditional mutual information calculated with a modified joint prior distribution. Based on qCMI with a uniform prior, we introduce *uniform RDI* (uRDI) to alleviate the effect of sample bias in causal inference and showcase its superior performance via numerical experiments.

(3) Based on RDI and uRDI we develop *Scribe*, a toolkit for detecting and visualizing causal regulatory interactions between genes, and explore the potential for single-cell experiments to power network reconstruction. We show via the numerical examples that inference is possible whenever there is coupling between samples in time and gene expression space domains. We show the supremacy of our method over conventional causal inference methods such as Granger Causality and CCM and discuss its shortcomings and potential caveats.

(4) We introduce *Dynode*, a deep-learning-based software package to broaden the scope of understanding the governing dynamics of cell development processes. The goal is not just to identify causal relationships but also to predict the future state of individual cells based on their current state. Dynode consists of two toolboxes named *Dynode Vectorfield* and *Dynode Spatiotemporal*. Dynode Vectorfield is designed to learn the temporal dynamics of individual cells as isolated ecosystems, considering only their internal state. It is demonstrated with synthetic and real datasets and offers practical insights through differential analysis tools. Dynode Spatiotemporal, on the other hand, extends the analysis to include interactions between cells, considering chemical signaling and therefore both time and space dynamics. It is also showcased with synthetic and real-world datasets while discussing its limitations and potential challenges.

TABLE OF CONTENTS

Page

List of Figures iii

List of Tables v

Chapter 1: Introduction 1

Chapter 2: Multivariate Information Measures in General Probability Spaces . . . 6
2.1 Introduction 6
2.2 Graph Divergence Measure 9
2.3 Estimators 13
2.4 Proof of Consistency 16
2.5 Empirical Results 18
2.6 Discussion and Future Work 26

Chapter 3: Causal Inference from Time Series Observational Data 27
3.1 Assumptions, Definitions and Notations 28
3.2 Restricted Directed Information 30
3.3 Potential Conditional Mutual Information 44

Chapter 4: Scribe: Inferring Causal Gene Regulatory Networks from Coupled Single-Cell Expression Dynamics 59
4.1 Causal inference with Scribe: an RDI-based algorithm 61
4.2 Results 64
4.3 Conclusion and Discussion 75

Chapter 5: Learning Temporal Cell Transcriptomic Dynamics via Artificial Intelligence 78
5.1 Dynode Vectorfield: A comprehensive AI-based toolbox for learning temporal transcriptomic dynamics 79

5.2 Results 85
5.3 Conclusion and Discussion 93

Chapter 6: Learning joint Spatial and Temporal Cellular Dynamics via Artificial Intelligence 95
6.1 Dynode Spatiotemporal: An AI-based toolbox based on Graph Neural Networks for learning joint spatial and temporal transcriptomic dynamics 96
6.2 Results 98
6.3 Conclusion and Discussion 105

Bibliography 107

Appendix A: Supplementary Material for Chapter 2 136
A.1 Multivariate Mutual Information 136
A.2 Directed Information 137
A.3 Investigating assumptions in Theorem 1 138
A.4 Consistency Proofs 141

Appendix B: Supplementary Material for Chapter 3 164
B.1 Proof of Theorem 3 164
B.2 Proof of Theorem 4 166
B.3 Proof of Theorem 6 and Corollary 6.1 168
B.4 Proof of Theorem 13 169
B.5 Proof of the Theorem 15 170
B.6 KSG estimator for qCMI: Proof of convergence 172
B.7 Proof of lemma 24 174
B.8 Proof of lemma 26 175

Appendix C: Supplementary Material for Chapters 5 and 6 186
C.1 Uncoupled Time-Course Data Samples and Divergence Measures 186
C.2 Technical Noise, Communication Channels, and Loss Function Design 188
C.3 Regularizers for Smoothness and Stability 194
C.4 C. Elegans Spatiotemporal and Lineage Data Preparation 195

LIST OF FIGURES

Figure Number Page

2.1 Examples of Bayesian Networks and their corresponding induced distributions 10
2.2 The method for estimating information theoretic measures 16
2.3 The results for the experiments versus the number of samples 25

3.1 The causality graph of the Lorenz system 32
3.2 The heatmap of the directed information (DI) and restricted directed information (RDI) for Lorenz system 42
3.3 Probability of successfully extracting the right causality graph for DI and RDI methods. 43
3.4 (a): A causal graph, where the interest is in determining the strength of X to Y. (b) A gene expression trace as a function of time for a few example genes. 45
3.5 The qCMI values calculated for $u^n(0,1)$ distributions for X and Z, and uniform noise 53
3.6 The qCMI values calculated for a system with beta distribution for X and Z and Gaussian additive noise 54
3.7 AUC values for the neuron cells' development process: a) versus the number of steps. b) versus the number of runs. (c) for the decaying linear system 57

4.1 Scribe, a Toolkit for Inferring and Visualizing Causal Regulations 62
4.2 Live Imaging Dataset of C. elegans Early Embryogenesis Captures Transcription Expression Dynamics Hierarchy 67
4.3 Scribe recovers a core regulatory network responsible for myelopoiesis 69
4.4 Temporally coupled gene expression measurements are necessary for inferring causal regulatory interactions 72
4.5 Causal inference in Scribe with RNA-velocity 74

5.1 Dynode Vectorfield, a universal framework to model single-cell dynamics . . 80
5.2 Various types of data simulated from synthetic two-gene systems 85
5.3 Dynode accurately reconstructs the vector field of the simulated toggle switch system. 86

5.4 Dynode reconstructs the vector field and provides the geometry analysis for the Mixture of Gaussians from various experimental data types. 88
5.5 Dynode Vectorfield accurately infers the gene regulatory network for hematopoiesis and enables genetic predictions 90
5.6 Dynode Vectorfield accurately infers the gene regulatory network for Pancreatic Endocrinogenesis and enables genetic perturbation predictions 92
6.1 Spatial and temporal graphs Representing the signaling between the cells over space and time 97
6.2 Dynode Spatiotemporal accurately learns the dynamics governing the digit formation process in mammals 100
6.3 Learning spurious interactions in digit formation as a result of correlations in the wild type data 102
6.4 The reconstruction of the embryo in silico 3D models of spatiotemporally resolved and lineage-resolved C. elegans dataset with 4D single-cell protein atlas of transcription factors. 104

LIST OF TABLES

Table Number Page

6.1 Parameters for Digit Formation system [228]. 99

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to all the committee members for their time and dedication, and to everyone who has been helping me through my PhD degree.

DEDICATION

To my dearest parents and my beloved sister
for their unwavering support and boundless love

Chapter 1

INTRODUCTION

Determining the network of causal[1] interactions between random variables from observations has a long history in graphical models. Being a fascinating topic of research, causality's mathematization has only relatively recently started, and many conceptual problems are still being discussed — often with considerable intensity. Many of these conceptual problems are extensively discussed in comprehensive textbooks such as [115, 203, 207] well known to the researchers of the field.

The problem of causal inference from observational data has been studied in two major contexts: 1) independent samples, and 2) temporal observations. Inferring the causality from the independently observed samples has been widely studied [203], and algorithms have been devised for it such as the PC algorithm [122, 257]. This approach is closely tied to the conditional independence testing problem [22, 303] as all but all the algorithms (including PC) test for conditional independence between the variables conditioned over all the already discovered cause-effect relationships and/or the rest of the variables. However, the dynamical systems are usually modeled by ordinary differential equations (ODEs) or partial differential equations (PDEs) which characterize the temporal or spatiotemporal changes in the system's state at each time. Thus a more natural way to discover the causal relationships would be to observe how a system evolves through time and collect time series data. Causal structures have to be consistent with the time order. In fact, after knowing a causal ordering of nodes and assuming that there are no hidden variables, finding the causal DAG does need milder

[1] *Causality* and *Causal Inference* concepts have particularly strong meanings being debated extensively by researchers. Throughout this work, we slightly abuse the term to point out the more relaxed concept of functional relationships and consequential dependence between the variables of the system, i.e. X causes Y will mean Y is a functional of X.

assumptions compared to those required for independent observations, making it easier to do reasoning about causal relations among variables that refer to different time instances [207]. This temporal approach is widely studied, resulting in some of the well-known methods such as Granger Causality [91], Transfer Entropy [241] which is closely related to Directed Information [167], and Convergent Cross-Mapping [265].

In this work, we study the problem of causal inference in different settings. Our study consists of 1) proposing and analyzing methods to measure the compatibility of the independently observed samples to a hypothetical graphical structure, 2) A theory analysis of causal inference from the time series observations, 3) studying causal inference from the interventional observations from an information-theoretic perspective, and 4) Proposing practical methods to infer the causal relationships and learn the underlying dynamics using theoretical methods and deep learning techniques. We direct our work to be focused mostly on the applications in genomics. Single-cell transcriptome sequencing experiments (scRNA-seq) have attracted the attention of researchers working on inferring gene regulatory networks and learning the underlying dynamics governing cell development processes. However, current scRNA-seq techniques come with fundamental shortcomings and limitations such as not being able to generate true time series or extensive interventional (perturbation) data. Nevertheless, we show that our approach is able to extract the causal structures up to some extent even in such handicapped settings.

The rest of the thesis is organized as follows:

In Chapter 2, we introduce the *Graph Divergence Measure* (GDM), which measures the incompatibility between the observed distribution of a given set of samples and a hypothetical graphical model structure. Based on the Kulback-Leibler Divergence [43], GDM will be reduced to well-known information theoretic measures such as mutual information, conditional mutual information, and total correlation for specific graphical models. We design novel estimators for these graph divergence measures based on the direct estimate of the corresponding Radon-Nikodym derivatives. This makes us able to estimate the GDM for general probability spaces such as purely discrete, purely continuous, discrete-continuous mixtures, and the

real-valued data defined on a low-dimensional manifold, with the same estimator. We prove the consistency of our estimator and show the superiority of its performance to the existing estimators in finite-sample regimes via extensive numerical experiments.

In Chapter 3, we address the problem of inferring the causal relationships in a dynamical system from the time series data. We first introduce the *Restricted Directed Information* and show that while the existing methods are designed exclusively for either stochastic or deterministic systems, we show that RDI can address both systems. We demonstrate its efficacy via numerical experiments. In particular, we show RDI recovers the graph correctly in all deterministic or stochastic instances, where there is enough information to recover the graph uniquely. However, since RDI is based on conditional mutual information, it directly depends on the joint distribution $P_{X,Y,Z}$, while in practice the causality measure from the variable X to Y given the rest of the factors Z is more desirable to depend only on the conditional distribution $P_{Y|X,Z}$. Hence we define the *Potential Conditional Mutual Information* (qCMI) as the conditional mutual information calculated with a modified joint distribution $p_{Y|X,Z}q_{X,Z}$ where $q_{X,Z}$ is a potential distribution. We then propose a k nearest neighbor (KNN) based estimator for this function, employing importance sampling, and a coupling trick, and prove the finite k consistency of such an estimator. We demonstrate that the estimator has excellent practical performance and show an application in dynamical system inference. We employ qCMI with *uniform* distribution over X, Y to replace the ordinary CMI in RDI method and call the updated method uniform RDI (uRDI). We then apply the RDI and uRDI methods to gene expression scRNA-seq data, to determine the functional relationships between the genes by estimating the strength of information transferred from a potential regulator to its downstream target, and therefore extracting the Gene Regulatory Networks, as will be explored in Chapter 4.

In Chapter 4 we focus on a practical application of the measures introduced in Chapter 3 to discover the casual interaction of the genes in a cell development process by utilizing single-cell transcriptomic data. We introduce *Scribe*, a toolkit for detecting and visualizing causal regulatory interactions between genes, and explore the potential for single-cell experiments

to power network reconstruction. Scribe employs RDI and uRDI to determine causality from a potential regulator gene to its downstream target. We apply Scribe to several types of single-cell measurements and show that there is a dramatic drop in performance for "pseudotime"-ordered single-cell data compared with true time-series data. We demonstrate that performing causal inference requires temporal coupling between measurements. We show that methods such as "RNA velocity" restore some degree of coupling through an analysis of chromaffin cell fate commitment. These analyses in Chapter 4 highlight a shortcoming in experimental and computational methods for analyzing gene regulation at single-cell resolution and suggest ways of overcoming it. Therefore we show via the numerical examples that inference is possible whenever there is coupling between samples in time and gene expression space domains. We show the supremacy of our method over conventional causal inference methods such as Granger Causality and CCM.

In Chapters 5 and 6, we broaden the scope of the problem discussed above to comprehensively investigate the governing dynamics in cell development processes. In other words, we aim not only to uncover the causal relationships but also to predict the future transcriptomic state of each cell based on the current state of the cells. To achieve this objective, we shift our approach and leverage Deep Learning methods to learn the governing dynamics of dynamical systems from the observational data. We introduce *Dynode*, a deep-learning-based package developed using *PyTorch* to learn the underlying dynamics of these processes. Dynode comprises two toolboxes: *Dynode Vectofield* and *Dynode Spatiotemporal*.

Dynode Vectorfield is designed to learn the temporal dynamics for each cell as an isolated system. In this context, each cell is treated as an independent closed ecosystem where its future state is determined solely by its own current state. Extensively discussed in Chapter 5 with its capabilities showcased via two synthetic and two real datasets, Dynode Vectorfield is able to infer these *intra-cell* temporal dynamics, along with its corresponding vector field, within the constraints of the data. It also offers practical insights through the differential analysis tools integrated into the package.

Dynode Spatiotemporal, introduced and extensively explored in Chapter 6, further extends

the functionality of Dynode Vectorfield to also consider the *inter-cell* connections, where the information exchange via chemical signaling is known to be a crucial factor in addition to the internal state of each to determine the future state of the cells, thus learning the dynamics in both time and space domains. We showcase the capabilities of Dynode Spatiotemporal using both synthetic and real-world datasets and discuss its limitations and potential caveats.

Chapter 2

MULTIVARIATE INFORMATION MEASURES IN GENERAL PROBABILITY SPACES

In this chapter, we define a general Graph Divergence Measure ($\mathbb{GDM}$), as a measure of incompatibility between the observed distribution and a given graphical model structure. While this information-theoretic quantity is well defined in arbitrary probability spaces, existing estimators add or subtract entropy terms (we call them ΣH methods). These methods work only in either purely discrete spaces or purely continuous cases since entropy (or differential entropy) is well-defined only in that regime. Hence we construct a novel estimator via a coupling trick that directly estimates this multivariate information measure using the Radon-Nikodym derivative. The estimator is proven to be consistent in a general setting which includes several cases where the existing estimators fail, thus providing the only known estimators for the following settings: (1) the data has some discrete and some continuous valued components (2) some (or all) of the components themselves are discrete-continuous *mixtures* (3) the data is real-valued but does not have a joint density on the entire space, rather is supported on a low-dimensional manifold. We show that our proposed estimator significantly outperforms known estimators on synthetic and real data sets.

2.1 Introduction

Information theoretic quantities, such as mutual information and its generalizations, play an important role in various settings in machine learning and statistical estimation and inference. Here we summarize briefly the role of some generalizations of mutual information in learning (cf. Sec. 2.2.1 for a mathematical definition of these quantities).

1. **Conditional mutual information** measures the amount of information between two variables X and Y given a third variable Z and is zero iff X is independent of Y given Z. CMI finds a wide range of applications in learning including causality testing [49, 303], structure inference in graphical models [296], feature selection [65] as well as in providing privacy guarantees [47].

2. **Total correlation** measures the degree to which a set of N variables are independent of each other, and appears as a natural metric of interest in several machine learning problems, for example, in independent component analysis, the objective is to maximize the independence of the variables quantified through total correlation [114]. In feature selection, ensuring the independence of selected features is one goal, pursued using total correlation in [174, 286].

3. **Multivariate mutual information** measures the amount of information shared between multiple variables [29, 171] and is useful in feature selection [25, 144] and clustering [30].

4. **Directed information** measures the amount of information between two random processes [206, 292] and is shown to be the correct metric in identifying time series graphical models [4, 105, 222, 224, 225, 266].

Estimation of these information-theoretic quantities from observed samples is a non-trivial problem that needs to be solved in order to utilize these quantities in the aforementioned applications. While there has been long history in estimation of entropy [17, 132, 176, 297], and renewed recent interest [97, 143, 261], much less effort has been spent on the multivariate versions. A standard approach to estimating general information theoretic quantities is to write them out as a sum or difference of entropy (denoted H usually) terms which are then directly estimated; we term such a paradigm as ΣH paradigm. However, the ΣH paradigm is applicable only when the variables involved are all discrete or there is a joint density on the space of all variables (in which case, differential entropy h can be utilized). The underlying information measures themselves are well defined in very general probability

spaces, for example, involving mixtures of discrete and continuous variables; however, no known estimators exist.

We motivate the requirement of estimators in general probability spaces by some examples in contemporary machine learning and statistical inference.

1. It is common place in machine learning to have data-sets where **some variables are discrete, and some are continuous**. For example, in recent work on utilizing information bottleneck to understand deep learning [272], an important step is to quantify the mutual information between the training samples (which are discrete) and the layer's output (which is continuous). The employed methodology was to quantize the continuous variables; this is common practice, even though highly sub-optimal.

2. Some variables involved in the calculation **may be mixtures of discrete and continuous variables**. For example, the output of ReLU activation will not have a density even when the input data has a density. Instead, the neuron will have a discrete mass at 0 (or wherever the ReLU breakpoint is) but will have a continuous distribution on the positive values. This is also the case in gene expression data, where a gene may have a discrete mass at expression 0 due to an effect called drop-out [153] but have a continuous distribution elsewhere.

3. The variables involved may have a joint density **only on a low dimensional manifold**. For example, when calculating mutual information between input and output of a neural network, some of the neurons are deterministic functions of the input variables and hence they will have a joint density supported on a low-dimensional manifold rather than the entire space.

In the aforementioned cases, no existing estimators are known to work. It is not merely a matter of having provable guarantees either. When we plug in estimators that assume a joint density into data that does not, the estimated information measure can be strongly negative.

In this chapter, we will discuss the following contributions:

1. **General paradigm** (Section 2.2): We define a general paradigm of graph divergence measures which captures the aforementioned generalizations of mutual information as special cases. Given a directed acyclic graph (DAG) between n variables, the graph divergence is defined as the Kullback-Leibler (KL) divergence between the true data distribution $\mathbb{P}_X$ and a restricted distribution $\overline{\mathbb{P}}_X$ defined on the Bayesian network and can be thought of as a measure of incompatibility with the given graphical model structure. These graph divergence measures are defined using the Radon Nikodym derivatives which are well-defined for general probability spaces.

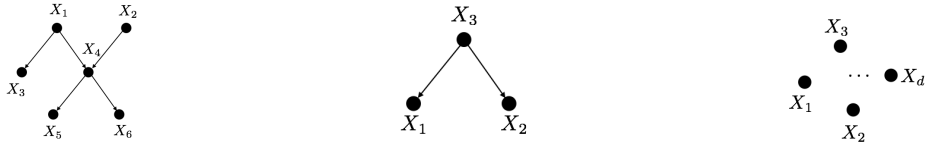
2. **Novel estimators** (Section 2.3): We propose novel estimators for these graph divergence measures, which directly estimate the corresponding Radon-Nikodym derivatives. To the best of our knowledge, these are the first family of estimators that are well defined for general probability spaces (breaking the ΣH paradigm).

3. **Consistency proofs** (Section 2.4): We prove that the proposed estimators converge to the true value of the corresponding graph divergence measures as the number of observed samples increases in a general setting which includes several cases: (1) the data has some discrete and some continuous valued components (2) some (or all) of the components themselves are discrete-continuous mixtures (3) the data is real-valued but does not have a joint density on the entire space but is supported on a low-dimensional manifold.

4. **Numerical results** (Section 2.5): Extensive numerical results demonstrate that (1) existing algorithms have severe failure modes in general probability spaces (strongly negative values, for example), and (2) our proposed estimator achieves consistency as well as significantly better finite-sample performance.

2.2 Graph Divergence Measure

In this section, we define the family of graph divergence measures. To begin with, we first define some notational preliminaries. We denote any random variable by an uppercase letter



(a) (b) (c)

Figure 2.1: (a) An example of Bayesian Network $\mathcal{G}$ with $\overline{\mathbb{P}}_X$ as the induced distribution $\mathbb{P}_{X_1}\mathbb{P}_{X_2}\mathbb{P}_{X_3|X_1}$ $\mathbb{P}_{X_4|X_1,X_2}\mathbb{P}_{X_5|X_4}\mathbb{P}_{X_6|X_4}$. (b) A Bayesian Network $\mathcal{G}$ inducing a Markov chain $\mathbb{P}_{X_3}\mathbb{P}_{X_1|X_3}\mathbb{P}_{X_2|X_3}$. (c) A Bayesian Network $\mathcal{G}$ with $\overline{\mathbb{P}}_X$ as the induced distribution $\mathbb{P}_{X_1}\mathbb{P}_{X_2}\cdots\mathbb{P}_{X_d}$.

such as X. The sample space of the variable X is denoted by $\mathcal{X}$ and any value in $\mathcal{X}$ is denoted by the lowercase letter x. For any subset $A \subseteq \mathcal{X}$, the probability of A for a given distribution function $\mathbb{P}_X(.)$ over X is denoted by $\mathbb{P}_X(A)$. We note that the random variable X can be a d-dimensional *vector* of random variables, i.e. $X \equiv (X_1, \ldots, X_d)$. The N observed samples drawn from the distribution $\mathbb{P}_X$ are denoted by $x^{(1)}, x^{(2)}, \ldots, x^{(N)}$, i.e. $x^{(i)}$ is the i^{th} observed sample.

Sometimes we might be interested in a subset of components of a random variable, $S \subseteq \{X_1, \ldots, X_d\}$ instead of the entire vector X. Accordingly, the sample space of the variable S is denoted by $\mathcal{S}$. For instance, $X = (X_1, X_2, X_3, X_4)$ and $S = (X_1, X_2)$. Throughout the entire chapter, unless otherwise stated, there is a one-to-one correspondence between the notations of X and any S. For example for any value $x \in \mathcal{X}$, the corresponding value in $\mathcal{S}$ is simply denoted by s. Further, $s^{(i)} \in \mathcal{S}$ represents the lower-dimensional sample corresponding to the ith observed sample $x^{(i)} \in \mathcal{X}$. Furthermore, any marginal distribution defined over S with respect to $\mathbb{P}_X$ is denoted by $\mathbb{P}_S$.

Consider a directed acyclic graph (DAG) $\mathcal{G}$ defined over d nodes (corresponding to the d components of the random variable X). A probability measure Q over X is said to be compatible with the graph $\mathcal{G}$ if it is a Bayesian network on $\mathcal{G}$. Given a graph $\mathcal{G}$ and a distribution $\mathbb{P}_X$, there is a natural measure $\overline{\mathbb{P}}_X(.)$ which is compatible with the graph and is

defined as follows:

$$\overline{\mathbb{P}}_X = \prod_{l=1}^{d} \mathbb{P}_{X_l|\mathrm{pa}(X_l)} \tag{2.1}$$

where $\mathrm{pa}(X_l) \subset X$ is the set of the *parent* nodes of the random variable X_l, with the sample space denoted by $\mathcal{X}_{\mathrm{pa}(l)}$, and the sample values $x_{\mathrm{pa}(l)}$ corresponding to x. The distribution $\mathbb{P}_{X_l|\mathrm{pa}(X_l)}$ is the *conditional* distribution of X_l given $\mathrm{pa}(X_l)$. Throughout the chapter, whenever mentioning the variable X_l with its own parents $\mathrm{pa}(X_l)$ we indicate it by $\mathrm{pa}+(X_l)$, i.e. $\mathrm{pa}+(X_l) \equiv \Big(X_l, \mathrm{pa}(X_l)\Big)$. An example is shown in Fig. 2.1a.

We note the fact that $\mathbb{P}_{S|X\setminus S}$ is well defined for any subset of variables $S \subset X$. Therefore if we let $S = X \setminus \mathrm{pa}(X_l)$, then $\mathbb{P}_{X\setminus\mathrm{pa}(X_l)|\mathrm{pa}(X_l)}$ is well defined for any $l \in \{1, \ldots, d\}$. By marginalizing over $X \setminus \mathrm{pa}+(X_l)$ we see that $\mathbb{P}_{X_l|\mathrm{pa}(X_l)}$ and thus the distribution $\overline{\mathbb{P}}_X$ is well defined.

The graph divergence measure is now defined as a function of the probability measure $\mathbb{P}_X$ and the graph $\mathcal{G}$. In this chapter, we will focus only on the KL Divergence as being the distance metric, hence unless otherwise stated $D(\cdot \parallel \cdot) = D_{KL}(\cdot \parallel \cdot)$. Let's first consider the case where $\mathbb{P}_X$ is absolutely continuous with respect to $\overline{\mathbb{P}}_X$ and hence the *Radon-Nikodym* derivative $d\mathbb{P}_X/d\overline{\mathbb{P}}_X$ exists. Therefore for a given set of random variables X and a Bayesian Network $\mathcal{G}$, we define **Graph Divergence Measure ($\mathbb{GDM}$)** as :

$$\mathbb{GDM}(X, \mathcal{G}) = D(\mathbb{P}_X \| \overline{\mathbb{P}}_X) = \int_{\mathcal{X}} \log \frac{d\mathbb{P}_X}{d\overline{\mathbb{P}}_X} d\mathbb{P}_X \tag{2.2}$$

Here we implicitly assume that $\log\left(d\mathbb{P}_X/d\overline{\mathbb{P}}_X\right)$ is measurable and integrable with respect to the measure $\mathbb{P}_X$. The $\mathbb{GDM}$ is set to infinity wherever Radon-Nikodym derivative does not exist. It is clear that $\mathbb{GDM}(X, \mathcal{G}) = 0$ if and only if the data distribution is compatible with the graphical model, thus the $\mathbb{GDM}$ can be thought of as a *measure of incompatibility with the given graphical model structure.*

We now have relevant variational characterization as below on our graph divergence measure, which can be harnessed to compute upper and lower bounds:

Proposition 2.2.1. *For a random variable X, a DAG $\mathcal{G}$, let $\Pi(\mathcal{G})$ be the set of measures $\mathbb{Q}_X$ defined on the Bayesian Network $\mathcal{G}$, then $\mathbb{GDMM}(X, \mathcal{G}) = \inf_{\mathbb{Q}_X \in \Pi(\mathcal{G})} D(\mathbb{P}_X \| \mathbb{Q}_X)$.*

Furthermore, let $\mathcal{C}$ denote the set of functions $h : \mathcal{X} \to \mathbb{R}$ such that $\mathbb{E}_{\mathbb{Q}_X}[\exp(h(X))] < \infty$. If $\mathbb{GDMM}(X, \mathcal{G}) < \infty$, then for every $h \in \mathcal{C}$, $\mathbb{E}_{\mathbb{P}_X}[h(X)]$ exists and:

$$\mathbb{GDMM}(X, \mathcal{G}) = \sup_{h \in \mathcal{C}} \mathbb{E}_{\mathbb{P}_X}[h(X)] - \log \mathbb{E}_{\mathbb{Q}_X}[\exp(h(X))]. \tag{2.3}$$

2.2.1 Special cases

For specific choices of X and Bayesian Network, $\mathcal{G}$, the Equation 2.2 is reduced to the well-known information measures. Some examples of these measures are as follows:
Mutual Information (MI): $X = (X_1, X_2)$ and $\mathcal{G}$ has no directed edge between X_1 and X_2. Thus $\overline{\mathbb{P}}_X = \mathbb{P}_{X_1}\mathbb{P}_{X_2}$, and we get, $\mathbb{GDMM}(X, \mathcal{G}) = I(X_1; X_2) = D(\mathbb{P}_{X_1X_2} \| \mathbb{P}_{X_1}\mathbb{P}_{X_2})$.

Conditional Mutual Information (CMI): We recover the conditional mutual information of X_1 and X_2 given X_3 by constraining $\mathcal{G}$ to be the one in Fig. 2.1b, since $\overline{\mathbb{P}}_X = \mathbb{P}_{X_3}\mathbb{P}_{X_2|X_3}\mathbb{P}_{X_1|X_3}$, i.e., $\mathbb{GDMM}(X, \mathcal{G}) = I(X_1; X_2|X_3) = D(\mathbb{P}_{X_1X_2X_3} \| \mathbb{P}_{X_1|X_3}\mathbb{P}_{X_2|X_3}\mathbb{P}_{X_3})$.

Total Correlation (TC): When $X = (X_1, \cdots, X_d)$, and $\mathcal{G}$ is the graph with no edges as in Fig. 2.1c, we recover the total correlation of the random variables $X_1, \ldots, X_d$ since $\overline{\mathbb{P}}_X = \mathbb{P}_{X_1} \ldots \mathbb{P}_{X_d}$, i.e., $\mathbb{GDMM}(X, \mathcal{G}_{dc}) = TC(X_1, \ldots, X_d) = D(\mathbb{P}_{X_1 \ldots X_d} \| \mathbb{P}_{X_1} \ldots \mathbb{P}_{X_d})$

Multivariate Mutual Information (MMI) : While the MMI defined by [171] is not positive in general, there is a different definition by [29] which is both non-negative and has an operational interpretation. Since MMI can be defined as the optimal total correlation after clustering, we can utilize our definition to define MMI (please see Appendix A Section A.1).

Directed Information : Suppose there are two stationary random processes X and Y,

the directed information rate from X to Y as first introduced by Massey [167] is defined as:

$$I(X \to Y) = \frac{1}{T} \sum_{t=1}^{T} I\left(X^t; Y_t \middle| Y^{t-1}\right)$$

It can be seen that the directed information can be written as:

$$I(X \to Y) = \mathbb{GDMM}\Big((X^T, Y^T), \mathcal{G}_I\Big) - \mathbb{GDMM}\Big((X^T, Y^T), \mathcal{G}_C\Big)$$

where the graphical model $\mathcal{G}_I$ corresponds to the *independent* distribution between X^T and Y^T, and $\mathcal{G}_C$ corresponds to the *causal* distribution from X to Y (more details provided in Appendix A Section A.2).

2.3 Estimators

2.3.1 Prior Art

Estimators for entropy date back to Shannon, who guesstimated the entropy rate of English [248]. Discrete entropy estimation is a well-studied topic and minimax rate of this problem is well-understood as a function of the alphabet size [121, 199, 300]. The estimation of differential entropy is considerably harder and also studied extensively in literature [17, 143, 146, 176, 190, 253, 259] and can be broadly divided into two groups; based on either Kernel density estimates [81, 123] or based on k-nearest-neighbor estimation [118, 197, 261]. In a remarkable work, Kozachenko and Leonenko suggested a nearest neighbor method for entropy estimation [132] which was then generalized to a kth nearest neighbor approach [252]. In this method, the distance to the kth nearest neighbor (KNN) is measured for each data-point, and based on this the probability density around each data point is estimated and substituted into the entropy expression. When k is fixed, each density estimate is noisy and the estimator accrues a bias and a remarkable result is that the bias is distribution-independent and can be

subtracted out [254].

While the entropy estimation problem is well-studied, mutual information and its generalizations are typically estimated using a sum of signed entropy (H) terms, which are estimated first; we term such estimators as ΣH estimators. In the discrete alphabet case, this principle has been shown to be worst-case optimal [97]. In the case of distributions with a joint density, an estimator that breaks the ΣH principle is the KSG estimator [134], which builds on the KNN estimation paradigm but couples the estimates in order to reduce the bias. This estimator is widely used and has excellent practical performance. The original paper did not have any consistency guarantees and its convergence rates were recently established [74]. There have been some extensions to the KSG estimator for other information measures such as conditional mutual information [67, 237], directed information [284] but none of them show theoretical guarantees on consistency of the estimators, furthermore they fail completely in mixture distributions.

When the data distribution is neither discrete nor admits a joint density, the ΣH approach is no longer feasible as the individual entropy terms are not well defined. This is the regime of interest in our paper. Recently, Gao et al [79] proposed a mutual-information estimator based on KNN principle, which can handle such continuous-discrete mixture cases, and the consistency was demonstrated. However it is not clear how it generalizes to even Conditional Mutual Information (CMI) estimation, let alone other generalizations of mutual information. In this paper, we build on that estimator in order to design an estimator for general graph divergence measures and establish its consistency for generic probability spaces.

2.3.2 Proposed Estimator

The proposed estimator is given in Algorithm 1 where $\psi(\cdot)$ is the digamma function and $\mathbf{1}_{\{\cdot\}}$ is the indicator function. The process is schematically shown in Fig. 2.2. We used the ℓ_∞-norm everywhere in our algorithm and proofs.

The estimator intuitively estimates the $\mathbb{GDMM}$ by the *resubstitution estimate* $\frac{1}{N}\sum_{i=1}^{N}\log\hat{f}(x^{(i)})$ in which $\hat{f}(x^{(i)})$ is the estimation of Radon-Nikodym derivative at each sample $x^{(i)}$. If $x^{(i)}$

.

Input: Parameter: $k \in \mathbf{Z}^+$, **Samples:** $x^{(1)}, x^{(2)}, \ldots, x^{(N)}$, **Bayesian Network:** $\mathcal{G}$ on **Variables:** $\mathcal{X} = (X_1, X_2, \cdots, X_d)$

Output: $\widehat{\mathbb{GDMM}}^{(N)}(X, \mathcal{G})$

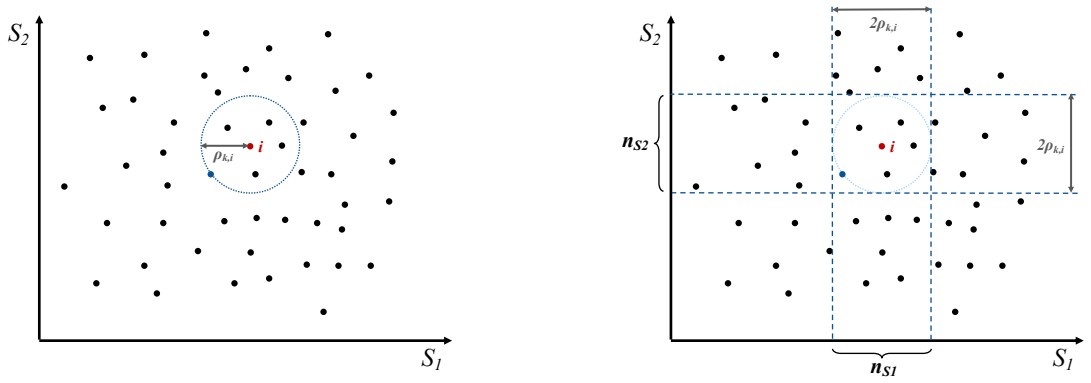
1: **for** $i = 1$ `to` N **do**
2: **Query:**
3: $\rho_{k,i} = \ell_\infty$-distance to the kth nearest neighbor of $x^{(i)}$ in the space $\mathcal{X}$
4: **Inquire:**
5: $\tilde{k}_i = \#$ points within the $\rho_{k,i}$-neighborhood of $x^{(i)}$ in the space $\mathcal{X}$
6: $n^{(i)}_{\mathrm{pa}(X_l)} = \#$ points within the $\rho_{k,i}$-neighborhood of $x^{(i)}$ in the space $\mathcal{X}_{\mathrm{pa}(l)}$
7: $n^{(i)}_{\mathrm{pa+}(X_l)} = \#$ points within the $\rho_{k,i}$-neighborhood of $x^{(i)}$ in the space $\mathcal{X}_{\mathrm{pa+}(l)}$
8: **Compute:**
9: $\zeta_i = \psi(\tilde{k}_i) + \sum_{l=1}^{d} \left(\mathbf{1}_{\{\mathrm{pa}(X_l) \neq \emptyset\}} \log \left(n^{(i)}_{\mathrm{pa}(X_l)} + 1 \right) - \log \left(n^{(i)}_{\mathrm{pa+}(X_l)} + 1 \right) \right)$
10: **end for**
11: Final Estimator:

$$\widehat{\mathbb{GDMM}}^{(N)}(X, \mathcal{G}) = \frac{1}{N} \sum_{i=1}^{N} \zeta_i + \left(\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l) = \emptyset\}} - 1 \right) \log N \tag{2.4}$$

Algorithm 1: Estimating **Graph Divergence Measure** $\mathbb{GDMM}(X, \mathcal{G})$

lies in a region where there is a density, the RN derivative is equal to $g_X(x^{(i)})/\bar{g}_X(x^{(i)})$ in which $g_X(.)$ and $\bar{g}_X(.)$ are density functions corresponding to $\mathbb{P}_X$ and $\overline{\mathbb{P}}_X$ respectively. On the other hand, if $x^{(i)}$ lies on a point where there is a discrete mass, the RN derivative will be equal to $h_X(x^{(i)})/\bar{h}_X(x^{(i)})$ in which $h_X(.)$ and $\bar{h}_X(.)$ are mass functions corresponding to $\mathbb{P}_X$ and $\overline{\mathbb{P}}_X$ respectively.

The density function $\bar{g}_X(x^{(i)})$ can be written as $\prod_{l=1}^{d} \left(g_{\mathrm{pa+}(X_l)}(x_{\mathrm{pa+}(l)}{}^{(i)}) / g_{\mathrm{pa}(X_l)}(x_{\mathrm{pa}(l)}{}^{(i)}) \right)$ for continuous components. Equivalently, the mass function $\bar{h}_X(x^{(i)})$ can be written as $\prod_{l=1}^{d} \left(h_{\mathrm{pa+}(X_l)}(x_{\mathrm{pa+}(l)}{}^{(i)}) / h_{\mathrm{pa}(X_l)}(x_{\mathrm{pa}(l)}{}^{(i)}) \right)$ for discrete components. Thus we need to estimate the density functions $g(.)$ and the mass functions $h(.)$ according to the type of $x^{(i)}$. The existing kth nearest neighbor algorithms will suffer while estimating the mass functions $h(.)$, since $\rho_{n_S,i}$ (the distance to the n_S-th nearest neighbor in subspace $\mathcal{S}$) for such points will be equal to zero for large N. Our algorithm, however, is designed in a way that it's capable of



(a) (b)

Figure 2.2: The method for estimating information theoretic measures. **(a)** Step 1: **Query** for the distance to the kth nearest neighbor of each point i in the space $\mathcal{X}$. **(b)** Step 2: **Inquire** for the number of points lying within the $\rho_{k,i}$-neighborhood of each point i in the subspaces of $\mathcal{X}$ including itself.

approximating both $g(.)$ functions as $\approx \frac{n_S}{N}\frac{1}{(\rho_{n_S,i})^{d_S}}$ and $h(.)$ functions as $\approx \frac{n_S}{N}$ dynamically for any subset $S \subseteq X$. This is achieved by setting $\rho_{n_S,i}$ terms such that all of them cancel out, yielding the estimator as in Eq. (2.4).

2.4 Proof of Consistency

The proof of consistency for our estimator consists of two steps: First we prove that the expected value of the estimator in Eq. (2.4) converges to the true value as $N \to \infty$, and second we prove that the variance of the estimator converges to zero as $N \to \infty$. Let's begin with the definition of $P_X(x, r)$:

$$P_X(x,r) = \mathbb{P}_X\big\{a \in \mathcal{X} : \|a - x\|_\infty \leq r\big\} = \mathbb{P}_X\Big\{B_r(x)\Big\} \tag{2.5}$$

Thus $P_X(x, r)$ is the probability of a hypercube with the edge length of $2r$ centered at the point x. We then state the following assmuptions:

Assumption 1. *We make the following assumptions to prove the consistency of our method:*

1. *k is set such that* $\lim_{N\to\infty} k = \infty$ *and* $\lim_{N\to\infty} \frac{k \log N}{N} = 0$.

2. *The set of discrete points* $\{x : P_X(x, 0) > 0\}$ *is finite.*

3. $\int_{\mathcal{X}} \big|\log f(x)\big| d\mathbb{P}_X < +\infty$, *where* $f \equiv d\mathbb{P}_X / d\overline{\mathbb{P}}_X$ *is Radon-Nikodym derivative.*

The Assumption 1.1 with 1.2 controls the boundary effect between the continuous and the discrete regions; with this assumption, we make sure that all the k nearest neighbors of each point belong to the same region almost surely (i.e. all of the points in the neighborhood are either continuous or discrete). Assumption 1.3 is the log-integrability of the Radon-Nikodym derivative. These assumptions are satisfied under mild technical conditions whenever the distribution $\mathbb{P}_X$ over the set $\mathcal{X}$ is (1) finitely discrete; (2) continuous; (3) finitely discrete over some dimensions and continuous over some others; (4) a mixture of the previous cases; (5) has a joint density supported over a lower dimensional manifold. These cases represent almost all the real world data.

As an example of a case not conforming to the aforementioned cases, we can consider singular distributions, among which the *Cantor distribution* is a significant example whose cumulative distribution function is the Cantor function. This distribution has neither a probability density function nor a probability mass function, although its cumulative distribution function is a continuous function. It is thus neither a discrete nor an absolutely continuous probability distribution, nor is it a mixture of these.

The Theorem 1 formally states the mean-convergence of the estimator while Theorem 2 formally states that convergence of the variance to zero.

Theorem 1. *Under the Assumptions 1, we have* $\lim_{N\to\infty} \mathbb{E}\left[\widehat{\mathbb{GDM}}^{(N)}(X, \mathcal{G})\right] = \mathbb{GDM}(X, \mathcal{G})$.

Theorem 2. *In addition to the Assumptions 1, assume that we have* $(k_N \log N)^2 / N \to 0$ *as* N *goes to infinity. Then we have* $\lim_{N\to\infty} Var\left[\widehat{\mathbb{GDM}}^{(N)}(X, \mathcal{G})\right] = 0$.

The Theorems 1 and 2 combined yield the consistency of the estimator 2.4.

The proof of the Theorem 1 starts with writing the Radon-Nikodym derivative explicitly. Then we need to upper-bound the term $\left|\mathbb{E}\big[\widehat{\mathbb{GDM}}^{(N)}(X,\mathcal{G})\big] - \mathbb{GDM}(X,\mathcal{G})\right|$. To achieve this goal, we segregate the domain of $\mathcal{X}$ into three parts as $\mathcal{X} = \Omega_1 \cup \Omega_2 \cup \Omega_3$ where $\Omega_1 = \{x : f(x) = 0\}$, $\Omega_2 = \{x : f(x) > 0, P_X(x,0) > 0\}$ and $\Omega_3 = \{x : f(x) > 0, P_X(x,0) = 0\}$. We will show that $\mathbb{P}_X(\Omega_1) = 0$. The sets Ω_2 and Ω_3 correspond to the discrete and continuous regions respectively. Then for each of the two regions, we introduce an upper bound that goes to zero as N grows boundlessly. Thus equivalently we show the mean of the estimate ζ_1 is close to $\log f(x)$ for any x.

The proof of the Theorem 2 is based on the Efron-Stein inequality, which upper bounds any estimator for any quantity from the observed samples $x^{(1)}, \ldots, x^{(N)}$. For any sample $x^{(i)}$, we then upper bound the term $\left|\zeta_i(X) - \zeta_i(X_{\setminus j})\right|$ by segregating the samples into various cases and examining each case separately. $\zeta_i(X)$ is the estimate using all the samples $x^{(1)}, \ldots, x^{(N)}$ and $\zeta_i(X_{\setminus j})$ is the estimate when the jth sample is removed. Summing up over all the i's, we obtain an upper bound which will converge to 0 as N goes to infinity.

The detailed proofs can be found in Appendix A.

2.5 Empirical Results

In this section, we evaluate the performance of our proposed estimator in comparison with other estimators via numerical experiments. The estimators evaluated here are our estimator referred to as *GDM*, the plain KSG-based estimators for continuous distributions to which we refer by *KSG*, the *binning* estimators and the noise-induced ΣH estimators. The parameter of the nearest neighbor k for all the GDM, KSG and ΣH methods is set to $\sqrt{N}/5$ to conform with the Assumptions 1 and keep the computational complexity at an acceptable level. The number of bins in binning method at each dimension is kept at $\sqrt[d]{N/m}$ for all the algorithms so we roughly have m samples per each bin, and we choose $m \in \{10, 20, 100\}$ whichever giving the best precision.

2.5.1 Experiment 1: Markov chain model with continuous-discrete mixture

For the first experiment, we simulated an X-Z-Y Markov chain model in which the random variable X is chosen as $X = \min\left(\tilde{X}, \alpha_1\right)$ where $\tilde{X} \sim \mathcal{U}(0,1)$ represents an auxiliary random variable uniformly distributed between 0 and 1. This means that we first generate a sample from $\tilde{X}$ denoted by $\tilde{x}$ and then let $x = \min\{\tilde{x}, \alpha_1\}$. Subsequently:

$$Z = \min\left(X, \alpha_2\right) \tag{2.6}$$

$$Y = \min\left(Z, \alpha_3\right) \tag{2.7}$$

We assume that $\alpha_3 < \alpha_2 < \alpha_1$. The three variables X, Y and Z represent a mixture of continuous and discrete random variables.

We simulated this system for various numbers of samples while setting $\alpha_1 = 0.9$, $\alpha_2 = 0.8$ and $\alpha_3 = 0.7$. For each set of samples $I(X;Y|Z)$ is estimated via different methods and its theory value is obviously equal to 0. The results are shown in Figure 2.3a. We can see that in this regime, only the GDM estimator can correctly converge. The KSG estimator and the ΣH estimator show high negative biases and the binning estimator shows a positive bias.

2.5.2 Experiment 2: Mixture of AWGN and BSC channels with variable error probability

As the second experiment, we considered an Additive White Gaussian Noise (AWGN) Channel in parallel with a Binary Symmetric Channel (BSC) where only one of them can be activated at a time. The random variable $0 < Z < 1$ controls which channel is activated; i.e. if Z is lower than the threshold β, then the AWGN channel is activated, otherwise the BSC channel will be activated.

The AWGN channel is modeled as $Y = X + N$ where $X \sim \mathcal{N}(0, \sigma_X^2)$ and $N \sim \mathcal{N}(0, \sigma_N^2)$. BSC channel is modeled as $Y = X \oplus E$, where X and E are two binary random variables $X \sim \text{Bern}(p)$ and $E \sim \text{Bern}(Z)$ denoting the input and the error respectively. This means that the probability of error in the BSC channel is controlled by the variable Z at each

time-point. Equivalently, if we suppose that the sample z_i is observed from Z at the moment i, the output of the BSC at time i is characterized by:

$$y_i = \begin{cases} x_i & \text{with probability } 1 - z_i \\ \neg x_i & \text{with probability } z_i \end{cases} \tag{2.8}$$

Let's assume $Z = \min\left(\alpha, \tilde{Z}\right)$ where $\tilde{Z} \sim \mathcal{U}(0,1)$ is an auxiliary uniform random variable, similar to the previous experiments. The theory value for $I(X : Y|Z)$ is obtained as follows:

$$\begin{aligned} I(X;Y|Z) &= \int_{z=0}^{1} I(X;Y|Z=z) f_Z(z) dz && (2.9) \\ &= \int_{z=0}^{\beta} I_{\text{AWGN}}(X;Y|Z=z) f_Z(z) dz + \int_{z=\beta}^{1} I_{\text{BSC}}(X;Y|Z=z) f_Z(z) dz && (2.10) \\ &= \beta I_{\text{AWGN}}(X;Y) + \int_{z=\beta}^{\alpha} I_{\text{BSC}}(X;Y|Z=z) dz \\ &\quad + (1-\alpha)\, I_{\text{BSC}}(X;Y|Z=\alpha) && (2.11) \\ &= \frac{\beta}{2} \log\left(1 + \frac{\sigma_X^2}{\sigma_N^2}\right) + \int_{z=\beta}^{\alpha} \Big(h(z\bar{p} + p\bar{z}) - h(z)\Big) dz \\ &\quad + (1-\alpha)\Big(h(\alpha\bar{p} + p\bar{\alpha}) - h(\alpha)\Big) && (2.12) \end{aligned}$$

in which $h(x)$ is the binary entropy function defined as $h(x) = -x \log x - (1-x)\log(1-x)$ for $x \in [0,1]$.

In our experiment discussed in Section 2.5 we set $p = 0.5$, $\alpha = 0.3$, $\beta = 0.2$, $\sigma_X = 1$ and $\sigma_N = 0.1$. The theory value for the conditional mutual information can be readily calculated as $I(X;Y|Z) = 0.53241$. We simulated the system for various number of samples, and obtained the estimated CMI values $I(X;Y|Z,Z^2,Z^3)$. Its theory value is obviously equal to $I(X;Y|Z)$, yet it's conditioned over a low-dimensional manifold in a high-dimensional space, and we are interested in examining the effect of it over the estimators' accuracy. The results are shown in Figure 2.3b. Similar to the previous experiment, the GDM estimator can correctly converge to the true value. The ΣH and binning estimators show a negative bias, and the KSG estimator gets totally lost.

2.5.3 Experiment 3: Total Correlation for independent mixtures

In the third set of experiments, we estimated the total correlation of three independent random variables X, Y and Z each of which is created independently from a mixture distribution as follows: First we generate an auxiliary random variable $\tilde{X} \sim \text{Bern}(0.5)$ then the random variable X is generated as follows:

$$X = \begin{cases} \alpha_X & \text{if } \tilde{X} = 0 \\ \sim \mathcal{U}(0,1) & \text{if } \tilde{X} = 1 \end{cases} \tag{2.13}$$

which means we toss a fair coin, if heads appears we will fix X at α_X, otherwise we will draw X from a uniform distribution between 0 and 1. samples from Y and Z are also generated independently in the same fashion. For this setup, the theory value of the total correlation of the three variables X, Y and Z is obviously equal to 0.

In our experiment we set $\alpha_X = 1$, $\alpha_Y = 1/2$ and $\alpha_Z = 1/4$, and generated various datasets with different number of samples. Then we estimated the total correlation via different approaches. The results are shown in the Figure 2.3c.

2.5.4 Experiment 4: Total Correlation for independent uniforms with correlated zero-inflation

In this experiment, we examine a system of random variables, which can be *clustered* into independent subsets, while inside each of the subsets, the variables are dependent. As a simple case we consider 4 random variables X_1, X_2, X_3 and X_4 which can be clustered as $\{X_1, X_2\}$ and $\{X_3, X_4\}$. Suppose $\tilde{X}_1$, $\tilde{X}_2$, $\tilde{X}_3$ and $\tilde{X}_4$ are four independent auxiliary random variables identically distributed as $\mathcal{U}(0.5, 1.5)$. Then we erase samples ($\tilde{X}_1$, $\tilde{X}_2$) and ($\tilde{X}_3$, $\tilde{X}_4$) independently, to generate the random variables X_1, X_2, X_3 and X_4; i.e. $X_1 = \alpha_1 \tilde{X}_1$, $X_2 = \alpha_1 \tilde{X}_2$ and $X_3 = \alpha_2 \tilde{X}_3$, $X_4 = \alpha_2 \tilde{X}_4$ in which $\alpha_1 \sim \text{Bern}(p_1)$ and $\alpha_2 \sim \text{Bern}(p_2)$. As we see, after this erasure process resulting in zero-inflation, X_1 and X_2 become correlated, so do

X_3 and X_4 while still $(X_1, X_2) \perp\!\!\!\perp (X_3, X_4)$. Thus the total correlation can be written as:

$$\begin{aligned} TC\left(X_1, X_2, X_3, X_4\right) &= h(X_1) + h(X_2) + h(X_3) + h(X_4) - h\left(X_1, X_2, X_3, X_4\right) && (2.14) \\ &= h(X_1) + h(X_2) + h(X_3) + h(X_4) - h\left(X_1, X_2\right) - h\left(X_3, X_4\right) && (2.15) \\ &= I\left(X_1; X_2\right) + I\left(X_3; X_4\right) && (2.16) \end{aligned}$$

To calculate $I\left(X_1; X_2\right)$, after applying the chain rule to $I\left(X_1; X_2, \alpha_1\right)$ twice, we have:

$$\begin{aligned} I\left(X_1; X_2\right) &= I\left(X_1; \alpha_1\right) + I\left(X_1; X_2 | \alpha_1\right) - I\left(X_1; \alpha_1 | X_2\right) && (2.17) \\ &= h(\alpha_1) - \underbrace{h\left(\alpha_1 | X_1\right)}_{0} && (2.18) \\ &\quad + p(\alpha_1 = 0) \underbrace{I\left(X_1; X_2 | \alpha_1 = 0\right)}_{0} + p(\alpha_1 = 1) \underbrace{I\left(X_1; X_2 | \alpha_1 = 1\right)}_{0} \\ &\quad - h\left(X_1 | X_2\right) + \underbrace{h\left(X_1 | X_2, \alpha_1\right)}_{= h(X_1 | X_2)} \\ &= h(\alpha_1) && (2.19) \end{aligned}$$

Similarly, $I\left(X_3; X_4\right) = h(\alpha_2)$. Thus:

$$TC\left(X_1, X_2, X_3, X_4\right) = h(\alpha_1) + h(\alpha_2) \tag{2.20}$$

In the experiment, we set $p_1 = p_2 = 0.6$. The theory value for the total correlation is equal to 1.34602. The results of running different algorithms over the data can be seen in Figure 2.3d. For the total correlation experiments 3 and 4, similar to that of conditional mutual information in experiments 1 and 2, only the GDM estimator can best estimate the true value. The estimator ΣH was removed from the figures due to its high inaccuracy.

2.5.5 Experiment 5: Gene Regulatory Networks

In this experiment, we briefly test our algorithm in doing causal inference, which is going to be examined more extensively later in the thesis. We use different estimators to do Gene

Regulatory Network inference based on the Restricted Directed Information (RDI) measure.

In a simplified model of a dynamical system $X = (X_1, \ldots, X_d)$, each X_l is a time series of length T written in the form $\{X_l(t)\}_{t=0}^{T}$, and the system's evolution through time is characterized as $X_l(t) = g_l\big(X(t-1)\big) + N_l(t)$ in which $g_l(.) : \mathcal{X} \to \mathcal{X}_l$ is a deterministic function and $N_l(t)$ is an independent random noise. In the causal inference, the goal is to infer the set of $\text{pa}(X_l)$ for each X_l. To reach this end, many researchers have studied the information theoretic measures. One of such measures for time series is the directed information, and a variation of it is the restricted directed information defined as $RDI(X_i \to X_j) := I\big(X_i(t-1), X_j(t) | X_j(t-1)\big)$ and the conditional version of it (cRDI) is defined as $RDI(X_i \to X_j | Z) := I\big(X_i(t-1), X_j(t) | X_j(t-1), Z(t-1)\big)$. It's shown that for the first-order Markov systems and under some mild conditions, $RDI(X_i \to X_j | \{X_i, X_j\}^c) \neq 0$ if and only if $X_i \in pa(X_j)$ [224]. Since RDI (cRDI) is in fact a conditional mutual information, its performance is directly related to that of the CMI estimator, which is used to obtain cRDI from the samples. Hence we are interested in evaluating the performance of various estimators in causal inference. The RDI measure is discussed in more details in Chapter 3.

We do our test on the simulated neuron cells' development process, based on a model from [216]. In this model, the time series vector X consists of 13 random variables each of which corresponding to a single gene in the development process. We simulated the development process for various values of T in which the additive noise is independent and identically distributed as $\mathcal{N}(0, .03)$ for all the genes, and every single sample is then subject to erasure (i.e. be replaced by 0s) with a probability of 0.5. Then we applied the cRDI method to the data to discover the pairwise causal relationships. In our method, we first applied the RDI method (with no conditioning) and obtained the pairwise values. Then we repeated the process to obtain cRDI values, and for every pair X_i and X_j, we conditioned the RDI over the X_{k^*} in which $k^* = \text{argmax}_{k \neq i} RDI(X_k \to X_j)$ (This method is called *Scribe* with parameter $l = 1$. See Chapter 4).

To calculate RDI and cRDI, various CMI estimators are utilized and then the Area-Under-ROC curve (AUROC) is calculated for each estimator. More detailed results are shown in

Figure 2.3e. We notice that the zero-inflation is highly detrimental to the causal signals, so we won't expect high performance of the causal inference over the data. As we see, RDI/cRDI methods implemented with the GDM estimator outperform the other estimators by at least %10 in terms of AUROC. We did not include the ΣH estimator results due to its very low performance.

2.5.6 Experiment 6: Feature Selection by Conditional Mutual Information Maximization

Selecting the best features for a learning task using information theoretic measures is well studied in the literature [286]. Among the well-known methods is the conditional mutual information maximization (CMIM) first introduced by Flueret [65], a variation of which was later introduced called CMIM-2 [285]. Suppose we have observed samples from the data (X, Y), and would like to select the set $S \subset X$ which describes Y the best. Suppose we start at $S = \emptyset$ and we add features to S in a recursive greedy fashion as:

$$\begin{aligned} \text{If } S = \emptyset \quad &: \quad S \leftarrow S \cup \{\text{argmax}_{X_i} I(X_i; Y)\} \\ \text{Otherwise} \quad &: \quad S \leftarrow S \cup \{\text{argmax}_{X_i} f(X_i, S)\} \end{aligned} \tag{2.21}$$

The algorithm 2.21 is equivalent to CMIM when $f(X_i, S) := \min_{X_j \in S} I(X_i, Y | X_j)$, and it's equivalent to CMIM-2 if we let $f(X_i, S) := \sum_{X_j \in S} I(X_i, Y | X_j)$ instead.

In our experiment, we generated a vector $X = (X_1, \ldots, X_{15})$ of 15 random variables in which all the random variables are taken from $\mathcal{N}(0, 1)$ and then each random variable X_i is clipped from above at α_i which is initially taken randomly from $\mathcal{U}(0.25, 0.3)$ and then kept constant during the sample generation. Then Y is generated as $Y = \cos\left(\sum_{i=1}^{5} X_i\right)$. So only the first 5 features are relevant and the other features are independent of Y. Thus an ideal feature selection method should be able to recover the first 5 features first. We implemented and ran both CMIM and CMIM-2 algorithms with various CMI estimators to evaluate the performance of the estimators and see how well they can extract the true features $X_1, \ldots, X_5$. A more detailed version of the AUROC plot including both CMIM and CMIM-2 is shown

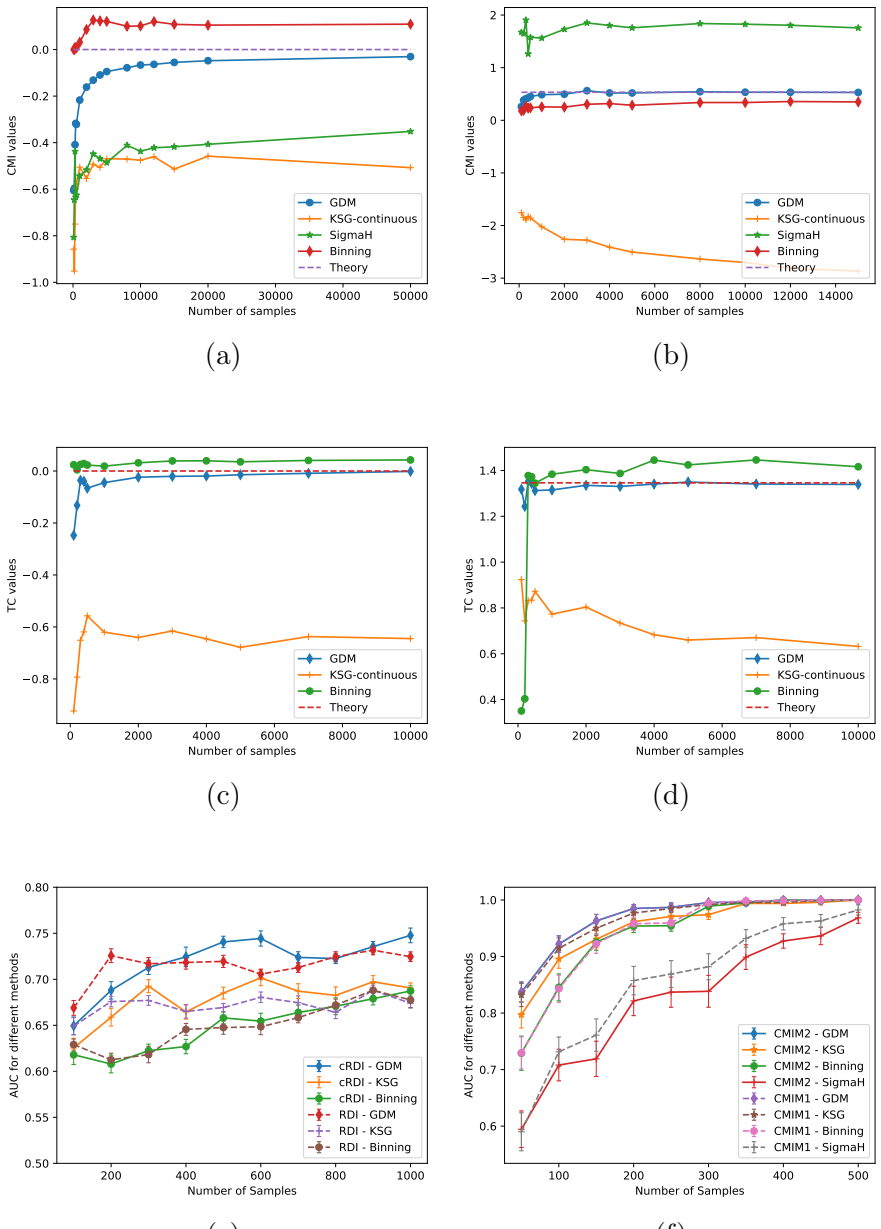


Figure 2.3: The results for the experiments versus the number of samples: 2.3a: The estimated CMI for the X-Z-Y Markov chain. 2.3b: CMI for the AWGN+BSC channels with low-dimensional Z manifold. 2.3c: The estimated TC values for three independent variables. 2.3d: The estimated TC for zero-inflated variables. 2.3e: The AUROC values for gene regulatory network inference. The error bars show the standard deviation scaled down by 0.2. 2.3f: The AUROC values for feature selection accuracy. The error bars show the standard deviations scaled down by 0.2.

in Figure 2.3f. We can see that the feature selection methods implemented with the GDM estimator outperform the other estimators. We notice that the performances of CMIM and CMIM-2 with the GDM estimator are almost identical.

2.6 Discussion and Future Work

In this chapter, a general paradigm of graph divergence measures and novel estimators thereof, for general probability spaces are proposed, which estimate several generalizations of mutual information. In the future, we would like to derive further efficient estimators for high-dimensional data. In the current work, estimators are shown to be consistent with infinite scaling of parameter k. In the future, we would like to understand the finite k performance of the estimators as well as guarantees on sample complexity and rates of convergence. Another potential direction to follow is to study the variational characterization of the graph divergence measure to design estimators. Improving the computational efficiency of the estimator is another direction of future work. Recent literature including [191] provides a new methodology to estimate mutual information in a computationally efficient manner and leveraging these ideas for the generalized measures and general probability distributions can be a promising direction ahead.

Chapter 3

CAUSAL INFERENCE FROM TIME SERIES OBSERVATIONAL DATA

In the previous chapter, we briefly introduced Restricted Directed (RDI) Information as a Graph Divergence Measure and how it can be utilized to infer causal relationships from observational time series data. In this chapter, we study the causal inference problem more extensively and introduce and analyze RDI as a measure to infer causality from time series data.

Determining the network of interactions between random variables from observations has a long history, and with the emergence of pervasive measurement techniques in different domains, including finance, social networks, computational biology, and smart cities, it has become increasingly commonplace to observe time series of measurements. These time series are usually well modeled by random processes that interact and evolve, and a common problem of interest in many domains is the inference of the underlying network structure of interactions from the observed time series. One of the areas in which discovering these interactions is of massive interest is genomics, where people study the network of gene interactions in a cell development process. With single-cell transcriptome sequencing now routinely sampling thousands of cells, potentially providing enough data to reconstruct causal gene regulatory networks from observational data, this area is becoming more and more trendy, which intrigues us to explore new means of discovering causal relationships. We particularly focus on *Directed Information* which has been recently proposed as a generalization of Granger causality to accurately infer the structure of the network.

We start our study by observing a curious phenomenon in Section 3.2: in the limit that the relationships become purely deterministic, directed information loses power completely.

We explore its connection to Taken's delay embedding theorem, propose a remedy called *Restricted Directed Information* (RDI); and finally, demonstrate its efficacy in simulations. In particular, we show that RDI recovers the graph correctly in all instances, deterministic or stochastic, where there is enough information to recover the graph uniquely.

While RDI would recover graphs correctly if there were infinite samples at hand, its performance in finite-sample can deteriorate with more samples if the samples don't properly represent the transitional states of the system. In other words, it is more desirable to have a functional purely of the distribution of *effect* conditioned on the *cause*, rather than that of the joint distribution between cause and effect. We therefore define the *Potential Conditional Mutual Information* (qCMI) as the conditional mutual information calculated with a modified joint distribution $\mathbb{P}(Y|X,Z)\,\mathbb{Q}(X,Z)$, where $\mathbb{Q}(X,Z)$ is an assumed potential distribution. We develop a k nearest neighbor based estimator for this function similar to that of GDM in Chapter 2; employing importance sampling, and the coupling trick used in KSG estimator [134], and prove the finite-k consistency of such an estimator. We demonstrate that the estimator has excellent practical performance and show an application in dynamical system inference.

Later in this thesis, we will see the application of the methods introduced in this Chapter, in the particular context of gene transcriptomic data. Based on RDI, we develop *Scribe*, a toolkit for detecting and visualizing causal regulatory interactions between genes and explore the potential for single-cell experiments to power network reconstruction. As mentioned, Scribe employs RDI to determine causality by estimating the strength of information transferred from a potential regulator to its downstream target. The details are discussed extensively in Chapter 4.

3.1 Assumptions, Definitions and Notations

A dynamical system is a system of m random processes $\{X_i(t)\}_{1\leq i\leq m}$ for $t = 1, 2, .., T$. At each time t, $X_i(t)$ is a random variable taking values in $\mathcal{X}_i$. We denote all these variables altogether by the $m \times 1$ vector $\underline{X}(t)$. A specific realization of each random variable at time t

is denoted by $x_i(t)$ for $i \in [m]$ and the whole *state* of the system at time t is shown by $\underline{x}(t)$. We define $[m] := \{1 \leq i \leq m\}$.

The dynamical system starts evolving from an initial state $\underline{X}(0)$ which we assume is chosen randomly according to a distribution $\mathbb{P}_{\underline{X}}(0)$. The probability distribution at time t is then denoted as $\mathbb{P}_{\underline{X}}(t)$. The mechanism of the system's evolution is defined by a *vector transition function* at each time moment, i.e. for each $t > 0$, $\underline{X}(t) = g\left(\underline{X}(t-1), \underline{N}(t)\right)$ in which g is the m-by-m transition function and $\underline{N}$ is an $m \times 1$ vector of mutually-independent random variables called *noise vector at time t* which is independent of the past states of the system. Therefore the system is first-order Markov. Note that g does not depend on the time index, hence the Markov chain is time-homogeneous.

Furthermore we assume that each variable $X_i(t)$ is a function of only its corresponding noise element. In other words, if $g = [g_1, g_2, \ldots, g_m]^T$, then $X_i(t) = g_i\left(\underline{X}(t-1), N_i(t)\right)$. Note that both the transition functions g_i and the initial state distribution $\mathbb{P}_{\underline{X}}(0)$ are required to fully specify the distribution of the random process. We will assume that the system is stationary, i.e., $\mathbb{P}_{\underline{X}}(t) = \mathbb{P}_{\underline{X}}(0)$ for all t.

Causal relationship and causality graph of a dynamical system:

We expressed the state of each variable X_l as $X_l(t) = g_l\left(\underline{X}(t-1), N_l(t)\right)$. But in general, g_l might be not a function of all the variables $X_l(t-1)$. In particular, it may depend only on a subset of variables, denoted by a set $\mathrm{pa}(X_l)$. So for each variable X_l we introduce a set of parent nodes denoted by $\mathrm{pa}(X_l) \subset [m]$ which influence X_l, in other words $X_l(t)$ is a function of $X_i(t-1)$ for $\forall X_i \in \mathrm{pa}(X_l)$. We say there is a causal relationship from X_i to X_l or "X_i causes X_l" if $X_i \in \mathrm{pa}(X_l)$. We can also encapsulate these causal relationships on a graph, with nodes being random processes and edges depending on the causal relationship. The directed graph $G_C = (V, E)$ is called the causality graph of a dynamic system, in which $V = [m]$ and $(i, l) \in E$ iff $X_i \in \mathrm{pa}(X_l)$.

The goal of network inference is to infer the casual graph of the system from observations of the time series $\underline{x}(0), ..., \underline{x}(T)$. In some contexts, one may observe multiple independent runs

of the random process from different initial states, which may provide additional information.

3.2 Restricted Directed Information

We first look at the Directed Information based measures for causality appropriate for the 1st-order markov systems. In a series of recent work [54, 222], methods utilizing directed information, originally proposed in communications theory literature [167], have been proposed as the appropriate extension to non-linear problems in order to infer the network correctly. The directed information between a pair of stochastic processes conditioned on the rest of the variables is given as,

$$\begin{aligned}\mathrm{DI}(X \to Y|Z) := \frac{1}{T}\sum_{t=1}^{T}\Big\{ \quad & H(Y_t|Y_{t-1},...,Y_1,Z_{t-1},...,Z_1) \\ - \quad & H(Y_t|Y_{t-1},...,Y_1,X_{t-1},...,X_1,Z_{t-1},...,Z_1)\Big\}.\end{aligned}$$

In [222], it is shown that there is a causal link from X_i to X_j if and only if $\mathrm{DI}(X_i \to X_j|X_{\{i,j\}^c}) > 0$, provided all transition probabilities are non-zero and there are no latent variables.

A particularly simple case of causal inference problems occurs when the evolution of the time series is purely deterministic; and this is indeed the scope of discussion in this section. For example, consider a popular non-linear dynamical system, referred to as the Lorenz system [159]. It is a system with 3 time series, x, y, z which evolves according to the following ordinary differential equations with parameters σ, ρ, β.

$$\dot{x}(t) = \sigma(y(t) - x(t)) \tag{3.1}$$

$$\dot{y}(t) = x(t)(\rho - z(t)) - y(t) \tag{3.2}$$

$$\dot{x}(t) = x(t)y(t) - \beta z(t) \tag{3.3}$$

This equation system is particularly interesting for some parameter regimes, for example

$\sigma = 10, \beta = \frac{8}{3}, \rho = 28$, in the sense that it has no fixed points or orbits of fixed period; rather the system exhibits a "strange" attractor inside which the system remains. One may wish to infer the system's connectivity, shown in Figure 3.1, from the observations of the time series. It is pointed out in a recent work on inferring causality in ecological systems [265] that for inferring the connectivity of such systems, Granger causality and its extensions cannot be satisfactorily employed. This is, because, in the case of dynamical systems, Taken's theorem [269] guarantees that the information about the state of a system $(x(t), y(t), z(t))$ is contained already in the lags of any one of its variables, for example, $x(t-1), x(t-2), x(t-3)$. Indeed in that case,

$$\mathrm{DI}(Y \to X) \;:=\; \frac{1}{T}\sum_{t=1}^{T}\Big(H(X_t|X_{t-1},...,X_1) - H(X_t|Y_{t-1},...,Y_1,X_{t-1},...,X_1)\Big) = 0.$$

Therefore all directed information terms may be zero, and indeed the system is not inferable using directed-information graphs. In this regime, [265] proposed some algorithms which can infer the underlying graph using lagged embeddings. These algorithms are guaranteed to return the correct answer when the system has only two interacting variables and when the system is purely deterministic. However, this method works only in the presence of deterministic relationships and cannot infer multi-variable relationships correctly. Here we try to bridge the gap between the two extremes: in the purely stochastic extreme, directed-information graphs seem to work well and in the purely deterministic setting, alternative methods seem to be needed. This problem is also interesting because as the SNR increases, i.e., the relative noise level in the system decreases, one may intuitively expect better performance but directed-information-based methods deteriorate (see Figure 3.3). The question that we study is the following: is there a universal algorithm that can recover the graph correctly in both the deterministic as well as stochastic settings?

Consider a system that is undergoing a deterministic evolution; in such a case directed information is not even well defined. In order to make this quantity well-defined, let us

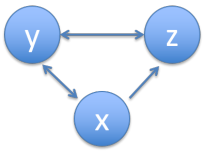


Figure 3.1: The causality graph of the Lorenz system

assume that the initial conditions are random so that there is some randomness in the system. As already discussed, directed information does not infer this graph correctly. We propose a natural metric called the *Restricted Directed Information*, which is the same as the directed information but only takes into account the past time-step.

$$\begin{aligned} RDI(X \to Y) &= H\left(Y(t)|Y(t-1)\right) - H\left(Y(t)|Y(t-1), X(t-1)\right) && (3.4) \\ &= I\left(X(t-1); Y(t)|Y(t-1)\right) && (3.5) \end{aligned}$$

Clearly, such a metric is tailored to first-order Markov relationships; and one cannot hope to infer higher order Markov relationships using such a metric (unlike directed information). However, under the first-order Markov assumption, this quantity can be estimated with far fewer samples, lending practical viability to such an estimator. In our implementations, we have adapted estimators based on the GDM estimator proposed in Chapter 2 which was in turn inspired by the KSG estimator [134], the properties of which were analyzed and generalized in Chapter 2 as well as [75, 82]. We study only stationary dynamic systems and the associated random processes X_i so that for all $t > 0$, $I\left(X(t-1), Y(t)|Y(t-1)\right)$ is unique and hence $RDI\left(X \to Y\right)$ is meaningful.

We can also define the conditional RDI similarly. The restricted directed information from X to Y given Z is defined as:

$$RDI\left(X \to Y|Z\right) = I\left(X(t-1); Y(t)|Y(t-1), Z(t-1)\right) \tag{3.6}$$

The definitions above are readily extendable to vector random processes $\underline{X}$, $\underline{Y}$ and $\underline{Z}$.

We also define an RDI graph. The directed graph $G_{RDI} = (V, E)$ is called the RDI graph of a dynamic system, in which $V = [m]$ and $(i, j) \in E$ iff $RDI\left(X_i \to X_j | \{X_k\}_{k \in [m]-\{i,j\}}\right) > 0$. We overload the notation $H(X)$ to denote the entropy of a discrete variable X or to denote the differential entropy of an absolutely continuous real-valued random variable.

If the system is ergodic, then it is possible to estimate RDI and therefore the RDI graph from a single run of the time series. In cases where the system is stationary but not ergodic, then multiple runs of the dynamical system with independent initializations are needed in order to estimate the RDI.

The question that we would like to answer is when can RDI infer the causal graph of a system correctly. It turns out that if the system indeed has non-trivial noise at all the nodes, it is easy to show that under the first order Markov assumption the metric infers the graph correctly. In this section, therefore, the focus is on the deterministic or semi-deterministic cases, where only some nodes may have noise added to them.

Our main result here is that in the deterministic case, RDI infers the graph correctly in cases where there is enough information to do so (see Theorem 3, Theorem 4 and Theorem 13). We also characterize the minimum number of "noisy" nodes required for a linear system to be inferred correctly; and show that RDI is optimal in this case as well (see Theorem 9). Throughout the rest of the section, we are interested in finding the causal relationships between the variables by using RDI. In other words, for different scenarios, we try to find conditions under which $G_C = G_{RDI}$. We study linear systems with and without noise in Subsection 3.2.1. We study deterministic non-linear systems with discrete alphabet in Subsection 3.2.2. Finally, we demonstrate the performance of RDI in simulations in Subsection 3.2.3.

3.2.1 Linear Systems

In general, a linear dynamical system is a system in which all the alphabets $\mathcal{X}_i$ are real lines, and the functions g_i are linear. Equivalently, a linear system can be described as:

$$\underline{X}(t) = A\underline{X}(t-1) + \underline{N}(t) \tag{3.7}$$

in which A is an $m \times m$ matrix and $\underline{X}(t)$ and $\underline{N}(t)$ are $m \times 1$ vectors, expressing the state of the system and the additive noise respectively. We assume that $\underline{N}(t)$ follows a Gaussian distribution $\mathcal{N}(0, \Sigma_N)$. Note that $A_{ji} \neq 0 \iff (i,j) \in E$, where E is the edge-set of G_C, the casual graph of the dynamical system.

Deterministic linear systems

A linear deterministic dynamical system is described as $\underline{X}(t) = A\underline{X}(t-1)$, i.e, the noise variance is zero. Thus one can accurately predict the system states for all $t > 0$ if the initial system state $\underline{X}(0)$ and the matrix A are known. We think of such deterministic systems initialized with a random state $\underline{X}(0) \sim \mathcal{N}(0, \Sigma_{X(0)})$. Then the system evolves as a Gaussian random process whose distribution at each time t is specified by $\mathbb{P}_{\underline{X}}(t) = \mathcal{N}(0, \Sigma_{X(t)})$.

As discussed before, the RDI is meaningful only when the system is stationary. For a general system to have a stationary distribution, it's necessary that the equation $\Sigma_{X(t)} = \Sigma_{X(t-1)}$ has a non-trivial answer. If the system is linear and deterministic with the transition matrix A, this condition is reduced to $\Sigma_X = A\Sigma_X A^T$. Furthermore, Σ_X will be identity if and only if A is orthonormal. Note that, even when A is orthonormal, there are other solutions to the equation as well, for example $\Sigma_X = 0$ is a valid solution; which stationary distribution the system will converge to depends on the initial distribution.

Now for the linear determinsitic system described above, we're looking for the conditions under which the RDI method will return the correct causality graph.

Definition 3.2.1. *A graph G_C and stationary covariance Σ_X satisfy* Property A *if for all*

edges (i, j) *in* G_C *we have*

$$\forall \underline{u} \in \mathbb{R}^m, \quad u_i \neq 0 : \quad \underline{u}^T \Sigma_X \underline{u} \neq 0. \tag{3.8}$$

Theorem 3. *Correctness of RDI for Linear Detrministic Systems.*

(a) If for a linear deterministic system property A is satisfied, then $G_{RDI} = G_C$.

(b) If Property A is not satisfied, then there will be an ambiguity in the correct system and no method will be able to return the causailty graph correctly.

Proof. Please see Section B.1. □

Linear Systems with Additive Noise

In the previous subsection, in the proof of Theorem 3 for a deterministic linear system we can see that Σ_X being P.D is sufficient and nearly necessary to have $G_C = G_{RDI}$ (See Lemma 21 in Appendix B). Now we'd like to study the effect of noise present only in the evolution of certain variables. So, the modified system model can be described as $\underline{X}(t) = A\underline{X}(t-1) + \underline{N}(t)$ in which $\underline{N}(t)$ is an $m \times 1$ additive noise vector. We note that in general, the noise vector can be any random variable, but throughout the Section 3.2.1 we assume it to be a vector of mutually-independent zero-mean Gaussian noises with arbitrary variance. For a specific node $i \in [m]$, $\mathrm{Var}(N_i) = 0$ means there is no noise injected to the node.

For such a linear dynamical system with additive noise, the stationary condition $\Sigma_{X(t)} = \Sigma_{X(t-1)}$ can be expressed as $\Sigma_X = A\Sigma_X A^T + \Sigma_N$. The answer to this equation, if it exists, is:

$$\Sigma_X = \sum_{t=0}^{\infty} A^t \Sigma_N (A^t)^T \tag{3.9}$$

Similar to the linear deterministic systems, we will show that RDI is the correct method to discover the causal relationships.

Theorem 4. *Correctness of RDI for Linear Systems with Additive Noise.*

(a) If for a linear system with independent noise, property A is satisfied, then $G_{RDI} = G_C$.

(b) If Property A is not satisfied, then there will be an ambiguity in the correct system and no method will be able to return the causailty graph correctly.

Proof. The proof of part (b) follows similar to Theorem 3. For part (a) please refer to Section B.2. □

We study a simple case where we can utilize Theorem 4 to prove that Σ_X is non-singular and hence, RDI can infer the correct graph. In other words, if we simply inject independent noises to all nodes, the we can infer the graph coorrectly:

Theorem 5. *Let* A , Σ_X *and* Σ_N *be* $m \times m$ *matrices, in such a way that* $\Sigma_X = \sum_{t=0}^{\infty} A^t \Sigma_N (A^t)^T$ *exists. If* $\Sigma_N = I$*, then* Σ_X *will be positive definite.*

Proof.

$$\forall u \neq 0 : u^T \Sigma_X u = \sum_{t=0}^{\infty} u^T A^t (A^t)^T u = \|u\|^2 + \sum_{t=1}^{\infty} \|u^T A^t\|^2 > 0 \tag{3.10}$$

$$\Rightarrow \quad \Sigma_X > 0 \tag{3.11}$$

□

Linear Systems in Semi-deterministic setting

Suppose there is independent noise injected at a group of nodes S_m, a subset of $[m]$. We want to understand when the matrix Σ_X will be positive-definite (P.D.) and RDI method will work. The next theorem states the condition under which a single-node noise injection will result in Σ_X being P.D. Then an immediate result will be stated as a generalization, for the situations when multi-node noise injection will result in a P.D Σ_X.

Theorem 6. *In Theorem 5, let's assume $\Sigma_N = e_j e_j^T$ where e_j is an arbitrary standard unit vector. So Σ_X will be positive definite if and only if the rank of the matrix $[\dots|A^{k-1}e_j|\dots|Ae_j|e_j]$ is m.*

Proof. Please refer to Section B.3. □

Corollary 6.1. *Assume that Σ_N is the covariance matrix of a generalized independent noise injected, i.e. let $\Sigma_N = \sum_{j\in S_m} r_j e_j e_j^T$ where S_m is a subset of $[m]$. Let's define $B_j = [A^{k-1}e_j|\dots|Ae_j|e_j]$. Then $\Sigma_x > 0$ if and only if the rank of the matrix $[B_{j_1}|\dots|B_{j_{|S_m|}}]$ in which $(j_1, \dots, j_{|S_m|} \in S_m)$ is m.*

Proof. Please refer to Section B.3. □

The theorem 6 states the condition for having a P.D Σ_X. We now examine whether this condition is likely to be satisfied in a randomized setup. In other words, we are interested in knowing how much probable it is to have a dynamic system with this property. For this purpose, we consider graphs whose topologies are fixed, and then the coefficients are randomly generated.

Suppose $G = (V, E)$ is a given directed graph. We generate a *random stable system* with matrix A generated in the following procedure:

$\forall i, j \in E : \hat{A}_{i,j} = Y_{ij}$ in which Y_{ij} is a continuous random variable such that $Y_{ij} \sim \mathcal{N}(0, 1)$. To make sure the system is stable and the covariance matrix of the stationary distribution $\Sigma_x = \sum_{t=0}^{\infty} A^t \Sigma_N (A^t)^T$ for Σ_N exists, we define $A = \frac{1-\epsilon}{\lambda_{max}(\hat{A})}\hat{A}$ and then A will be the transition matrix of the system, i.e. $\forall t > 0 : \underline{X}(t) = A\underline{X}(t-1) + \underline{N}(t)$.

Now we show that the *Hamiltonian* property of a graph determines whether Σ_X is non-singular. G is Hamiltonian if it has a Hamiltonian cycle, a cycle with the length of m which meets each vertex exactly once. The Theorem 9 states that for a randomly-generated linear system, if the graph of the generative model is Hamiltonian, then a single-point noise injection is sufficient for Σ_X to be positive definite with probability 1. Before that, we express two lemmas which is used in the proof of this theorem.

Lemma 7. *Let $f(X_1, X_2, \ldots, X_k)$ be a polynomial such that $deg(f)$ is limited. Suppose $\forall i : X_i \sim \mathcal{N}(0,1)$. Then $Prob\{f(X_1, X_2, \ldots, X_k) = 0\} = 0$ if there exists an assignment $X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k$ such that $f(x_1, x_2, \ldots, x_k) \neq 0$.*

Proof. This follows from standard arguments; see Lemma 2.6 in [258] for example. □

Lemma 8. *Suppose the Hamiltonian graph $G = (V, E)$ is given. Then if we assume G to be the causality graph of a linear dynamic system, there is a valid linear transition matrix A according to this graph so that:*

$$det\left([A^{m-1}e_i|\ldots|Ae_i|e_i]\right) \neq 0 \text{ for all } i \in [m] \tag{3.12}$$

Proof. Let's denote the Hamiltonian cycle of G with $\mathcal{H}(G) \subset V$. So we define A as the following:

$$A_{j,i} = \begin{cases} 1 & (i,j) \in \mathcal{H}(G) \\ 0 & (i,j) \notin \mathcal{H}(G) \end{cases} \tag{3.13}$$

So it can be seen that for all $i, j \in [m]$, there is a $k \in [m-1]$ such that $A^k e_i = e_j$. This implies that the matrix $[A^{m-1}e_i|\ldots|Ae_i|e_i]$ is a permutation of $I_{m\times m}$ and thus it will be full rank which is equivalent to $det\left([A^{m-1}e_i|\ldots|Ae_i|e_i]\right) \neq 0$. □

Theorem 9. *Let's assume that the causality graph of a linear dynamic system with additive noise is Hamiltonian. Suppose that a single-point noise injection scheme is used, i.e. $\Sigma_N = e_j e_j^T$ for some $j \in [m]$, and the matrix A is generated in the fashion described above so the system is stable. Then for all $j \in [m]$, Σ_x is positive definite with probability 1.*

Proof. Following the same steps as in 6, $\Sigma_x > 0$ if and only if the rank of the matrix $B = [\ldots|A^2e_j|Ae_j|e_j]$ is m. For the matrix B to have a rank of m it's sufficient that $B_m = [A^{m-1}e_j|\ldots|A^2e_j|Ae_j|e_j]$ be full rank, which is equivalent to proving $det(B_m) \neq 0$.

It can be seen that $det(B_m)$ is a polynomial in $A_{i,j}$ with a finite degree. Since the graph is Hamiltonian, from the Lemma 8 there is an assignment to the matrix A so that $det(B_m) \neq 0$. from the the Lemma 7, Prob $\{det(B_m) = 0\} = 0$ which yields the theorem. □

In the next theorem, we prove that if the graph $G = (V, E)$ is not *strongly connected*, i.e. there exists a pair of nodes $i, j \in [m]$ that there is no path from i to j, then the single-point noise injection is not necessarily sufficient for Σ_X to be P.D .

Theorem 10. *If the causality graph of a linear dynamic system with additive noise is not strongly connected, then there exists $j \in [m]$ such that for $\Sigma_N = e_j e_j^T$, Σ_X is not P.D .*

Proof. We know the graph is not strongly connected. So we can conclude that: $\exists i, j \in [m] \forall k \in \mathbb{N} : \left(A^k e_j\right)_i = 0$. In other words, the ith row of B is all zeros.

Let's put $\Sigma_N = e_j e_j^T$. Similar to the Theorem 9, showing Σ_X is not P.D is equivalent to showing that $rank\left(B = [\dots | A^2 e_j | A e_j | e_j]\right) < m$. From the graph not being strongly connected we concluded that the ith row of B is all zeros. Hence $rank(B) < m$. □

The Theorem 9 expresses the sufficiency of the graph being Hamiltonian for Σ_X to be positive definite when the graph is fixed and the coefficients are chosen according to a random distribution. Now, we study a model where both the graph is randomly generated and the coefficients are randomly generated given the graph, as before. The question is how much is it likely for a random graph to be Hamiltonian? This problem is well studied and here we only mention the results. Suppose that G is a random directed graph of m nodes based on the Erdos-Renyi model [57]. The graph is generated simply by considering a complete directed graph with m nodes, and including each edge in G with a probability of p, or removing it with a probability of $1 - p$. We denote such a graph with $D = (m, p)$. The theorem below states a condition for which the graph will surely have a Hamiltonian graph.

Theorem 11. [170] [69] *Let $D = (m, p)$ be an Erdos-Renyi graph. If $p \geq (1 + o(1)) \frac{logn}{n}$, then D has a directed Hamiltonian cycle with probability 1.*

Similarly, the Theorem 10 expresses the necessity of strong connectivity for sufficiency of an arbitrary noise injection to result in a P.D Σ_X. The theorem below states the condition under which the graph will almost always be not strongly connected and hence a single-node noise injection won't necessarily be sufficient.

Theorem 12. [198] *Let $D = (m, p)$ be an Erdos-Renyi graph. If $p \leq (1 - o(1)) \frac{logn}{n}$, then D is disconnected with probability 1.*

3.2.2 Non-linear Systems

In the section 3.2.1 we discussed the linear systems. Here we focus on a family of more general functions: Non-linear functions. Similar to the Section 3.2.1, we will concentrate on purely deterministic systems, as in the presence of independent noise at each node, the proofs follow similar to existing work.

In general, a deterministic non-linear dynamic system can be modified by $\underline{X}(t) = g\left(\underline{X}(t-1)\right) = \left[g_1\left(\underline{X}(t-1)\right), \ldots, g_m\left(\underline{X}(t-1)\right)\right]^T$ in which $\{g_i\}_{i\in[m]}$ are fixed deterministic functions and the alphabet $\mathcal{X}$ is finite. For this system, we would like to study the possibility of inferring the causality graph from RDI method.

The Theorem 13 determines the conditions under which the non-linear deterministic system can be inferred by RDI method.

Definition 3.2.2. Property B*: A causal graph G_c and a probability distribution $\mathbb{P}_{\underline{X}}$ is said to have property B, if, for all the edges (i, j) in the edge-set of the G_C of a non-linear deterministic system, $H\left(X_j|\{X_u\}_{u\in[m]-\{i\}}\right) > 0$ under $\mathbb{P}_{\underline{X}}$.*

Theorem 13. *Consider a deterministic non-linear system in which $\forall t > 0 : \underline{X}(t) = g\left(\underline{X}(t-1)\right)$. Let $\mathbb{P}_{\underline{X}}$ be the stationary distribution of the system.*

(a) If Property B is true, then $G_{RDI} = G_C$.

(b) If Property B is not true, then no method can return the correct G_C.

Proof. Please see Section B.4. □

While Property B proves the Theorem 13, The following lemma will introduce conditions under which the property B is satisfied. First, we define two concepts which we'll use throughout the proof of the lemma.

Definition 3.2.3. *A distribution $P(X)$ with $\{X_1, X_2, \ldots, X_n\}$ is said to have a deterministic component X_i if $\exists g : X_i = g(\{X_j\}_{j\in[m]-\{i\}})$, equivalently $H(X_i|\{X_j\}_{j\in[m]-\{i\}}) = 0$.*

Definition 3.2.4. *Suppose that y is defined as a function of the $m \times 1$ vector $\underline{x}$, i.e. $y = g(\underline{x})$. For a particular x_i, y is called a fully sensitive function of x_i if:*

$$\begin{aligned} &\forall \{x_j\}_{j\in[m]-\{i\}} \forall x_i, \tilde{x_i} : x_i \neq \tilde{x_i} \iff \\ &g(\{x_j\}_{j\in[m]-\{i\}}, x_i) \neq g(\{x_j\}_{j\in[m]-\{i\}}, \tilde{x}_i) \end{aligned} \tag{3.14}$$

In other words, for all fixed $\{x_j\}_{j\in[m]-i}$, there's a bijection between y and x_i.

Lemma 14. *For a dynamic non-linear deterministic system, if for all edges (i, j) in the causality graph, X_j is a fully sensitive function of X_i, and the variable X_i is not a deterministic component for the stationary distribution, then property B is satisfied.*

Proof.

$$\begin{aligned} &H\left(X_j(t)|\{X_u(t-1)\}_{u\in[m]-\{i\}}\right) = \\ &H\left(X_i(t-1)|\{X_u(t-1)\}_{u\in[m]-\{i\}}\right) > 0 \end{aligned} \tag{3.15}$$

□

3.2.3 Simulations

We conclude our discussion over RDI with some simulation results. We show that the RDI method works well in both deterministic and stochastic setups.

We applied RDI method on the *Lorenz System*, as described earlier in Section 3.2 and compared its performance to DI method. Before describing the simulation setup in greater details, we note that there are many stationary ergodic measures on the Lorenz system; this is because the Lorenz system does have periodic orbits in addition to the strange attractor. Since we are most interested in the strange attractor, there is an invariant measure called the

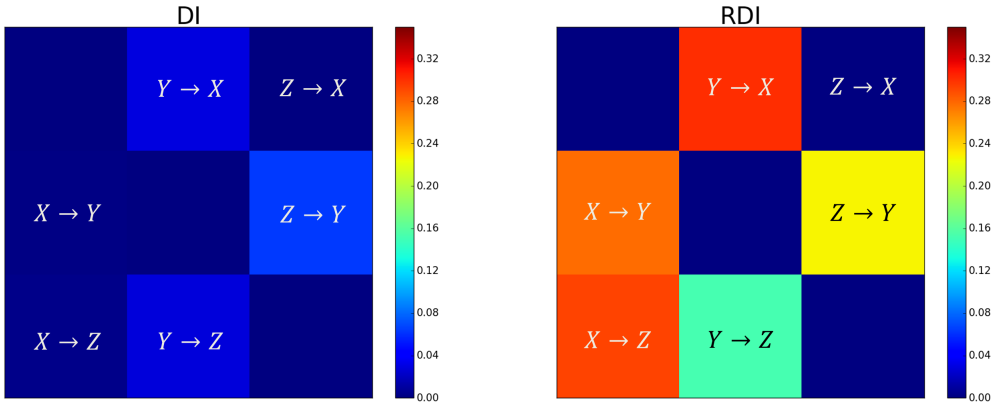


Figure 3.2: The heatmap of the directed information (DI) and restricted directed information (RDI) for Lorenz system. Each row represents an inbound node (from up to down: x, y, and z respectively) and Each column represents an outbound node (from left to right: x, y, and z respectively).

Sinai-Ruelle-Bowen measure [23, 251], which applies to the systems satisfying Property A, an example of which is the Lorenz system [279]. Most points near the attractor are attracted towards the strange attractor and result in this particular stationary ergodic measure, which is what we observed in our simulations as well. Empirically, this dataset seems to satisfy the conditions of Lemma 14 and therefore satisfies Property B; in particular, the iterations are fully sensitive to the inputs and the measure seems to have no deterministic component. Proving this formally for the SRB measure is left for future work. By Theorem 13, we expect the RDI method to work well in inferring the causal graph of the Lorenz system. We test this hypothesis with a simulation study.

The causality graph of the Lorenz System is shown in Figure 3.1. As we notice, all nodes influence each other except z to x. The standard values for the parameters of the system are $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$ as well as the initial point $\underline{x}(0) = [2, 3, 4]^T$ which are kept constant throughout the simulations. As the first observation, we simulated one run of the system with 1000000 samples and calculated the pairwise conditioned DI's and RDI's. The heatmap of the values is shown in Figure 3.2. The heatmap qualitatively shows that RDI

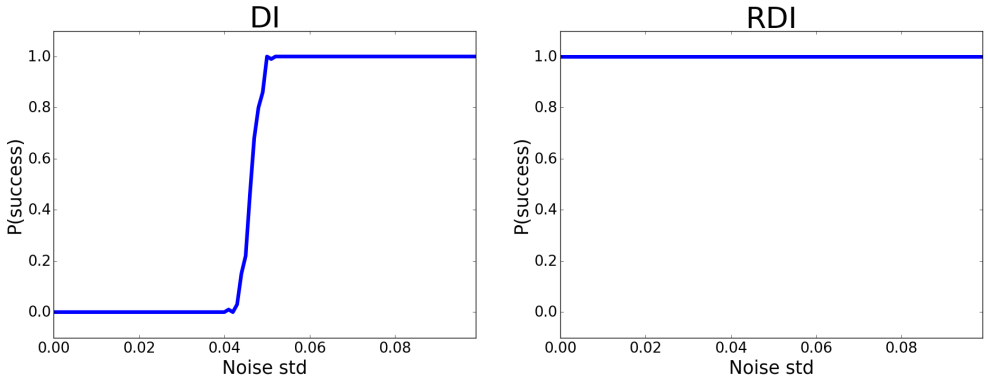


Figure 3.3: Probability of successfully extracting the right causality graph for DI and RDI methods.

returns all the corresponding edges in the causality graph of the Lorez system (G_C) correctly, but the DI method does not return the correct answer. Note that the diagonal in both methods is irrelevant (and is set to zero).

In the next simulation, we study the performance of network inference, when i.i.d. Gaussian noise is added to each variable in the system iteration. Recall that both DI and RDI are guaranteed to return the correct solution when noise is present in the system. However, this may require an infinite number of samples. Here, we fix the number of samples to 100000 and compare the reconstruction performance of the two systems in the presence of noise.

While both methods return non-zero values for all the pairs of variables, it becomes necessary to set a threshold to declare the causal graph. To calculate these zero thresholds, we calculated the values of conditioned DI and RDI for two independent variables conditioned over a third variable, all of which were Gaussian zero-mean unit-variance, and from each of which 100000 samples were generated. We repeated the calculations for 100 times and took the top 5 percentile as the zero threshold.

Then we defined a *success event* as the event of G_{DI} or G_{RDI} being the same as G_C of the Lorenz system. We changed the standard deviation of the noises from 0.001 to 0.099 with increments 0.001 and calculated the DI and RDI values and observed whether a success happened or not. For each deviation value, we repeated the simulation for 100 times and

found the number of total successes as a fraction of total trials (100) and took the resulting value as the *Probability of Success*. The results of simulations is shown in Figure 3.3.

It can be seen that for small powers of noise, the DI method doesn't show any success. As the power of the noise increases, around the deviation value of 0.05 we see that DI starts performing well. It shows that the DI method needs a specific amount of noise added at each stage to perform well, and indeed at high SNR the performance of DI method is not satisfactory, while RDI method doesn't need to rely on the noise to have a good performance and it shows a success probability of 1 for the Lorenz system.

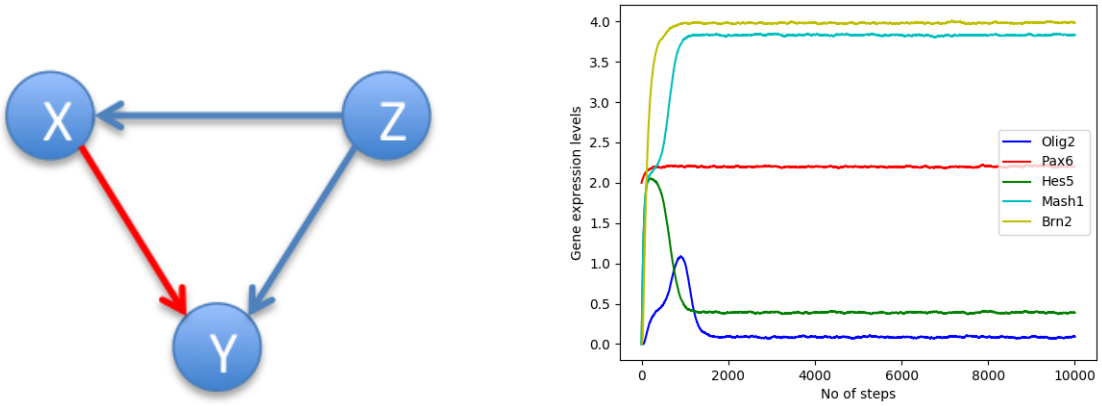
3.3 Potential Conditional Mutual Information

In Section 3.2 we studied the RDI algorithm, Which we saw was based on *Conditional Mutual Information* (CMI). Let's now take a deeper look into CMI. Given three random variables X, Y, Z, the conditional mutual information $I(X;Y|Z)$ is the expected value of the mutual information between X and Y given Z, and can be expressed as follows [43],

$$CMI_{X\leftrightarrow Y|Z}(\mathbb{P}_{X,Y,Z}) := I(X;Y|Z) = D(\mathbb{P}_{X,Y,Z}||\mathbb{P}_Z\mathbb{P}_{X|Z}\mathbb{P}_{Y|Z}). \tag{3.16}$$

Thus CMI is a function of the joint distribution $\mathbb{P}_{X,Y,Z}$. A basic property of CMI, and a key application, is that $I(X;Y|Z) = 0$ iff X is independent of Y given Z. This measure depends on the joint distribution between the three variables $\mathbb{P}_{X,Y,Z}$. There are certain circumstances where such a dependence on the entire joint distribution is not favorable, and a measure that depends purely on the conditional distribution $\mathbb{P}_{Y|X,Z}$ is more useful. This is because, in a way, conditional independence can be well defined purely in terms of the conditional distribution and the measure $\mathbb{P}_{X,Z}$ is extraneous. One case, as we will see later, is when we are testing for whether X "causes" Y conditioned over all the other variables Z in a time series setup using RDI, studied in the previous section.

This motivates the direction that we explore in this section: we define potential conditional mutual information as a function purely of $\mathbb{P}_{Y|X,Z}$ evaluated with a distribution $\mathbb{Q}_{X,Z}$ that is



(a) (b)

Figure 3.4: (a): A causal graph, where the interest is in determining the strength of X to Y. (b) A gene expression trace as a function of time for a few example genes.

fixed a-priori.

An Example: Consider the following causal graph where $X \to Y$, $Z \to X$ and $Z \to Y$ shown in Figure 3.4a. Let us say $\mathbb{P}_{Y|X,Z}$ has a strong dependence on both X and Z, say defined by the structural equation $Y = X + Z$. We would like to measure the strength of the edge $X \to Y$ in this causal graphical model. One natural measure in this context is $I(X;Y|Z)$. However, if we use $I(X;Y|Z)$ as the "strength" metric, the strength goes to zero when $Z \approx X$ and this is undesirable. In such a case, Janzing et al [117] pointed out that a better strength of causal influence is given by the following:

$$C(X \to Y) := D\left(\mathbb{P}_{X,Y,Z} \| \mathbb{P}_Z \mathbb{P}_{X|Z}(\sum_{x'} \mathbb{P}_{Y|Z,x'} \mathbb{P}_{x'})\right). \tag{3.17}$$

This causal measure satisfies certain axioms laid out in that paper and is nonzero in the aforesaid example. However, in the case that the distribution $\mathbb{P}_X$ approaches a deterministic distribution (X is approximately a constant), this measure becomes zero, irrespective of the fact that the relationship from X and Z to Y remains unaltered. We would like to

define a *potential dependence measure* that is dependent purely on $\mathbb{P}_{Y|X,Z}$ and which has no dependence on the observed $\mathbb{P}_{X,Z}$. We note that such a measure should give a (strong) non-zero result if $Y = X + Z$.

The Measure: We define potential conditional information measure as the conditional mutual information evaluated under a predefined distribution $q_{X,Z}$, and denote it as $qCMI_{X\to Y|Z}$, and express it as follows.

$$qCMI_{X\to Y|Z}(\mathbb{P}_{Y|X,Z}) := CMI_{X\leftrightarrow Y|Z}(\mathbb{Q}_{X,Z}\mathbb{P}_{Y|X,Z}). \tag{3.18}$$

A Simple Property: The main question here is how to choose $\mathbb{Q}_{X,Z}$. A simple property that may be of interest is the following, which can be easily stated in case that all three variables X, Y, Z are discrete. In such a case, it will be useful if we can have that $qCMI_{X\to Y|Z}(\mathbb{P}_{Y|X,Z}) = 0$ if and only if $\mathbb{P}_{Y|X,Z}$ depends purely only on Z. Such a property will be true for qCMI as long as $\mathbb{Q}_{X,Z}$ is non-zero for every value of X, Z. In case that all three variables are real valued, a similar statement can be asserted when $\mathbb{Q}_{X,Z}$ is a positive everywhere density, under the assumption that $\mathbb{P}_{X,Y,Z}$ induces a joint density.

Instantiations: We will propose some instantiations of the potential CMI by giving examples of the distribution $\mathbb{Q}_{X,Z}$.

- CMI: $\mathbb{Q}_{X,Z} = \mathbb{P}_{X,Z}$. Here the $q_{X,Z}$ is the factual measure and hence qCMI devolves to pure CMI.

- uCMI: $\mathbb{Q}_{X,Z} = u_{X,Z}$, where $u_{X,Z}$ is the product uniform distribution on (X, Z). This is well defined when (X, Z) is either discrete or has a joint density with a bounded support.

- nCMI: $\mathbb{Q}_{X,Z} = \mathcal{N}_{X,Z}$, where $\mathcal{N}_{X,Z}$ is the i.i.d. Gaussian distribution on (X, Z). This is well defined when X and Z are real-valued (whether scalar or vector).

- maxCMI $= \max_{\mathbb{Q}_{XZ}} \text{CMI}(\mathbb{Q}_{X,Z}\mathbb{P}_{Y|X,Z})$ is defined as an analog of the Shannon capacity in the conditional case, where we maximize the CMI over all possible distributions on (X, Z). This is akin to tuning the input distribution to maximize the signal in the graph. Note that uCMI or nCMI is not invariant to invertible functional transformations on (X, Z), whereas maxCMI is indeed invariant to such functional transformations.

- iCMI: $\mathbb{Q}_{X,Z} = \mathbb{P}_X\mathbb{P}_Z$ is the CMI evaluated not under the true joint distribution of (X, Z) but under the product distribution on (X, Z). This measure is related to the causal strength measure proposed in [117], though not identical.

Note that uCMI, nCMI, maxCMI all satisfy the property that they are zero if and only if $\mathbb{P}_{Y|X,Z}$ has no dependence on X, whereas CMI and iCMI do not.

Application in Causal Inference: As this entire work is concerned about, a key application of the potential information measures is in testing graphical models, where conditional independence tests are the basic primitive by which models are built [49, 107, 257]. To give a concrete example of the setting, which motivated us to pursue this line of study, consider the problem of causal inference, specifically in a genomics context, where we want to infer the gene regulatory network from time series data. We observe a set of m time series, $X_i(t)$ for $t = 1, 2, ...T$ with $i = 1, 2, ..m$ and wish to infer the causal graph G_C of the dynamical system. The underlying model assumption is that $\underline{X}$ is a Markov chain with $X_l(t)$ depending only on $X_i(t-1)$ for $i \in \text{pa}(X_l)$ and the goal is to determine $\text{pa}(X_l)$, the set of parents of a given node X_l. This was originally studied in the setting when the variables are jointly Gaussian and hence the dependence is linear (see [89] for the original treatment, and [84, 108] for versions with latent variables). This problem was generalized to the setting with arbitrary probability distributions and temporal dependences in [54] and studied further in [222], for one-step Markov chains in [266] and deterministic relationships in Section 3.2 of this thesis. From these works, under some technical condition, we can assert that the following method

is guaranteed to be consistent (See Section 3.2),

$$X_i \to X_j \iff I\left(X_i(t-1); X_j(t) | X_{i^c}(t-1)\right) > 0. \tag{3.19}$$

Thus to solve this problem, we estimate the CMI between the aforesaid variables. However, we observed while experimenting with gene regulatory network data (from [216]), that there is a strange phenomenon; the performance of the inference worsens as we collect more data: the number of data points *increases*.

An example of a gene expression time series for a few genes is shown in Figure 3.4b. It is clear that as the number of time points increases, the system is moving into equilibrium with very little change in gene expression values. This induces a bias on any $X_i(t)$ which then will look more and more like a deterministic distribution.

In such a case, an information measure such as CMI which depends on the "input" distribution $\mathbb{P}_{\underline{X}(t-1)}$ will converge to zero and thus its performance will deteriorate as the number of samples increases, due to this so-called induced bias in the sample distribution. However, a measure that depends on the conditional distribution $\mathbb{P}_{X_j(t)|\underline{X}(t-1)}$ need not deteriorate with increasing number of samples. Thus qCMI is more appropriate in this context (see Section 3.3.3 for the performance of qCMI on this problem). [1]

Related work: In the case that there is a pair of random variables X, Y, recent work [75] explored conditional dependence measures which depend only on $\mathbb{P}_{Y|X}$. Again in the two-variable case, a measure that had weak dependence on $\mathbb{P}_X$ was studied in [130]. The proposal there was to use the strong data processing constant and hypercontractivity [9, 213] to infer causal strength; this has strong relationships to information bottleneck [271]. In this paper, we extend [76] to conditional independence (rather than independence studied there). In a

[1] In Causal Inference, this a-priori distribution modification and bias removal is conceptually close to "perturbation", where the observant won't rely solely on the wild-type data collected from a system's purely natural course in time, but would instead intentionally modify the variables in the system to observe the downstream effect of each modification and thus improve the discovery of causal relationships. qCMI, however, is more of an indirect post-collection distribution balancing method and doesn't directly interfere with the distribution of the observational data in the practical fashion that perturbation would do.

related but different direction, Shannon capacity, which is a potential dependence metric, was proposed in [129] to infer causality from observational data [180].

In this section:

1. We propose *potential conditional mutual information* as a way of quantifying conditional independence, but depending only on the conditional distribution $\mathbb{P}_{Y|X,Z}$.

2. We propose *new estimators* in the real-valued case that combines ideas from importance sampling, a coupling trick and k-nearest neighbors estimation to estimate potential CMI similar to that of KSG method [134] also used in Chapter 2.

3. We prove that the proposed estimator is *consistent* for a fixed k, which does not depend on the number of samples N.

4. We demonstrate by simulation studies that the proposed estimator has excellent performance when there are a finite number of samples, as well as an application in gene network inference, where we show that qCMI can solve the non-monotonicity problem.

3.3.1 Estimator

To estimate the qCMI, we follow the coupling trick used in KSG estimator [134] as well as the GDM estimator discussed in Chapter 2. Specifically, this trick was applied in the context of conditional mutual information estimation in [67]. However, we note that this estimator of CMI does not have proof of consistency to the best of our knowledge. As we know, the CMI estimator essentially fixes a k for the (X, Y, Z) vector and calculates for each sample (x_i, y_i, z_i), the distance $\rho_{k,i}$ to the k-th nearest neighbor. The estimator fixes this $\rho_{k,i}$ as the distance and calculates the number of nearest neighbors within $\rho_{k,i}$ in the Z, (X, Z) and (Y, Z) dimensions as $n_{z,i}, n_{xz,i}, n_{yz,i}$ respectively. The CMI estimator is then given by,

$$\hat{CMI}_{X\leftrightarrow Y|Z} := \frac{1}{N}\sum_{i=1}^{N}\left(\psi(k) - \log(n_{xz,i}) - \log(n_{yz,i}) + \log(n_{z,i})\right) + \log\left(\frac{c_{d_x+d_z}c_{d_y+d_z}}{c_{d_x+d_y+d_z}c_{d_z}}\right) \quad (3.20)$$

where c_d terms are the volume of the unit ball in the d-dimensional space for the given norm. In the case that ℓ_∞-norm is used, all c_d terms will be equal to 1 and hence the term $\log\left(\frac{c_{d_x+d_z}c_{d_y+d_z}}{c_{d_x+d_y+d_z}c_{d_z}}\right)$ will be equal to 0.

qCMI estimator

Here, we adapt the aforementioned estimator to calculate the qCMI for a given potential distribution $q_{X,Z}$. The major difference is the utilization of an importance sampling estimator to get the importance of each sample i estimated as follows,

$$\omega_i := \frac{q_{XZ}(x_i, z_i)}{\hat{f}_{XZ}(x_i, z_i)}. \tag{3.21}$$

However, importance sampling based re-weighting alone is insufficient to handle qCMI estimation, since there is a logarithm term that depends on the density also. We handle this effect by appropriately re-weighting the number of nearest neighbors for the (y, z) and z terms carefully using the importance sampling estimators. The estimation algorithm is described in detail in Algorithm 2.

3.3.2 Properties

Our main technical result is the consistency of the proposed potential conditional mutual information estimator. This proof requires combining several elements from importance sampling, and accounting for the correlation induced by the coupling trick, in addition to handling the fact that the k is fixed and hence introduces a bias into the estimation.

Assumption 2. *We make the following assumptions.*

a) $\int f_{XYZ}(x,y,z)\left(\log f_{XYZ}(x,y,z)\right)^2 dxdydz < \infty$.

b) All the probability density functions (PDF) are absolutely integrable, i.e. for all $A, B \subset \{X, Y, Z\}$, $\int |f_{A|B}(a|b)|da < \infty$ *and* $\int |f^q_{A|B}(a|b)|da < \infty$.

Input: Data Samples (x_i, y_i, z_i) for $i = 1, \ldots, N$ and $q_{X,Z}$
Output: $qC\hat{M}I$ an estimate of $qCMI$

Step 1: Calculate weights ω_i
for $i = 1, \ldots, N$ **do**
    Estimate $\hat{f}_{XZ}(x_i, z_i)$ using a Kernel density estimator [51, 249].
    $\omega_i := \frac{q_{XZ}(x_i,z_i)}{\hat{f}_{XZ}(x_i,z_i)}$, the importance sampling estimate of sample i.
end for
Step 2: Calculate information samples I_i
for $i = 1, \ldots, N$ **do**
    $\rho_{k,i} :=$ Distance of k-th nearest neighbor of (x_i, y_i, z_i).
    $n_{xz,i} := \sum_{j \neq i : \|(x_i,z_i)-(x_j,z_j)\| < \rho_{k,i}} 1$, the number of neighbors of (x_i, z_i) within distance $\rho_{k,i}$.
    $n_{yz,i} := \sum_{j \neq i : \|(y_i,z_i)-(y_j,z_j)\| < \rho_{k,i}}^{N} \omega_j$, the *weighted* number of neighbors of (y_i, z_i) within distance $\rho_{k,i}$.
    $n_{z,i} := \sum_{j \neq i : \|z_i - z_j\| < \rho_{k,i}} \omega_i$, the *weighted* number of neighbors of z_i within distance $\rho_{k,i}$.
    $I_i := \psi(k) - \log(n_{xz,i}) - \log(n_{yz,i}) + \log(n_{z,i})$.
end for
Return $qC\hat{M}I_{X \to Y|Z} := \frac{1}{N} \sum_{i=1}^{N} \omega_i I_i + \log\left(\frac{c_{d_x+d_z} c_{d_y+d_z}}{c_{d_x+d_y+d_z} c_{d_z}}\right)$.

Algorithm 2: qCMI algorithm

c) There exists a finite constant C such that the hessian matrices of f_{XYZ} and f^q_{XYZ} exist and it's true that $\max\{\|h(f_{XYZ})\|_2, \|h(f^q_{XYZ})\|_2\} < C$ almost everywhere.

d) All the PDFs are upper-bounded, i.e. there exists a positive constant C' such that for all $A, B \subset \{X, Y, Z\}$, $f_{A|B} < C'$ and $f^q_{A|B} < C'$ almost everywhere.

e) f_{XZ} is upper and lower-bounded, i.e. there exist positive constants $C1$ and $C2$ such that $C_1 q_{XZ}(x, z) < f_{XZ}(x, z) < C_2 q_{XZ}(x, z)$ almost everywhere.

f) There bandwidth h_N of kernel density estimator is chosen as $h_N = \frac{1}{2} N^{-1/(2d_x+2d_z+3)}$.

g) The k for the KNN estimator is chosen satisfying $k > \max\{\frac{d_z}{d_x+d_y}, \frac{d_x+d_y}{d_z}, \frac{d_x+d_z}{d_y}\}$

Theorem 15. *Under the Assumption 2, the qCMI estimator expressed in Algorithm 2 converges to the true value qCMI.*

$$q\hat{CMI}_{X\to Y|Z} \xrightarrow{p} qCMI_{X\to Y|Z} \tag{3.22}$$

Proof. Please see Section B.5 for the proof. □

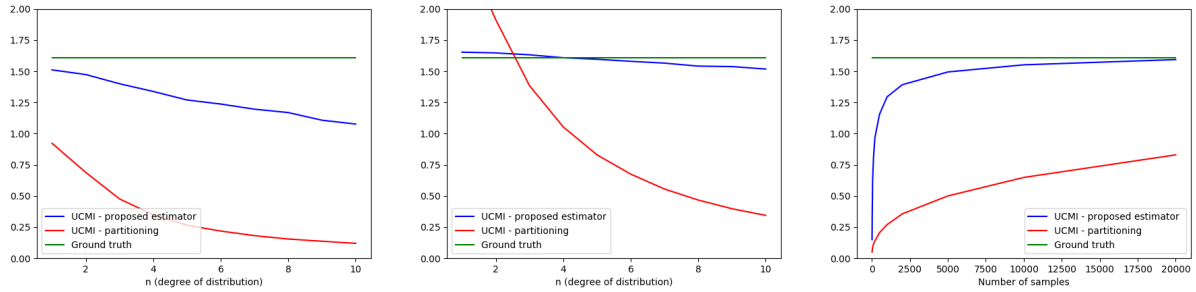
3.3.3 Simulation study

In this section, we describe some simulated experiments we did to test the qCMI algorithm. The reader should notice that all the tests we have done are taking $\mathbb{Q}_{XZ}$ as $\mathcal{U}_{XZ}$, i.e. all the tests are done for the special case of uCMI. So by exploiting the qCMI notations, we mean uCMI everywhere.

qCMI consistency

The first numerical experiment we do is to test the consistency of our qCMI estimation algorithm. We set up a system of three variables X, Y, and Z. The variables X and Z are independently taken from $\mathcal{U}^n(0,1)$ distribution, i.e. X and Z are taken from a uniform distribution and then raised to a power of n. When $n=1$, the variables X and Z are already uniform. When n is large, the $\mathcal{U}^n(0,1)$ distribution skews more towards 0. For simplicity, we apply identical n for both X and Z here. Then Y is generated as $Y=(X+Z+W)$ mod 1 in which the noise term W is sampled from $\mathcal{U}(0,0.2)$. From elementary information theory calculation, we can deduce that $I(X;Y|Z)=\log\left(\frac{1}{.2}\right)=1.609$ if $n=1$. Thus $qCMI_{X\to Y|Z}=I^q(X;Y|Z)=1.609$ for all n. We plot the estimated value against the ground truth.

As the first part of the experiment, we keep the number of samples constant at 1000 and 20000, and change the degree n from 1 to 10. We compare the results of our KSG-based method with the simple partitioning method and the theoretical value of qCMI. For the partitioning method, the number of partitions at each dimension is determined by $\sqrt[3]{N/100}$,



(a) (b) (c)

Figure 3.5: The qCMI values calculated for $u^n(0,1)$ distributions for X and Z, and uniform noise: (a) N=1000 samples and (b) N =20000 samples. (c) Degree $n = 5$.

so that we observe on average 100 samples inside each quantization bin.

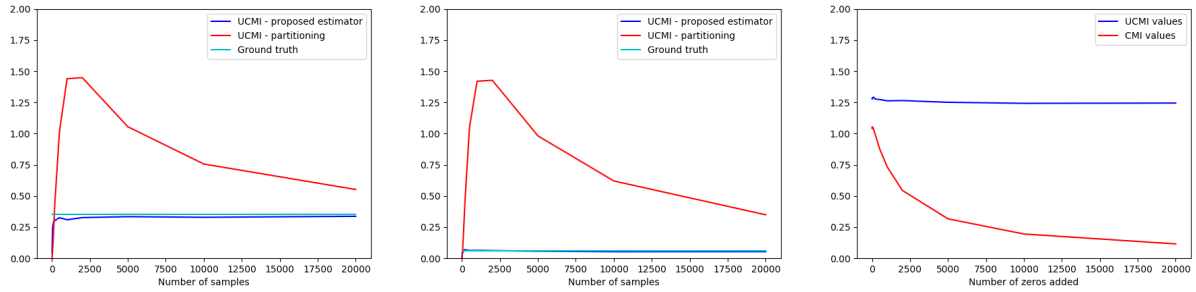
The results are shown in Figure 3.5a and Figure 3.5b. Our expectation is that qCMI remains constant as n (degree of distribution) changes. We see that with a relatively high number of samples, the accuracy of the proposed qCMI is satisfactorily high.

In the second part of the experiment, we do the same experiment as the first part, but this time we keep $n = 5$ and change the number of samples. The result is shown in Figure 3.5c. We can see the convergence of the KSG-based qCMI estimator to the true value and how it outperforms the partitioning-based qCMI method.

In the third part of the experiment, we repeated the process for the first part but replaced the $\mathcal{U}^n(0,1)$ distributions with $\beta(1.5, 1.5)$ and the noise distribution with $N(0, \sigma^2)$ and repeated the experiment for $\sigma = 0.3, 1.0$. For this part, we kept the number of partitions at 25 for each dimension. The results of calculated qCMI values are shown in Figure 3.6a and Figure 3.6b.

Dealing with discrete components

As we discussed before, the qCMI algorithm replaces the observed distribution $\mathbb{P}_{XZ}$ distribution with a distribution $\mathbb{Q}_{XZ}$. This property comes in handy when we want to remove the bias



(a) (b) (c)

Figure 3.6: The qCMI values calculated for a system with beta distribution for X and Z and Gaussian additive noise: (a) $\sigma = 0.3$, and (b) $\sigma = 1.0$. (c) The qCMI and CMI values versus the number of zeros added.

caused by repeated samples. For example, as discussed earlier, suppose that we want to measure the mutual information of two coupled variables in a dynamical system evolving through time. Such systems usually start from an initial state, go through a transient state and eventually reach a steady state. If one takes samples of the system's state at a constant rate to study the interaction of two variables, they might end up taking too many samples from the initial and steady states while the transient phase which usually happens in a relatively short time might be more informative. The conditional mutual information is not able to deal with this undesirable bias caused by the initial and steady states, while qCMI inherently deals with the effect by compensating for the samples which are less likely to happen.

To better observe the effect, we repeat the first experiment of the previous section, but this time we generate 1000 samples from the scenario, and then add zeros to the X, Y, and Z to create a high probability of occurrence at $(0, 0, 0)$. The proof of consistency of the estimator holds only when there is a joint density, i.e., the joint measure is absolutely continuous with respect to the Lebesgue measure, and hence does not directly apply to this case. We refer to Chapter 2 of this thesis and [80] for an analysis of a similar coupled KNN estimator for mutual information in the discrete-continuous mixture case.

Changing the number of zero points added from 0 to 20000, we apply the conditional MI and qCMI to the data generated and compare the results. As we can see in figure 3.6c, with the number of zeros increasing, the value of conditional MI falls down to zero, unable to capture the inter-dependence of X and Y given Z, while qCMI value remains unchanged, properly discovering the inter-dependence from the transient values.

Non-linear Neuron Cells' Development Process

In this section, we apply the RDI and uRDI algorithms to the dynamical system of neuron cells' development process simulated based on a model from [216] (A brief model of this system can be found in STAR Methods section in [219], Equation 1). It describes the evolution of 13 genes through the development process. The non-linear equations governing the development process approximate a continuous development process, in which $\dot{\underline{x}}(t) = g(\underline{x}(t-1))$. In other words, $\underline{x}(t) = \underline{x}(t-1) + dt.g\left(\underline{x}(t-1)\right) + \underline{n}(t)$ in which $\underline{n}$ are independent Gaussian noises $n_i \sim N(0, \sigma^2)$.

For this system, we want to infer the true network of causal inferences. We first apply the RDI algorithm (Section 3.2) to extract the *pairwise unconditioned causality* between the variables by calculating $I\left(X_i(t-1), X_j(t) | X_j(t-1)\right)$. Then we apply the *uRDI* algorithm. uRDI is basically RDI, except that the conditional mutual information $I(X;Y|Z)$ in RDI is replaced with qCMI as $I^q(X;Y|Z)$, where $\mathbb{Q}_{X,Z}$ is set to be uniform.

This system is a good example for when the genes undergo a rather short transient state compared to the initial and steady states, and hence we expect an improvement in the performance of causal inference by applying uRDI (see Figure 3.4b for an example run of the system). The details of the system model and analysis are given in [216].

We simulated the system for discretization $dt = 0.1$ and $\sigma = .001$ and changed the number of steps until which the system continues developing, and then applied the RDI and uRDI algorithms to evaluate the performance of each of the algorithms in terms of the area-under-the-ROC-Curve (AUC). The results are shown in Figure 3.7a. As we can see, with the number of steps increasing (which implies the number of samples captured in the

steady state is increased), the uRDI algorithm outperforms RDI. In another test scenario, we fixed the number of steps at 200, but concatenated several runs of the same process. The results and the improvement of performance by uRDI can be seen in the Figure 3.7b.

Decaying Linear Dynamical System

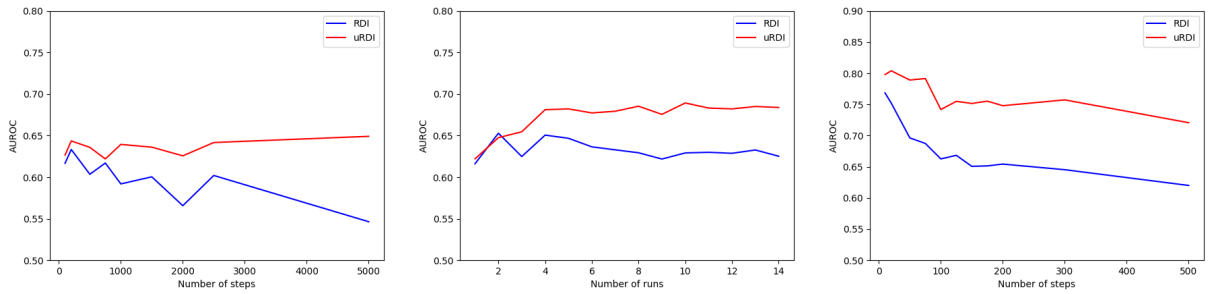
In this section, we simulate a linear decaying dynamical system. A dynamical system in the simple case of a deterministic linear system can be described as $\underline{X}(t) = A\underline{X}(t-1)$ in which A is a square matrix.

Here we simulate a system of 13 variables, all of which are initialized from a $\mathcal{U}(0.5, 2)$ distribution. The first 6 variables $(X_1, \ldots, X_6)$ evolve through a linear deterministic process in which A is a square 6×6 matrix initialized as:

$$A = \begin{bmatrix} \mathcal{U}(0.75, 1.25) & 0 & 0 & 0 & \mathcal{U}(0.75, 1.25) & 0 \\ \mathcal{U}(0.75, 1.25) & \mathcal{U}(0.75, 1.25) & 0 & 0 & 0 & \mathcal{U}(0.75, 1.25) \\ 0 & \mathcal{U}(.75, 1.25) & \mathcal{U}(.75, 1.25) & 0 & 0 & 0 \\ 0 & \mathcal{U}(.75, 1.25) & 0 & \mathcal{U}(.75, 1.25) & 0 & 0 \\ 0 & 0 & \mathcal{U}(.75, 1.25) & \mathcal{U}(.75, 1.25) & \mathcal{U}(.75, 1.25) & 0 \\ \mathcal{U}(.75, 1.25) & \mathcal{U}(.75, 1.25) & 0 & 0 & 0 & \mathcal{U}(.75, 1.25) \end{bmatrix} \tag{3.23}$$

Then the matrix A is divided by $5 * \lambda_{\max}(A)$ in which $\lambda_{\max}(A)$ is the greatest eigenvalue of A. It's done to make sure that all the variables decay exponentially to 0. After initialization, the matrix A is kept constant throughout the development process, i.e. it doesn't change with time t. The other 7 variables $(X_7, \ldots, X_{13})$ are random independent Gaussian variables.

In this experiment, we simulate the system described above for various numbers of time steps, keeping the standard deviation of the Gaussian variables at $\sigma = 0.1$, and applied both RDI and uRDI algorithms to infer the true causal relationships. Then we calculate the AUC values, the results are shown in Figure 3.7c. As we can see, the uRDI algorithm outperforms RDI by a margin of 0.1 in terms of AUC.



(a) (b) (c)

Figure 3.7: AUC values for the neuron cells' development process: a) versus the number of steps. b) versus the number of runs. (c) for the decaying linear system

3.3.4 Future Directions

Quantifying causal strength: As pointed out earlier, potential conditional mutual information can be used as a metric for quantifying causal strength when the graph is a simple three-node network (shown in Figure 3.4a). However, further work is needed in order to generalize the definition to deduce the causal strength of an edge or a set of edges in an arbitrary graph, akin to the formulation in [117] and to study the relative advantages and disadvantages of such a formulation.

Discrete qCMI estimators: It has been shown in recent work that such estimators are not optimal even for determining mutual information in the discrete alphabet case [121, 281, 300]. A very interesting question is how such minimax-rate optimal estimators can be developed in the potential measures problem.

maxCMI estimation: While we have developed efficient estimators for qCMI, in maxCMI, there is a further maximization over potential distributions q, which leads to some interesting interactions between estimation and optimization. Recent work has studied the estimation of Shannon capacity on continuous alphabets, however, the formulation is not convex leading to possible local minima [75]. Further work is needed in order to find provably optimal estimators for maxCMI in the continuous case.

Other conditional measures: Recent work [130] has used strong data processing constants as a way for quantifying dependence between two variables, with relationships to information bottleneck. These measures depend *partially* on the factual measure p_X, and are implicitly regularized. One direction of future work is to develop multi-variable versions of such estimators to estimate the strength of conditional independence, for example.

Ensemble estimation: Another approach exploiting k-nearest-neighbors for mutual information is the so-called ensemble estimation approach, where estimators for different k are combined together to get a stronger estimator, with fast convergence [181]. An interesting direction of research is to obtain ensemble estimators for potential measures.

Chapter 4

SCRIBE: INFERRING CAUSAL GENE REGULATORY NETWORKS FROM COUPLED SINGLE-CELL EXPRESSION DYNAMICS

In Chapter 3 we introduced Restricted Directed Information (RDI) and the RDI with uniform prior distribution (uRDI) for measuring causality in a dynamical system. In this chapter, we turn our focus to one of the motives for studying and developing such tools: We study the application of the RDI and uRDI methods in gene regulatory network inference using single-cell transcriptomic data.[1]

Most biological processes, either in development or disease progression [62,68,139,166,173], are governed by complex gene regulatory networks. In the past few decades, numerous algorithms for inferring networks from observational gene expression data [62, 68, 139, 166, 173] have been developed. Inferring a network of regulatory interactions between genes is challenging for two main reasons:

- The first challenge is that adding even a handful of genes to a network inference analysis requires that an algorithm consider many additional interactions between them (Figure 4.1A). Each of these potential regulatory interactions must be accepted or rejected on the basis of data. If a network that includes a particular gene regulatory interaction does not statistically "explain" the observed data substantially better than the network that excludes it, the interaction should be rejected. Deciding whether to include an interaction in a network is especially difficult because adding interactions risks overfitting to a particular dataset. Ultimately, the number of edges explodes as the number of

[1]The main work with extended experiments and results is published in [219]. The code is available at: https://github.com/aristoteleo/Scribe

genes grows, and so does the algorithms' demand for input data.

- A second challenge in regulatory network inference is distinguishing upstream regulatory genes from their direct downstream targets. Most methods that aim to do so are predicated on the notion that changes in regulators should precede changes in their targets in time (Figure 4.1B) [15]. Granger causality (GC) [89] is a statistical hypothesis test for determining whether one time series (X_1) is useful in forecasting another(X_2), which has been applied to infer biological networks [309]. However, GC assumes a linear relationship between the regulator and the target, which is violated in many biological settings [103]. Convergent Cross Mapping (CCM) [265], a more recent technique based on state-space reconstruction [269] can detect pairwise non-linear interactions. However, this method is limited to deterministic systems and thus may be poorly suited for many cellular processes (e.g., cell differentiation), which are inherently stochastic.

Single-cell RNA sequencing (scRNA-seq) experiments are attractive for gene regulatory network inference for two reasons. First, scRNA-seq experiments now routinely produce thousands of independent measurements, which may open the door to sufficiently-powered inference [153]. Second, algorithms that order the cells along "trajectories" that describe development or disease progress offer a tremendously high "pseudo-temporal" view of gene expression kinetics [94, 218, 243, 275]. The recently introduced SCENIC method [2] combines GENIE3 [113] with regulatory binding motif enrichment to simultaneously cluster cells and infer regulatory networks. Other studies have inferred regulatory networks from scRNA-seq data using differential equations [169, 192], information measures [31], bayesian network analysis [238], boolean network methods [96] or linear regression techniques [113, 200, 293]. However, most methods don't explicitly leverage time series data to identify causal interactions, and more importantly, most fail to recover the correct network even in simple settings [14, 64].

Here, we introduce *Scribe*, a scalable toolkit for inferring causal regulatory networks which relies on Restricted Directed Information (RDI). In contrast to GC and CCM, Scribe learns both linear and non-linear causality in deterministic and stochastic systems. It

also incorporates rigorous procedures to alleviate the sampling bias and builds upon novel estimators and regularization techniques to facilitate inference of large-scale causal networks. We apply Scribe to a variety of different types of real and simulated single-cell gene expression data, including single-cell RNA-seq and live imaging. In concordance with the theory, we demonstrate that Scribe has superior performance compared to existing methods when the observations consist of true time-series data. However, current scRNAseq protocols do not follow the same cells over time, breaking temporal coupling between measurements. We demonstrate that there is a dramatic drop in performance in causal network accuracy when the temporal coupling is lost. We then demonstrate that "RNA velocity", a recently developed analytic technique for scRNA-seq analysis, restores temporal coupling and improves causal regulatory network inference. Our results suggest that preserving this coupling should be a major objective of the next generation of single-cell measurement technologies.

4.1 Causal inference with Scribe: an RDI-based algorithm

Causal inference in Scribe is based on RDI. The method we implemented basically tries to calculate the RDI value for each pair of genes (i, j) conditioned over the top L genes (default is 0 or no conditioning and 1 for cases where we used conditioning) which are candidates of being regulators of the gene j.

To reach this goal, it first calculates all the pairwise unconditioned RDI values, for all the potential delays specified by the user in vector d (by default, it is a vector including 5, 10, 20, 25). Note that for the RNA-velocity dataset, since we assume the time delays Δt for the current and predicted future RNA expression level are constant across the cell and genes, there is no need to scan for a window of potential time delays. Then for each pair (i, j), it treats the delay corresponding to the largest RDI value as the "true" delay of effect, i.e. the actual time delay by which the effect of i appears in j. Having identified the "true" delays, the method then re-calculates the pairwise RDI values for each pair of genes (i, j), this time conditioned over the top L (L can be specified by the user) genes with the highest incoming RDI values to j associated with their corresponding true delays, treating

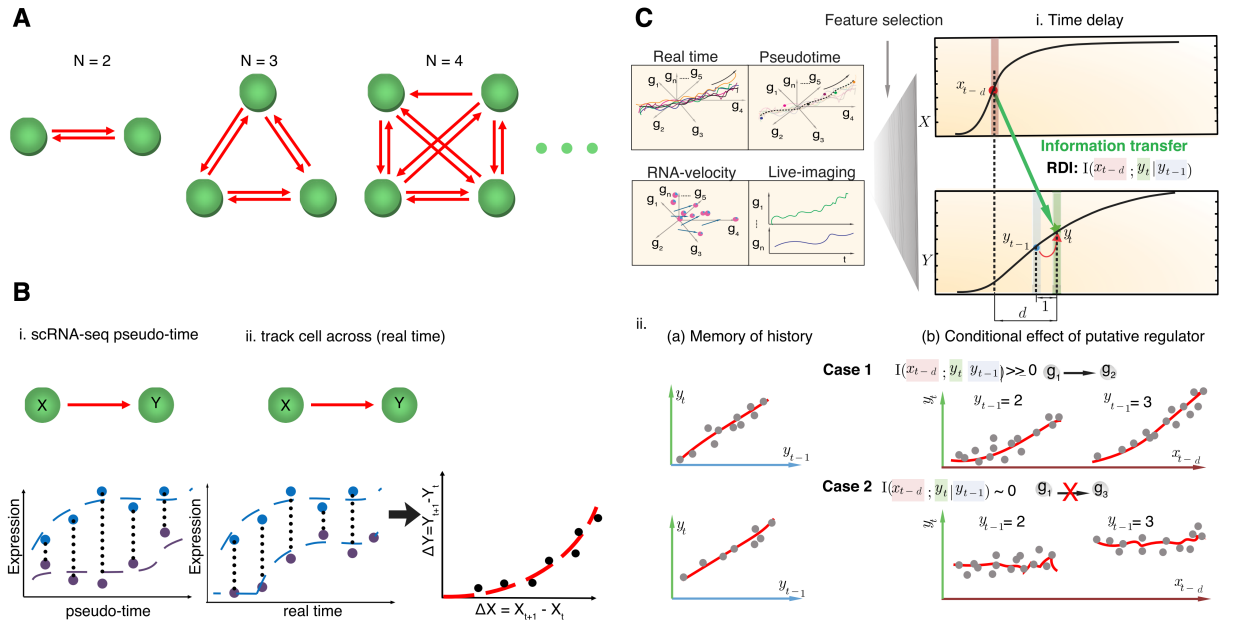


Figure 4.1: Scribe, a Toolkit for Inferring and Visualizing Causal Regulations. (A) Inferring regulatory networks from gene expression data is challenging because the number of regulatory interactions that must be evaluated grows much more quickly than the number of genes in the analysis. (B) Ordering single-cell data in "pseudo-time" or tracking how fluctuations in a regulatory network are followed by changes in a putative target in the same individual cells could boost the power to detect causal regulatory interactions. Here, vertical dash lines indicate paired gene expression from the same cell, and horizontal dash lines indicate smoothed gene expression across cells from different pseudo-time points (left) or across real-time from the same cell (middle). The right panel represents the relationship of X/Y's gene expression difference between any two consecutive pseudo-time or real-time points across cells or from the same cell. (C) Scribe detects causality from four types of single-cell measurement ("pseudo-time", "live image", "RNA velocity", and "real-time") datasets with the metric, restricted directed information (RDI). Scribe relies on RDI [224] to quantify the information transferred from the potential regulator to the target under some time delay while conditioned over its past on this pseudo-time-series data (See Section 3.2). A gene often has a strong memory of its intermediate previous state Y_{t-1}, but RDI will only give a highly positive causality score from the putative regulator to its target in cases where there is still a strong relationship between the regulator's history and the target's present condition on the target's history (case 1 versus case 2).

.

Input: gene expression time series (either based on pseudotime series, "RNA-velocity" or live imaging data, among others) X_i for each gene i

Output: A matrix of pairwise causality scores

Unconditioned RDI:
for each pair of genes (i, j) **do**
  for all delays $\delta \in d$ **do**
    Calculate $\mathrm{RDI}_\delta\left(X_i \to X_j\right) := I\left(X_i(t-\delta); X_j(t) | X_j(t-1)\right)$
  end for
  Set $\delta_{i,j}^{\max} := \mathrm{argmax}_{\delta \in d} \mathrm{RDI}_\delta\left(X_i \to X_j\right)$
end for
Finding the $L+1$ highest regulators:
for each gene j **do**
  Sort $\mathrm{RDI}_{\delta_{i,j}^{\max}}\left(X_i \to X_j\right)$ for all i's descendingly
  Take the $L+1$ nodes i with the highest incoming RDI values to j and store them in a set called $\mathrm{inc}_j^{\max}$. Store their corresponding delays $\delta_{i,j}^{\max}$ in a set called $d_j^{\max}$.
end for
Calculate conditional RDI:
for each pair of genes (i, j) **do**
  If $i \in \mathrm{inc}_j^{\max}$, remove i from $\mathrm{inc}_j^{\max}$. Otherwise, remove the $L+1$-th node from it.
  Output $\mathrm{RDI}_{\delta_{i,j}^{\max}}\left(X_i \to X_j \middle| \left\{X_l(t-\delta_{l,j}^{\max})\right\}_{l \in \mathrm{inc}_j^{\max}}\right)$
end for

Algorithm 3: Causal Inference in **Scribe**

them as the potential regulators of j. The algorithm of causal inference in Scribe is shown in Algorithm 3.

Parameters of RDI

The parameters of Scribe can be summarized as follows:

- d (Vector of positive integers - Default: 5, 20, 40): The vector of potential delays, for which the corresponding RDI values are calculated. Setting this argument too small may limit the ability of Scribe to detect causal relationships, while setting it too large can result in the discovery of incorrect or indirect causal relationships, resulting in false delays and conditioning.

- L (Non-negative integer - Default: 0): The number of the top incoming node(s) to the

target, excluding the source, over which RDI is conditioned. $L = 0$ corresponds to no conditioning (Plain pair-wise RDI). Any $L > 0$ corresponds to conditional RDI (cRDI). Conditioning over more nodes approaches the theoretical prerequisite of conditioning over all genes, excluding the source and target, needed for inferring the true causal network. However, it imposes more computational burden and undesirably reduces the accuracy of the RDI estimator with a fixed number of samples N, as it exponentially increases the dimension of the state space used to calculate the k-nearest neighbors.

- k (Positive integer - Default: 5): Number of the nearest neighbors in the kNN estimator for the conditional mutual information. The parameter should be set in such a way that the neighborhood captures an adequate number of samples for a good estimate of the probability corresponding to each sample.

- $Uniformization$ (Boolean - Default: False): If True, uRDI instead of RDI will be used. While imposing a higher computational burden over the same data than RDI, uRDI is expected to improve the causal inference in the cases with highly-biased sampling distributions.

4.2 Results

In this section, we extensively demonstrate and discuss the results of applying the Scribe algorithm to a variety of experimental settings and their corresponding data for the purpose of inferring *causality*, as the strength of information transferred from one variable, a potential regulator, to another time-delayed response variable, a potential target, where a higher score implies stronger evidence for a causal interaction and vice versa.

Scribe aims to be agnostic to the particular measurement technology used in an experiment, simply requiring as input time-ordered gene expression profiles for each cell as they progress along a time axis. It estimates causality scores using RDI for pairs of genes which are in turn used to build a causal regulatory network (Fig 4.1A, B). Scribe can calculate RDI in several different ways, each of which is designed to address challenges posed by single-cell

expression data. In order to alleviate sampling biases, for example, key transitory states are under-represented while stationary states are over-represented, Scribe can calculate two adjusted RDI scores: termed uniformization of (conditional) mutual information (uRDI or ucRDI, respectively) to quantify the potential causality as discussed in Chapter 3 (See Section 3.3). These scores capture how much influence a regulator can potentially exert on a target without cognizance of the regulator's distribution.

From pairwise RDI scores, Scribe assembles and refines a regulatory network between genes. However, many causal interactions could be indirect, and although Scribe could remove them by computing RDI between each pair of genes conditional on other potential regulators, this greatly increases the required number of samples and its running time and is typically impractical. Scribe therefore first refines the inferred network using regularization based on Context Likelihood of Relatedness (CLR regularization) [62] and can be further sparsified with an optional method for directed graph regularization on large networks (see STAR Methods in [219] for more details). Finally, Scribe offers the user a variety of ways to visualize and plot the resulting network or explore individual regulatory interactions in more detail.

4.2.1 Causal network inference of C. elegans' early embryogenesis with live-imaging

In order to assess the performance of Scribe, we examined Caenorhabditis elegans' early embryogenesis, where live imaging has been used to measure nearly half of all Transcription Factors (TFs) protein expression dynamics in every single cell in an embryo [189]. This dataset consists of 265 time series each of which tracks the expression dynamics of a TF using fluorescent reporter constructs. Measurements were collected at 1-min intervals in every cell of the developing embryo for the first 350 min of embryogenesis (Figure 4.2A).

We tested whether Scribe was able to learn validated genetic interactions that govern worm development. For example, it is understood that in the intestinal cell lineage *Ealap* the TFs *end-1* and *end-3* were up-regulated prior to their targets *elt-2* and *elt-7* (Figure 4.2B and well before most other up-regulated factors in this lineage (Figure 4.2C) [298]. We ran Scribe on these four genes to determine whether it could correctly infer the causal regulatory

interactions between them.

Although Scribe captured some known causal interactions among the core TFs that specify this lineage [196], it also reported both false positive and false negative interactions based on previously curated networks [196, 298]. For example, Scribe reports that *end-1* also strongly regulates *end-3*, which is not supported by previous studies [196, 298] (Figure 4.2D). Although accurate network inference surely depends on many factors, we hypothesize that the false positives and false negatives in the live imaging analysis (Fig 4.2D) were mainly due to the inability of the system to report measurements for more than one gene in the same individual cell. Because in the live-imaging dataset we are using, the expression values come from cells in different embryos, each of which has a single gene wired to a GFP reporter. That is, the experiment did not report two genes (i.e. a regulator and a potential target) in the same cell. So while the values of a single gene are coupled across time, values for two genes are not coupled, because they are never recorded in the same cell. Therefore, fluctuations in expression levels of a regulator are not "coupled" to corresponding fluctuations in its targets.

The entire *Ealap* lineage-specific network of C. elegans early embryogenesis constructed by Scribe is shown in Figures 4.2E–4.2G; zoomed-in versions of each network state is available in the Supplemental Information and Scribe's GitHub repository. Overall, Scribe was able to accurately infer known regulatory hierarchy (Figure 4.2F) [189].

Additional insights provided by Scribe: We find that Scribe can potentially provide new insights into gene regulation by revealing temporal/lineage/spatial specific gene regulations. For example, focusing on one cell lineage, the intestine cell *Ealap*, we found that known regulators of *E* lineage, *end-1*, *end-3*, *ref-1*, *elt-2* and *elt-7* sequentially appear in our cell maintenance network ordered by their known onset timing [189]. Interestingly, this analysis also suggests a putative regulation of *ref-1* by *end-1* which is further causally regulated by *aly-1* (a factor thought to be related to mRNA binding and export [66]) in the *Ea* cell maintenance network. In general, we find that both the maintenance and the commitment networks become increasingly complex as cells divide along the lineage tree. As each cell or cell division is associated with a particular time period and space dimension, the causal

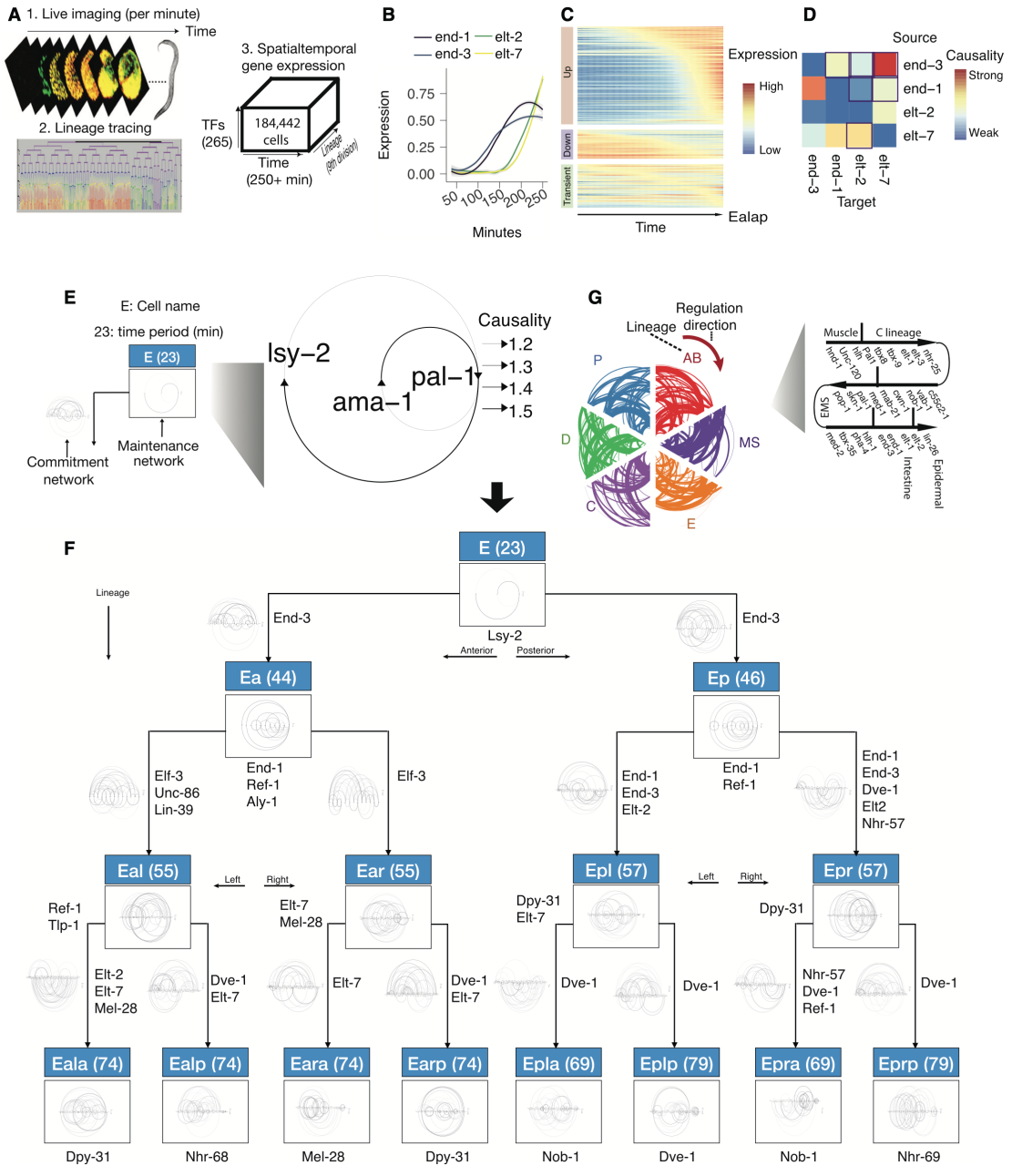


Figure 4.2: Live Imaging Dataset of C. elegans Early Embryogenesis Captures Transcription Expression Dynamics Hierarchy. (A) Scheme used by Murray et al. for measuring transcription factor's protein expression dynamics in real-time for every cell during early C. elegans embryogenesis. (B) Single-cell lineage-resolved fluorescence data capture temporal dynamics of E lineage master regulators during C. elegans embryogenesis. The expression for each gene is scaled to be between 0 and 1 and then smoothed using LOESS (locally estimated scatterplot smoothing) regression as in [209], the same as in (C).

networks recovered by Scribe recover both time and spatially-dependent gene regulation. For example, an *E2F*-like gene *elt-3* has strong putative causal interaction in both *Eal* or *Ear*'s commitment networks and gains additional causal interactions in the left axis of the worm (*Eal*) including a key neuronal TF *unc-86* and a previously identified posterior-anterior asymmetric gene *lin-39*, indicating potential novel regulation in the left-right axis of *E* lineage for the latter. This analysis demonstrates the power of Scribe in building a compendium of the causal regulatory network for early C. elegans embryogenesis and further highlights the usefulness of Scribe for casual regulatory network inference from diverse time series datasets.

4.2.2 Accurate causal network inference requires temporally coupled expression data

Next, we explored Scribe's ability to recover causal interactions using single-cell RNA-seq which in contrast to live-imaging measures many genes in each cell. We first collected publicly available datasets from several biological systems including developing airway epithelium [277], dendritic cell response to antigen stimulation [246], and myelopoiesis [193]. We then pseudo-temporally ordered these cells as previously described using Monocle 2 [218]. Next, we ran Scribe on these pseudo-time series (Fig 4.3, Supplementary Figure 3 in [219]) and examined the regulatory interactions reported for known transcriptional regulators of these systems. For each gene, we summed the causal interaction scores to all other genes, deriving a measure of its aggregate influence on the system. These aggregate causality scores were significantly higher for known transcriptional regulators than for genes believed to be target genes by the authors of the original studies ((unpaired two-sample t-test, Supplementary Figure 3 in [219]).

We next explored whether Scribe can accurately reconstruct causal regulatory networks. Recently, Olsson and colleagues suggested a core network of transcription factors for regulating myelopoiesis [193] by performing bulk ATAC-seq, ChIP-seq, perturbation experiments and profiling the transcriptomes of 382 cells from flow-sorted populations undergoing the transition (Fig 4.3A). We used Scribe to calculate causal scores for each regulator-target pair from the *Irf8* and *Gfi1* master regulators of the monocyte or granulocyte lineage, respectively, to

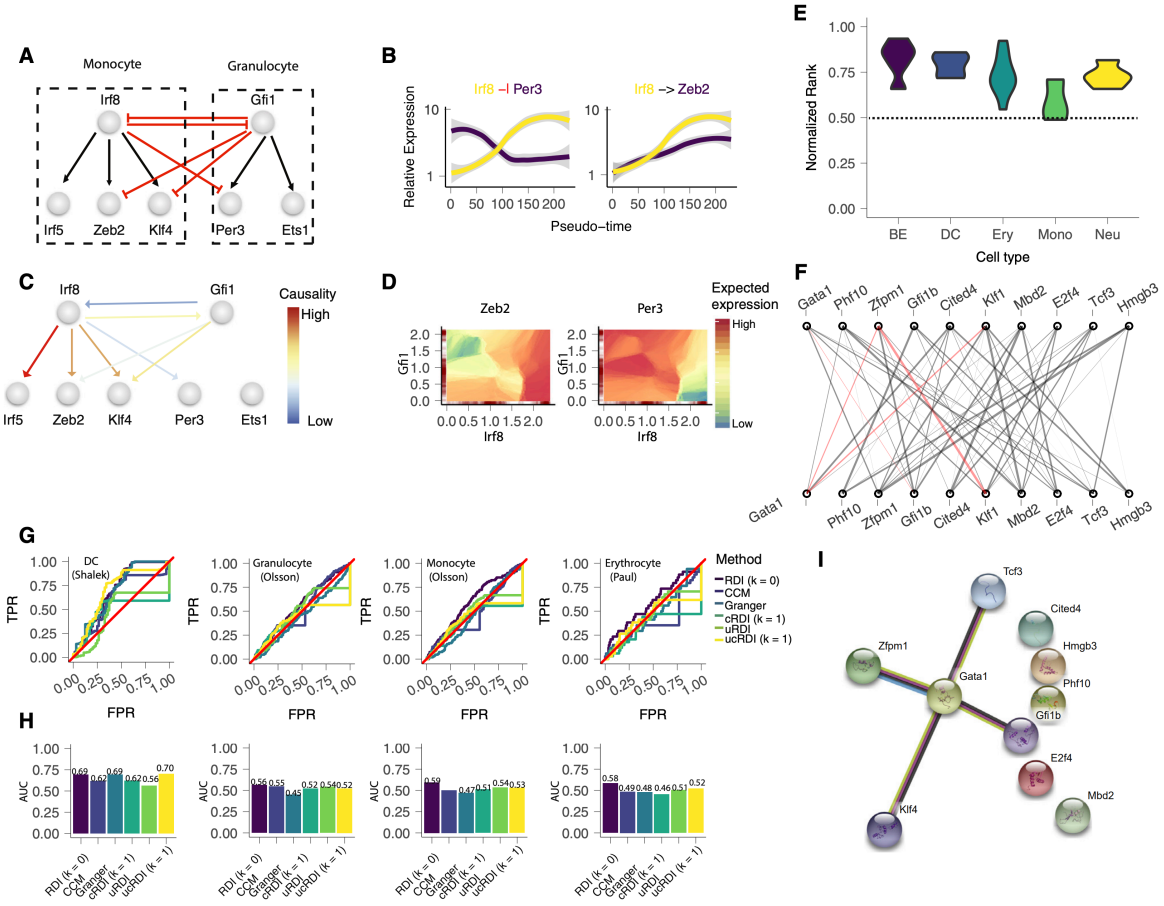


Figure 4.3: Scribe recovers a core regulatory network responsible for myelopoiesis. (A) A core network describes key regulators during the specification of monocytes and granulocytes [193]. (B) Examples of gene-target pair kinetic curves over pseudotime along the monocyte lineage. (C) Scribe infers the expected core regulatory network interactions for myelopoiesis. (D) Visualization of combinatorial gene regulation from Irf8 and Gfi1 to Zeb2 or Per3. (E) The normalized rank of lineage-specific genes' total outgoing RDI sum. (F) Lineage-specific network of significant regulators during erythropoiesis. Edges supported by the spring database are colored as red lines.

For (E) and (F), BEAM analysis was used to identify significant branching genes associated with the four (one) lineage bifurcation events shown in the hematopoietic trajectory from [218] based on the Paul dataset [201]. The top 1,000 differentially expressed genes associated with each bifurcation were chosen to build a causal network for each relevant lineage. A set of TFs relevant to specific lineages described previously is used for (E) or (F). Neu, neutrophil; Ery, erythroid; Mk, megakaryocyte; mono, monocyte; DC, dendritic cell; BE, basophil and eosinophil.

(G and H) Receiver operating curves (ROC) (G, top) and area under the curve (AUC) (H, bottom) of the inferred causal network based on Scribe, GC, and CCM, from left to right, on the dendritic cell (DC) dataset, granulocyte or monocyte branch of the Olsson dataset [193], and erythroid branch of the Paul dataset. Four different variants of causal inference implemented in Scribe are tested: RDI (L =0), the default RDI method without conditioning on any other gene; RDI (L = 1), the RDI method based on conditioning on the incoming gene with the highest causality score, except the current target; uRDI, the method based on the uniformization technique applied to the actual distribution in RDI; and uRDI (L = 1), the uRDI method but also with the conditioning on the incoming gene with the highest causality score, except the current target.

(I) The network of the gene set as included in (F) retrieved from the STRING database.

the other six genes in the core network. We hypothesized that Scribe would return strong causal scores for the targets ascribed to each regulator but not others. We observed that expression kinetics over pseudo-time correctly reflect the network architecture (Figures 4.3A and 4.3B). For example, as *Irf8* expression increases in the monocyte lineage, *Zeb2* expression also increases while expression of *Per3* decreases (Fig 4.3B), reflecting the fact that *Irf8* activates expression of *Zeb2* while inhibiting Per3's expression. We represent the causal network inferred by Scribe as a heatmap where each row corresponds to the causal score from the regulator to all other genes and the color corresponds to the magnitude of the causal score (Fig 4.3C). Scribe assigns a high causality score for all targets of *Irf8* (*Gfi1*, *Irf5*, *Klf4*, *Per3*, *Zeb2*) but the lowest causality score to *Irf8* and *Ets1*, which are not its direct targets. Similarly, Scribe assigns high causality score for the majority of *Gfi1*'s targets (*Irf8*, *Klf4*, *Per3*) even though *Gfi1* has low expression values (Fig 4.3C). The visualization of the combinatorial regulation of *Irf8* and *Gfi1* to either *Zeb2* or *Per3*, based on the Scribe visualization toolkit, captures the conflicting regulation pattern between two regulators and their two targets (Fig 4.3D).

To determine Scribe's capabilities to reconstruct transcriptome-level causal networks containing edges between transcription factors (TFs) as well as from TFs to putative downstream targets, we applied Scribe to scRNA-seq data of hematopoiesis [201]. We find that the lineage specific genes tend to have a high total outgoing RDI sum among all significant transcription factors (Fig 4.3E). When restricting to a small subset of previously identified erythropoiesis-associated TFs, we find Scribe identified several regulatory interactions, such as *Gata1*-*Gfi1*-*Klf4*, which are known to play an important role in myeolopoeisis [140, 263, 270] (See Fig 4.3F). However, in recovering known regulatory interactions in each system based on the manually curated network from the literature, Scribe marginally outperformed GC and CCM but all three methods generally performed poorly, with no method reaching an AUC of greater than 0.7 (Figures 4.3G and 4.3I).

We hypothesized that similar to live imaging datasets, a lack of coupling between the expression measurements in pseudo-temporally ordered single-cell RNA-seq data leads to

poor accuracy during regulatory network inference. In contrast to a true time series in which an individual cell is tracked and measured longitudinally, in pseudo-temporal datasets, each expression measurement comes from a different cell. Therefore, although pseudo-time reveals the overall trend of the gene expression dynamics, the real-time gene expression "micro-fluctuations" (fluctuations that happen within short time scales) of a regulator to a target are not captured in pseudo-time.

To test whether causal network inference requires temporal coupling between genes across measurements, we ran Scribe on simulated data based on a core network of neurogenesis (see STAR Methods section in [219]) collected using four strategies for obtaining longitudinal measurements from individual cells. First, we consider "real-time", an ideal theoretical technology in which all genes are tracked in each individual cell as that cell differentiates. We therefore consider a second setting "live imaging", in which each cell is tracked over time but only one gene is measured - as mentioned earlier in Figure 4.2. Third, we examine pseudo-time, where all genes are measured only once in distinct cells that have been sampled from a population undergoing differentiation. Finally, we tested Scribe on RNA velocity data, which consists of a snapshot measurement of each cell's current transcriptome along with a prediction of that same cell's expression levels at a short time in the future (Fig 4.4A).

Using pseudo-temporal measurements, Granger causality, convergent cross mapping, and Scribe all performed very poorly in recovering direct, causal interactions between genes in the hypothetical network (Fig 4.4B, Supplementary Figure 4 [219]). The inability of these methods to recover regulatory interactions is unlikely to be due to the undersampling of the system, as the performance was insensitive to varying the number of cells captured in the simulated datasets (Figures 4.4C, Supplementary Figure 4 in [219]). The performance of the three methods was only modestly better when using data captured by "live imaging".

We next evaluated two alternative modes of measuring gene expression dynamics in single cells in which fluctuations are coupled. Using conditional RDI, Scribe produced highly accurate reconstructions from "real-time" measurements of gene expression (AUC: 0.859 ± 0.0283), in which every gene is measured repeatedly in a set of cells as they differentiate. This

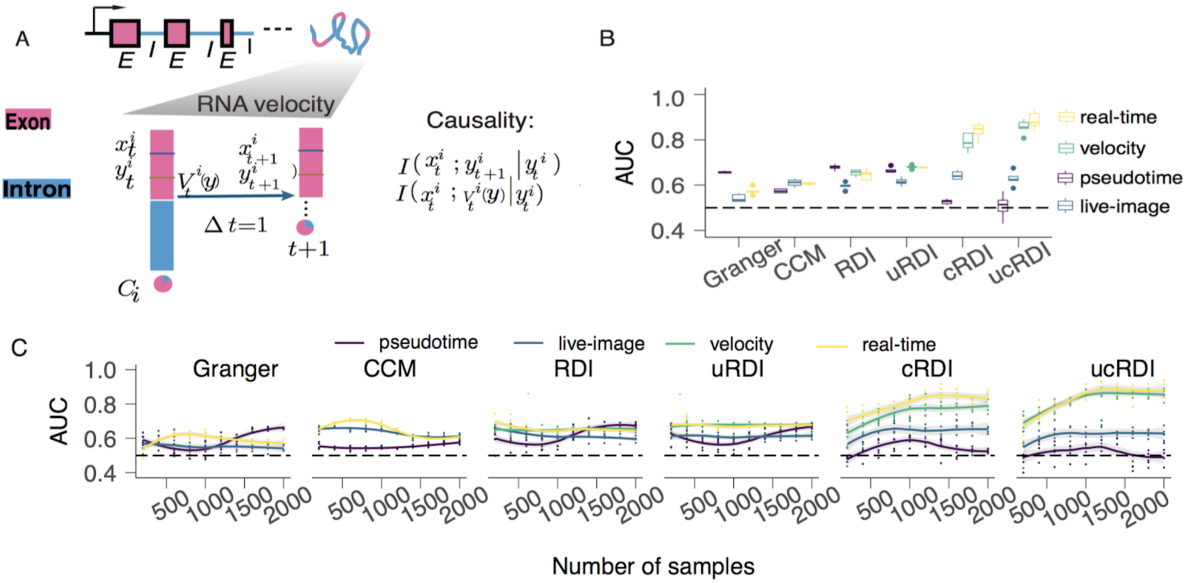


Figure 4.4: Temporally coupled gene expression measurements are necessary for inferring causal regulatory interactions. Four possible types of single-cell time series datasets, pseudotime (no dynamics coupling across genes over time points), live-imaging (dynamics coupled for each gene across time points but not across genes), "RNA-velocity" (dynamics coupled for all genes between two time points measured in each cell) and real-time (dynamics coupled for all gene across all time-points in each cell), are simulated (see Methods in [219] for details). (A) Incorporating RNA velocity analysis into Scribe for causal inference. A gene with multiple exons (pink box, E) and introns (blue line, I) is transcribed into immature RNA and then spliced into mature RNA, both of which can be quantified by scRNA-seq (See Methods in [219] section "Benchmarking Scribe with alternative algorithms on inferring causal regulatory network"). (B) Dynamics coupling in real-time and "RNA velocity" datasets enables Scribe to correctly infer causal regulatory networks. Scribe (including four variants, RDI ($k = 0$), uRDI ($k = 0$), RDI ($k = 1$) and uRDI ($k = 1$)) and two alternative causal inference methods, Granger and CCM are used to reconstruct network based on 2,000 data points from each of the four different single-cell time series datasets. The results from each method are then compared with the known network architecture to obtain the AUC score. Five replicates are performed for each method. (C) Dependence of causal inference on the number of samples. The same analysis as in A is performed but with down-sampled datasets on a sequence from 200 to 2000, incremented by 200, data points (see Methods in [219] for more details). RDI: restricted directed information. uRDI: uniform RDI which replaces the biased data distribution with a uniform distribution to remove the dependence of the sample distribution.

demonstrates that when measurements are fully coupled across time, and fluctuations in a regulator can propagate to its targets, restricted directed information correctly reveals causal regulatory interactions. Scribe also recovered accurate networks (AUC: 0.837 ± 0.0189) with "RNA velocity" measurements (Figure 4.4A). Although RNA velocity does not repeatedly measure cells, it provides a "prediction" of the future expression levels of each gene based on comparing mature to immature transcript levels, in effect introducing a form of temporal coupling to the data. These simulations show that methods for regulatory inference based on information transfer fail using data from measurement modalities in which fluctuation of a regulator's expression across cells is "uncoupled" from fluctuations in its targets.

4.2.3 Causal Network Inference with "RNA Velocity" Reveals Regulatory Interactions that Drive Chromaffin Cell Differentiation

We next sought to test whether Scribe could recover causal network interactions using real RNA velocity measurements. Recently, La Manno and colleagues applied RNA-velocity to study the chromaffin cells differentiation as well as their associated cell cycle dynamics [137]. We used this chromaffin dataset as a proof-of-principle for incorporating "RNA velocity" into Scribe. We first reconstructed a developmental trajectory from mature mRNA expression levels from each cell in this dataset and then applied BEAM [217] to identify genes that significantly bifurcate between Schwann and chromaffin cell branches (Figure 4.5B). These genes were enriched in processes related to neuron differentiation along the path from SCPs (Schwann Cell Progenitors) to mature chromaffin cells (Supplementary Figure 4E [219]).

We then applied Scribe to the RNA velocity measurements from the 3,665 significantly branch-dependent genes (qval < 0.01, Benjamini-Hochberg correction) (Figure 4.5C and Supplementary Figure 4 in [219]). We first built a network between significant branching TFs as well as from TFs to the significant targets in chromaffin lineage and found that only 0.75% of TFs interact with each other while 8.40% TFs regulate potential targets (causality score > 0.05) (Figures 4.5E-4.5G).

We then inferred a core network between fourteen TFs believed to drive chromaffin

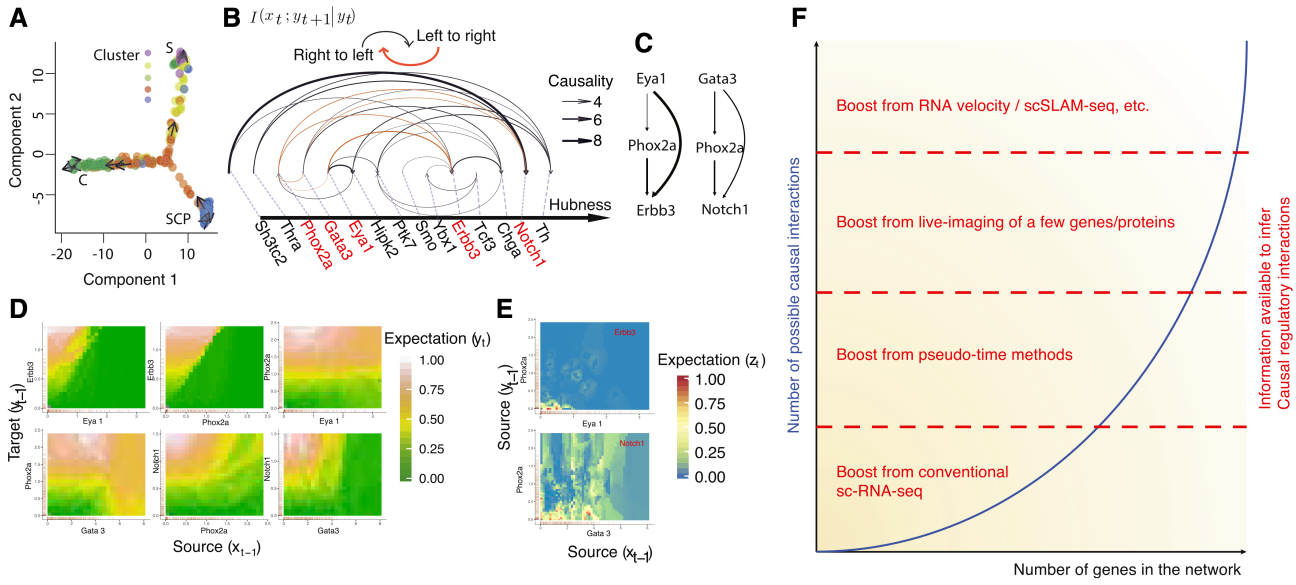


Figure 4.5: Causal inference in Scribe with RNA-velocity. (A) RNA velocity vector projected onto the first two latent dimensions. A small subset of arrows is used to visualize the velocity field of the cells. S, sympathoblasts; C, chromaffin; SCP, Schwann cell progenitor. The color of each cell corresponds to the cluster id from Figure 5B of [70]. (B) A core causal network for chromaffin cell commitment inferred based on RNA velocity. Gene set is collected from [70]. Context likelihood of relatedness (CLR) regularization is used to remove spurious causal edges in the network (see STAR Methods in [219]). (C) Two potential coherent feed-forward loop (FFL) motifs of chromaffin differentiation are discovered from the core network. Edge width corresponds to causal regulation strength. (D) Visualization of the six causal regulation pairs in the feed-forward loops of Eya1-Phox2a-Erbb3 and Gata3-Phox2a-Notch1 (see STAR Methods in [219] for details). (E) Visualizing combinatorial regulation logic for the two feed-forward loops in (C) with Scribe. For both (D) and (E), a grid with 625 cells (25 on each dimension) is used. Similarly, expected values are scaled by the maximum to obtain a range from 0 to 1. (F) Scribe's ability to detect causal regulatory interactions is limited by the single-cell measurement technology used. Technologies that provide measurements that are coupled across time and between genes provide more power for inference than conventional single-cell RNA-seq experiments

cell differentiation [70]. Within this core network, Scribe identified two feed-forward loop (FFL) motifs [3]: *Eya1-Phox2a-Erbb3* and *Gata3-Phox2a-Notch1* (Figures 4.5C-4.5E). The STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) database of genetic and molecular interactions [268] provided additional support for these regulatory motifs (Supplementary Figure 4H in [219]). From the RNA-velocity network, we also find that SCP-related TFs, such as *Sh3tc2*, tend to have stronger causal regulation (ranked higher in terms of hubness as shown in the arc plot) while chromaffin cell-related TFs, including *Chga* and *Th*, have much smaller causal regulations, reflecting the network captures transition from SCPs to chromaffin cells [70].

4.3 Conclusion and Discussion

Despite extensive research into gene regulatory network inference over the past several decades, the fundamental source of poor performance by these methods on single-cell data remains uncertain. One possibility is that even with the tremendous gains in the throughput achieved by the developers of scRNA-seq technology over the past decade [267], these methods still have not been provided with sufficient data to accurately reconstruct networks. Alternatively, the basic approach of inferring genetic interactions based on statistical interactions between their measured expression levels may be fundamentally limited.

We developed Scribe, which uses information-theoretic metrics RDI and uRDI to infer complex causal regulatory interactions between genes, overcoming limitations inherent to GC and CCM. Scribe also provides several ways to visualize causal information transfer, helping users distinguish between direct and indirect interactions and unravel combinatorial regulatory logic.

Although Scribe correctly infers causal regulatory interactions in simulated measurements that track all the genes in an individual cell over time, it performs poorly on live imaging or pseudo-temporally ordered single-cell datasets. We demonstrate that poor performance is due to the loss of temporal coupling between measurements of genes that interact, in which fluctuations in the levels of a regulator propagate to measurements of its targets. This may

explain poor performance by a broad class of information-theoretic or statistical approaches for inferring regulatory networks from scRNA-seq data. If so, then simply improving the throughput of scRNA-seq protocols will not be sufficient to power inference methods. Pseudo-temporally ordering scRNA-seq data provides a boost to the number of genes that may be considered, and the temporal coupling provided from joint measurement via live imaging of pairs of genes could boost power further.

Improvements to single-cell expression assays that produce measurements for multiple genes that are coupled across time may enable the accurate regulatory network inference possible using Scribe or similar approaches. Although methods for non-destructively tracking expression levels of many genes in single cells over time have not been described, several assays have been reported that provide snapshot estimates of both steady-state mRNA levels along with their rates of synthesis. These assays report measurements of the current and future transcriptome of individual cells, essentially providing temporal coupling over a short time horizon. For example, SLAM-seq (thiol(SH)-linked alkylation for the metabolic sequencing of RNA) [102, 188] or TUC-seq (thiouridine-to-cytidine sequencing) [233] assay mature RNA levels and estimate the rate of their synthesis via nucleotide-labeling or conversion-based approaches. Importantly, single-cell versions of those technologies [27, 58, 100] have recently been developed when this work was under review awaiting integrating Scribe with those technologies as future investigation. Sequential, multiplex RNA fluorescence in-situ hybridization (FISH) or "Seq-FISH" [244] which probes both exons and introns of RNAs can also provide similar measurements. RNA velocity, which analyzes scRNA-seq reads falling within introns and estimates both mature mRNA levels and their immature intermediates to predict the transcriptome over a short time in the future, also generates coupled measurements. Accordingly, using RNA velocity measurements greatly improves Scribe's accuracy compared to running it on pseudo-temporal scRNA-seq measurements. These assays and algorithmic improvements boost Scribe's ability to recover causal interactions because they provide increasingly comprehensive and temporally coupled measurements across the transcriptome. Concentrating efforts to improve temporal coupling in new experimental methods should, in

our view, be a priority for the field.

scRNA-seq holds great promise for powering various algorithms for network inference but as we have shown, major obstacles remain in the way of doing so in practice. Once provided with temporally coupled measurements, Scribe accurately reconstructs networks of modest scale. As experimental and computational improvements to single-cell expression techniques couple measurements across time, we expect Scribe to be increasingly capable of dissecting the complex genetic circuits that drive development and disease.

Chapter 5

LEARNING TEMPORAL CELL TRANSCRIPTOMIC DYNAMICS VIA ARTIFICIAL INTELLIGENCE

In Chapter 4 we introduced *Scribe* to reconstruct the gene regulatory networks from transcriptomic single-cell data relying on RDI and uRDI methods, and demonstrated its capabilities and discussed its limits. In this and the next chapter, we expand the scope of the problem and try to uncover the governing dynamics in cell development processes more comprehensively. To achieve this goal, we switch gears and employ the Artificial Intelligence (AI) techniques to *learn* the cell transcriptomic dynamics, i.e. our goal is to train *Neural Networks* that are able to predict each cell's future from its current state.

In this chapter, we focus on learning the temporal dynamics of how genes interact in an evolving cell where the cell is assumed to be a closed system, i.e. we won't consider any possible external input to the cell from its surroundings. More accurately, the input to our toolkit is the current state of each cell, and the output would be the future state of the same cell - or equivalently the instantaneous velocity vector of the cell's state.

The dynamics that govern changes in the gene expression state of a cell are assumed to be described as:

$$v(t) = \dot{X}(t) = f\left(X(t)\right) \tag{5.1}$$

which could be thought of as an *Ordinary Differential Equation* (ODE). In the equation above, $X(t)$ is the current state of the cell (Which is defined as the vector of the current expression level of all the genes in the cell), and $v(t)$ is the vector of the instantaneous changes (i.e. velocity) at time t. Our goal is to learn the function $f(.)$ from the empirical observations, which can be equivalently thought of as learning the continuous *Vectorfield* of

the gene velocity over the gene expression space.

The so-called *Transcriptomic Vectorfield Reconstruction* is a relatively new problem in genomics made possible by recent advances in single-cell RNA sequencing as well as newly developed powerful tools and methods in data science. Qiu et al developed Dynamo [220] based on the SparseVFC method [160] to learn the underlying vector field from expression-velocity sample pairs. The work is then followed by both developing algorithmic methods [71], and AI-based vector field learners using either RNA-velocity [36] or time-course data [56, 147]. These methods, however, are designed to work only in a specific setting, don't support a wide variety of data types, and are not equipped with extensive downstream analysis functionalities.

Inferring the function $f(.)$ not only provides us with the instantaneous cell velocity vectors for any given state but also enables us to predict and track the temporal changes of any cell by solving the ordinary differential equation 5.1 given the initial state $X(0)$. This solution is usually denoted by the numerical integral $X(t) = \int_0^t f(\tau, X(\tau))\, d\tau$ since solving an ODE is in fact an integration. [1]

5.1 Dynode Vectorfield: A comprehensive AI-based toolbox for learning temporal transcriptomic dynamics

In this chapter, we present *Dynode Vectorfield* [2] , an AI-based toolbox developed in PyTorch and a part of the Dynode package that is capable of inferring the function $f(.)$ in Equation 5.1 via training a neural network $f_\theta(.)$ trained in a supervised fashion over the observed transcriptomic data, where θ denotes the neural network parameters. The trained network predicts the instantaneous velocity of the system $\hat{v} = f_\theta(x)$ for any given state x as input.

Throughout the rest of the section, we briefly highlight the features of Dynode Vectorfield which are also summarized in Figure 5.1. Some detailed discussions regarding loss function choice and regularization to enforce stability and smoothness over the learned dynamics can

[1] Note that the aforementioned integral is not an ordinary integral as $X(.)$ is not an explicit invariant function of τ but it's in fact a recursive function of $f(\tau - d\tau)$. The notion of the integral is used to emphasize the notion of de-differentiation that naturally exists in solving ODE equations.

[2] The package is published here: https://github.com/aristoteleo/dynode-release

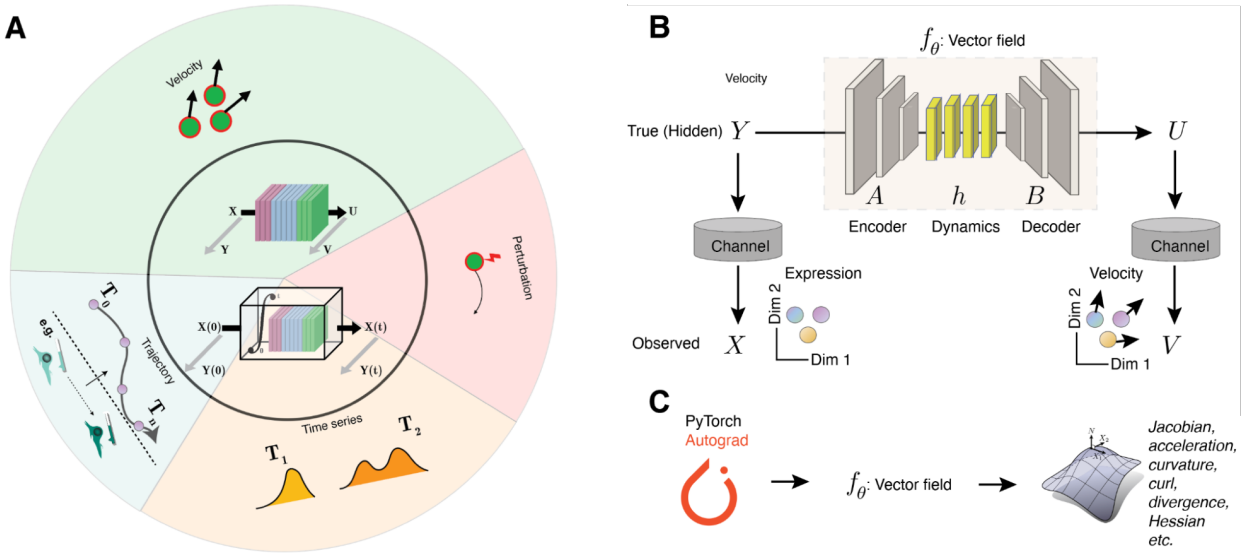


Figure 5.1: Dynode Vectorfield, a universal framework to model single-cell dynamics with differentiable vector field learning approaches. (A) Single-cell genomics data types and core neural network architectures to model them. Four major data types exist, including RNA velocity, single-cell perturbation, and single-cell trajectories with longitudinal measurement of the same cell and population time series. (B) The core neural net architecture of Dynode, including the Autoencoder, dynamics, and channel model to learn vector fields for various data types. The Autoencoder encodes and decodes the gene expression and velocity to lower dimensional space where the vector field lives. The (optional) channel model will model the observation noise and the dropouts of the single-cell data. (C) Dynode calculates differential geometry with PyTorch Autograd to make sense of the reconstructed vector field.

be found in Appendix C.

5.1.1 Dynode Vectorfield supports a wide variety of neural network models

Dynode Vectorfield is designed as a highly flexible and configurable deep-learning framework supporting a wide variety of neural network models inside the PyTorch environment. Even though the default model implemented in the toolbox is a Dense Neural Network (DNN), the user may use any standard neural net module in PyTorch such as ResNet [99], Recurrent models such as RNN, or Convolutional Neural Networks (CNNs) [149] as they see fit, or even implement their own proprietary model with a compatible *Application Programming*

Interface(API) to that of Dynode.

Periodic Sinusoidal Activation: The implemented DNN network is capable of supporting all the common activation functions such as *ReLU*, *Sigmoid*, and *Tanh*. While all these activations work relatively well in the task of learning the large-scale non-linear mapping f_θ between the input and the output, they almost fully fail to capture the finer small-scale trends. One could observe and reason that any higher-order derivative of the neural network (e.g. the Hessian matrix) will be equal to 0 for almost any input point if any activation with linear local behavior (such as ReLU or Leaky-ReLU) is employed, where the first-order derivative of DNN's with respect to input is almost surely constant at any given input value. To alleviate this issue, we also support periodic sinusoidal activation functions as introduced in [255] where the networks with such activation functions are dubbed as *Sinusoidal Representation Networks* or *Sirens*. Sirens should be able to capture any order of derivatives for the dynamics at any given input point. The drawback is that the periodicity parameter ω_0 needs to be set properly to avoid exerting false frequencies and wobbles in the learned dynamics. The detailed discussion can be found in [255].

5.1.2 Dynode Vectorfield supports multiple types of datasets and multi-modal training

The Dynode Vectorfield toolbox is designed in such a way that it can adapt to various types of datasets it's provided with and select the proper method of training accordingly. To be more specific, it can be trained in *Regression mode* (For pairwise expression-velocity observations), or *Neural ODE mode* [33] (for wild-type time-course expression data), or both of them simultaneously if the corresponding data is provided. Furthermore, Dynode can be trained with *perturbation time-series* and *uncoupled (population) time-series* data, where the feed-forward path still would mainly be calculated via Neural ODE, with some key differences compared to the training procedure of wild-type time-course data (Figure 5.1A).

- **RNA-Velocity Regression:** This is the most basic mode of training when the gene expression and RNA-velocity data are jointly available for each cell in the form of

$(x^{(i)}, v^{(i)})$ (i.e. the input-output pair is provided for each data sample i). In this case, the network simply fits the output to the given RNA-velocity values minimizing the appropriate loss function.

- **Coupled time-course expression:** In this mode, it's assumed the gene expression values of a single cell are provided over a time period. So the data provided would look like a non-uniformly sampled time series over a period of time in the form of $\big(x^{(i)}(0), x^{(i)}(t_1), ..., x^{(i)}(t_N)\big)$. The Neural ODE would therefore try to fit the future predicted expressions to that of the given observations starting from the initial expression $x^{(i)}(0)$ for each sample i.

- **Population time-course expression:** In the aforementioned time-course experimental data, tracking the gene expressions of a single cell over time is not always possible. What is provided in the most advanced datasets consists of a few instances of the state of a system (e.g. a group of cells in a tissue sequenced together) at each time point t_i, where each two different time point t_i and t_j encompass independent (i.e. uncoupled) group of observations. In other words, we're provided with a *population of cells* at each time point.

  Even though the lack of coupling between the temporal samples usually proves costly as we lose a great deal of information on how individual points would evolve over time, one may partially infer the dynamics by tracking the temporal changes in the *distribution* of the population at different time points, i.e. one may ask "will the temporal changes in how the observations $X(t)$ are scattered over the state space provide any information about how the system evolves through time?". Therefore in such cases, instead of minimizing an ordinary pairwise loss such as Mean-Squared-Error, one would try to match the distribution of predictions in the form of $\mathbb{P}_{\hat{Y}}$ with $\mathbb{P}_Y$. Therefore, the loss function would be defined in the form of a *divergence* between the two distributions. In the Dynode Vectorfield toolbox, we include two of the most popular of such measures, Wasserstein Distance and Kullback-Leibler Divergence which can be used as the loss

functions for such uncoupled observations. [3] The details on how these losses are implemented can be found in Section C.1.

- **Single-cell Perturbation:** As discussed extensively in [207] (See Sections 1.4.1 and 1.4.2 for example), and as we'll see in Section 5.2, intervening with the natural order of phenomena and observing the outcomes which would remain unobserved otherwise, is of much interest in inferring the interactions of the variables in a dynamical system. Intentionally Perturbing an input variable in a dynamic system and observing its subsequent effect(s) provides otherwise non-available crucial information on the internal dynamics of the system and helps any inference platform uncover the true dynamics more accurately. The Dynode Vectorfield toolbox is capable of training and testing on such datasets, where any gene perturbed, either fully knocked out or fixed at a non-zero certain level, can be marked at the input which so its values forced to be constant (i.e. the corresponding RNA-velocity is rendered zero), and the downstream effect is observed at the output accordingly, at both training and inference phases.

  The importance of such data is increasingly appreciated and therefore conducting perturbation experiments and collecting data is becoming more commonplace. This so-called *Gene Perturbation* approach has been gaining high attention recently among researchers. As a remarkable breakthrough, Dixit et al [52] recently designed and carried out such an experiment and collected data. They developed *Perturb-seq*, combining single-cell RNA sequencing (RNA-seq) and Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR)-based perturbations to perform many such assays in a pool. This data, however, is only collected at the steady state of the cells and does not include time series. These experiments, overall, are still relatively expensive and challenging to design, with a limited number and pattern of perturbations possible per each cell, hence the collected data is not as widespread as regular wild-type scRNA-seq data.

[3] In most of the use cases, Wasserstein Distance is recommended over KL Divergence.

5.1.3 Dynode Vectorfield can learn the dynamics in a low-dimensional space

Dynode is able to learn the low-dimensional dynamics, assuming the cell dynamics lie on a low-dimensional manifold. [4]

If the low-dimensional manifold's dimension is set to be smaller than the original space, Dynode will initialize two additional neural network modules to map the high-dimensional space to the low-dimensional one and vice versa, respectively called *encoder* and *decoder*. Dynode then first transforms the input x to a low-dimensional vector via the encoder, which then passes through the low-dimensional dynamics neural net, the output of which is then transformed back to the original space via the decoder, rendering the velocity vector v. Therefore, the total forward pass consists of all three blocks, and the training is carried out on all of them simultaneously.

Since the mapping from the higher dimension to the lower dimension naturally needs to be invertible, we require the encoder and decoder block to be the inverse of each other, hence the two blocks make an *auto-encoder* together. At each iteration, therefore, we perform a training step over the auto-encoder (i.e. encoder+decoder) for state x and velocity v samples.

5.1.4 Dynode calculates differential geometry to make sense of the reconstructed vector field

The Dynode Vectorfield toolbox is able to calculate various derivatives of the trained neural network's output with respect to the input, to provide deeper insights into the reconstructed vectorfield. The derivatives implemented in Dynode are implemented via PyTorch's Autograd toolbox and include but are not limited to Jacobian, Hessian, curl, divergence, acceleration, and curvature (Figure 5.1C). This functionality enables us to carry out insightful analysis and reveal interesting gene-level interactions inside the learned cell dynamics. Examples of these analyses are discussed in Section 5.2.

---

[4] To be more precise, if the dynamical system is d-dimensional (Equivalently the gene expression and RNA-velocity vectors of each cell $x, v \in \mathbb{R}^d$, and the function $f : \mathbb{R}^d \to \mathbb{R}^d$) then there exists $d_l < d$ for which exists the one-to-one transform $T : \mathbb{R}^d \to \mathbb{R}^{d_l}$ and the low-dimensional dynamics $h : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ such that $T(v) = h\left(T(X)\right)$.

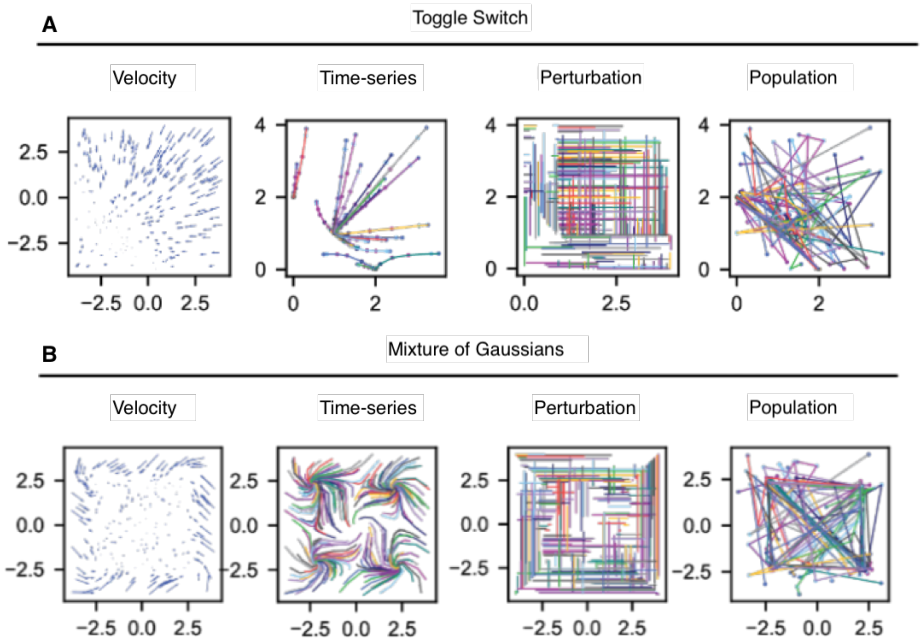


Figure 5.2: Various types of data simulated from synthetic two-gene systems (A) Toggle-switch (B) Mixture of Gaussians.
from left to right: RNA-velocity, coupled time-course expression, perturbation, and uncoupled population time-course expression.

5.2 Results

In this section, we apply the Dynode Vectorfield toolbox to a variety of synthetic and experimental datasets. We first verify the performance of Dynode Vectorfield using two synthetic systems 2-gene systems, namely *toggle-switch system* [112, 289] and *Mixture of Gaussians* [160]. We then apply the toolbox to two real datasets collected from scRNA-seq experiments carried out over *Human Hematopoiesis process* [294, 295] and *Pancreatic Endocrinogenesis* [16] and analyze the learned dynamics to gain biological insights about the aforementioned processes.

In all the tests carried out, we used a five-layer DNN network to learn the core dynamics, and anywhere the dynamics are learned in a lower-dimensional space than the original space,

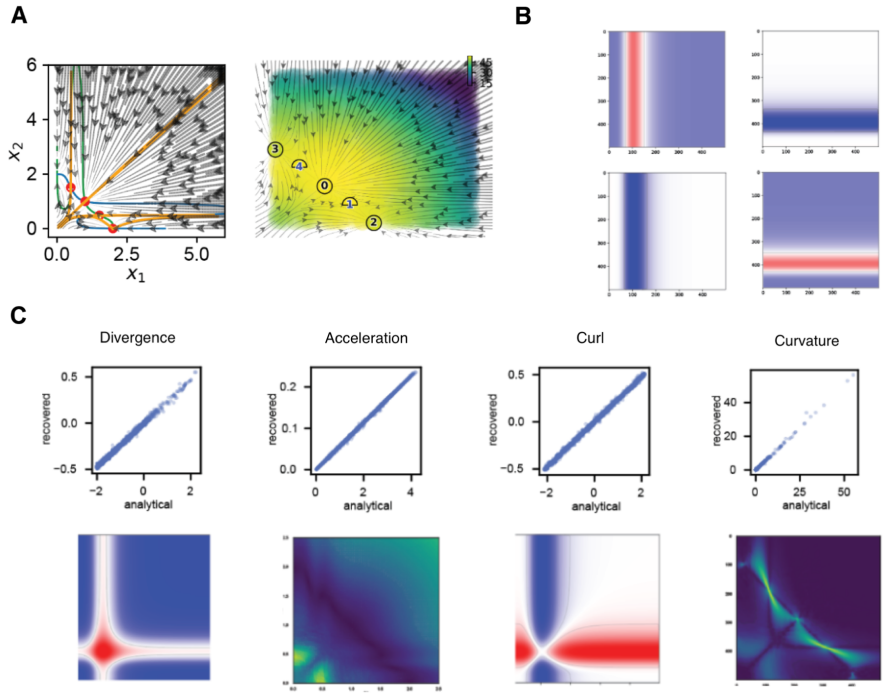


Figure 5.3: (A) Dynode accurately reconstructs the vector field and maps the fixed points of the simulated toggle switch system. (B) The heatmaps of gene-to-gene Jacobians. (C) Dynode accurately calculates divergence, acceleration, curl, and curvature. The X-axis is analytical values derived from the equation while Y-axis is the recovered values based on the learned vector field. Note that curvature and accelerations are vectors and we calculate the norm of the vectors for plotting.

the initialized encoder and decoder consist of three DNN layers. We used Sirens activation for all the tests except for the Pancreatic Endocrinogenesis where using Tanh activation helped denoise the data more effectively and yielded smoother output dynamics.

5.2.1 Dynode accurately learns synthetic dynamics modeling real gene interactions

We first apply Dynode Vectorfield on two synthetic systems and verify it correctly learning the dynamics and properly revealing the differential geometry of the learned dynamics. Dynode was tested with four different types of data mentioned in Subsection 5.1.2, as shown in Figure 5.2.

The first system tested is a two-gene toggle-switch motif [112, 289] which models a self-

activation mutual-inhibition pattern between two genes recurring in many cell differentiation processes, such as the ones we will see in Subsection 5.2.2. The ODE describing the system is defined as:

$$\begin{aligned} \dot{x}_1 &= a_1 \frac{{x_1}^n}{{K_{a_1}}^n + {x_1}^n} + b_1 \frac{{K_{b_1}}^n}{{K_{b_1}}^n + {x_2}^n} - x_1 \\ \dot{x}_2 &= a_2 \frac{{x_2}^n}{{K_{a_2}}^n + {x_2}^n} + b_2 \frac{{K_{b_2}}^n}{{K_{b_2}}^n + {x_1}^n} - x_2 \end{aligned} \tag{5.2}$$

where $a_1 = a_2 = b_1 = b_2 = 1$, $K_{a_1} = K_{a_2} = K_{b_1} = K_{b_2} = 1$, and $n = 4$. We simulated 10000 independent expression-velocity pair samples from the system, with expressions x_1 and x_2 being picked uniformly in the $(0, 4)$ range. The learned vector field, with the discovered fixed points and the differential geometry analysis, is shown in Figure 5.3. Figure 5.3A shows the accurate reconstruction of the vector field for the toggle switch data. Figure 5.3B shows the gene-to-gene Jacobian values over the two-dimensional region of interest for all four combinations, correctly reflecting the self-activation mutual-inhibition interactions. Figure 5.3C the differential analysis of the learned vector field and how accurately it's able to recover all the values versus the ground truth.

The second system we simulate and analyze is a synthetic vector field defined by a mixture of five Gaussians described in [160]. This system's ODE is defined as:

$$\begin{aligned} \dot{x}_1 &= 9x_1 - 2{x_1}^3 + 9x_2 - 2{x_2}^3 - 1 \\ \dot{x}_2 &= -11x_1 + 2{x_1}^3 + 11x_2 - 2{x_2}^3 + 1 \end{aligned} \tag{5.3}$$

The aforementioned dynamics form a repeller at the origin $(0, 0)$ and four attractors at $(\pm 2, \pm 2)$ and four saddle points at ± 2 on each axis. We used Dynode Vectorfiled to recover the dynamics from various types of data generated from Equation 5.3. We generated 10000 uniform random samples from each of the four experiment types mentioned in Subsection 5.1.2 with each sample simulated for 10 time points for time course, perturbation, and population data types.

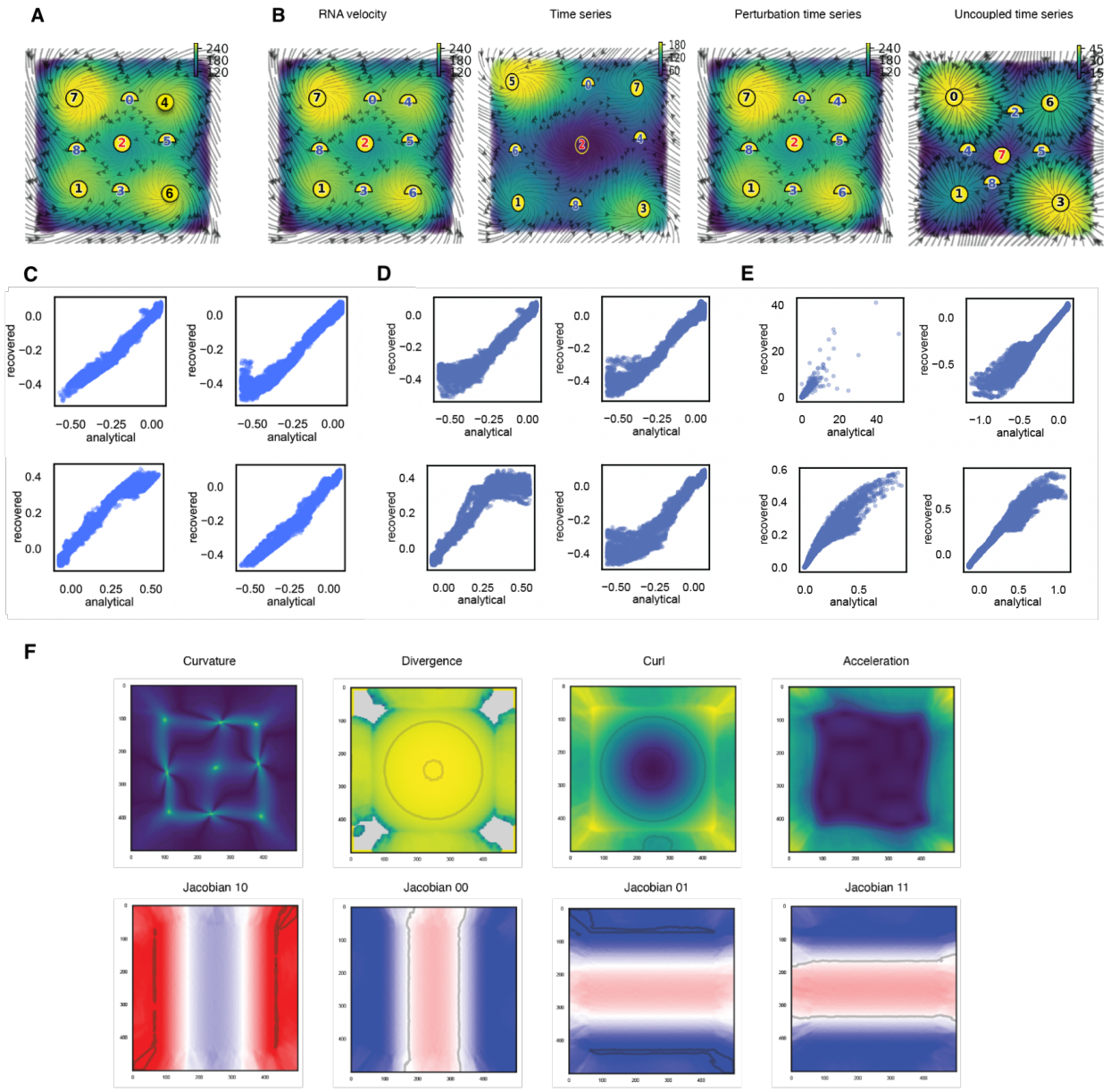


Figure 5.4: Dynode reconstructs the vector field and provides the geometry analysis for the Mixture of Gaussians from various experimental data types. (A) The stream plot of the ground truth vector field. (B) The learned vector field from various types of data: 1) RNA Velocity, 2) time course, 3) Perturbation, and 4) Population time course. (C) The learned Jacobian (y-axis) versus the ground truth Jacobian for RNA velocity data. (D) The learned Jacobian (y-axis) versus the ground truth Jacobian for time course data. (E) The learned Jacobian (y-axis) versus the ground truth Jacobian for perturbation data. (F) The detailed geometry analysis for the learned dynamics from RNA velocity.

Figures 5.4A-B visually show how accurate the reconstructed vector fields from each of the data types are, with the heatmap color representing the norm of the velocity vector at each point (bright: high, dark: low). As we can see, RNA velocity and Perturbation data provide the most accurate recovery of the dynamics, while with time series we get the shape of the curves of the vector filed correctly with the attractors being discovered at somewhat inaccurate locations. The population data, as expected, provides information regarding the overall trend of the data (i.e. the location of the repeller and the attractors) while the finer shape of the streamlines is missing from the reconstructed vector field. Figures 5.4C-E show the accuracy of the calculated Jacobian values from the learned dynamics, versus the ones calculated analytically from Equation 5.3 as scatter plots, which properly align with the accuracy of the vector field reconstruction for each data type discussed above. In Figure 5.4F various differential quantities are plotted for the model learned from the RNA velocity data, revealing the behavior of the vector field such as the location of the attractors and the repeller from divergence, or the locations with high curl and acceleration which are located close to the attractors.

5.2.2 Dynode Vectorfield recovers underlying interactions during Hematopoiesis and Pancreatic Endocrinogenesis from single-cell RNA-seq data

We applied the Dynode Vectorfield to two real datasets to learn the underlying dynamics governing the cell development process and reveal insightful gene interactions via the differential geometry analysis of the trained neural network.

The first dataset analyzed is *Human Hematopoiesis* clone tracing, the formation of blood cells in human individuals [294, 295]. The data consists of 1947 expression-velocity samples and we follow an identical preparation, processing, and analysis to that of [220] and present the results in Figure 5.5.

The underlying regulatory network of a few important genes in the process and the corresponding cell lineages are shown in Figure 5.5A where the red and blue connectors represent activation and repression relationships between the genes respectively. As we can see in

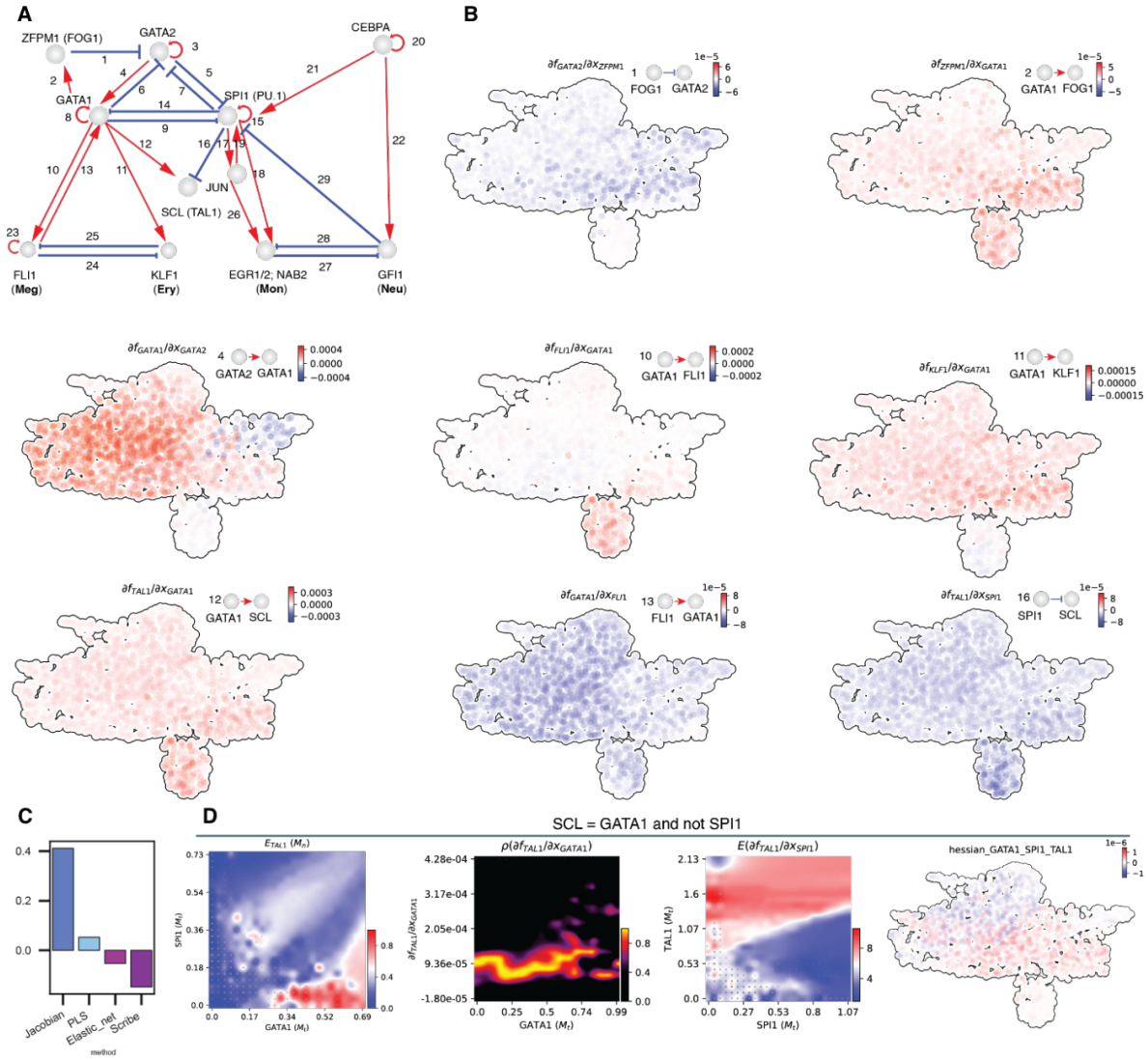


Figure 5.5: Dynode Vectorfield accurately infers the gene regulatory network for hematopoiesis and enables genetic predictions (A) The manually curated gene regulatory network of hematopoiesis. (B) Scatterplots of the RNA Jacobian for different regulatory pairs across cells. A cartoon is shown on the top left of each subpanel to reveal the gene regulations. The red arrows indicate activation while the blue blunt arrows indicate repression. Similarly, the color of the cells indicates the Jacobian values with a negative value corresponding to blue color (activation) and a positive value of red color (repression). (C) The correlation between the predicted gene regulation and the ground-truth regulation based on literature, averaged over all the cells. (D) The heatmap of expected expression of TAL1 under different levels of GATA1 (x-axis) and SPI1 (y-axis) (first); the probability of the Jacobian from GATA1 to TAL1 under different levels of GATA1 (second); the expected Jacobian from SPI1 to TAL1 under different level of SPI1 and TAL1 (third); The scatterplot of the heatmap from both GATA1 and SPI1 to TAL1 (fourth).

5.5B, Dynode Vectorfield is able to capture the majority of these regulatory relationships via the trained neural network, with the exception of for example *FLI1* to *GATA1*. To make the discoveries further robust, collecting more samples and training the neural network with data types such as perturbation is strongly suggested, as they potentially rule out spurious/incorrect links between the genes naturally arising from the correlations existing in the wild-type data (See Section 5.3). Figure 5.5C demonstrates Dynode Vectorfield's superior performance in revealing the types of gene interactions (i.e. activation or repression) versus Scribe (Chapter 4), Partial Least Squares (PLS) regression [299], and Elastic Net regression [305]. Figure 5.5D showcases Dynode's capability of revealing an example of combinatorial regulatory relationships from *GATA1* (activation) and *SPI1* (repression) to *SCL (TAL1)*.

The second dataset studied is the *Pancreatic Endocrinogenesis*, where the formation of the endocrine system of the pancreas is studied and samples are collected during the differentiation process [16]. [5] The major steps of cell development during endocrinogenesis, the corresponding vector field, and the key genes involved are shown in Figure 5.6A-B.
We then trained the neural net model in Dynode Vectorfield over the expression-velocity samples collected from around 3696 cells, in which the top 4000 significant genes were selected. We learned the dynamics in low-dimensional space of 30 dimensions and then did some in-depth analysis of the learned vector field focusing on the key genes involved in the endocrinogenesis procedure to reveal some biological insights. Figure 5.6C reveals some of the interactions between the key factors in the process. Specifically, we reveal that *Neurog3* activates *Pax4* in the progenitor early stages, which in turn would repress *Arx* during the pre-endocrine stage. Specifically, *Pax4* would activate the *Ins2* in β cells which are responsible for producing insulin, as verified by [40].
We then further studied the effect of knocking out the key genes on how the cells would evolve

[5] An extensive analysis of the dataset is conducted using Dynamo [220] here:
https://dynamo-release.readthedocs.io/en/latest/notebooks/tutorial_pancreatic_endocrinogenesis.html
In our analysis, we followed the same data processing and preparation procedure.

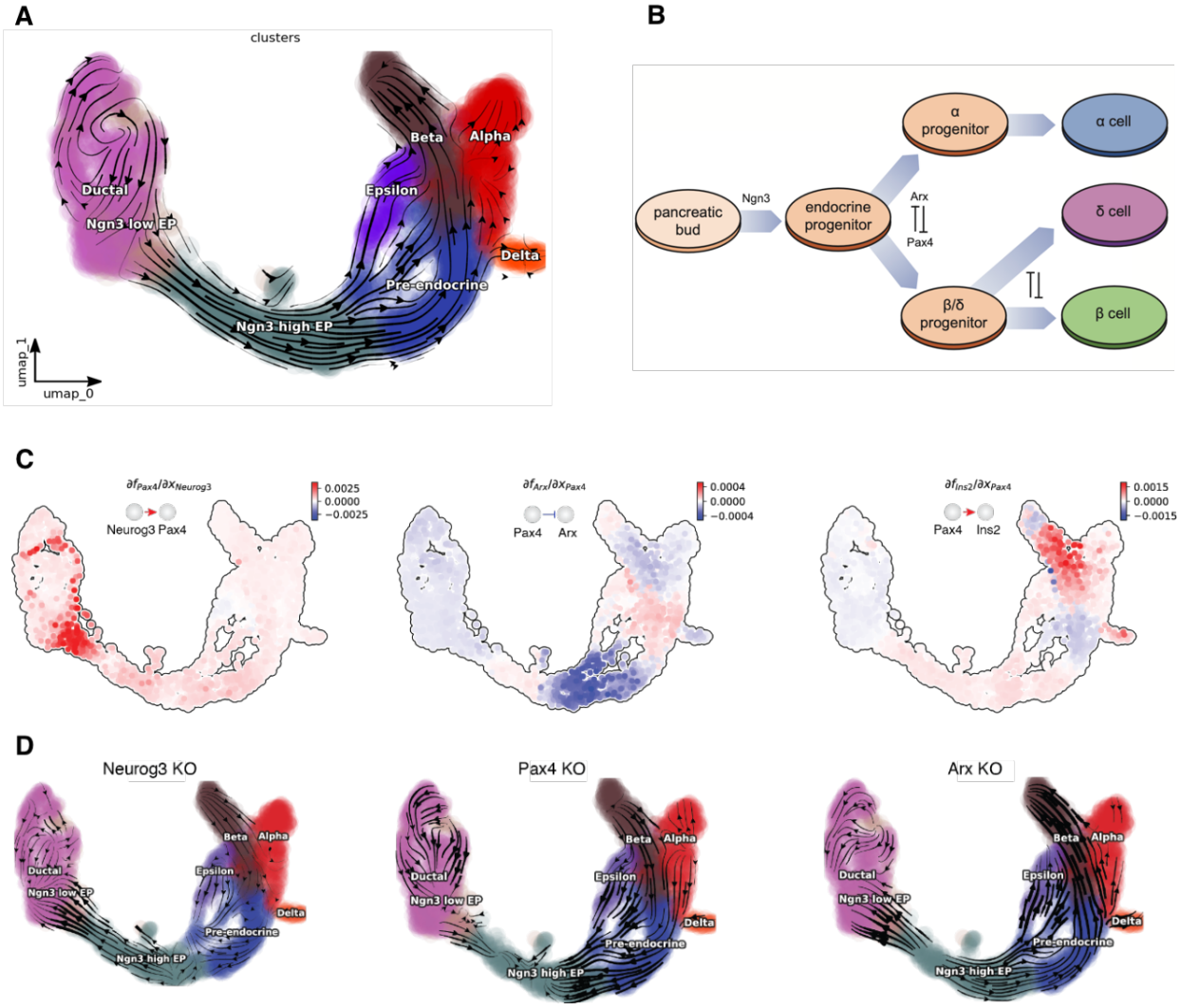


Figure 5.6: Dynode Vectorfield accurately infers the gene regulatory network for Pancreatic Endocrinogenesis and enables genetic perturbation predictions. (A) The two-dimensional UMAP projection of pancreatic expression-velocity vector field with the corresponding cell types. [16, 220] (B) The differentiation process of pancreatic endocrine cells and key regulatory genes/motifs. [16, 220] (C) The scatter plot of the Jacobian from Neurog3 to Pax4 (left), or Pax4 to Arx (middle), and Pax4 to Ins2 (right) across cells. (D) The predicted cell fate changes after knocking out Neurog3 (left), Pax4 (middle), or Arx (right).

during endocrinogenesis from the trained neural network by using the perturbation capability of Dynode Vectorfield. As we expect, knocking out *Neurog3* would reverse the entire process from stable cell types to progenitor cells. Furthermore, knocking out *Pax4* would prevent β cells from developing, while Arx being knocked out would expedite the development process of β and ϵ cells as it's a repressor of *Pax4*.

5.3 Conclusion and Discussion

In this chapter, we introduced Dynode Vectorfield, an AI-based software toolbox and a part of the Dynode package that learns temporal dynamics governing cell evolution and development processes, and described its capabilities and functionalities. In the process of developing the software, we committed ourselves to designing a comprehensive framework rather than a specific package with a limited scope of use cases, ensuring it meets three criteria: (a) Flexibility, (b) Versatility, and (c) Longevity. We made sure that Dynode is able to support a wide variety of possible experimental data types, and it's highly configurable by the user to best fit any experimental setup and conditions. Most importantly, we developed Dynode so it not only doesn't lose its power as time passes but its performance with the technological advances and as the available experimental data quality, abundance, and diversity improve in the future.

That being said, the capabilities of Dynode are not still fully utilized as the quality of transcriptomic data, the amount of data available, and the feasible experiments are limited, which will only improve with time. For instance, the full-scale tracking of all the gene's expressions in a single cell over time (i.e. coupled time-course data) is not quite realizable. As another example, gene perturbation experiments are not widespread and are only feasible in a limited fashion, such as only knocking out a few genes in a limited number of cells. Therefore, these drawbacks could fundamentally limit the amount of information available from which the Dynode Vectorfield toolbox would reveal the true dynamics.

Even though wild-type transcriptomic data is relatively abundant and is becoming more and more available, it comes with its own caveats too. As already discussed throughout the

thesis, and was seen in Section 5.2, the non-uniqueness of the dynamics attributable to the observed wild-type data is an inherent flaw of such experiments, where the existing correlations between the variables of the system could lead to more than one possible underlying dynamics correctly describing the observed data. It's evident that any neural network would learn only one of the possibilities and hence the discovery of spurious/incorrect causal relationships with respect to the ground truth is expected. [6] One powerful way to rule out all these spurious causal relationships would be complementing the training process with single-cell perturbations which would provide the neural network with extra side information, which in turn could rule out the spurious/incorrect regulatory relationships and result in more accurate dynamics reconstructed. The access to such experimental data in genomics, as mentioned, is still quite limited. With the imminent upcoming technical innovations in transcriptomic experiments and higher quality data becoming more widely available, Dynode Vectorfield will further stand out and lead to insightful discoveries for the foreseeable future.

[6]Lots of examples can be made in which relying solely on the observational data can lead to wrong or no conclusions in inferring how various variables influence each other in a dynamical system, thereby proving the importance of stronger evidence for such inference tasks such as *interventions* and *counterfactuals*. As a quick simple example, consider the simple tri-variate system described by $Y = 2X$ and $Z = X + Y$. By just looking at the numerical samples of the system, there would be infinite possible ways of indistinguishably describing the second equation, for example, $Z = 2X + 0.5Y$ would be as valid.

Chapter 6

LEARNING JOINT SPATIAL AND TEMPORAL CELLULAR DYNAMICS VIA ARTIFICIAL INTELLIGENCE

In Chapter 5, we introduced the first toolbox included in the Dynode package named Dynode Vectorfield to learn the temporal transcriptomic dynamics and went over its features and capabilities and then showcased its performance over synthetic and real datasets. In this chapter, we expand the scope of the problem to learn the dynamics in both time and space domains, introducing the *Dynode Spatiotemporal* toolbox.

In many natural phenomena such as heat transfer and electromagnetic wave propagation, the dynamical systems describing such phenomena are defined in both the time and space domain. In other words, the state of the system X at each time point t is, in fact, a function defined over a *Euclidean Space* $\mathcal{S}$ rather than being a finite-dimensional vector, or more precisely it's a *topological manifold* defined over space $\mathcal{S}$, therefore denoted by $X(t, s)$ to indicate the state is a function of both time and space. Trivially, the underlying dynamics defining temporal evolution of phenomena, model and incorporate spatial geometry alongside time trends to fully describe the system, thus expanding the scope of the ODE Equation 5.1:

$$v(t, s) = \dot{X}(t, s) = f\left(X(t, s)\right) \tag{6.1}$$

[1] This equation in many cases would take the form of a *Partial Differential Equation* (PDE) where the function $f(.)$ typically incorporates spatial differential terms such as gradient or Laplacian.

Specifically, in a cellular development process, assuming that each cell's evolutionary path

[1]Note that $\dot{X}(t, s) = \frac{d}{dt}X(t, s)$.

is solely driven by its own expression (as done in Chapter 5) would be too restrictive. In fact, each cell's future state is influenced by its neighbors as well. From a biochemical standpoint, it's known that cells exchange significant amounts of information with their nearby cells via signaling chemicals called *morphogens* which are governing factors for their destination cell's state and fate [262]. Even though not quite conforming to Equation 6.1 as PDEs define the dynamics over a continuous space, the notion of time and space are therefore incorporated in cellular dynamics. Thus, a more comprehensive approach to modeling the governing cell differentiation dynamics over both space and time, where the inter-cell signaling and intra-cell gene interactions are learned is desirable. Especially, the experimental data needed for such modeling is becoming more abundant with the recent innovations in spatiotemporal transcriptomic profiling of the cells such as stereo-seq applied to mouse organogenesis [32], C. Elegance embryogenesis profiling [162], oral squamous cell carcinoma spatial profiling [12], and *TEMPOmap* for various human cells from transcription and translocation to degradation [229], with the computational methods being developed for such data to do various tasks such as spatial clustering or data imputation [63].

6.1 Dynode Spatiotemporal: An AI-based toolbox based on Graph Neural Networks for learning joint spatial and temporal transcriptomic dynamics

We introduce *Dynode Spatiotemporal*, the second toolbox included in the Dynode package implemented using PyTorch. The toolbox tries to model the cell dynamics by jointly learning inter-cell signaling and intra-cell gene interactions which would describe the dynamics in both space and time. To model spatial and temporal signalings in the dynamics to be learned, we employ *Graph Neural Networks* (GNNs) [307] at the core of Dynode Spatiotemporal.

As mentioned above, the cells in a tissue or an organism exchange messages with their neighboring cells via chemical signaling which partially governs the cell procedures. These signal exchanges can be modeled with a graph, where each node represents a cell and each edge represents the spatial proximity of the cells (i.e. the presence of biochemical signaling between each pair of cells) as shown in Figure 6.1A. Furthermore, the past state(s) of each

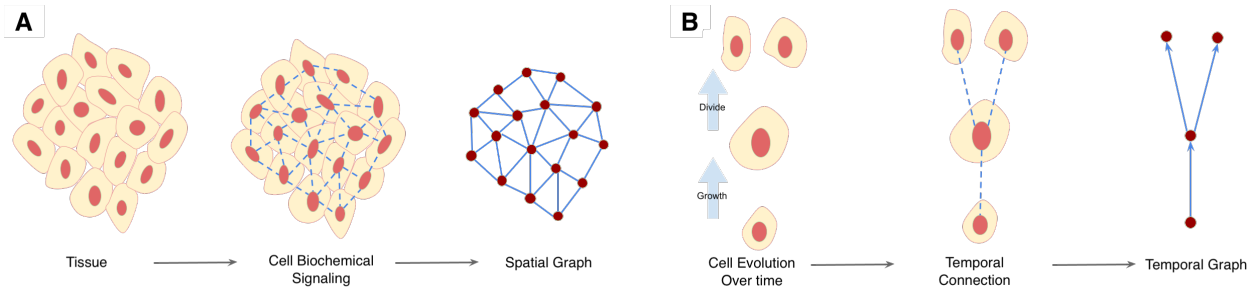


Figure 6.1: Spatial and temporal graphs Representing the signaling between the cells over space and time. (A) Spatial Graph Construction. (B) Temporal Graph Construction.

cell would also provide information about the temporal procedures and trends inside each cell. Hence, such temporal connections could be represented by a directional graph with edges along the direction of time, to model the messages transmitting the information of the past to the present states of the cells (Figure 6.1B). Therefore, A GNN would be the proper tool to learn (A) The temporal and spatial signaling between the cells as a *message-passing* procedure, and (B) The internal dynamics of the cell which would give rise to the desirable output (e.g the transcriptomic path, or its fate) as *aggregation* or *update* procedure.

Therefore, Dynode Spatiotemporal encodes the states of the system (i.e. cells) at the past time points $t-1,\ldots,t-l$ and passes them along the given temporal edges to the present state of the system at time point t. Parameter l is the configurable memory length of the network and is set to 1 as default. Then, the current state of the cells at time t along with the encoded and aggregated messages from the past are further encoded as spatial messages and passed along to the adjacent cells via spatial graph edges, with the optional edge features, yielding a *spatiotemporal hidden state* at each present node. Finally, an output stage consisting of DNN layers decodes the spatiotemporal hidden state into the desired output.

Similar to Dynode Vectorfiled, the neural network structure is highly configurable and any graph neural network architecture and activation can be used based on the task to be accomplished. We, however, set the GNN architecture as *Graph Transformer Operator* or *TransformerConv* [250] as it exhibited superior performance in our benchmarks. We also

set the intermediate activations as Exponential Linear Unit (ELU) [41] and the hidden state encoders (both temporal and spatial) as *Tanh*. The Dynode Spatiotemporal is also able to train in both direct regression and Neural ODE modes, similar to that of Dynode Vectorfield. We showcase both in Section 6.2.

6.2 Results

In this section, we apply the Dynode Spatiotemporal toolbox to two test datasets to learn the dynamics and gain some insights into the differential geometry of the dynamics both in time and space. We first apply Dynode Spatiotemporal to *Mammalian Digit Formation* process [228] based on *Turing Pattern* [280] responsible for many alternating patterns in nature, and show its capability of learning the dynamics. Then, in an extensive test case study, we apply the toolbox to the early-stage embryonic spatiotemporally resolved and lineage-resolved C. elegans dataset with 4D single-cell protein atlas of transcription factors [162] to learn the embryo's growth comprehensively consisting of predicting cells' migration over space, cell fates across the differentiation process, and gene expression of all the cells along the lineage tree.

6.2.1 Dynode Spatiotemporal learns the dynamics governing the mammalian digit formation process

The formation of digits in animals is proposed to be controlled by the Turing pattern, a self-regulated reaction-diffusion model that is believed to describe the procedure of many natural alternating patterns such as animal skin colors and even limbs and teeth development [280]. Simply put, Turing's model describes the interaction between two homogeneously distributed substances, the first of which acts as an activator for the second substance, while the second one is an inhibitor for the first material. With proper parameter choice for the equation set describing the mechanism, and with proper initialization of the aforementioned substances, one can yield stable alternating patterns across one or two-dimensional spaces.

The same model is proposed by [228] to explain the procedure of digit formation over

	k_2	k_3	k_4	k_5	k_7	k_9	d_b	d_w
In-phase dynamics	1	-1	1.27	-0.1	1.59	-0.1	1	2.5
Opposite-phase dynamics	1	-1	-1.59	-0.1	-1.27	-0.1	2.5	1

Table 6.1: Parameters for Digit Formation system [228].

animals' limbs. The mechanism is described as the interaction between the gene *Sox9* and two morphogens *Bmp* and *Wnt* over time and space, and is mathematically defined by a PDE over time and the one-dimensional space as the following:

$$\begin{aligned} \frac{\partial}{\partial t} sox9 &= \alpha_{sox9} + k_2 bmp + k_3 wnt - sox9^3 \\ \frac{\partial}{\partial t} bmp &= \alpha_{bmp} + k_4 sox9 + k_5 bmp + d_b \nabla^2 bmp \\ \frac{\partial}{\partial t} wnt &= \alpha_{wnt} + k_7 sox9 + k_9 wnt + d_w \nabla^2 wnt \end{aligned} \tag{6.2}$$

In [228], the stability of the dynamics above is discussed extensively, based on which two stable sets of parameters are proposed for the system as shown in Table 6.1. We use the first set for $Sox9$ to be in phase with respect to bmp and wnt (Figure 6.2C). Note that $\alpha_{sox9} = \alpha_{bmp} = \alpha_{wnt} = 0$. [2]

To showcase the performance of Dynode Spatiotemporal, we simulated the above dynamics and generated samples from it for 20 independent runs each run with a different random initialization $(sox9(0), bmp(0), wnt(0))$, across a linear space of 60 units long with resolution 0.1, and for 500 time units with resolution 1. Hence a total of 500 time points for every run, and 600 linear space points at every time point, with Neumann boundary condition (i.e. the first-order derivative at the first and last spatial points is 0) [37] enforced at every time point. The spatial proximity graph at each time point is constructed by connecting each of the 600 spatial points to its n-th right and left immediate neighbors. The parameter n should be chosen in such a way that the connected points are sufficiently close to have meaningful effect

[2]Note that the coefficients in the PDE equation from [228] have different signs from those in Equation 6.2 as can be seen in Figure 6.2B. The values in Table 6.1 are corresponding to Equation 6.2.

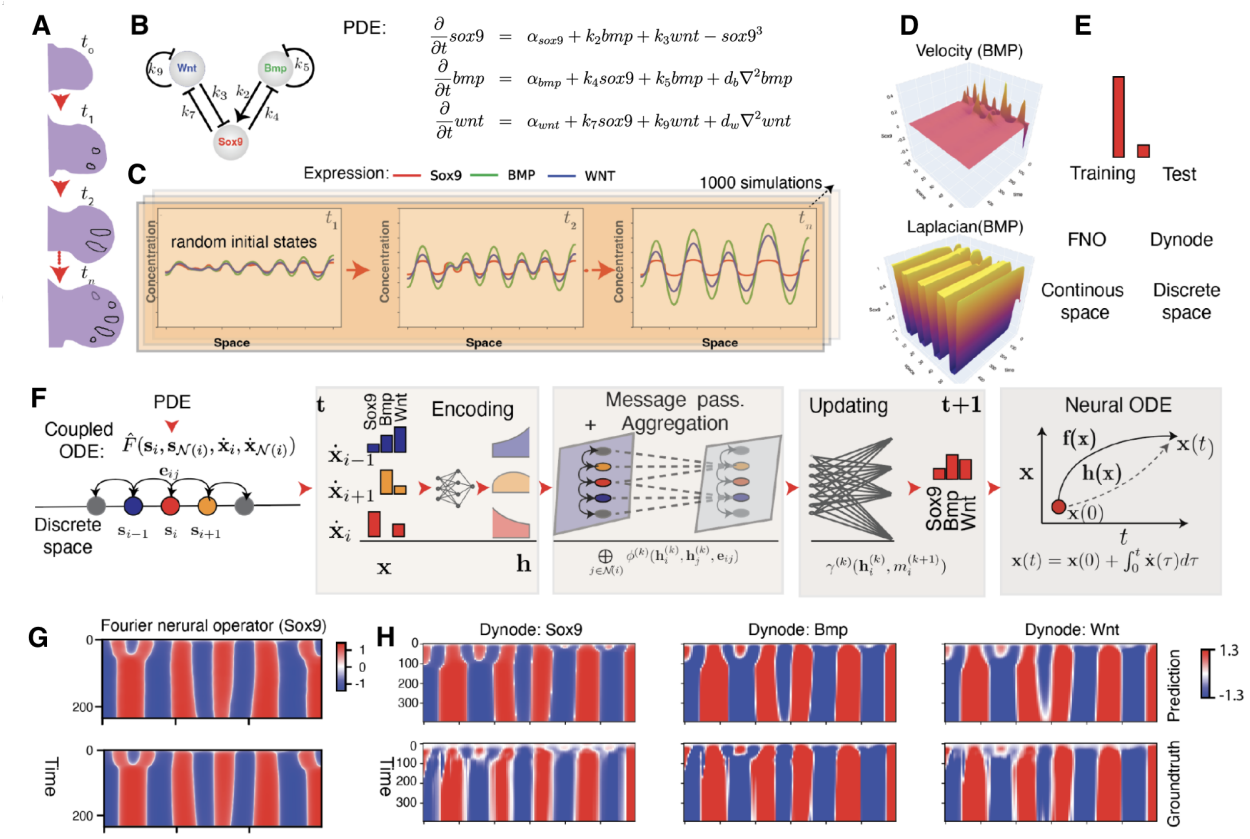


Figure 6.2: Dynode Spatiotemporal accurately learns the dynamics governing the digit formation process in mammals. (A) Schematic of digit formation in mammalian species. (B) Reaction diffusion model of Wnt, Bmp and Sox2. (C) Example expression kinetics of three genes across space for three random time points starting from a random initial state of the simulation of the one-dimensional spatial PDE model of Wnt, BMP, and Sox9 underlying the digit formation. (D) The 3D portrait of the velocity and Laplacian of BMP across time and space. (E) The relative fraction of the training and test dataset. The two major methods used are the Fourier Neural Operator (FNO) [150], suitable for continuous spaces, and the Dynode Spatiotemporal (Graph Neural Network + Neural ODE), suitable for non-uniform discrete spaces. (F) Graph Neural Networks (GNNs) that incorporate graph aggregation and message-passing over spatial graphs are integrated with Neural ODE to sparsely learn high dimensional PDEs with continuous time and non-uniform discrete spaces. (G) FNO recovers the spatiotemporal dynamics on the test dataset. (H). Dynode Spatiotemporal (GNN + Neural ODE) also well learns the spatiotemporal kinetics of the digit formation system.

over each other, but far enough for the spatial gradient (i.e. Laplacian) to be significant enough for the neural network to capture. In this experiment, we chose $n = 5$ (Figure 6.2F).

The generated data along with the spatial graph mentioned above is then used to train three TransformerConv layers in Dynode to learn the digit formation dynamics. We tried both the direct regression (predict state $X(t+1, s)$ given the input $X(t, s)$) and the Neural ODE (Given $X(t = 0, s)$ predict every $X(t > 0, s)$ using Neural ODE) approaches and observed similar results, hence we plot only the Neural ODE results. After we train the neural network, we test it by initializing the dynamics randomly (i.e. different from those of the training data) and compare the predicted output of the neural network versus the ground truth, and can see the reconstruction results in Figure 6.2H. The normalized MSE for the test is 1.01e-3. [3] We compare the performance of the algorithm against *Fourier Neural Operator* (FNO) [150] which to our knowledge exhibits the best performance for learning PDEs and it can be seen that the reconstruction is very good as shown in Figure 6.2G with MSE equal to 2.3e-4. This method, however, is only suitable for the PDEs defined over the continuous space where the data is collected over the uniformly quantized grid of space, and hence it wouldn't be directly applicable to spatiotemporally-resolved transcriptomic cell data.

Learning spurious interactions as a result of correlations in the wild type data: Similar to what we saw in Subsection 5.2.2 and further discussed in Section 5.3, we're bound to the non-uniqueness of dynamics defining the generated wild-type data here as well. Like many other practical cases, the data collected from Equation 6.2 follows its natural trends and limits and doesn't necessarily have an arbitrary distribution of values for all the input features. Specifically, this means not all the arbitrary combinations of input values for $(sox9, bmp, wnt)$ are observable from the simulated data over time. One can notice that the values $sox9$, bmp, and wnt for any run exhibit a very high correlation, with bmp and

[3] Note that since the PDE for digit formation (Equation 6.2) has a specific form (i.e. a linear combination of 1st and 3rd-order polynomials), it's possible to design a very specific network to match it, e.g one may input all the features to a fully-connected deep network to extract non-linear features from the input gene-morphogen expressions, then combine the extracted feathers over a single-layer GNN with linear activation. We tested both the generic and specific approaches, and even though training the latter is simpler than the generic, the final fully-trained performance is identical.

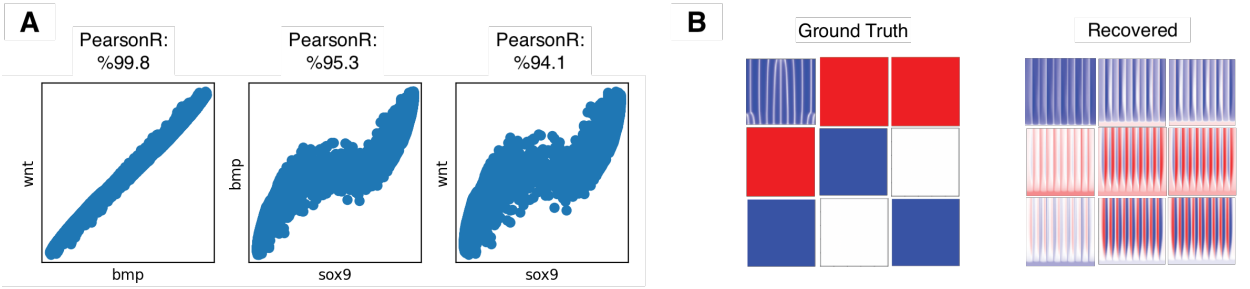


Figure 6.3: Learning spurious interactions in digit formation as a result of correlations in the wild type data. (A) The correlation of the three factors in the wild-type data collected from the dynamics of digit formation. (B) The 3×3 Jacobian matrix from the ground truth (Left) vs learned (Right). Blue and Red values indicate negative and positive interactions respectively. In each smaller color square, the y-axis indicates time and the x-axis indicates space.

wnt having an almost linear correlation. That means any of these variables of the system could be written as infinite possible combinations of each other, especially $\alpha bmp + \beta wnt = 0$. Therefore, the original dynamics 6.2 and hence dynamics learned by Dynode Spatiotemporal could be defined in infinite equally-valid forms. This could lead to the discovery of incorrect activation/repression relationships from the differential analysis (Figure 6.3).

As suggested before, one effective way to deal with such cases is to enforce independent input features via more advanced techniques such as controlled and carefully designed *perturbation* experiments, which are still not available for spatiotemporally resolved transcriptomic data. This direction remains open for future research and development.

6.2.2 The reconstruction of the embryo of spatiotemporally resolved and lineage-resolved C. elegans from 4D single-cell protein atlas of transcription factors

We applied the Dynode Spatiotemporal toolbox to a dataset collected from the proteins of transcriptomic factors in C. Elegans to learn the process of embryogenesis. We wonder whether we can leverage this spatially and temporally resolved information to train a GNN model to create an in silico model of C. elegans embryogenesis. To achieve this goal, we used the live imaging data from a 4D single-cell protein atlas of transcription factors of the

embryogenesis of C. Elegans, where the data is collected from 291 embryos, in each of which a single gene is tracked over time [162]. We combined the data with the lineage tree of C. Elegans to curate proper training and testing data (Figure 6.4A). The details of the data preparation procedure are provided in Section C.4.

The network used here consists of one layer of the TransformerConv network for temporal message-passing from the past cells at $t-1$ to cells at t. Then the current transcriptomic state of each cell at time t along with the hidden state passed from the previous cell state is fed to a three-layer TransformerConv network spatial message passing, along with the physical distance and the expression difference of each cell pair as edge features, to create a new hidden state at each cell at time t. The hidden state is then fed to a two-layer dense neural network as the output stage to render the desired output based on the tasks described below (Figure 6.4B).

- **Cell Fate Prediction:** The first task is to predict the fate of each cell at each time point t given the expressions of all the cells at t along with the messages passed from past time points. Each cell can take one of the three possible fates: growth, division, or death, shown in Figure 6.4C. The number in the bracket corresponds to the number of the corresponding event for each category. We train Dynode Spatiotemporal to predict growth and division events only, as the number of death events is too small for any training task. We use a weighted binary cross-entropy loss, with weights inversely proportional to the number of each event's occurrence, to account for the far smaller number of division events. As the test precision and recall of each of the categories in Figure 6.4D indicates, the performance of the network is remarkable in recovering the correct fate of each cell, especially over the division events given they comprise less than %1 of all events.

- **Cell Migration Prediction:** We then tried to predict how the cells move (i.e. migrate) in the physical space over time. Given each cell's current coordinates (x, y, z) at time t, and using the Graph Neural ODE method, we applied Dynode Spatiotemporal to

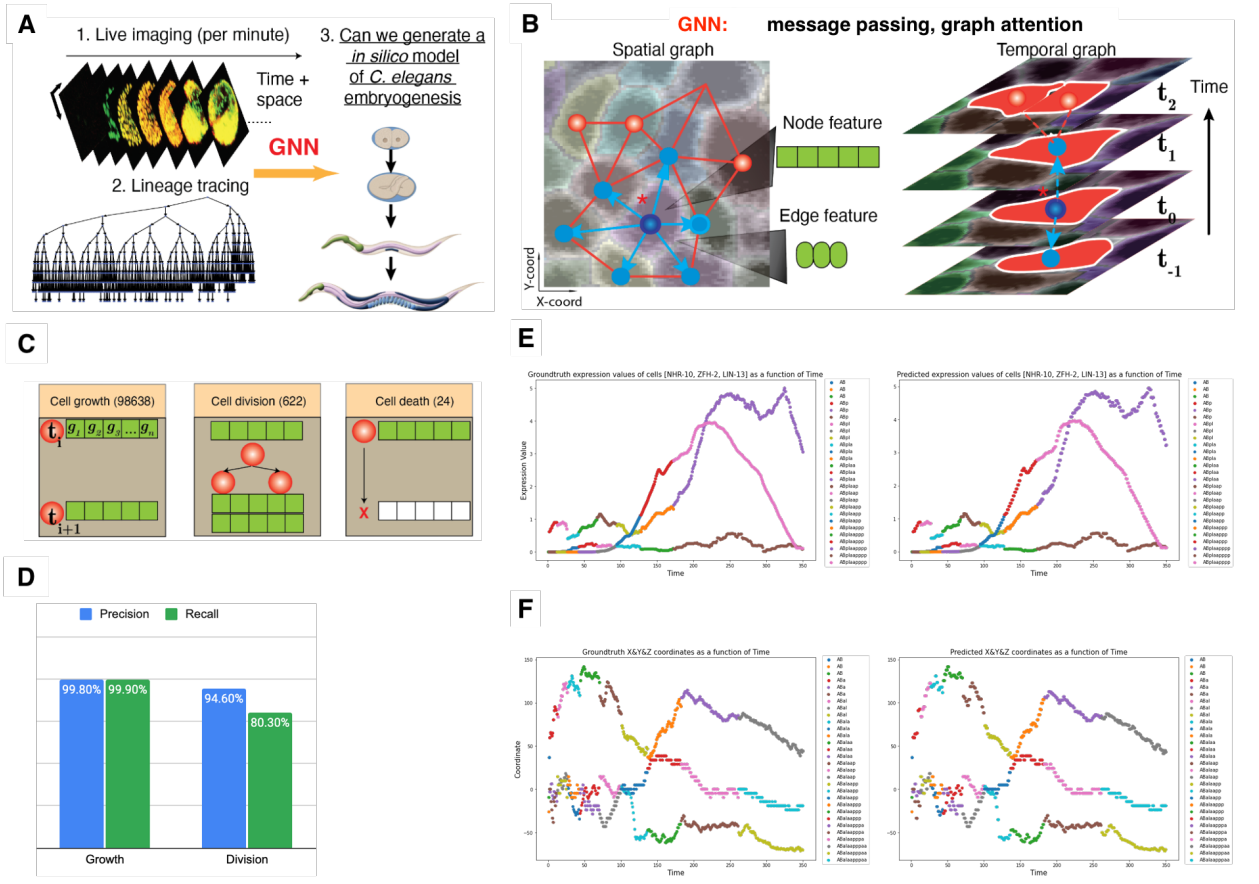


Figure 6.4: The reconstruction of the embryo in silico 3D models of spatiotemporally resolved and lineage-resolved C. elegans dataset with 4D single-cell protein atlas of transcription factors. (A) Live imaging of C. elegans that tracks spatial coordinates of each cell and the expression of a fused protein expression over the time course of embryogenesis. The lineage tree of C. elegans was reconstructed based on tracking of the nucleus. (B) Using GNNs to build 3D in silico models of C. elegans with spatially, temporally, and lineage-resolved 4D protein atlas dataset. The graph neural network models the spatial and temporal message passing. Each node corresponds to a cell while each edge corresponds to the spatial connection in the spatial or the temporal coupling (either from the same cell or progenitor cell from a prior time point) graph. (C) The three different fates of C. elegans embryogenesis. The number in the bracket corresponds to the number of the corresponding event for each category. (D) The precision and recall of each cell fate prediction. Note that *death* was not included due to the small set of cells in the category. (E) The trajectory of single cells in the gene expression space for *NRH10*, *ZFH2*, *LIN13* that reveals the intrinsic gene expression state of an example cell *ABplaappppp* and all its precursors up to cell *AB*. Left: Ground Truth, Right: Predicted. (F) The trajectory of single cells in the physical space that reveals the cell migration path of an example cell *ABalaappppaa* and all its precursors up to cell *AB*. Left: Ground Truth, Right: Predicted.

predict the next coordinates of each cell at the next time point. The predicted trajectory versus the ground truth is shown in Figure 6.4F. The normalized test loss is 4.46e-5 for this task.

- **Gene Expression Prediction:** The gene expression prediction task over time is carried out similarly to the cell migration task, except the output would be the projected gene expression at the future time point $t+1$ given each cell's current transcriptomic state at time t. Similar to migration, we used Dynode Spatiotemporal with the Graph Neural ODE method to predict the expression. The predicted trajectory versus the ground truth is shown in Figure 6.4E. The normalized test loss is 8.96e-4 for this task.

6.3 Conclusion and Discussion

In this chapter, we introduced the second toolbox of the Dynode package called Dynode Spatiotemporal to learn the joint temporal and spatial dynamics governing cellular processes. Similar to Dynode Vectofield, we committed ourselves to developing a toolbox with flexibility, versatility, and longevity. As discussed earlier, the spatiotemporally resolved transcriptomic data collection experiments are relatively new and limited, and Dynode Spatiotemporal is to our best knowledge the first toolbox capable of learning the joint gene expression and inter-cell signaling dynamics, with no counterpart to be benchmarked against.

Furthermore, due to these studies and experiments still being in their early stages, the abundance, quality, and diversity of the data available are limited. As discussed in Sections 6.2.2 and C.4, It's still not possible to track every gene in every cell at all time points as the sample grows and evolves over time, and thus the true temporal and spatial interactions are not preserved. Even though the experiment in Section 6.2.2 partially represents the typical cell development, cell migration, and gene expression's temporal trend in a standard C. Elegans embryo, more advanced experiments are required to preserve the finer gene interactions and inter-cell signaling dynamics. Furthermore, neural networks' learning capability is trivially bound by the abundance of the data, and their performance only improves as more training

data becomes available.

As extensively discussed in Section 6.2.1 as well as Section 5.3, the non-uniqueness of the dynamics is a universal problem across the causal inference field, which is also present here and can lead to learning the wrong gene interaction and inter-cell signaling, and thus requires advanced experiments such as perturbation to be dealt with.

Similar to the Dynode Vectorfield toolbox, Dynode Spatiotemporal will further stand out and lead to insightful discoveries as the data quality and quantity improve over time with the imminent upcoming technical innovations in transcriptomic experiments.

BIBLIOGRAPHY

[1] Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms. *arXiv preprint arXiv:1503.00609*, 2015.

[2] Sara Aibar, Carmen Bravo González-Blas, Thomas Moerman, Hana Imrichova, Gert Hulselmans, Florian Rambow, Jean-Christophe Marine, Pierre Geurts, Jan Aerts, Joost van den Oord, et al. Scenic: single-cell regulatory network inference and clustering. *Nature methods*, 14(11):1083, 2017.

[3] Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450, 2007.

[4] Pierre-Olivier Amblard and Olivier JJ Michel. On directed information theory and granger causality graphs. *Journal of computational neuroscience*, 30(1):7–16, 2011.

[5] Pierre-Olivier Amblard and Olivier JJ Michel. On directed information theory and granger causality graphs. *Journal of computational neuroscience*, 30(1):7–16, 2011.

[6] Pierre-Olivier Amblard and Olivier JJ Michel. On directed information theory and granger causality graphs. *Journal of computational neuroscience*, 30(1):7–16, 2011.

[7] Fatemeh Amiri, MohammadMahdi Rezaei Yousefi, Caro Lucas, Azadeh Shakery, and Nasser Yazdani. Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications*, 34(4):1184–1199, 2011.

[8] Anima Anandkumar, Furong Huang, Daniel J Hsu, and Sham M Kakade. Learning mixtures of tree graphical models. In *Advances in Neural Information Processing Systems*, pages 1052–1060, 2012.

[9] Venkat Anantharam, Amin Gohari, Sudeep Kamath, and Chandra Nair. On maximal correlation, hypercontractivity, and the data processing inequality studied by erkip and cover. *arXiv preprint arXiv:1304.6133*, 2013.

[10] Suguru Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *Information Theory, IEEE Transactions on*, 18(1):14–20, 1972.

[11] Barry C Arnold, Narayanaswamy Balakrishnan, and Haikady Navada Nagaraja. *A first course in order statistics*, volume 54. Siam, 1992.

[12] Rohit Arora, Christian Cao, Mehul Kumar, Sarthak Sinha, Ayan Chanda, Reid McNeil, Divya Samuel, Rahul K Arora, T Wayne Matthews, Shamir Chandarana, et al. Spatial transcriptomics reveals distinct and conserved tumor core and edge architectures that predict survival and targeted therapy response. *Nature Communications*, 14(1):5029, 2023.

[13] Kin Fai Au, Vittorio Sebastiano, Pegah Tootoonchi Afshar, Jens Durruthy Durruthy, Lawrence Lee, Brian A. Williams, Harm van Bakel, Eric E. Schadt, Renee A. Reijo-Pera, Jason G. Underwood, and Wing Hung Wong. Characterization of the human ESC transcriptome by hybrid sequencing. *Proceedings of the National Academy of Sciences*, 110(50):E4821–E4830, December 2013.

[14] Ann C Babtie, Thalia E Chan, and Michael PH Stumpf. Learning regulatory models for cell development from single cell transcriptomic data. *Current Opinion in Systems Biology*, 5:72–81, 2017.

[15] Ziv Bar-Joseph, Anthony Gitter, and Itamar Simon. Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8):552, 2012.

[16] Aimée Bastidas-Ponce, Sophie Tritschler, Leander Dony, Katharina Scheibner, Marta Tarquis-Medina, Ciro Salinno, Silvia Schirge, Ingo Burtscher, Anika Böttcher, Fabian J Theis, et al. Comprehensive single cell mrna profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development*, 146(12):dev173849, 2019.

[17] Jan Beirlant, Edward J Dudewicz, László Györfi, and Edward C Van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997.

[18] Sean C Bendall, Kara L Davis, El-ad David Amir, Michelle D Tadmor, Erin F Simonds, Tiffany J Chen, Daniel K Shenfeld, Garry P Nolan, and Dana Pe'er. Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development. *Cell*, 157(3):714–725, 2014.

[19] Sean C Bendall, Erin F Simonds, Peng Qiu, D Amir El-ad, Peter O Krutzik, Rachel Finck, Robert V Bruggner, Rachel Melamed, Angelica Trejo, Olga I Ornatsky, and R.S. Balderas. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science*, 332(6030):687–696, 2011.

[20] Jose M Bernardo. Algorithm as 103: Psi (digamma) function. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 25(3):315–317, 1976.

[21] Richard E Blahut. Computation of channel capacity and rate-distortion functions. *Information Theory, IEEE Transactions on*, 18(4):460–473, 1972.

[22] Taoufik Bouezmarni, Jeroen VK Rombouts, and Abderrahim Taamouti. Nonparametric copula-based test for conditional independence with applications to granger causality. *Journal of Business & Economic Statistics*, 30(2):275–287, 2012.

[23] Rufus Bowen and David Ruelle. The ergodic theory of axiom a flows. In *The Theory of Chaotic Attractors*, pages 55–76. Springer, 1975.

[24] Guy Bresler, Ma'ayan Bresler, and David Tse. Optimal assembly for high throughput shotgun sequencing. *BMC Bioinformatics*, 14(Suppl 5):S18, July 2013.

[25] Gavin Brown. A new perspective for information theoretic feature selection. In *Artificial Intelligence and Statistics*, pages 49–56, 2009.

[26] Sharon R Browning and Brian L Browning. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *The American Journal of Human Genetics*, 81(5):1084–1097, 2007.

[27] Junyue Cao, Wei Zhou, Frank Steemers, Cole Trapnell, and Jay Shendure. Characterizing the temporal dynamics of gene expression in single cells with sci-fate. *bioRxiv*, page 666081, 2019.

[28] Mark JP Chaisson, Richard K Wilson, and Evan E Eichler. Genetic variation and the de novo assembly of human genomes. *Nature Reviews Genetics*, 2015.

[29] Chung Chan, Ali Al-Bashabsheh, Javad B Ebrahimi, Tarik Kaced, and Tie Liu. Multivariate mutual information inspired by secret-key agreement. *Proceedings of the IEEE*, 103(10):1883–1913, 2015.

[30] Chung Chan, Ali Al-Bashabsheh, Qiaoqiao Zhou, Tarik Kaced, and Tie Liu. Info-clustering: A mathematical theory for data clustering. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):64–91, 2016.

[31] Thalia E Chan, Michael PH Stumpf, and Ann C Babtie. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell systems*, 5(3):251–267, 2017.

[32] Ao Chen, Sha Liao, Mengnan Cheng, Kailong Ma, Liang Wu, Yiwei Lai, Xiaojie Qiu, Jin Yang, Jiangshan Xu, Shijie Hao, et al. Spatiotemporal transcriptomic atlas of mouse organogenesis using dna nanoball-patterned arrays. *Cell*, 185(10):1777–1792, 2022.

[33] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[34] Rui Chen, George I. Mias, Jennifer Li-Pook-Than, Lihua Jiang, Hugo Y. K. Lam, Rong Chen, Elana Miriami, Konrad J. Karczewski, Manoj Hariharan, Frederick E. Dewey, Yong Cheng, Michael J. Clark, Hogune Im, Lukas Habegger, Suganthi Balasubramanian, Maeve O'Huallachain, Joel T. Dudley, Sara Hillenmeyer, Rajini Haraksingh, Donald Sharon, Ghia Euskirchen, Phil Lacroute, Keith Bettinger, Alan P. Boyle, Maya Kasowski, Fabian Grubert, Scott Seki, Marco Garcia, Michelle Whirl-Carrillo, Mercedes Gallardo, Maria A. Blasco, Peter L. Greenberg, Phyllis Snyder, Teri E. Klein, Russ B. Altman, Atul J. Butte, Euan A. Ashley, Mark Gerstein, Kari C. Nadeau, Hua Tang, and Michael Snyder. Personal omics profiling reveals dynamic molecular and medical phenotypes. *Cell*, 148(6):1293–1307, March 2012.

[35] Yuxin Chen, Govinda Kamath, Changho Suh, and David Tse. Community recovery in graphs with locality. *arXiv preprint arXiv:1602.03828*, 2016.

[36] Zhanlin Chen, William C King, Aheyon Hwang, Mark Gerstein, and Jing Zhang. Deepvelo: Single-cell transcriptomic deep velocity field learning with neural ordinary differential equations. *Science Advances*, 8(48):eabq3745, 2022.

[37] Alexander H-D Cheng and Daisy T Cheng. Heritage and early history of the boundary element method. *Engineering analysis with boundary elements*, 29(3):268–302, 2005.

[38] Myung Jin Choi, Vincent YF Tan, Animashree Anandkumar, and Alan S Willsky. Learning latent tree graphical models. *The Journal of Machine Learning Research*, 12:1771–1812, 2011.

[39] C Chow and C Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

[40] Chieh Min Jamie Chu, Honey Modi, Cara Ellis, Nicole AJ Krentz, Søs Skovsø, Yiwei Bernie Zhao, Haoning Cen, Nilou Noursadeghi, Evgeniy Panzhinskiy, Xiaoke Hu, et al. Dynamic ins2 gene activity defines β-cell maturity states. *Diabetes*, 71(12):2612–2631, 2022.

[41] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[42] Jean Cornuet, Jean-Michel Marin, Antonietta Mira, and Christian P Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, 2012.

[43] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[44] Imre Csiszár. Generalized cutoff rates and renyi's information measures. *Information Theory, IEEE Transactions on*, 41(1):26–34, 1995.

[45] Imre Csiszár. Axiomatic characterizations of information measures. *Entropy*, 10(3):261–273, 2008.

[46] Imre Csiszár and Paul C Shields. *Information theory and statistics: A tutorial*. Now Publishers Inc, 2004.

[47] Paul Cuff and Lanqing Yu. Differential privacy as a mutual information constraint. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 43–54. ACM, 2016.

[48] Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Optimal phylogenetic reconstruction. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 159–168. ACM, 2006.

[49] A Philip Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–31, 1979.

[50] Chand Desai, Gian Garriga, Steven L Mclntire, and H Robert Horvitz. A genetic pathway for the development of the caenorhabditis elegans hsn motor neurons. *Nature*, 1988.

[51] Luc Devroye and Clark S Penrod. The consistency of automatic kernel density estimates. *The Annals of Statistics*, pages 1231–1249, 1984.

[52] Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *Cell*, 167(7):1853–1866, 2016.

[53] Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on pure and applied mathematics*, 36(2):183–212, 1983.

[54] Michael Eichler. Graphical modelling of multivariate time series. *Probability Theory and Related Fields*, 153(1-2):233–268, 2012.

[55] Abbas El Gamal and Young-Han Kim. *Network information theory*. Cambridge university press, 2011.

[56] Rossin Erbe, Genevieve Stein-O'Brien, and Elana J Fertig. Transcriptomic forecasting with neural ordinary differential equations. *Patterns*, 4(8), 2023.

[57] Paul Erdös and Alfréd Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[58] Florian Erhard, Marisa AP Baptista, Tobias Krammer, Thomas Hennig, Marius Lange, Panagiota Arampatzi, Christopher S Jürges, Fabian J Theis, Antoine-Emmanuel Saliba, and Lars Dölken. scslam-seq reveals core features of transcription dynamics in single cells. *Nature*, 571(7765):419–423, 2019.

[59] Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

[60] Jalal Etesami, Negar Kiyavash, and Todd P Coleman. Learning minimal latent directed information trees. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 2726–2730. IEEE, 2012.

[61] LawrenceCraig Evans. *Measure theory and fine properties of functions*. Routledge, 2018.

[62] Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology*, 5(1):e8, 2007.

[63] Shuangsang Fang, Bichao Chen, Yong Zhang, Haixi Sun, Longqi Liu, Shiping Liu, Yuxiang Li, and Xun Xu. Computational approaches and challenges in spatial transcriptomics. *Genomics, Proteomics & Bioinformatics*, 2022.

[64] Mark WEJ Fiers, Liesbeth Minnoye, Sara Aibar, Carmen Bravo González-Blas, Zeynep Kalender Atak, and Stein Aerts. Mapping gene regulatory networks from single-cell omics data. *Briefings in functional genomics*, 17(4):246–254, 2018.

[65] François Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5(Nov):1531–1555, 2004.

[66] Puri Fortes, Dasa Longman, Susan McCracken, Joanna Y Ip, Raymond Poot, Iain W Mattaj, Javier F Cáceres, and Benjamin J Blencowe. Identification and characterization of red120: a conserved pwi domain protein with links to splicing and 3'-end formation. *FEBS letters*, 581(16):3087–3097, 2007.

[67] Stefan Frenzel and Bernd Pompe. Partial mutual information for coupling analysis of multivariate time series. *Physical review letters*, 99(20):204101, 2007.

[68] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[69] Alan M. Frieze. An algorithm for finding hamilton cycles in random directed graphs. *Journal of Algorithms*, 9(2):181–204, 1988.

[70] Alessandro Furlan, Vyacheslav Dyachuk, Maria Eleni Kastriti, Laura Calvo-Enrique, Hind Abdo, Saida Hadjab, Tatiana Chontorotzea, Natalia Akkuratova, Dmitry Usoskin, Dmitry Kamenev, et al. Multipotent peripheral glial cells generate neuroendocrine cells of the adrenal medulla. *Science*, 357(6346):eaal3753, 2017.

[71] Mingze Gao, Chen Qiao, and Yuanhua Huang. Unitvelo: temporally unified rna velocity reinforces single-cell trajectory inference. *Nature Communications*, 13(1):6586, 2022.

[72] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. *arXiv preprint arXiv:1411.2003*, 2014.

[73] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Estimating mutual information by local gaussian approximation. *arXiv preprint arXiv:1508.00536*, 2015.

[74] W. Gao, S. Oh, and P. Viswanath. Demystifying fixed k-nearest neighbor information estimators. *IEEE Transactions on Information Theory*, pages 1–1, 2018.

[75] Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Causal strength via shannon capacity: Axioms, estimators and applications. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[76] Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Conditional dependence via shannon capacity: Axioms, estimators and applications. *arXiv preprint arXiv:1602.03476*, 2016.

[77] Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Conditional dependence via shannon capacity: Axioms, estimators and applications. *arXiv preprint arXiv:1602.03476*, 2016.

[78] Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Conditional dependence via shannon capacity: Axioms, estimators and applications. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2780–2789, 2016.

[79] Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Estimating mutual information for discrete-continuous mixtures. In *Advances in Neural Information Processing Systems*, pages 5988–5999, 2017.

[80] Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Estimating mutual information for discrete-continuous mixtures. *arXiv preprint arXiv:1709.06212*, 2017.

[81] Weihao Gao, Sewoong Oh, and Pramod Viswanath. Breaking the bandwidth barrier: Geometrical adaptive entropy estimation. In *Advances in Neural Information Processing Systems*, pages 2460–2468, 2016.

[82] Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k-nearest neighbor information estimators. *arXiv preprint arXiv:1604.03006*, 2016.

[83] Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k-nearest neighbor information estimators. In *Information Theory (ISIT), 2017 IEEE International Symposium on*, pages 1267–1271. IEEE, 2017.

[84] Philipp Geiger, Kun Zhang, Bernhard Schoelkopf, Mingming Gong, and Dominik Janzing. Causal inference by identification of vector autoregressive processes with hidden components. In *International Conference on Machine Learning*, pages 1917–1925, 2015.

[85] AmirEmad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Kun Zhang. Learning causal structures using regression invariance. In *Advances in Neural Information Processing Systems*, pages 3015–3025, 2017.

[86] David Gordon, John Huddleston, Mark JP Chaisson, Christopher M Hill, Zev N Kronenberg, Katherine M Munson, Maika Malig, Archana Raja, Ian Fiddes, LaDeana W

Hillier, and C. Dunn. Long-read sequence assembly of the gorilla genome. *Science*, 352(6281):aae0344, 2016.

[87] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110:393–416, 2021.

[88] Manfred G. Grabherr, Brian J. Haas, Moran Yassour, Joshua Z. Levin, Dawn A. Thompson, Ido Amit, Xian Adiconis, Lin Fan, Raktima Raychowdhury, Qiandong Zeng, Zehua Chen, Evan Mauceli, Nir Hacohen, Andreas Gnirke, Nicholas Rhind, Federica di Palma, Bruce W. Birren, Chad Nusbaum, Kerstin Lindblad-Toh, Nir Friedman, and Aviv Regev. Full-length transcriptome assembly from RNA-seq data without a reference genome. *Nature Biotechnology*, 29(7):644–652, July 2011.

[89] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[90] Clive WJ Granger. Testing for causality: a personal viewpoint. *Journal of Economic Dynamics and control*, 2:329–352, 1980.

[91] Clive WJ Granger. Causality, cointegration, and control. *Journal of Economic Dynamics and Control*, 12(2-3):551–559, 1988.

[92] Brenton R. Graveley, Angela N. Brooks, Joseph W. Carlson, Michael O. Duff, Jane M. Landolin, Li Yang, Carlo G. Artieri, Marijke J. van Baren, Nathan Boley, Benjamin W. Booth, James B. Brown, Lucy Cherbas, Carrie A. Davis, Alex Dobin, Renhua Li, Wei Lin, John H. Malone, Nicolas R. Mattiuzzo, David Miller, David Sturgill, Brian B. Tuch, Chris Zaleski, Dayu Zhang, Marco Blanchette, Sandrine Dudoit, Brian Eads, Richard E. Green, Ann Hammonds, Lichun Jiang, Phil Kapranov, Laura Langton, Norbert Perrimon, Jeremy E. Sandler, Kenneth H. Wan, Aarron Willingham, Yu Zhang, Yi Zou, Justen Andrews, Peter J. Bickel, Steven E. Brenner, Michael R. Brent, Peter Cherbas, Thomas R. Gingeras, Roger A. Hoskins, Thomas C. Kaufman, Brian Oliver, and Susan E. Celniker. The developmental transcriptome of drosophila melanogaster. *Nature*, 471(7339):473–479, March 2011.

[93] Dominic Grün and Alexander van Oudenaarden. Design and analysis of single-cell sequencing experiments. *Cell*, 163(4):799–810, 2015.

[94] Laleh Haghverdi, Maren Buettner, F Alexander Wolf, Florian Buettner, and Fabian J Theis. Diffusion pseudotime robustly reconstructs lineage branching. *Nature methods*, 13(10):845, 2016.

[95] Bruce Hajek, Yihong Wu, and Jiaming Xu. Computational lower bounds for community detection on random graphs. *arXiv preprint arXiv:1406.6625*, 2014.

[96] Fiona K Hamey, Sonia Nestorowa, Sarah J Kinston, David G Kent, Nicola K Wilson, and Berthold Göttgens. Reconstructing blood stem cell regulatory network models from single-cell molecular profiles. *Proceedings of the National Academy of Sciences*, 114(23):5822–5829, 2017.

[97] Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Adaptive estimation of shannon entropy. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 1372–1376. IEEE, 2015.

[98] Sridhar Hannenhalli and Pavel A Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM (JACM)*, 46(1):1–27, 1999.

[99] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[100] Gert-Jan Hendriks, Lisa A Jung, Anton JM Larsson, Michael Lidschreiber, Oscar Andersson Forsman, Katja Lidschreiber, Patrick Cramer, and Rickard Sandberg. Nasc-seq monitors rna synthesis in single cells. *Nature communications*, 10(1):3138, 2019.

[101] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE, 2007.

[102] Veronika A Herzog, Brian Reichholf, Tobias Neumann, Philipp Rescheneder, Pooja Bhat, Thomas R Burkard, Wiebke Wlotzka, Arndt Von Haeseler, Johannes Zuber, and Stefan L Ameres. Thiol-linked alkylation of rna to assess expression dynamics. *Nature methods*, 14(12):1198–1204, 2017.

[103] Steven M Hill, Laura M Heiser, Thomas Cokelaer, Michael Unger, Nicole K Nesser, Daniel E Carlin, Yang Zhang, Artem Sokolov, Evan O Paull, Chris K Wong, et al. Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature methods*, 13(4):310, 2016.

[104] Nils Lid Hjort and MC Jones. Locally parametric nonparametric density estimation. *The Annals of Statistics*, pages 1619–1647, 1996.

[105] Katerina Hlaváčková-Schindler, Milan Paluš, Martin Vejmelka, and Joydeep Bhattacharya. Causality detection based on information-theoretic approaches in time series analysis. *Physics Reports*, 441(1):1–46, 2007.

[106] Siu-Wai Ho and Sergio Verdú. Convexity/concavity of renyi entropy and α-mutual information. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 745–749. IEEE, 2015.

[107] Paul W Holland, Clark Glymour, and Clive Granger. Statistics and causal inference. *ETS Research Report Series*, 1985(2), 1985.

[108] Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. Learning temporal dependence from time-series data with latent variables. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 253–262. IEEE, 2016.

[109] Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. Learning temporal dependence from time-series data with latent variables. *Proc.IEEE Data Science and Advanced Analytics, 2016.*, 2016.

[110] Bryan Howie, Christian Fuchsberger, Matthew Stephens, Jonathan Marchini, and Gonçalo R Abecasis. Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. *Nature genetics*, 44(8):955–959, 2012.

[111] Sui Huang, Gabriel Eichler, Yaneer Bar-Yam, and Donald E Ingber. Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Physical review letters*, 94(12):128701, 2005.

[112] Sui Huang, Yan-Ping Guo, Gillian May, and Tariq Enver. Bifurcation dynamics in lineage-commitment in bipotent progenitor cells. *Developmental biology*, 305(2):695–713, 2007.

[113] Vân Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. Inferring regulatory networks from expression data using tree-based methods. *PloS one*, 5(9):e12776, 2010.

[114] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[115] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[116] D. Janzing, B. Steudel, N. Shajarisales, and B. Schölkopf. *Justifying Information-Geometric Causal Inference*, chapter 18, pages 253–265. Springer International Publishing, 2015.

[117] Dominik Janzing, David Balduzzi, Moritz Grosse-Wentrup, and Bernhard Schölkopf. Quantifying causal influences. *The Annals of Statistics*, 41(5):2324–2358, 2013.

[118] Jiantao Jiao, Weihao Gao, and Yanjun Han. The nearest neighbor information estimator is adaptively near minimax rate-optimal. *arXiv preprint arXiv:1711.08824*, 2017.

[119] Jiantao Jiao, Haim H Permuter, Lei Zhao, Young-Han Kim, and Tsachy Weissman. Universal estimation of directed information. *IEEE Transactions on Information Theory*, 59(10):6220–6242, 2013.

[120] Jiantao Jiao, Haim H Permuter, Lei Zhao, Young-Han Kim, and Tsachy Weissman. Universal estimation of directed information. *IEEE Transactions on Information Theory*, 59(10):6220–6242, 2013.

[121] Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman. Minimax estimation of functionals of discrete distributions. *IEEE Transactions on Information Theory*, 61(5):2835–2885, 2015.

[122] Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(Mar):613–636, 2007.

[123] Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, et al. Nonparametric von mises estimators for entropies, divergences and mutual informations. In *Advances in Neural Information Processing Systems*, pages 397–405, 2015.

[124] Sreeram Kannan, Joseph Hui, and Kayvon Mazooji. Shannon: Open source rna-seq assembly software. *https://github.com/sreeramkannan/Shannon/*, 2016.

[125] Sreeram Kannan, Joseph Hui, Kayvon Mazooji, Lior Pachter, and David Tse. Shannon: An information-optimal de novo rna-seq assembler. *Nature Biotechnology, in review, available at http://biorxiv.org/content/early/2016/02/09/039230*, 2016.

[126] Leonid Vasilevich Kantorovich and SG Rubinshtein. On a space of totally additive functions. *Vestnik of the St. Petersburg University: Mathematics*, 13(7):52–59, 1958.

[127] JHB Kemperman. On the shannon capacity of an arbitrary channel. In *Indagationes Mathematicae (Proceedings)*, volume 77, pages 101–115. Elsevier, 1974.

[128] Shiraj Khan, Sharba Bandyopadhyay, Auroop R Ganguly, Sunil Saigal, David J Erickson III, Vladimir Protopopescu, and George Ostrouchov. Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. *Physical Review E*, 76(2):026209, 2007.

[129] Rahul Kidambi and Sreeram Kannan. On shannon capacity and causal estimation. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pages 988–992. IEEE, 2015.

[130] Hyeji Kim, Weihao Gao, Sreeram Kanan, Sewoong Oh, and Pramod Viswanath. Discovering potential correlations via hypercontractivity. *arXiv preprint arXiv:1709.04024*, 2017.

[131] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[132] LF Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

[133] LF Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

[134] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[135] Smita Krishnaswamy, Matthew H Spitzer, Michael Mingueneau, Sean C Bendall, Oren Litvin, Erica Stone, Dana Pe'er, and Garry P Nolan. Conditional density-based analysis of t cell signaling in single-cell data. *Science*, 346(6213):1250689, 2014.

[136] Nojun Kwak and Chong-Ho Choi. Input feature selection by mutual information based on parzen window. *IEEE transactions on pattern analysis and machine intelligence*, 24(12):1667–1671, 2002.

[137] Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E Kastriti, Peter Lönnerberg, Alessandro Furlan, et al. Rna velocity of single cells. *Nature*, 560(7719):494, 2018.

[138] Ka-Kit Lam, Asif Khalak, and David Tse. Near-optimal assembly for shotgun sequencing with noisy reads. In *RECOMB-Seq, to appear*, 2014. PMCID not available.

[139] Peter Langfelder and Steve Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):559, 2008.

[140] Peter Laslo, Chauncey J Spooner, Aryeh Warmflash, David W Lancki, Hyun-Jun Lee, Roger Sciammas, Benjamin N Gantner, Aaron R Dinner, and Harinder Singh. Multilineage transcriptional priming and determination of alternate hematopoietic cell fates. *Cell*, 126(4):755–766, 2006.

[141] Hai-Son Le, Marcel H. Schulz, Brenna M. McCauley, Veronica F. Hinman, and Ziv Bar-Joseph. Probabilistic error correction for RNA sequencing. *Nucleic Acids Research*, 41(10):e109–e109, May 2013.

[142] Byunghan Lee, Taesup Moon, Sungroh Yoon, and Tsachy Weissman. Dude-seq: Fast, flexible, and robust denoising of nucleotide sequences. *arXiv preprint arXiv:1511.04836*, 2016.

[143] Intae Lee. Sample-spacings-based density and entropy estimators for spherically invariant multidimensional data. *Neural Computation*, 22(8):2208–2227, 2010.

[144] Jaesung Lee and Dae-Won Kim. Feature selection for multi-label classification using multivariate mutual information. *Pattern Recognition Letters*, 34(3):349–357, 2013.

[145] Rasko Leinonen, Hideaki Sugawara, and Martin Shumway. The sequence read archive. *Nucleic acids research*, page gkq1019, 2010.

[146] Marek Leśniewicz. Expected entropy as a measure and criterion of randomness of binary sequences. *Przegląd Elektrotechniczny*, 90(1):42–46, 2014.

[147] Qian Li. sctour: a deep learning architecture for robust inference and accurate prediction of cellular dynamics. *Genome Biology*, 24(1):1–33, 2023.

[148] Wei Li, Jianxing Feng, and Tao Jiang. IsoLasso: A LASSO regression approach to RNA-seq based transcriptome assembly. *Journal of Computational Biology*, 18(11):1693–1707, November 2011.

[149] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 2021.

[150] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[151] Aleksandr Mikhailovich Liapunov. *Stability of motion*. Elsevier, 2016.

[152] Huawen Liu, Jigui Sun, Lei Liu, and Huijie Zhang. Feature selection with dynamic mutual information. *Pattern Recognition*, 42(7):1330–1339, 2009.

[153] Serena Liu and Cole Trapnell. Single-cell transcriptome sequencing: recent advances and remaining challenges. *F1000Research*, 5, 2016.

[154] Clive R Loader. Local likelihood density estimation. *The Annals of Statistics*, 24(4):1602–1618, 1996.

[155] John C Loehlin. *Latent variable models: An introduction to factor, path, and structural analysis* . Lawrence Erlbaum Associates Publishers, 1998.

[156] M Loève. Probability theory. 1977, 1977.

[157] Damiano Lombardi and Sanjay Pant. Nonparametric k-nearest-neighbor entropy estimator. *Physical Review E*, 93(1):013310, 2016.

[158] John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, Barbara Foster, Mike Moser, Ellen Karasik, Bryan Gillard, Kimberley Ramsey, Susan Sullivan, Jason Bridge, Harold Magazine, John Syron, Johnelle Fleming, Laura Siminoff, Heather Traino, Maghboeba Mosavel, Laura Barker, Scott Jewell, Dan Rohrer, Dan Maxim, Dana Filkins, Philip Harbach, Eddie Cortadillo, Bree Berghuis, Lisa Turner, Eric Hudson, Kristin Feenstra, Leslie Sobin, James Robb, Phillip Branton, Greg Korzeniewski, Charles Shive, David Tabor, Liqun Qi, Kevin Groch, Sreenath Nampally, Steve Buia, Angela Zimmerman, Anna Smith, Robin Burges, Karna Robinson, Kim Valentino, Deborah Bradbury, Mark Cosentino, Norma Diaz-Mayoral, Mary Kennedy, Theresa Engel, Penelope Williams, Kenyon Erickson, Kristin Ardlie, Wendy Winckler, Gad Getz, David DeLuca, Daniel MacArthur, Manolis Kellis, Alexander Thomson, Taylor Young, Ellen Gelfand, Molly Donovan, Yan Meng, George Grant, Deborah Mash, Yvonne Marcus, Margaret Basile, Jun Liu, Jun Zhu, Zhidong Tu, Nancy J. Cox, Dan L. Nicolae, Eric R. Gamazon, Hae Kyung Im, Anuar Konkashbaev, Jonathan Pritchard, Matthew Stevens, Timothèe Flutre, Xiaoquan Wen, Emmanouil T. Dermitzakis, Tuuli Lappalainen, Roderic Guigo, Jean Monlong, Michael Sammeth, Daphne Koller, Alexis Battle, Sara Mostafavi, Mark McCarthy, Manual Rivas, Julian Maller, Ivan Rusyn, Andrew Nobel, Fred Wright, Andrey Shabalin, Mike Feolo, Nataliya Sharopova, Anne Sturcke, Justin Paschal, James M. Anderson, Elizabeth L. Wilder, Leslie K. Derr, Eric D. Green, Jeffery P. Struewing, Gary Temple, Simona Volpi, Joy T. Boyer, Elizabeth J. Thomson, Mark S. Guyer, Cathy Ng, Assya Abdallah, Deborah Colantuoni, Thomas R. Insel, Susan E. Koester, A. Roger Little, Patrick K. Bender, Thomas Lehner, Yin Yao,

Carolyn C. Compton, Jimmie B. Vaught, Sherilyn Sawyer, Nicole C. Lockhart, Joanne Demchok, and Helen F. Moore. The genotype-tissue expression (GTEx) project. *Nature Genetics*, 45(6):580–585, June 2013.

[159] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[160] Jiayi Ma, Ji Zhao, Jinwen Tian, Xiang Bai, and Zhuowen Tu. Regularized vector field learning with sparse approximation for mismatch removal. *Pattern Recognition*, 46(12):3519–3532, 2013.

[161] Wenzhe Ma, Ala Trusina, Hana El-Samad, Wendell A Lim, and Chao Tang. Defining network topologies that can achieve biochemical adaptation. *Cell*, 138(4):760–773, 2009.

[162] Xuehua Ma, Zhiguang Zhao, Long Xiao, Weina Xu, Yahui Kou, Yanping Zhang, Gang Wu, Yangyang Wang, and Zhuo Du. A 4d single-cell protein atlas of transcription factors delineates spatiotemporal patterning during embryogenesis. *Nature Methods*, 18(8):893–902, 2021.

[163] Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, and J.J. Trombetta. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.

[164] Christopher A. Maher, Chandan Kumar-Sinha, Xuhong Cao, Shanker Kalyana-Sundaram, Bo Han, Xiaojun Jing, Lee Sam, Terrence Barrette, Nallasivam Palanisamy, and Arul M. Chinnaiyan. Transcriptome sequencing to detect gene fusions in cancer. *Nature*, 458(7234):97–101, March 2009.

[165] Guillaume Marçais, James A Yorke, and Aleksey Zimin. Quorum: an error corrector for illumina reads. *PloS one*, 10(6):e0130821, 2015.

[166] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In *BMC bioinformatics*, volume 7, pages 1–15. BioMed Central, 2006.

[167] James Massey. Causality, feedback and directed information. In *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*, pages 303–305. Citeseer, 1990.

[168] James Massey. Causality, feedback and directed information. In *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*, pages 303–305. Citeseer, 1990.

[169] Hirotaka Matsumoto, Hisanori Kiryu, Chikara Furusawa, Minoru SH Ko, Shigeru BH Ko, Norio Gouda, Tetsutaro Hayashi, and Itoshi Nikaido. Scode: an efficient regulatory network inference algorithm from single-cell rna-seq during differentiation. *Bioinformatics*, 33(15):2314–2321, 2017.

[170] COLIN McDIARMID. Clutter percolation and random graphs. In *Combinatorial Optimization II*, pages 17–25. Springer, 1980.

[171] William McGill. Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory*, 4(4):93–111, 1954.

[172] Marina Meila and Michael I Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1(Oct):1–48, 2000.

[173] Patrick E Meyer, Frederic Lafitte, and Gianluca Bontempi. minet: Ar/bioconductor package for inferring large transcriptional networks using mutual information. *BMC bioinformatics*, 9(1):461, 2008.

[174] Patrick Emmanuel Meyer, Colas Schretter, and Gianluca Bontempi. Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):261–274, 2008.

[175] Olgica Milenkovic, Gil Alterovitz, Gerard Battail, Todd P Coleman, Joachim Hagenauer, Sean P Meyn, Nathan Price, Marco F Ramoni, Ilya Shmulevich, and Wojciech Szpankowski. Introduction to the special issue on information theory in molecular biology and neuroscience, 2010.

[176] Erik G Miller. A new class of entropy estimators for multi-dimensional densities. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–297. IEEE, 2003.

[177] Rudy Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal processing*, 16(3):233–248, 1989.

[178] Rudy Moddemeijer. A statistic to estimate the variance of the histogram-based mutual information estimator based on dependent pairs of observations. *Signal Processing*, 75(1):51–63, 1999.

[179] J.M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research*, 2015.

[180] Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, 17(1):1103–1204, 2016.

[181] Kevin R Moon, Kumar Sricharan, and Alfred O Hero III. Ensemble estimation of mutual information. *arXiv preprint arXiv:1701.08083*, 2017.

[182] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5(7):621–628, 2008.

[183] Elchanan Mossel and Sebastien Roch. Incomplete lineage sorting: consistent phylogeny estimation from multiple loci. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(1):166–171, 2010.

[184] A. Motahari, G. Bresler, and D. Tse. Information theory for DNA sequencing: Part i: A basic model. In *2012 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 2741–2745, July 2012. PMCID not available.

[185] Abolfazl Motahari, Kannan Ramchandran, David Tse, and Nan Ma. Optimal DNA shotgun sequencing: Noisy reads are as good as noiseless reads. In *IEEE International Symposium on Information Theory*, 2013. PMCID not available.

[186] A.S. Motahari, G. Bresler, and D.N.C. Tse. Information theory of DNA shotgun sequencing. *IEEE Transactions on Information Theory*, 59(10):6273–6289, October 2013. PMCID not available.

[187] Jonas Mueller, Tommi Jaakkola, and David Gifford. Modeling trends in distributions. *arXiv preprint arXiv:1511.04486*, 2015.

[188] Matthias Muhar, Anja Ebert, Tobias Neumann, Christian Umkehrer, Julian Jude, Corinna Wieshofer, Philipp Rescheneder, Jesse J Lipp, Veronika A Herzog, Brian Reichholf, et al. Slam-seq defines direct gene-regulatory functions of the brd4-myc axis. *Science*, 360(6390):800–805, 2018.

[189] John Isaac Murray, Thomas J Boyle, Elicia Preston, Dionne Vafeados, Barbara Mericle, Peter Weisdepp, Zhongying Zhao, Zhirong Bao, Max Boeck, and Robert H Waterston. Multidimensional regulation of gene expression in the c. elegans embryo. *Genome research*, 22(7):1282–1294, 2012.

[190] Ilya Nemenman, Fariel Shafee, and William Bialek. Entropy and inference, revisited. In *Advances in neural information processing systems*, pages 471–478, 2002.

[191] Morteza Noshad and Alfred O Hero III. Scalable hash-based estimation of divergence measures. *arXiv preprint arXiv:1801.00398*, 2018.

[192] Andrea Ocone, Laleh Haghverdi, Nikola S Mueller, and Fabian J Theis. Reconstructing gene regulatory dynamics from high-dimensional single-cell snapshot data. *Bioinformatics*, 31(12):i89–i96, 2015.

[193] Andre Olsson, Meenakshi Venkatasubramanian, Viren K Chaudhri, Bruce J Aronow, Nathan Salomonis, Harinder Singh, and H Leighton Grimes. Single-cell analysis of mixed-lineage states leading to a binary cell fate choice. *Nature*, 537(7622):698, 2016.

[194] Mícheál O'Searcoid. *Metric spaces*. Springer Science & Business Media, 2006.

[195] Art B. Owen. *Monte Carlo theory, methods and examples*. Stanford, 2013.

[196] Melissa Owraghi, Gina Broitman-Maduro, Thomas Luu, Heather Roberson, and Morris F Maduro. Roles of the wnt effector pop-1/tcf in the c. elegans endomesoderm specification gene network. *Developmental biology*, 340(2):209–221, 2010.

[197] Dávid Pál, Barnabás Póczos, and Csaba Szepesvári. Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graphs. In *Advances in Neural Information Processing Systems*, pages 1849–1857, 2010.

[198] I Palásti. On the strong connectedness of directed random graphs. *Studia Sci. Math. Hungar*, 1:205–214, 1966.

[199] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003.

[200] Nan Papili Gao, SM Minhaz Ud-Dean, Olivier Gandrillon, and Rudiyanto Gunawan. Sincerities: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics*, 34(2):258–266, 2017.

[201] Franziska Paul, Ya'ara Arkin, Amir Giladi, Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Deborah Winter, David Lara-Astiaso, Meital Gury, Assaf Weiner, et al. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell*, 163(7):1663–1677, 2015.

[202] Judea Pearl. *Causality*. Cambridge university press, 2009.

[203] Judea Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[204] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[205] Fernando Pérez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 1666–1670. IEEE, 2008.

[206] Haim H Permuter, Young-Han Kim, and Tsachy Weissman. Interpretations of directed information in portfolio theory, data compression, and hypothesis testing. *IEEE Transactions on Information Theory*, 57(6):3248–3259, 2011.

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

[208] Pavel Pevzner. *Computational molecular biology: an algorithmic approach*. MIT press, 2000.

[209] Hannah Pliner, Jonathan Packer, Jose McFaline-Figueroa, Darren Cusanovich, Riza Daza, Sanjay Srivatsan, Xiaojie Qiu, Dana Jackson, Anna Minkina, Andrew Adey, et al. Chromatin accessibility dynamics of myogenesis at single cell resolution. *BioRxiv*, page 155473, 2017.

[210] Barnabás Póczos, Liang Xiong, and Jeff Schneider. Nonparametric divergence estimation with applications to machine learning on distributions. *arXiv preprint arXiv:1202.3758*, 2012.

[211] Yury Polyanskiy and Sergio Verdú. Arimoto channel coding converse and rényi divergence. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1327–1333. IEEE, 2010.

[212] Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. *Lecture Notes for ECE563 (UIUC) and*, 6(2012-2016):7, 2014.

[213] Yury Polyanskiy and Yihong Wu. Strong data-processing inequalities for channels and bayesian networks. In *Convexity and Concentration*, pages 211–249. Springer, 2017.

[214] Bernd Pompe, Pierre Blidh, Dirk Hoyer, and Michael Eiselt. Using mutual information to measure coupling in the cardiorespiratory system. *IEEE Engineering in Medicine and Biology Magazine*, 17(6):32–39, 1998.

[215] Robert J Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K Sorger, Leonidas G Alexopoulos, Xiaowei Xue, Neil D Clarke, Gregoire Altan-Bonnet, and Gustavo Stolovitzky. Towards a rigorous assessment of systems biology models: the dream3 challenges. *PloS one*, 5(2):e9202, 2010.

[216] Xiaojie Qiu, Shanshan Ding, and Tieliu Shi. From understanding the development landscape of the canonical fate-switch pair to constructing a dynamic landscape for two-step neural differentiation. *PloS one*, 7(12):e49271, 2012.

[217] Xiaojie Qiu, Andrew Hill, Jonathan Packer, Dejun Lin, Yi-An Ma, and Cole Trapnell. Single-cell mrna quantification and differential analysis with census. *Nature methods*, 14(3):309, 2017.

[218] Xiaojie Qiu, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah A Pliner, and Cole Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nature methods*, 14(10):979, 2017.

[219] Xiaojie Qiu, Arman Rahimzamani, Li Wang, Bingcheng Ren, Qi Mao, Timothy Durham, José L. McFaline-Figueroa, Lauren Saunders, Cole Trapnell, and Sreeram Kannan. Inferring causal gene regulatory networks from coupled single-cell expression dynamics using scribe. *Cell Systems*, 10(3):265–274.e11, 2020.

[220] Xiaojie Qiu, Yan Zhang, Jorge D Martin-Rufino, Chen Weng, Shayan Hosseinzadeh, Dian Yang, Angela N Pogson, Marco Y Hein, Kyung Hoi Joseph Min, Li Wang, et al. Mapping transcriptomic vector fields of single cells. *Cell*, 185(4):690–711, 2022.

[221] Christopher J Quinn, Todd P Coleman, Negar Kiyavash, and Nicholas G Hatsopoulos. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *Journal of computational neuroscience*, 30(1):17–44, 2011.

[222] Christopher J Quinn, Negar Kiyavash, and Todd P Coleman. Directed information graphs. *IEEE Transactions on information theory*, 61(12):6887–6909, 2015.

[223] Christopher J Quinn, Negar Kiyavash, and Todd P Coleman. Directed information graphs. *IEEE Transactions on Information Theory*, 61(12):6887–6909, 2015.

[224] Arman Rahimzamani and Sreeram Kannan. Network inference using directed information: The deterministic limit. In *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*, pages 156–163. IEEE, 2016.

[225] Arman Rahimzamani and Sreeram Kannan. Potential conditional mutual information: Estimators and properties. In *Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on*, pages 1228–1235. IEEE, 2017.

[226] Arman Rahimzamani and Sreeram Kannan. Potential conditional mutual information: Estimators and properties. *arXiv preprint*, 2017.

[227] Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian processes for machine learning*, volume 1. Springer, 2006.

[228] Jelena Raspopovic, Luciano Marcon, Laura Russo, and James Sharpe. Digit patterning is controlled by a bmp-sox9-wnt turing network modulated by morphogen gradients. *Science*, 345(6196):566–570, 2014.

[229] Jingyi Ren, Haowen Zhou, Hu Zeng, Connie Kangni Wang, Jiahao Huang, Xiaojie Qiu, Xin Sui, Qiang Li, Xunwei Wu, Zuwan Lin, et al. Spatiotemporally resolved transcriptomics reveals the subcellular rna kinetic landscape. *Nature Methods*, pages 1–11, 2023.

[230] Alfréd Rényi. On measures of dependence. *Acta mathematica hungarica*, 10(3-4):441–451, 1959.

[231] David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. Detecting novel associations in large data sets. *science*, 334(6062):1518–1524, 2011.

[232] Robin J Richardson and Thomas S Evans. Non-parametric causal models. *UAI Tutorial*, 2015.

[233] Christian Riml, Thomas Amort, Dietmar Rieder, Catherina Gasser, Alexandra Lusser, and Ronald Micura. Osmium-mediated transformation of 4-thiouridine to cytidine as key to study rna dynamics by sequencing. *Angewandte Chemie International Edition*, 56(43):13479–13483, 2017.

[234] Gordon Robertson, Jacqueline Schein, Readman Chiu, Richard Corbett, Matthew Field, Shaun D. Jackman, Karen Mungall, Sam Lee, Hisanaga Mark Okada, Jenny Q. Qian, Malachi Griffith, Anthony Raymond, Nina Thiessen, Timothee Cezard, Yaron S. Butterfield, Richard Newsome, Simon K. Chan, Rong She, Richard Varhol, Baljit Kamoh, Anna-Liisa Prabhu, Angela Tam, YongJun Zhao, Richard A. Moore, Martin Hirst, Marco A. Marra, Steven J. M. Jones, Pamela A. Hoodless, and Inanc Birol. De novo assembly and analysis of RNA-seq data. *Nature Methods*, 7(11):909–912, November 2010.

[235] Édgar Roldán. Estimating the kullback–leibler divergence. In *Irreversibility and Dissipation in Microscopic Systems*, pages 61–85. Springer, 2014.

[236] Alexander B Rosenberg, Rupali P Patwardhan, Jay Shendure, and Georg Seelig. Learning the sequence determinants of alternative splicing from millions of random sequences. *Cell*, 163(3):698–711, 2015.

[237] Jakob Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. *arXiv preprint arXiv:1709.01447*, 2017.

[238] Manuel Sanchez-Castillo, David Blanco, Isabel M Tienda-Luna, MC Carrion, and Yufei Huang. A bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data. *Bioinformatics*, 34(6):964–970, 2017.

[239] Eren Sasoglu and David Tse. DNA assembly from paired reads as 2-d jigsaw puzzles. In *International Symposium on Information Theory, submitted*, 2014. PMCID not available.

[240] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[241] Thomas Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461, 2000.

[242] Marcel H. Schulz, Daniel R. Zerbino, Martin Vingron, and Ewan Birney. Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics*, 28(8):1086–1092, April 2012.

[243] Manu Setty, Michelle D Tadmor, Shlomit Reich-Zeliger, Omer Angel, Tomer Meir Salame, Pooja Kathail, Kristy Choi, Sean Bendall, Nir Friedman, and Dana Pe'er. Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nature biotechnology*, 34(6):637, 2016.

[244] Sheel Shah, Yodai Takei, Wen Zhou, Eric Lubeck, Jina Yun, Chee-Huat Linus Eng, Noushin Koulena, Christopher Cronin, Christoph Karp, Eric J Liaw, et al. Dynamics and spatial genomics of the nascent transcriptome by intron seqfish. *Cell*, 174(2):363–376, 2018.

[245] N. Shajarisales, D. Janzing, B. Schölkopf, and M. Besserve. Telling cause from effect in deterministic linear dynamical systems. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 285–294. JMLR, 2015.

[246] Alex K Shalek, Rahul Satija, Joe Shuga, John J Trombetta, Dave Gennert, Diana Lu, Peilin Chen, Rona S Gertner, Jellert T Gaublomme, Nir Yosef, et al. Single-cell rna-seq reveals dynamic paracrine control of cellular variation. *Nature*, 510(7505):363, 2014.

[247] C.E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423 and 623–656, 1948.

[248] Claude E Shannon. Prediction and entropy of printed english. *Bell Labs Technical Journal*, 30(1):50–64, 1951.

[249] Simon J Sheather and Michael C Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690, 1991.

[250] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

[251] Yakov G Sinai. Gibbs measures in ergodic theory. *Russian Mathematical Surveys*, 27(4):21, 1972.

[252] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.

[253] Shashank Singh and Barnabás Póczos. Exponential concentration of a density functional estimator. In *Advances in Neural Information Processing Systems*, pages 3032–3040, 2014.

[254] Shashank Singh and Barnabás Póczos. Finite-sample analysis of fixed-k nearest neighbor density functional estimators. In *Advances in Neural Information Processing Systems*, pages 1217–1225, 2016.

[255] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[256] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[257] Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. *Causation, prediction, and search*. MIT press, 2000.

[258] K Sreeram, S Birenjith, and P Vijay Kumar. Dmt of multihop networks: end points and computational tools. *IEEE Transactions on Information Theory*, 58(2):804–819, 2012.

[259] Kumar Sricharan, Raviv Raich, and Alfred O Hero. Estimation of nonlinear functionals of densities with confidence. *IEEE Transactions on Information Theory*, 58(7):4135–4159, 2012.

[260] Kumar Sricharan, Raviv Raich, and Alfred O Hero III. Empirical estimation of entropy functionals with confidence. *arXiv preprint arXiv:1012.4188*, 2010.

[261] Kumar Sricharan, Dennis Wei, and Alfred O Hero. Ensemble estimators for multivariate entropy estimation. *IEEE transactions on information theory*, 59(7):4374–4388, 2013.

[262] Angelike Stathopoulos and Dagmar Iber. Studies of morphogens: keep calm and carry on, 2013.

[263] Tomas Stopka, Derek F Amanatullah, Michael Papetti, and Arthur I Skoultchi. Pu. 1 inhibits the erythroid program by binding to gata-1 on dna and creating a repressive chromatin structure. *The EMBO journal*, 24(21):3712–3723, 2005.

[264] Milan Studenỳ and Jirina Vejnarová. The multiinformation function as a tool for measuring stochastic dependence. In *Learning in graphical models*, pages 261–297. Springer, 1998.

[265] George Sugihara, Robert May, Hao Ye, Chih-hao Hsieh, Ethan Deyle, Michael Fogarty, and Stephan Munch. Detecting causality in complex ecosystems. *science*, 338(6106):496–500, 2012.

[266] Jie Sun, Dane Taylor, and Erik M Bollt. Causal network inference by optimal causation entropy. *SIAM Journal on Applied Dynamical Systems*, 14(1):73–106, 2015.

[267] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nature protocols*, 13(4):599–604, 2018.

[268] D Szklarczyk, JH Morris, H Cook, M Kuhn, S Wyder, M Simonovic, et al. The 1110 string database in 2017: quality-controlled protein-protein association networks, 1111 made broadly accessible. *Nucleic Acids Res. Oxford University Press*, 2017.

[269] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[270] Tomohiko Tamura, Daisuke Kurotaki, and Shin-ichi Koizumi. Regulation of myelopoiesis by the transcription factor irf8. *International journal of hematology*, 101(4):342–351, 2015.

[271] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[272] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *Information Theory Workshop (ITW), 2015 IEEE*, pages 1–5. IEEE, 2015.

[273] Alexandru I. Tomescu, Anna Kuosmanen, Romeo Rizzi, and Veli Mäkinen. A novel min-cost flow method for estimating transcript expression with RNA-seq. *BMC Bioinformatics*, 14(Suppl 5):S15, April 2013.

[274] Cole Trapnell. Defining cell types and states with single-cell genomics. *Genome research*, 25(10):1491–1498, 2015.

[275] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381–386, 2014.

[276] Cole Trapnell, Brian A. Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J. van Baren, Steven L. Salzberg, Barbara J. Wold, and Lior Pachter. Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):511–515, May 2010.

[277] Barbara Treutlein, Doug G Brownfield, Angela R Wu, Norma F Neff, Gary L Mantalas, F Hernan Espinoza, Tushar J Desai, Mark A Krasnow, and Stephen R Quake. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell rna-seq. *Nature*, 509(7500):371, 2014.

[278] Alexandre B Tsybakov and EC Van der Meulen. Root-n consistent estimators of entropy for densities with unbounded support. *Scandinavian Journal of Statistics*, pages 75–83, 1996.

[279] Warwick Tucker. A rigorous ode solver and smale's 14th problem. *Foundations of Computational Mathematics*, 2(1):53–117, 2002.

[280] Alan Mathison Turing. The chemical basis of morphogenesis. *Bulletin of mathematical biology*, 52:153–197, 1990.

[281] Gregory Valiant and Paul Valiant. Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new clts. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 685–694. ACM, 2011.

[282] Tim Van Erven and Peter Harremoës. Rényi divergence and kullback-leibler divergence. *Information Theory, IEEE Transactions on*, 60(7):3797–3820, 2014.

[283] Benedicte Vatinlen, Fabrice Chauvet, Philippe Chrétienne, and Philippe Mahey. Simple bounds and greedy algorithms for decomposing a flow into a minimal set of paths. *European Journal of Operational Research*, 185(3):1390–1401, 2008.

[284] Martin Vejmelka and Milan Paluš. Inferring the directionality of coupling with conditional mutual information. *Physical Review E*, 77(2):026214, 2008.

[285] Jorge R Vergara and Pablo A Estévez. Cmim-2: an enhanced conditional mutual information maximization criterion for feature selection. *Journal of Applied Computer Science Methods*, 2, 2010.

[286] Jorge R Vergara and Pablo A Estévez. A review of feature selection methods based on mutual information. *Neural computing and applications*, 24(1):175–186, 2014.

[287] Hoi-To Wai, Anna Scaglione, Uzi Harush, Baruch Barzel, and Amir Leshem. Rids: Robust identification of sparse gene regulatory networks from perturbation experiments. *arXiv preprint arXiv:1612.06565*, 2016.

[288] Chih-Chun Wang and David Tse. Assembly with confidence. Draft, Personal communication.

[289] Jin Wang, Li Xu, Erkang Wang, and Sui Huang. The potential landscape of genetic circuits imposes the arrow of time in stem cell differentiation. *Biophysical journal*, 99(1):29–39, 2010.

[290] Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation for multidimensional densities via-nearest-neighbor distances. *Information Theory, IEEE Transactions on*, 55(5):2392–2405, 2009.

[291] Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63, 2009.

[292] Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.

[293] Jiangyong Wei, Xiaohua Hu, Xiufen Zou, and Tianhai Tian. Reverse-engineering of gene networks for regulating early blood development from single-cell measurements. *BMC medical genomics*, 10(5):72, 2017.

[294] Caleb Weinreb, Alejo Rodriguez-Fraticelli, Fernando D Camargo, and Allon M Klein. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367(6479):eaaw3381, 2020.

[295] Caleb Weinreb, Samuel Wolock, Betsabeh K Tusi, Merav Socolovsky, and Allon M Klein. Fundamental limits on dynamic inference from single-cell snapshots. *Proceedings of the National Academy of Sciences*, 115(10):E2467–E2476, 2018.

[296] Joe Whittaker. *Graphical models in applied multivariate statistics*. Wiley Publishing, 2009.

[297] Robert Wieczorkowski and Przemysiaw Grzegorzewski. Entropy estimators-improvements and comparisons. *Communications in Statistics-Simulation and Computation*, 28(2):541–567, 1999.

[298] Tobias Wiesenfahrt, Janette Y. Berg, Erin Osborne Nishimura, Adam G. Robinson, Barbara Goszczynski, Jason D. Lieb, and James D. McGhee. The function and regulation of the GATA factor ELT-2 in the C. elegans endoderm. *Development*, 143(3):483–491, 02 2016.

[299] Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2):109–130, 2001.

[300] Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6):3702–3720, 2016.

[301] Aaron D Wyner. A definition of conditional mutual information for arbitrary ensembles. *Information and Control*, 38(1):51–59, 1978.

[302] Yinlong Xie, Gengxiong Wu, Jingbo Tang, Ruibang Luo, Jordan Patterson, Shanlin Liu, Weihua Huang, Guangzhu He, Shengchang Gu, Shengkang Li, Xin Zhou, Tak-Wah Lam, Yingrui Li, Xun Xu, Gane Ka-Shu Wong, and Jun Wang. Soapdenovo-trans: De novo transcriptome assembly with short rna-seq reads. *Bioinformatics*, 2014.

[303] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*, 2012.

[304] Xiuwei Zhang, Chenling Xu, and Nir Yosef. Simulating multiple faceted variability in single cell rna sequencing. *Nature communications*, 10(1):2611, 2019.

[305] Zheng Zhang, Zhihui Lai, Yong Xu, Ling Shao, Jian Wu, and Guo-Sen Xie. Discriminative elastic-net regularized linear regression. *IEEE Transactions on Image Processing*, 26(3):1466–1481, 2017.

[306] Lizhong Zheng and David N. C. Tse. Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels. *IEEE Transactions on information theory*, 49(5):1073–1096, 2003.

[307] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[308] Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, 24(5):530–536, 1978.

[309] Cunlu Zou and Jianfeng Feng. Granger causality vs. dynamic bayesian network inference: a comparative study. *BMC bioinformatics*, 10(1):122, 2009.

Appendix A

SUPPLEMENTARY MATERIAL FOR CHAPTER 2

A.1 Multivariate Mutual Information

In [29], Multivariable Mutual Information (MMI) is defined as follows: let $\Pi(\mathcal{X})$ be the collection of all possible partitions of $\mathcal{P}$ which split $\mathcal{X}$ into at least two non-empty disjoint subsets. For any partition $\mathcal{P} \in \Pi(\mathcal{X})$, the product distribution $\Pi_{C\in\mathcal{P}}\mathbb{P}_{X_C}$ specifies an independence relation, i.e., X_C's are treated as agglomerated random variables and are mutually independent. Given a particular partition, define an information measure $I_{\mathcal{P}}(X)$ as :

$$I_{\mathcal{P}}(X) = \frac{1}{\mathcal{P}-1} D(\mathbb{P}_X \parallel \Pi_{C\in\mathcal{P}}\mathbb{P}_{X_C}) \tag{A.1}$$

Then, MMI is defined as:

$$\text{MMI(X)} = \min_{\mathcal{P}\in\Pi(\mathcal{X})} I_{\mathcal{P}}(X) \tag{A.2}$$

This can cast as a functional of our Graph Divergence Measure by choosing for every partition, $\mathcal{P} \in \Pi(\mathcal{X})$, a DAG $\mathcal{G}_{\mathcal{P}}$ with all X_C's forming an aggregate node but disconnected from each other and thus inducing a measure $\overline{\mathbb{P}}_X^{\mathcal{P}}$. Thus,

$$I_{\mathcal{P}}(X) = \frac{1}{\mathcal{P}-1} \mathbb{GDMM}(\mathbb{P}_X \parallel \overline{\mathbb{P}}_X^{\mathcal{P}}) \tag{A.3}$$

which implies,

$$\text{MMI(X)} = \min_{\mathcal{P}\in\Pi(\mathcal{X})} \mathbb{GDMM}(\mathbb{P}_X \parallel \overline{\mathbb{P}}_X^{\mathcal{P}}) \tag{A.4}$$

A.2 Directed Information

In this section, we will derive the expression of the directed information from $X^T = (X_1, \ldots, X_T)$ to $Y^T = (Y_1, \ldots, Y_T)$ in terms of two graph divergence measures. For the simplicity of notations and logic we will only do it assuming X and Y are discrete. However, it's easily extendable to the case of mixture distributions using the notion of Radon-Nikodym derivative. From the definition of the directed information we have:

$$
\begin{aligned}
& I\left(X^T \to Y^T\right) && \text{(A.5)}\\
= & \sum_{t=1}^{T} I\left(X^t; Y_t \middle| Y_{t-1}\right) && \text{(A.6)}\\
= & \sum_{t=1}^{T} \sum_{x^t, y^t} \left(\mathbb{P}_{X^t Y^t}(x^t, y^t) \log \frac{\mathbb{P}_{Y_t|X^t, Y^{t-1}}(y_t|x^t, y^{t-1})}{\mathbb{P}_{Y_t|Y^{t-1}}(y_t|y^{t-1})} \right) && \text{(A.7)}\\
= & \sum_{x^T, y^T} \mathbb{P}_{X^T Y^T}(x^T, y^T) \sum_{t=1}^{T} \log \frac{\mathbb{P}_{Y_t|X^t, Y^{t-1}}(y_t|x^t, y^{t-1})}{\mathbb{P}_{Y_t|Y^{t-1}}(y_t|y^{t-1})} && \text{(A.8)}\\
= & \sum_{x^T, y^T} \mathbb{P}_{X^T Y^T}(x^T, y^T) \log \frac{\prod_{t=1}^{T} \mathbb{P}_{Y_t|X^t, Y^{t-1}}(y_t|x^t, y^{t-1})}{\prod_{t=1}^{T} \mathbb{P}_{Y_t|Y^{t-1}}(y_t|y^{t-1})} && \text{(A.9)}\\
= & \sum_{x^T, y^T} \mathbb{P}_{X^T Y^T}(x^T, y^T) \log \frac{\prod_{t=1}^{T} \mathbb{P}_{Y_t|X^t, Y^{t-1}}(y_t|x^t, y^{t-1})}{\mathbb{P}_{Y^T}(y^T)} && \text{(A.10)}\\
= & \sum_{x^T, y^T} \mathbb{P}_{X^T Y^T}(x^T, y^T) \log \frac{\prod_{t=1}^{T} \mathbb{P}_{Y_t|X^t, Y^{t-1}}(y_t|x^t, y^{t-1}) \mathbb{P}_{X_t|X^{t-1}}(x_t|x^{t-1})}{\mathbb{P}_{X^T}(x^T) \mathbb{P}_{Y^T}(y^T)} && \text{(A.11)}\\
= & \sum_{x^T, y^T} \mathbb{P}_{X^T Y^T}(x^T, y^T) \log \frac{\mathbb{P}_{X^T Y^T}(x^T, y^T)}{\mathbb{P}_{X^T}(x^T) \mathbb{P}_{Y^T}(y^T)} \\
& - \sum_{x^T, y^T} \mathbb{P}_{X^T Y^T}(x^T, y^T) \log \frac{\mathbb{P}_{X^T Y^T}(x^T, y^T)}{\prod_{t=1}^{T} \mathbb{P}_{Y_t|X^t, Y^{t-1}}(y_t|x^t, y^{t-1}) \mathbb{P}_{X_t|X^{t-1}}(x_t|x^{t-1})} && \text{(A.12)}\\
= & D\left(\mathbb{P}_{X^T Y^T} \| \mathbb{P}^I_{X^T Y^T}\right) - D\left(\mathbb{P}_{X^T Y^T} \| \mathbb{P}^C_{X^T Y^T}\right) && \text{(A.13)}\\
= & \mathbb{GDM}\Big((X^T, Y^T), \mathcal{G}_I\Big) - \mathbb{GDM}\Big((X^T, Y^T), \mathcal{G}_C\Big) && \text{(A.14)}
\end{aligned}
$$

The distributions $\mathbb{P}^I_{X^T Y^T} = \mathbb{P}_{X^T}\mathbb{P}_{Y^T}$ and $\mathbb{P}^C_{X^T Y^T} = \prod_{t=1}^T \mathbb{P}_{Y_t|X^t,Y^{t-1}}\mathbb{P}_{X_t|X^{t-1}}$ represent the *independent* distribution between X^T and Y^T, and the *causal* distribution from X^T to Y^T respectively.

A.3 Investigating assumptions in Theorem 1

In this section, we will investigate the validity of the assumptions for various types of distributions. In particular, we will investigate the different types of distributions we had mentioned: (1) finitely discrete; (2) continuous; (3) finitely discrete over some dimensions while continuous over others; (4) a mixture of the previous cases; (5) has a joint density supported over a lower dimensional manifold. As we had mentioned, these cases represent almost all the real world data. We note that the estimator is well defined only when $\mathcal{X} = \mathbb{R}^{d_X}$; i.e. all the alphabets be in real space.

We will examine the Assumptions 1.2 and 1.3 for each case separately. The Assumption 1.1 is a parametric assumption related to the convergence of the algorithm and is not directly related to the distribution of the data. Jointly considered with the Assumption 1.2, it controls the boundary effects between the continuous and the discrete regions.

A.3.1 Finitely discrete distribution

For a *finitely discrete* distribution the Assumption 1.2 holds by definition. The Assumption 1.3 trivially holds since the size of sample space $|\mathcal{X}|$ is finite.

A.3.2 Continuous distribution

For a continuous distribution, the Assumption 1.2 holds since there is no discrete component.

The distribution is absolutely continuous with respect to the *Lebesgue* measure λ meaning that $\mathbb{P}_X(A) = 0$ for any subset $A \subset \mathcal{X}$ implies $\lambda(A) = 0$. Thus we can conclude that there exists a density $g : \mathcal{X} \to \mathbb{R}^+$ such that $\mathbb{P}_X(A) = \int_A g \, d\lambda$. Naturally $\mathbb{P}_X(\mathcal{X}) = \int_{\mathcal{X}} g \, d\lambda = 1$.

Furthermore if the density function g is bounded everywhere and the variable has a bounded support, the Assumption 1.3 is fulfilled.

A.3.3 Finitely discrete over some dimensions while continuous over others

In this case, the variable set X can be decomposed to two sets X^D and X^C representing the finitely discrete and continuous dimensions respectively. So for any realization of the discrete dimensions $X^D = x^D$ the probability mass function $h_{X^D}(x^D)$ exists, and a conditional density $g_{X^C|X^D}(x^C|x^D)$ is well defined. We can define an auxiliary function $P_{X^C|X^D}(x^C, r|x^D) \equiv \mathbb{P}_{X^C|X^D}\left\{a \in \mathcal{X}^C : \|a - x^C\|_\infty \leq r \middle| X^D = x^D\right\}$. Thus we can write

$$P_X(x^C, x^D, r) = \mathbb{P}_X\left\{(a,b) \in \mathcal{X} : b = x^D, \|a - x^C\|_\infty \leq r\right\} = P_{X^C|X^D}(x^C, r|x^D)h_{X^D}(x^D) \tag{A.15}$$

We will show that if for any x^D the continuous distribution over X^C satisfies the Assumptions 1 as discussed in the Section A.3.2, then we can see that $\mathbb{P}_X$ will also satisfy the assumptions.

Let's define $f_{x^D}(x^C)$ for any fixed x^D as:

$$f_{x^D}(x^C) = \lim_{r\to 0} P_{X^C|X^D}(x^C, r|x^D) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)^C|\mathrm{pa}(X_l)^D}\left({x_{\mathrm{pa}(l)}}^C, r\middle|{x_{\mathrm{pa}(l)}}^D\right)}{P_{\mathrm{pa}+(X_l)^C|\mathrm{pa}+(X_l)^D}\left({x_{\mathrm{pa}+(l)}}^C, r\middle|{x_{\mathrm{pa}+(l)}}^D\right)} \tag{A.16}$$

Assuming the limit exists everywhere. Thus the Radon-Nikodym Derivative f can be written as:

$$\begin{aligned} f(x) &= \lim_{r\to 0} P_X(x^C, x^D, r) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)}\left({x_{\mathrm{pa}(l)}}^C, {x_{\mathrm{pa}(l)}}^D, r\right)}{P_{\mathrm{pa}+(X_l)}\left({x_{\mathrm{pa}+(l)}}^C, {x_{\mathrm{pa}+(l)}}^D, r\right)} && \text{(A.17)} \\ &= \lim_{r\to 0} P_{X^C|X^D}(x^C, r|x^D) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)^C|\mathrm{pa}(X_l)^D}\left({x_{\mathrm{pa}(l)}}^C, r\middle|{x_{\mathrm{pa}(l)}}^D\right)}{P_{\mathrm{pa}+(X_l)^C|\mathrm{pa}+(X_l)^D}\left({x_{\mathrm{pa}+(l)}}^C, r\middle|{x_{\mathrm{pa}+(l)}}^D\right)} \\ &\quad \times h_{X^D}(x^D) \prod_{l=1}^{d} \frac{h_{\mathrm{pa}(X_l)}\left({x_{\mathrm{pa}(l)}}^D\right)}{h_{\mathrm{pa}+(X_l)}\left({x_{\mathrm{pa}+(l)}}^D\right)} && \text{(A.18)} \\ &= f_{x^D}(x^C) \times h_{X^D}(x^D) \prod_{l=1}^{d} \frac{h_{\mathrm{pa}(X_l)}\left({x_{\mathrm{pa}(l)}}^D\right)}{h_{\mathrm{pa}+(X_l)}\left({x_{\mathrm{pa}+(l)}}^D\right)} && \text{(A.19)} \end{aligned}$$

Therefore for Assumption 1.3 we have:

$$\begin{aligned}\int_{\mathcal{X}} |\log f(x)|\, d\mathbb{P}_X \;\leq\; & \sum_{x^D} h_{X^D}(x^D) \int_{\mathcal{X}^C} \left|\log f_{x^D}(x^C)\right| g_{X^C|X^D}(x^C|x^D) dx^C \\ & + \sum_{x^D} h_{X^D}(x^D) \left| \log h_{X^D}(x^D) \prod_{l=1}^{d} \frac{h_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)}{}^D\right)}{h_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)}{}^D\right)} \right| \end{aligned} \tag{A.20}$$

In which we used the fact that $\mathbb{E}\left[\log f(x)\right] = \mathbb{E}_{X^D}\left[\mathbb{E}\left[\log f(x^C, x^D)\middle| X^D = x^D\right]\right]$. The first term above is upper-bounded since we assumed that continuous distribution over X^C satisfies the Assumption 1.3 for any x^D. The second term is upper-bounded since it's a finitely discrete distribution as discussed in Section A.3.1. Thus $\int_{\mathcal{X}} |\log f(x)|\, d\mathbb{P}_X < \infty$ and the Assumption 1.3 holds.

A.3.4 A mixture of the previous cases

In this case, we assume that the probability can be described as a linear combination of a continuous and a discrete distribution, i.e. without loss of generality we can assume that for any subset $A \subset \mathcal{X}$ the distribution can be described as $\mathbb{P}(A) = \alpha_C \mathbb{P}_X^C(A) + \alpha_D \mathbb{P}_X^D(A)$ in which $\alpha_C + \alpha_D = 1$. We note that the $\mathbb{P}_X^D$ does not represent a mass function here since the mass function is only defined over a discrete alphabet $\mathcal{X}^D = \{x_1, ..., x_m\} \subset \mathcal{X}$ while $\mathbb{P}_X^D$ needs to admit the continuous domain $\mathcal{X}$. Thus we relate $\mathbb{P}_X^D$ to the mass function $h_X^D(.)$ as $\mathbb{P}^D(A) = \sum_{x \in \mathcal{X}^D} \mathbf{1}_{\{x \in A\}} h_X^D(x)$. Furthermore we can see that for any $x \in \mathcal{X}^D$ and for r small enough:

$$P_X^D(x, r) \equiv \mathbb{P}_X^D\Big(B_r(x)\Big) = h_X^D(x) \tag{A.21}$$

If $x \notin \mathcal{X}^D$ then for small enouth r we have $P_X^D(x, r) = 0$.

The Assumption 1.2 holds since the discrete distribution $\mathbb{P}_X^D$ is finite, and hence the number of total discrete points will be finite.

For the Assumption 1.3 we have:

$$\int_{\mathcal{X}} |\log f(x)| \, d\mathbb{P}_X = \alpha_C \int_{\mathcal{X}} |\log f(x)| \, d\mathbb{P}_X^C + \alpha_D \sum_{x \in \mathcal{X}^D} h_X^D(x) \, |\log f(x)| \tag{A.22}$$

and if the continuous distribution complies with the assumptions mentioned in Section A.3.2 then the term above is upper-bounded and the Assumption 1.3 holds.

A.3.5 A distribution with a joint density supported over a lower dimensional manifold

This case simply means that the probability distribution $\mathbb{P}_X$ in $\mathcal{X}$ can be mapped to a probability distribution $\mathbb{P}_Y$ in a lower-dimensional space $\mathcal{Y}$ where $d_Y < d_X$, via a one-to-one continuous function $h : \mathcal{X} \to \mathcal{Y}$. If the lower-dimnesional distribution $\mathbb{P}_Y$ is continuous complying with the properties discussed in the Section A.3.2, then it will preserve all the properties through the inverse mapping $h^{-1} : \mathcal{Y} \to \mathcal{X}$ and $\mathbb{P}_X$ hence $\mathbb{P}_X$ will satisfy the Assumptions 1. If the distribution has finite discrete components either as a discrete or as a mixture distribution, the finiteness of the components will be preserved as well and hence $\mathbb{P}_X$ will satisfy the Assumptions 1.

Therefore this category of distributions will satisfy the Assumptions 1 if the distribution $\mathbb{P}_Y$ satisfies them.

A.4 Consistency Proofs

A.4.1 Proof of Theorem 1

First let's generalize the definition of $P_X(x, r)$ in Equation 2.5 to any subset $S \subseteq X$, i.e. for any point $s \in S$ we define:

$$P_S(s, r) = \mathbb{P}_S\big\{a \in \mathcal{S} : \|a - s\|_\infty \leq r\big\} = \mathbb{P}_S\Big\{B_r(s)\Big\} \tag{A.23}$$

Thus $P_S(s, r)$ is the probability of an ℓ_∞ ball of radius r centered at s, or equivalently, a hypercube with the edge length of $2r$ centered at the point s.

To prove the asymptotic unbiasedness of the estimator, we will first write the Radon-Nikodym derivative in an explicit form via the following lemma.

Lemma 16. *For almost every $x \in \mathcal{X}$:*

$$\frac{d\mathbb{P}_X}{d\overline{\mathbb{P}}_X}(x) = f(x) = \lim_{r\to 0} P_X(x,r) \prod_{l=1}^{d} \frac{P_{pa(X_l)}\left(x_{pa(l)}, r\right)}{P_{pa+(X_l)}\left(x_{pa+(l)}, r\right)} \tag{A.24}$$

Proof. Please see the section A.4.2. □

Now notice that $\widehat{\mathbb{GDMM}}^{(N)}(X,\mathcal{G}) = \frac{1}{N}\sum_{i=1}^{N}\zeta_i$ in which all the ζ_i's are identically distributed. Thus we have $\mathbb{E}[\widehat{\mathbb{GDMM}}^{(N)}(X,\mathcal{G})] = \mathbb{E}[\zeta_1]$. Therefore, the bias can be written as:

$$\left|\mathbb{E}\left[\widehat{\mathbb{GDMM}}^{(N)}(X,\mathcal{G})\right] - \mathbb{GDMM}(X,\mathcal{G})\right| = \left|\mathbb{E}_X\left[\mathbb{E}[\zeta_1|X]\right] - \int_{\mathcal{X}} \log f(X) d\mathbb{P}_X\right| \tag{A.25}$$

$$\leq \int_{\mathcal{X}} \left|\mathbb{E}[\zeta_1|X] - \log f(X)\right| d\mathbb{P}_X \tag{A.26}$$

Now we will give upper bounds for $|\mathbb{E}[\zeta_1|X] - \log f(X)|$ for every $x \in \mathcal{X}$. Similar to the technique used by [79], we divide the domain of $\mathcal{X}$ into three parts as $\mathcal{X} = \Omega_1 \bigcup \Omega_2 \bigcup \Omega_3$ where:

- $\Omega_1 = \{x : f(x) = 0\}$
- $\Omega_2 = \{x : f(x) > 0, P_X(x,0) > 0\}$
- $\Omega_3 = \{x : f(x) > 0, P_X(x,0) = 0\}$

For each of the domains, we show that $\lim_{N\to\infty}\int_{\Omega_i}\left|\mathbb{E}[\zeta_1|X=x] - \log f(x)\right| d\mathbb{P}_X = 0$.

For $x \in \Omega_1$:

The probability of Ω_1 is zero with respect to $\mathbb{P}_X$, since:

$$\mathbb{P}_X(\Omega_1) = \int_{\Omega_1} d\mathbb{P}_X = \int_{\Omega_1} f(x) d\overline{\mathbb{P}}_X = \int_{\Omega_1} 0 d\overline{\mathbb{P}}_X = 0 \tag{A.27}$$

In which the second equality is due to Lemma 16. Thus $\int_{\Omega_1} |\mathbb{E}[\zeta_1|X=x] - \log f(x)|\, d\mathbb{P}_X = 0$.

For $x \in \Omega_2$:

In this case $f(x)$ is obviously the same as $P_X(x,0)\prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)},0\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)},0\right)}$. We will first show that the probability of the k-nearest neighbor distance $\rho_{k,1}$ being non-zero is small, which means we will use the number of samples being equal to x as $\tilde{k}_i$, and we will show that the mean of the estimate ζ_1 is close to $\log f(x)$.

We notice that for x, the probability of $\rho_{k,1} > 0$ is equal to the probability that x is observed at most $k-1$ times. So it can be upper bounded as:

$$\begin{aligned}
& \mathbb{P}\Big(\rho_{k,1} > 0 \Big| X = x\Big) && \text{(A.28)}\\
= & \sum_{m=0}^{k-1} \begin{pmatrix} N-1 \\ m \end{pmatrix} P_X(x,0)^m \Big(1 - P_X(x,0)\Big)^{N-1-m} && \text{(A.29)}\\
\leq & \sum_{m=0}^{k-1} N^m \Big(1 - P_X(x,0)\Big)^{N-k} && \text{(A.30)}\\
\leq & kN^k \Big(1 - P_X(x,0)\Big)^{N-k} && \text{(A.31)}\\
\leq & kN^k e^{-(N-k)P_X(x,0)} && \text{(A.32)}
\end{aligned}$$

Now let's consider the case when $\rho_{k,1} = 0$. We can write the term ζ_1 in the form:

$$\zeta_1 = \psi(\tilde{k}_1) + \sum_{l=1}^{d} \Big(\mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}} \log(n^{(1)}_{\mathrm{pa}(X_l)} + 1) - \log(n^{(1)}_{\mathrm{pa}+(X_l)} + 1)\Big) + K_N$$

The term K_N depends only on N and the structure of the Bayesian model $\overline{\mathbb{P}}_X$ and is independent of the observed samples, and in general is equal to:

$$K_N = -\log C_{d_X} + \sum_{l=1}^{d} \Big(\log C_{d_{\mathrm{pa}+(X_l)}} - \log C_{d_{\mathrm{pa}(X_l)}}\Big) + \left(\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}} - 1\right) \log N. \quad \text{(A.33)}$$

In which the terms C_{d_S} indicate the volume of a unit ball in an d_S-dimensional space $\mathcal{S}$. Since we are using ℓ_∞-norm in our algorithm and proofs, all C_{d_S} terms will be equal to 1 and hence $K_N = \left(\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}} - 1\right) \log N$ as it appeared in Algorithm 1.

Then for the case of $\rho_{k,1} = 0$ we can write:

$$
\begin{aligned}
& \left|\mathbb{E}[\zeta_1 | X = x, \rho_{k,1} = 0] - \log f(x)\right| && \text{(A.34)} \\
= \; & \left|\mathbb{E}\Bigg[\psi(\tilde{k}_i) + \sum_{l=1}^{d} \left(\mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}} \log(n^{(1)}_{\mathrm{pa}(X_l)} + 1) - \log(n^{(1)}_{\mathrm{pa+}(X_l)} + 1)\right)\right. \\
& \left. + \left(\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}} - 1\right) \log N \Bigg| X = x, \rho_{k,1} = 0\Bigg] \right. \\
& \left. - \log P_X(x, 0) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)}, 0\right)}{P_{\mathrm{pa+}(X_l)}\left(x_{\mathrm{pa+}(l)}, 0\right)}\right| && \text{(A.35)} \\
\leq \; & \left|\mathbb{E}[\psi(\tilde{k}_1) | X = x, \rho_{k,1} = 0] - \log N P_X(x, 0)\right| && \text{(A.36)} \\
& + \sum_{l=1}^{d} \left|\mathbb{E}[\log(n^{(1)}_{\mathrm{pa+}(X_l)} + 1) | X = x, \rho_{k,1} = 0] - \log N P_{\mathrm{pa+}(X_l)}\left(x_{\mathrm{pa+}(l)}, 0\right)\right| && \text{(A.37)} \\
& + \sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}} \left|\mathbb{E}[\log(n^{(1)}_{\mathrm{pa}(X_l)} + 1) | X = x, \rho_{k,1} = 0] - \log N P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)}, 0\right)\right| && \text{(A.38)}
\end{aligned}
$$

We notice that $\tilde{k}_1$ is the number of samples among $\{x^{(i)}\}_{i=1}^{N}$ such that $X_i = x$, where each sample is independently equal to x with probability $P_X(x, 0)$. Therefore the distribution of $\tilde{k}_1$ is $\mathrm{Bino}\,(N, P_X(x, 0))$. Similarly, for any $S \subset X$, $n_{S,1} + 1$ is the number of samples among $\{x^{(i)}\}_{i=1}^{N}$ such that $s^{(i)} = s$; i.e. projection of $x^{(i)}$ over $\mathcal{S}$ is equal to the projection of x over $\mathcal{S}$. Thus $n_{S,1} + 1 \sim \mathrm{Bino}\,(N, P_S(s, 0))$. In addition to that, the event $\rho_{k,1}$=0 is equivalent to $\tilde{k}_1 \geq k$. Thus to upperbound term A.36 and any of the individual terms inside the summations of terms A.37 and A.38 we propose the lemma below:

Lemma 17. *If X is distributed as Bino(N,p) and $m \geq 0$, then:*

$$|\mathbb{E}\left[\log(X+m)|X \geq k\right] - \log(Np)| \leq U(k,N,m,p) \tag{A.39}$$

Where $U(k,N,m,p)$ is given by:

$$U(k,N,m,p) = \max\left\{\left|\log\left(\frac{1+\frac{m}{Np}}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)}\right)\right|, \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)}\frac{3}{2Np}\right\} \tag{A.40}$$

Proof. Please see Section A.4.3. □

Remark. *Since the Assumption 1.1 states that $k/N \to 0$ as $N \to \infty$, then $(Np-k)^2/N = N(p-k/N)^2 \to \infty$, and the upperbound $U(k,N,m,p)$ will converge to 0 as $N \to \infty$ for any p.*

From Lemma 17 we have:

$$\begin{aligned}
& \left|\mathbb{E}[\log(n_S^{(1)}+1)|X=x,\rho_{k,1}=0] - \log NP_S(s,0)\right| && \text{(A.41)}\\
= \; & \left|\mathbb{E}[\log(n_S^{(1)}+1)|X=x,n_S^{(1)}+1\geq k] - \log NP_S(s,0)\right| && \text{(A.42)}\\
\leq \; & U\Big(k,N,0,P_S(s,0)\Big) && \text{(A.43)}
\end{aligned}$$

For any of the individual terms inside the summations of terms A.37 and A.38. For the term A.36 we notice that $|\psi(\tilde{k}_1) - \log(\tilde{k}_1)| \leq 1/\tilde{k}_1 \leq 1/k$ [20]. Thus this term can be bounded similarly by $U\left(k,N,0,P_X(x,0)\right) + 1/k$. By combining all these bounds, we obtain:

$$\begin{aligned}
& \left|\mathbb{E}[\zeta_1|X=x,\rho_{k,1}=0] - \log f(x)\right| && \text{(A.44)}\\
\leq \; & \sum_{l=1}^{d}\Big(U\Big(k,N,0,P_{\text{pa}(X_l)}(x_{\text{pa}(l)},0)\Big) + U\Big(k,N,0,P_{\text{pa+}(X_l)}(x_{\text{pa+}(l)},0)\Big)\Big) \\
& + U\Big(k,N,0,P_X(x,0)\Big) + \frac{1}{k} && \text{(A.45)}
\end{aligned}$$

Assumption 1.2 implies that discrete probabilities and hence $U\Big(k, N, 0, p\Big)$ terms are bounded; i.e. there exists a $\hat{p}$ such that for any $x \in \Omega_2$:

$$\begin{cases} U\Big(k, N, 0, P_X(x, 0)\Big) \leq U\Big(k, N, 0, \hat{p}\Big) & \\ U\Big(k, N, 0, P_{\text{pa}(X_l)}(x_{\text{pa}(l)}, 0)\Big) \leq U\Big(k, N, 0, \hat{p}\Big) & \text{for } l = 1, \ldots, d \\ U\Big(k, N, 0, P_{\text{pa+}(X_l)}(x_{\text{pa+}(l)}, 0), 0)\Big) \leq U\Big(k, N, 0, \hat{p}\Big) & \text{for } l = 1, \ldots, d \end{cases} \tag{A.46}$$

Therefore:

$$\Big|\mathbb{E}[\zeta_1 | X = x, \rho_{k,1} = 0] - \log f(x)\Big| \leq (2d+1)\, U\Big(k, N, 0, \hat{p}\Big) + \frac{1}{k} \tag{A.47}$$

If we combine it with the case of $\rho_{k,i} > 0$, we obtain that:

$$\begin{aligned} & \Big|\mathbb{E}[\zeta_1 | X = x] - \log f(x)\Big| && \text{(A.48)} \\ \leq\ & \Big|\mathbb{E}[\zeta_1 | X = x, \rho_{k,1} > 0] - \log f(x)\Big| \times \mathbb{P}(\rho_{k,1} > 0) & \\ & + \Big|\mathbb{E}[\zeta_1 | X = x, \rho_{k,1} = 0] - \log f(x)\Big| \times \mathbb{P}(\rho_{k,1} = 0) && \text{(A.49)} \\ \leq\ & \Big((2d+1)\log N + |\log f(x)|\Big) k N^k e^{-(N-k)P_X(x,0)} + (2d+1)\, U\Big(k, N, 0, \hat{p}\Big) + \frac{1}{k} && \text{(A.50)} \end{aligned}$$

Where we used the fact that $|\zeta_1| \leq (2d+1)\log N$. Integrating over Ω_2, we can write:

$$\begin{aligned} & \int_{\Omega_2} \Big|\mathbb{E}[\zeta_1 | X = x] - \log f(x)\Big| d\mathbb{P}_X && \text{(A.51)} \\ \leq\ & \left((2d+1)\log N + \int_{\Omega_2} |\log f(x)|\, d\mathbb{P}_X\right) k N^k e^{-(N-k)\inf_{x \in \Omega_2} P_X(x,0)} && \text{(A.52)} \\ & + (2d+1)\, U\Big(k, N, 0, \hat{p}\Big) + \frac{1}{k} && \text{(A.53)} \end{aligned}$$

By Assumption 1.1, k goes to infinity as N goes to infinity, so $1/k$ vanishes as N grows boundlessly. The term $U\Big(k, N, 0, \hat{p}\Big)$ will also converge to zero as we saw. From assumption 1.3, $\int_{\Omega_2} |\log f(x)|\, d\mathbb{P}_X < +\infty$. Thus the first term also converges to 0 as $N \to \infty$. Thus:

$$\lim_{N\to\infty}\int_{\Omega_2}\Big|\mathbb{E}[\zeta_1|X=x]-\log f(x)\Big|d\mathbb{P}_X=0 \tag{A.54}$$

For $x\in\Omega_3$:

In this case, $P_X(x,r)$ is a monotonic function of r such that $P_X(x,0)=0$ and $\lim_{r\to\infty}P_X(x,r)=1$. Hence we can view $\log P_X(x,r)\prod_{l=1}^{d}\frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)},r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)},r\right)}$ as a function of $P_X(x,r)$ and it converges to $\log f(x)$ as $P_X(x,r)\to 0$, for almost every x. Since $\mathbb{P}_X(\Omega_3)\le 1<\infty$ and we know $\int_{\Omega_3}|\log f(x)|\,d\mathbb{P}_X<\infty$, By Egorov's theorem, for any $\epsilon>0$, there exists a subset $E\subseteq\Omega_3$ with $\mathbb{P}_X(E)<\epsilon$ and $\int_E|\log f(x)|\,d\mathbb{P}_X<\epsilon$, such that the term $\log P_X(x,r)\prod_{l=1}^{d}\frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)},r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)},r\right)}$ converges uniformly on $\Omega_3\setminus E$ as $P_X(x,r)\to 0$. Now let's assume ϵ_N is a sequence converging to 0. Consequently, The corresponding sets E_N will also create a sequence. For any fixed N and for $x\in E_N$, we notice that $|\zeta_1|\le(2d+1)\log N$, so we have:

$$\int_{E_N}\Big|\mathbb{E}[\zeta_1|X=x]-\log f(x)\Big|d\mathbb{P}_X \tag{A.55}$$

$$\le\ \int_{E_N}\Big(\,(2d+1)\log N+|\log f(x)|\,\Big)d\mathbb{P}_X<\epsilon_N\Big((2d+1)\log N+1\Big) \tag{A.56}$$

By choosing ϵ_N such that $\epsilon_N\log N\to 0$ as $N\to\infty$ (For example $\epsilon_N=1/N$), we will have $\lim_{N\to\infty}\int_E|\mathbb{E}[\zeta_1|X=x]-\log f(x)|\,d\mathbb{P}_X=0$.

For any $x\in\Omega_3\setminus E_N$, since $P_X(x,0)=0$, we know that $\mathbb{P}\left(\rho_{k,1}=0|X=x\right)=0$, so $\tilde{k}_1=k$ with probability 1. Conditioning on $\rho_{k,1}=r>0$, the term $|\mathbb{E}[\zeta_1|X=x]-\log f(x)|$ can be

decomposed as:

$$\Big|\mathbb{E}[\zeta_1|X=x]-\log f(x)\Big| \tag{A.57}$$

$$= \left|\int_{r=0}^{\infty}\Big(\mathbb{E}[\zeta_1|X=x,\rho_{k,1}=r]-\log f(x)\Big)dF_{\rho_{k,1}}(r)\right| \tag{A.58}$$

$$\leq \left|\int_{r=0}^{\infty}\left(\log P_X(x,r)\prod_{l=1}^{d}\frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)},r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)},r\right)}-\log f(x)\right)dF_{\rho_{k,1}}(r)\right| \tag{A.59}$$

$$+\left|\int_{r=0}^{\infty}\left(\psi(k)-\log N-\log P_X(x,r)\right)dF_{\rho_{k,1}}(r)\right| \tag{A.60}$$

$$+\sum_{l=1}^{d}\left|\int_{r=0}^{\infty}\Big(\mathbb{E}\left[\log(n^{(1)}_{\mathrm{pa}+(X_l)}+1)|(X,\rho_{k,1})=(x,r)\right]\right.$$
$$\left.-\log NP_{\mathrm{pa}+(X_l)}(x_{\mathrm{pa}+(l)},r)\Big)dF_{\rho_{k,1}}(r)\right| \tag{A.61}$$

$$+\sum_{l=1}^{d}\mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}}\left|\int_{r=0}^{\infty}\Big(\mathbb{E}\left[\log(n^{(1)}_{\mathrm{pa}(X_l)}+1)|(X,\rho_{k,1})=(x,r)\right]\right.$$
$$\left.-\log NP_{\mathrm{pa}(X_l)}(x_{\mathrm{pa}(l)},r)\Big)dF_{\rho_{k,1}}(r)\right| \tag{A.62}$$

In which $F_{\rho_{k,1}}(r)$ is the CDF of the k-nearest neighbor distance $\rho_{k,1}$ given $X=x$. The derivative of this CDF with respect to $P_X(x,r)$ is given by:

$$\frac{dF_{\rho_{k,1}}(r)}{dP_X(x,r)}=\frac{(N-1)!}{(k-1)!(N-k-1)!}P_X(x,r)^{k-1}\Big(1-P_X(x,r)\Big)^{N-k-1} \tag{A.63}$$

Upper bound for the term (A.59) :
Since $P_X(x,r)\prod_{l=1}^{d}\frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)},r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)},r\right)}$ converges to $f(x)$ uniformly over $\Omega_3\setminus E$ as $P_X(x,r)\to 0$, So for any δ_N and any $x\in\Omega_3\setminus E$, there exists r_1 such that for any $r<r_1$:

$$\left|\log P_X(x,r)\prod_{l=1}^{d}\frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)},r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)},r\right)}-\log f(x)\right|<\delta_N \tag{A.64}$$

Let r_2 be the value of r such that $P_X(x, r_2) = 4k \log N/N$, and take $r_N = \min\{r_1, r_2\}$. Thus r_N depends on x, but δ_N does not depend on x and $\lim_{N\to\infty} \delta_N = 0$. Therefore, (A.59) can be upper bounded as:

$$\left| \int_{r=0}^{\infty} \left(\log P_X(x,r) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)}, r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)}, r\right)} - \log f(x) \right) dF_{\rho_{k,1}}(r) \right| \tag{A.65}$$

$$\leq \int_{r=0}^{r_N} \left| \log P_X(x,r) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)}, r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)}, r\right)} - \log f(x) \right| dF_{\rho_{k,1}}(r)$$

$$+ \int_{r=r_N}^{\infty} \left| \log P_X(x,r) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)}, r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)}, r\right)} - \log f(x) \right| dF_{\rho_{k,1}}(r) \tag{A.66}$$

$$\leq \delta_N \mathbb{P}\left(\rho_{k,1} \leq r_N | X = x\right)$$

$$+ \left(\sup_{r \geq r_N} \left| \log P_X(x,r) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)}, r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)}, r\right)} - \log f(x) \right| \right) \mathbb{P}\left(\rho_{k,1} \geq r_N | X = x\right) \tag{A.67}$$

First, $\mathbb{P}\left(\rho_{k,1} \leq r_N | X = x\right)$ is smaller than 1. Secondly, since $P_X(x, r) \geq 4k \log N/N > 1/N$ for $r \geq r_N$, so we have $|\log P_X(x, r)| \leq \log N$. The same bounds apply for any $|P_S(s, r)|$ as well, as $P_S(s, r) \geq P_X(x, r)$. Thus by triangle inequality, the supremum is upper-bounded by $(2d + 1) \log N + |\log f(x)|$. Finally, the probability $\mathbb{P}\left(\rho_{k,1} \geq r_N | X = x\right)$ is upper bounded by:

$$\mathbb{P}\left(\rho_{k,1} \geq r_N | X = x\right) \tag{A.68}$$

$$= \sum_{m=0}^{k-1} \binom{N-1}{m} P_X(x, r_N)^m \left(1 - P_X(x, r_N)\right)^{N-1-m} \tag{A.69}$$

$$\leq \sum_{m=0}^{k-1} N^m \left(1 - P_X(x, r_N)\right)^{N-k} \tag{A.70}$$

$$= kN^k \left(1 - \frac{4k \log N}{N}\right)^{N/2} \tag{A.71}$$

$$\leq kN^k e^{-2k \log N} = \frac{k}{N^k} \tag{A.72}$$

for N large enough such that $N-k > N/2$. Therefore A.59 is upper-bounded by:

$$\left| \int_{r=0}^{\infty} \left(\log P_X(x,r) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)}, r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)}, r\right)} - \log f(x) \right) dF_{\rho_{k,1}}(r) \right| \tag{A.73}$$

$$\leq \quad \delta_N + k \frac{(2d+1)\log N + |\log f(x)|}{N^k} \tag{A.74}$$

Upper bound for the term (A.60) :

We have:

$$\int_{r=0}^{\infty} (\psi(k) - \log N - \log P_X(x,r))\, dF_{\rho_{k,1}}(r) \tag{A.75}$$

$$= \quad \psi(k) - \log N - \frac{(N-1)!}{(k-1)!(N-k-1)!}$$

$$\times \int_{r=0}^{\infty} \Big(P_X(x,r)^{k-1} (1 - P_X(x,r))^{N-k-1} \log P_X(x,r) \Big) dP_X(x,r) \tag{A.76}$$

$$= \quad \psi(k) - \log N - \frac{(N-1)!}{(k-1)!(N-k-1)!} \int_{t=0}^{1} \left(t^{k-1} (1-t)^{N-k-1} \log t \right) dt \tag{A.77}$$

$$= \quad \psi(k) - \log N - (\psi(k) - \psi(N)) \tag{A.78}$$

$$= \quad \psi(N) - \log(N) \tag{A.79}$$

We notice that for any N we have $\psi(N) < \log N$ and $\lim_{N\to\infty} (\psi(N) - \log N) = 0$.

Upper bound for the individual terms of (A.61) and (A.62) :

The upper bound for each of these terms follows the same logic, so we do the proof for an arbitrary subset $S \in \{\mathrm{pa}(X_l)\}_{l=1}^{d} \cup \{\mathrm{pa}+(X_l)\}_{l=1}^{d}$.

The distributions of $n_S^{(1)}$ for all $S \in \{\mathrm{pa}(X_l)\}_{l=1}^{d} \cup \{\mathrm{pa}+(X_l)\}_{l=1}^{d}$ are given by the lemma below:

Lemma 18. *Given $X = x$ and $\rho_{k,1} = r > 0$ and s being the projection of x over $\mathcal{S}$, the term*

$n_S^{(1)} - k$ *is distributed as Bino* $\left(N - k - 1, \frac{P_S(s,r) - P_X(x,r)}{1 - P_X(x,r)}\right)$.

Proof. Please see the proof of Lemma B.2 in [79]. □

The bound on $\mathbb{E}[\log(n_S^{(1)} + 1)|X = x, \rho_{k,1} = r]$ is given by the Lemma B.3 in [79] which we restate here:

Lemma 19. *If X is distributed as Bino(N, p), then $|\mathbb{E}[\log(X + k)] - \log(Np + k)| \leq C/(Np + k)$ for some constant C.*

Proof. Please see the proof of Lemma B.3 in [79]. □

Thus we can write:

$$
\begin{aligned}
& \left| \int_{r=0}^{\infty} \left(\mathbb{E}\left[\log(n_S^{(1)} + 1) | (X, \rho_{k,1}) = (x, r) \right] - \log N P_S(s, r) \right) dF_{\rho_{k,1}}(r) \right| && \text{(A.80)} \\
\leq \; & \left| \int_{r=0}^{\infty} \left(\mathbb{E}[\log(n_S^{(1)} + 1) | (X, \rho_{k,1}) = (x, r)] \right. \right. \\
& \left. \left. - \log \left((N - k - 1) \frac{P_S(s, r) - P_X(x, r)}{1 - P_X(x, r)} + k + 1 \right) \right) dF_{\rho_{k,1}}(r) \right| && \text{(A.81)} \\
& + \left| \int_{r=0}^{\infty} \left(\log \frac{(N - k - 1) \frac{P_S(s,r) - P_X(x,r)}{1 - P_X(x,r)} + k + 1}{N P_S(s, r)} \right) dF_{\rho_{k,1}}(r) \right| && \text{(A.82)} \\
\leq \; & \int_{r=0}^{\infty} \left| \left(\mathbb{E}\left[\log(n_S^{(1)} + 1) | (X, \rho_{k,1}) = (x, r) \right] \right. \right. \\
& \left. \left. - \log \left((N - k - 1) \frac{P_S(s, r) - P_X(x, r)}{1 - P_X(x, r)} + k + 1 \right) \right) \right| dF_{\rho_{k,1}}(r) && \text{(A.83)} \\
& + \left| \mathbb{E}_r \left[\log \frac{N(P_S(s, r) - P_X(x, r)) + (k + 1)(1 - P_S(s, r))}{N P_S(s, r)(1 - P_X(x, r))} \right] \right| && \text{(A.84)}
\end{aligned}
$$

Where $\mathbb{E}_r$ denotes the expectation over the distribution $F_{\rho_{k,1}}$. By the Lemma B.3 in [79],

the term in A.83 is upper bounded by:

$$\int_{r=0}^{\infty} \left| \left(\mathbb{E}\left[\log(n_S^{(1)} + 1) | (X, \rho_{k,1}) = (x, r) \right] \right.\right. \tag{A.85}$$
$$\left.\left. - \log\left((N-k-1)\frac{P_S(s,r) - P_X(x,r)}{1 - P_X(x,r)} + k + 1 \right) \right) \right| dF_{\rho_{k,1}}(r)$$

$$\leq \int_{r=0}^{\infty} \frac{C}{(N-k-1)\frac{P_S(s,r)-P_X(x,r)}{1-P_X(x,r)} + k + 1} dF_{\rho_{k,1}}(r) \tag{A.86}$$

$$\leq \int_{r=0}^{\infty} \frac{C}{k+1} dF_{\rho_{k,1}}(r) = \frac{C}{k+1} \tag{A.87}$$

The last inequality follows from the fact that $P_S(s,r) > P_X(x,r)$. For (A.84), using the fact that $\log(x/y) \leq (x-y)/y$ and Cauchy-Schwarz inequality, we have the following:

$$\mathbb{E}_r \left[\log \frac{N(P_S(s,r) - P_X(x,r)) + (k+1)(1 - P_S(s,r))}{NP_S(s,r)(1 - P_X(x,r))} \right] \tag{A.88}$$

$$\leq \mathbb{E}_r \left[\frac{N(P_S(s,r) - P_X(x,r)) + (k+1)(1 - P_S(s,r))}{NP_S(s,r)(1 - P_X(x,r))} - 1 \right] \tag{A.89}$$

$$= \mathbb{E}_r \left[\frac{(k+1 - NP_X(x,r))(1 - P_S(s,r))}{NP_S(s,r)(1 - P_X(x,r))} \right] \tag{A.90}$$

$$\leq \sqrt{\mathbb{E}_r \left[\left(\frac{k+1-NP_X(x,r)}{NP_X(x,r)} \right)^2 \right] \mathbb{E}_r \left[\left(\frac{P_X(x,r)(1-P_S(s,r))}{P_S(s,r)(1-P_X(x,r))} \right)^2 \right]} \tag{A.91}$$

Note that $P_S(s,r) \geq P_X(x,r)$ for all r, so the second expectation term is always smaller than or equal to 1. For the first expectation, let $t = P_X(x,r)$ then we have:

$$\mathbb{E}_r \left[\left(\frac{k+1-NP_X(x,r)}{NP_X(x,r)} \right)^2 \right] \tag{A.92}$$

$$= \int_{r=0}^{\infty} \left(\frac{k+1-NP_X(x,r)}{NP_X(x,r)} \right)^2 dF_{\rho_{k,1}}(r) \tag{A.93}$$

$$= \frac{(N-1)!}{(k-1)!(N-k-1)!} \int_{t=0}^{1} \frac{(k+1-Nt)^2}{N^2 t^2} t^{k-1}(1-t)^{N-k-1} dt \tag{A.94}$$

$$= \frac{(N-1)(N-2)(k+1)^2}{N^2(k-1)(k-2)} - \frac{2(N-1)(k+1)}{N(k-1)} + 1 \tag{A.95}$$

This term is upper bounded by $C_1(1/N + 1/k)$ for some constant C_1 for N and k large enough. Thus:

$$\mathbb{E}_r\left[\log \frac{N(P_S(s,r) - P_X(x,r)) + (k+1)(1 - P_S(s,r))}{NP_S(s,r)(1 - P_X(x,r))}\right] \leq \sqrt{C_1(\frac{1}{N} + \frac{1}{k})} \tag{A.96}$$

Similarly, by using the fact that $\log(x/y) > (x-y)/x$ and Cauchy-Schwarz inequality again, we conclude that there are some constant $C_2 > 0$ such that:

$$\mathbb{E}_r\left[\log \frac{N(P_S(s,r) - P_X(x,r)) + (k+1)(1 - P_S(s,r))}{NP_S(s,r)(1 - P_X(x,r))}\right] \geq -\sqrt{C_2(\frac{1}{N} + \frac{1}{k})} \tag{A.97}$$

Therefore by combining all these bounds we obtain

$$\left|\int_{r=0}^{\infty}\left(\mathbb{E}\left[\log(n_S^{(1)} + 1)|(X, \rho_{k,1}) = (x,r)\right] - \log NP_S(s,r)\right)dF_{\rho_{k,1}}(r)\right| \tag{A.98}$$

$$\leq \quad \frac{C}{k+1} + \sqrt{C'(\frac{1}{k} + \frac{1}{N})} \tag{A.99}$$

where $C' = \max\{C_1, C_2\}$.

Now putting all the bounds for $\Omega_3 \setminus E_N$, we have:

$$\int_{\Omega_3 \setminus E_N}\left|\mathbb{E}\left[\zeta_1|X = x\right] - \log f(x)\right|d\mathbb{P}_X \tag{A.100}$$

$$\leq \quad \delta_N + k\frac{(2d+1)\log N + \int_{\mathcal{X}}|\log f(x)|\, d\mathbb{P}_X}{N^k}$$

$$+\psi(N) - \log(N) + 2d\left(\frac{C}{k+1} + \sqrt{C'(\frac{1}{k} + \frac{1}{N})}\right) \tag{A.101}$$

From the assumptions 1, k increases as $N \to \infty$, and $\int_{\mathcal{X}}|\log f(x)|\, d\mathbb{P}_X < \infty$. Therefore the whole upperbound vanishes as N goes to infinity. Thus for the entire set Ω_3 we have:

$$\lim_{N\to\infty}\int_{\Omega_3}\left|\mathbb{E}[\zeta_1|X = x)] - \log f(x)\right|d\mathbb{P}_X = 0 \tag{A.102}$$

A.4.2 Proof of Lemma 16

This proof is based on the Lebesgue-Besicovitch differentiation theorem (For example look at Theorem 1.32 from [61]), stated as:

Theorem 20. *Let μ be a Radon measure on $\mathbb{R}^d$. For $f \in L^1_{loc}(\mu)$,*

$$\lim_{r\to 0} \frac{1}{\mu\left(B_r(x)\right)} \int_{B_r(x)} f d\mu = f(x) \tag{A.103}$$

for μ-a.e. x.

Let $f = \frac{d\mathbb{P}_X}{d\overline{\mathbb{P}}_X}$ and $\mu = \overline{\mathbb{P}}_X$. Since μ is a probability measure, it is a Radon measure on Euclidean space. Also, since $\int_{\mathcal{X}} |f| d\mu = 1$, so it's globally and therefore locally integrable with respect to μ. Thus the conditions of the Theorem 20 are satisfied, and we can write:

$$\lim_{r\to 0} P_X(x,r) \prod_{l=1}^{d} \frac{P_{\mathrm{pa}(X_l)}\left(x_{\mathrm{pa}(l)}, r\right)}{P_{\mathrm{pa}+(X_l)}\left(x_{\mathrm{pa}+(l)}, r\right)} \tag{A.104}$$

$$= \lim_{r\to 0} \frac{\mathbb{P}\Big\{B_r(x)\Big\}}{\prod_{l=1}^{d} \mathbb{P}_{X_l|\mathrm{pa}(X_l)}\Big\{B_r\left(x_l\right) \Big| B_r\left(x_{\mathrm{pa}(l)}\right)\Big\}} \tag{A.105}$$

$$= \lim_{r\to 0} \frac{\mathbb{P}\Big\{B_r(x)\Big\}}{\overline{\mathbb{P}}_X\Big\{B_r(x)\Big\}} \tag{A.106}$$

$$= \lim_{r\to 0} \frac{1}{\overline{\mathbb{P}}_X\Big\{B_r(x)\Big\}} \int_{B_r(x)} \frac{d\mathbb{P}_X}{d\overline{\mathbb{P}}_X} d\overline{\mathbb{P}}_X \tag{A.107}$$

$$= \frac{d\mathbb{P}_X}{d\overline{\mathbb{P}}_X}(x) \tag{A.108}$$

A.4.3 Proof of Lemma 17

First, we upperbound $\mathbb{E}\left[\log(X+m)|X\geq k\right]-\log(Np)$. We can see that:

$$\begin{aligned}
&\mathbb{E}\left[X+m|X\geq k\right] && \text{(A.109)}\\
=\;& \frac{1}{\mathbb{P}\left(X\geq k\right)}\sum_{i=k}^{N}(i+m)\begin{pmatrix}N\\ i\end{pmatrix}p^i(1-p)^{N-i} && \text{(A.110)}\\
\leq\;& \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)}\sum_{i=k}^{N}(i+m)\begin{pmatrix}N\\ i\end{pmatrix}p^i(1-p)^{N-i} && \text{(A.111)}\\
\leq\;& \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)}\sum_{i=1}^{N}(i+m)\begin{pmatrix}N\\ i\end{pmatrix}p^i(1-p)^{N-i} && \text{(A.112)}\\
=\;& \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)}\left(\mathbb{E}\left[X\right]+m\right)=\frac{Np+m}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)} && \text{(A.113)}
\end{aligned}$$

In which we used Hoeffding's inequality. We know that $\mathbb{E}\left[\log(X+m)|X\geq k\right]\leq\log\left(\mathbb{E}\left[X+m|X\geq k\right]\right)$, therefore:

$$\mathbb{E}\left[\log(X)|X\geq k\right]-\log(Np)\leq\log\left(\frac{1+\frac{m}{Np}}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)}\right) \tag{A.114}$$

Second, to give an upper bound over $\log(Np)-\mathbb{E}\left[\log(X+m)|X\geq k\right]$, we first notice that:

$$\log(Np)-\mathbb{E}\left[\log(X+m)|X\geq k\right]\leq\log(Np)-\mathbb{E}\left[\log(X)|X\geq k\right] \tag{A.115}$$

Then we upperbound $\log(Np)-\mathbb{E}\left[\log(X)|X\geq k\right]$ by applying Taylor's theorem around $x_0=Np$, where there exists ζ between x and x_0 such that:

$$\log(x)=\log(Np)+\frac{x-Np}{Np}-\frac{(x-Np)^2}{2\zeta^2} \tag{A.116}$$

since $\zeta\geq\min\left\{x,x_0\right\}=\min\left\{x,Np\right\}$, we have:

$$
\begin{aligned}
& -\log(x) + \log(Np) + \frac{x - Np}{Np} = \frac{(x - Np)^2}{2\zeta^2} \\
\leq \quad & \max\left\{ \frac{(x - Np)^2}{2x^2}, \frac{(x - NP)^2}{2(Np)^2} \right\} \leq \frac{(x - Np)^2}{2x^2} + \frac{(x - Np)^2}{2(Np)^2} \qquad \text{(A.117)}
\end{aligned}
$$

Now taking the conditional expectations from both sides, we have:

$$
\begin{aligned}
& -\mathbb{E}\left[\log(X) | X \geq k\right] + \log(Np) + \frac{\mathbb{E}\left[X | X \geq k\right] - Np}{Np} \\
\leq \quad & \mathbb{E}\left[\frac{(X - Np)^2}{2X^2} \middle| X \geq k \right] + \frac{\mathbb{E}\left[(X - Np)^2 | X \geq k\right]}{2(Np)^2} \qquad \text{(A.118)}
\end{aligned}
$$

First, we notice that $\mathbb{E}\left[X | X \geq k\right] \geq \mathbb{E}\left[X\right] = Np$.

Second, $\mathbb{E}\left[(X - Np)^2 | X \geq k\right] \leq \frac{1}{1 - \exp\left(-2\frac{(Np-k)^2}{N}\right)} \mathrm{Var}\left[X\right] = \frac{Np(1-p)}{1 - \exp\left(-2\frac{(Np-k)^2}{N}\right)}$.

Thus we can write:

$$
\begin{aligned}
& -\mathbb{E}\left[\log(X) | X \geq k\right] + \log(Np) \qquad \text{(A.119)} \\
\leq \quad & \frac{Np(1-p)}{1 - \exp\left(-2\frac{(Np-k)^2}{N}\right)} \frac{1}{2(Np)^2} + \mathbb{E}\left[\frac{(X - Np)^2}{2X^2} \middle| X \geq k \right] \qquad \text{(A.120)}
\end{aligned}
$$

To deal with the term $\mathbb{E}\left[\frac{(X-Np)^2}{2X^2} \middle| X \geq k \right]$, we have:

$$\mathbb{E}\left[\frac{(X-Np)^2}{2X^2}\middle| X \geq k\right] \tag{A.121}$$

$$\leq \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)} \sum_{i=k}^{N} \frac{(i-Np)^2}{2i^2} \binom{N}{i} p^i (1-p)^{N-i} \tag{A.122}$$

$$\leq \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)} \sum_{i=k}^{N} \frac{(i-Np)^2}{(i+1)(i+2)} \binom{N}{i} p^i (1-p)^{N-i} \tag{A.123}$$

$$= \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)} \sum_{i=k}^{N} \frac{(i-Np)^2}{(N+1)(N+2)p^2} \binom{N+2}{i+2} p^{2+i} (1-p)^{N-i} \tag{A.124}$$

$$\leq \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)} \frac{1}{(N+1)(N+2)p^2} \mathbb{E}_{Y\sim \text{Bino}(N+2,p)}\left[(Y-Np)^2\right] \tag{A.125}$$

$$= \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)} \frac{(N+2)p(1-p)+4p^2}{(N+1)(N+2)p^2} \tag{A.126}$$

$$\leq \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)} \frac{(N+2)p}{(N+1)(N+2)p^2} \leq \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)} \frac{1}{Np} \tag{A.127}$$

In which we used the fact that $2i^2 \geq (i+1)(i+2)$ for $i \geq 4$, and $(N+2)p \geq 4p$ for $N \geq 2$. Plugging it into Equation A.120, we have:

$$-\mathbb{E}\left[\log(X)|X \geq k\right] + \log(Np) \leq \frac{1}{1-\exp\left(-2\frac{(Np-k)^2}{N}\right)} \frac{3}{2Np} \tag{A.128}$$

And the desired result is yielded.

A.4.4 Proof of Theorem 2

For this part, we also follow the same procedure as followed for Theorem 2 in [79] using Efron-Stein inequality. Suppose $\widehat{\mathbb{GDMM}}^{(N)}(X, \mathcal{G})$ is the estimate based on the original samples $x^{(1)}, x^{(2)}, \ldots, x^{(N)}$. For the usage of Efron-Stein inequality, we suppose that there is another set of n i.i.d samples drawn from P_X denoted by $x'^{(1)}, x'^{(2)}, \ldots, x'^{(n)}$. Let $\widehat{\mathbb{GDMM}}^{(N)}(X^{(j)}, \mathcal{G})$

be the estimate based on the original samples' set in which the jth sample is replaced by another i.i.d sample $x'^{(j)}$ taken from the second set, i.e. $\widehat{\mathbb{GDM}}^{(N)}(X^{(j)}, \mathcal{G})$ is the estimate based on $x^{(1)}, \ldots, x^{(j-1)}, x'^{(j)}, x^{(j+1)}, \ldots, x^{(N)}$. Then the Efron-Stein inequality states that:

$$\text{Var}\left[\widehat{\mathbb{GDM}}^{(N)}(X, \mathcal{G})\right] \leq \frac{1}{2}\sum_{j=1}^{N}\mathbb{E}\left[\left(\widehat{\mathbb{GDM}}^{(N)}(X, \mathcal{G}) - \widehat{\mathbb{GDM}}^{(N)}(X^{(j)}, \mathcal{G})\right)^2\right] \tag{A.129}$$

Now we will find an upper bound for the difference $\left|\widehat{\mathbb{GDM}}^{(N)}(X, \mathcal{G}) - \widehat{\mathbb{GDM}}^{(N)}(X^{(j)}, \mathcal{G})\right|$ for a given index j by considering the worst case scenario. Let $\widehat{\mathbb{GDM}}^{(N)}(X_{\setminus j}, \mathcal{G})$ be the estimate based on the original samples' set in which the jth sample is removed, i.e. $x^{(1)}, \ldots, x^{(j-1)}, x^{j+1}, \ldots, x^{(N)}$. Then by triangle inequality, we have:

$$\begin{aligned}
& \sup_{x^{(1)},\ldots,x^{(N)},x'^{(j)}} \left|\widehat{\mathbb{GDM}}^{(N)}(X, \mathcal{G}) - \widehat{\mathbb{GDM}}^{(N)}(X^{(j)}, \mathcal{G})\right| \\
\leq & \sup_{x^{(1)},\ldots,x^{(N)},x'^{(j)}} \left(\left|\widehat{\mathbb{GDM}}^{(N)}(X, \mathcal{G}) - \widehat{\mathbb{GDM}}^{(N)}(X_{\setminus j}, \mathcal{G})\right| \right. \qquad \text{(A.130)} \\
& \left. + \left|\widehat{\mathbb{GDM}}^{(N)}(X_{\setminus j}, \mathcal{G}) - \widehat{\mathbb{GDM}}^{(N)}(X^{(j)}, \mathcal{G})\right| \right) \\
\leq & \sup_{x^{(1)},\ldots,x^{(N)}} \left|\widehat{\mathbb{GDM}}^{(N)}(X, \mathcal{G}) - \widehat{\mathbb{GDM}}^{(N)}(X_{\setminus j}, \mathcal{G})\right| \qquad \text{(A.131)} \\
& + \sup_{x^{(1)},\ldots,x^{(j-1)},x'^{(j)},x^{(j+1)},\ldots,x^{(N)}} \left|\widehat{\mathbb{GDM}}^{(N)}(X_{\setminus j}, \mathcal{G}) - \widehat{\mathbb{GDM}}^{(N)}(X^{(j)}, \mathcal{G})\right| \\
= & \; 2 \sup_{x^{(1)},\ldots,x^{(N)}} \left|\widehat{\mathbb{GDM}}^{(N)}(X, \mathcal{G}) - \widehat{\mathbb{GDM}}^{(N)}(X_{\setminus j}, \mathcal{G})\right| \qquad \text{(A.132)}
\end{aligned}$$

Remember that:

$$\begin{aligned}
&\widehat{\mathbb{GDM}}^{(N)}(X,\mathcal{G}) \\
=\ & \frac{1}{N}\sum_{i=1}^{N}\zeta_i(X) && \text{(A.133)}\\
=\ & \frac{1}{N}\sum_{i=1}^{N}\left(\psi(\tilde{k}_i)+\sum_{l=1}^{d}\left(\mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}}\log(n^{(i)}_{\mathrm{pa}(X_l)}+1)-\log(n^{(i)}_{\mathrm{pa}+(X_l)}+1)\right)\right. \\
& \left.+\left(\sum_{l=1}^{d}\mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}}-1\right)\log N\right) && \text{(A.134)}
\end{aligned}$$

Thus we can write:

$$\sup_{x^{(1)},\ldots,x^{(N)}}\left|\widehat{\mathbb{GDM}}^{(N)}(X,\mathcal{G})-\widehat{\mathbb{GDM}}^{(N)}(X^{(j)},\mathcal{G})\right|\leq\frac{2}{N}\sup_{x^{(1)},\ldots,x^{(N)}}\sum_{i=1}^{N}\left|\zeta_i(X)-\zeta_i(X_{\setminus j})\right| \quad \text{(A.135)}$$

Now we need to upper-bound $\left|\zeta_i(X)-\zeta_i(X_{\setminus j})\right|$ for all the different i's. The cases are as follows:

Case I: $i=j$. Since the upper bounds $|\zeta_i(X)|\leq(2d+1)\log N$ and $\left|\zeta_i(X_{\setminus j})\right|\leq(2d+1)\log(N-1)$ always holds, thus we have $\left|\zeta_i(X)-\zeta_i(X_{\setminus j})\right|\leq 2(2d+1)\log N$. Since there's only one i equal to j, we have $\sum_{\text{Case I}}\left|\zeta_i(X)-\zeta_i(X_{\setminus j})\right|\leq 2(2d+1)\log N$.

Case II: $i\neq j$ and $\rho_{k,i}=0$. Suppose $S\in\left\{\mathrm{pa}(X_l)\right\}_{l=1}^{d}\cup\left\{\mathrm{pa}+(X_l)\right\}_{l=1}^{d}\cup\{X\}$ is a subset of X. Since $\rho_{k,i}=0$, then $n_S^{(i)}=\left|\{i'\neq i:s^{(i)}=s^{(i')}\}\right|$. We recall that for $S=X$ we actually have $n_S^{(i)}=\tilde{k}_i$. Thus by removing the point $x^{(j)}$ is this case, for any subset S, if $s^{(i)}=s^{(j)}$,

then $n_S^{(i)}$ is decreased by 1, and if $s^{(i)} \neq s^{(j)}$ then $n_S^{(i)}$ will not change. Thus we can write:

$$
\begin{aligned}
& \left|\zeta_i(X) - \zeta_i(X_{\setminus j})\right| && \text{(A.136)}\\
\leq\ & \left|\psi\left(\tilde{k}_i\right) - \psi\left(\tilde{k}_i - 1\right)\right| && \text{(A.137)}\\
& + \sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}} \mathbf{1}_{\left\{x_{\mathrm{pa}(l)}{}^{(i)} = x_{\mathrm{pa}(l)}{}^{(j)}\right\}} \left|\log(n^{(i)}_{\mathrm{pa}(X_l)} + 1) - \log n^{(i)}_{\mathrm{pa}(X_l)}\right| && \text{(A.138)}\\
& + \sum_{l=1}^{d} \mathbf{1}_{\left\{x_{\mathrm{pa}+(l)}{}^{(i)} = x_{\mathrm{pa}+(l)}{}^{(j)}\right\}} \left|\log(n^{(i)}_{\mathrm{pa}+(X_l)} + 1) - \log n^{(i)}_{\mathrm{pa}+(X_l)}\right| && \text{(A.139)}\\
& + \left(\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}} + 1\right)\Big(\log N - \log(N-1)\Big) && \text{(A.140)}\\
\leq\ & \frac{1}{\tilde{k}_i - 1} + \sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}} \mathbf{1}_{\left\{x_{\mathrm{pa}(l)}{}^{(i)} = x_{\mathrm{pa}(l)}{}^{(j)}\right\}} \frac{1}{n^{(i)}_{\mathrm{pa}(X_l)}} && \text{(A.141)}\\
& + \sum_{l=1}^{d} \mathbf{1}_{\left\{x_{\mathrm{pa}+(l)}{}^{(i)} = x_{\mathrm{pa}+(l)}{}^{(j)}\right\}} \frac{1}{n^{(i)}_{\mathrm{pa}+(X_l)}} && \text{(A.142)}\\
& + \left(\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}} + 1\right)\frac{1}{N-1} && \text{(A.143)}
\end{aligned}
$$

Where we used the fact that $\log(1 + 1/x) \leq 1/x$. For any S, the number of nodes i satisfying $s^{(i)} = s^{(j)}$ is no more than $n_S^{(j)}$, and the total number of points in the case II is no more than $N - 1$. Thus we can write:

$$
\begin{aligned}
\sum_{i\in\text{Case II}} \left|\zeta_i(X) - \zeta_i(X_{\setminus j})\right| \ \leq\ & (\tilde{k}_i - 1)\frac{1}{\tilde{k}_i - 1} && \text{(A.144)}\\
& + \sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}} n^{(i)}_{\mathrm{pa}(X_l)} \frac{1}{n^{(i)}_{\mathrm{pa}(X_l)}} && \text{(A.145)}\\
& + \sum_{l=1}^{d} n^{(i)}_{\mathrm{pa}+(X_l)} \frac{1}{n^{(i)}_{\mathrm{pa}+(X_l)}} && \text{(A.146)}\\
& + \left(\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}} + 1\right)(N-1)\frac{1}{N-1} && \text{(A.147)}\\
=\ & 2(d+1) && \text{(A.148)}
\end{aligned}
$$

Case III: $i \neq j$ and $\rho_{k,i} > 0$. In this case, $\tilde{k}_i$ is always equal to k, and for any subset $S \in \big\{\text{pa}(X_l)\big\}_{l=1}^{d} \cup \big\{\text{pa+}(X_l)\big\}_{l=1}^{d} \cup \{X\}$ we have $n_S^{(i)} = \big|\{i' \neq i : \|s^{(i)} - s^{(j)}\| \leq \rho_{k,i}\}\big|$.

Case III.1: $\|x^{(i)} - x^{(j)}\| \leq \rho_{k,i}$. This means that the point x_j is in the k nearest neighbors of x_i and by eliminating it, $\rho_{k,i}$ will change. So we don't know how $n_S^{(i)}$'s will change, thus we will use the upper-bound $\big|\zeta_i(X) - \zeta_i(X_{\setminus j})\big| \leq 2(2d+1)\log N$. From the first part of the Lemma C.1 from [79], we can upper bound the number of such i's by $\gamma_d k$, in which γ_d is the minimum number of d-dimensional cones with the angle smaller than $\pi/6$ needed to cover the total space $\mathbb{R}^d$. Thus we have:

$$\sum_{\text{Case III.1}} \big|\zeta_i(X) - \zeta_i(X_{\setminus j})\big| \leq 2k(2d+1)\gamma_{d_X} \log N \tag{A.149}$$

where d_X is the dimension of the space $\mathcal{X}$.

Case III.2: $\|x^{(i)} - x^{(j)}\| > \rho_{k,i}$. This case is somewhat similar to the Case II. Here the $\rho_{k,i}$

will not change. But the $n_S^{(i)}$'s for all subsets $S \neq X$ can decrease by 1. We can write:

$$\begin{aligned}
& \sum_{\text{Case III.2}} \left|\zeta_i(X) - \zeta_i(X_{\setminus j})\right| && \text{(A.150)}\\
\leq\ & \sum_{i=1}^{N}\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}} \mathbf{1}_{\left\{x_{\mathrm{pa}(l)^{(i)}} = x_{\mathrm{pa}(l)^{(j)}}\right\}} \left|\log(n^{(i)}_{\mathrm{pa}(X_l)} + 1) - \log n^{(i)}_{\mathrm{pa}(X_l)}\right| && \text{(A.151)}\\
& + \sum_{i=1}^{N}\sum_{l=1}^{d} \mathbf{1}_{\left\{x_{\mathrm{pa}+(l)^{(i)}} = x_{\mathrm{pa}+(l)^{(j)}}\right\}} \left|\log(n^{(i)}_{\mathrm{pa}+(X_l)} + 1) - \log n^{(i)}_{\mathrm{pa}+(X_l)}\right| && \text{(A.152)}\\
& + \sum_{i=1}^{N-1} \left(\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}} + 1\right)\Big(\log N - \log(N-1)\Big) && \text{(A.153)}\\
\leq\ & \sum_{i=1}^{N}\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}} \mathbf{1}_{\left\{x_{\mathrm{pa}(l)^{(i)}} = x_{\mathrm{pa}(l)^{(j)}}\right\}} \frac{1}{n^{(i)}_{\mathrm{pa}(X_l)}} && \text{(A.154)}\\
& + \sum_{i=1}^{N}\sum_{l=1}^{d} \mathbf{1}_{\left\{x_{\mathrm{pa}+(l)^{(i)}} = x_{\mathrm{pa}+(l)^{(j)}}\right\}} \frac{1}{n^{(i)}_{\mathrm{pa}+(X_l)}} && \text{(A.155)}\\
& + \sum_{i=1}^{N-1} \left(\sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}} + 1\right)\frac{1}{N-1} && \text{(A.156)}\\
\leq\ & \sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)\neq\emptyset\}} \gamma_{d_{\mathrm{pa}(X_l)}} \log(N+1) && \text{(A.157)}\\
& + \sum_{l=1}^{d} \gamma_{d_{\mathrm{pa}+(X_l)}} \log(N+1) + \sum_{l=1}^{d} \mathbf{1}_{\{\mathrm{pa}(X_l)=\emptyset\}} + 1 && \text{(A.158)}\\
\leq\ & 2d\gamma_{d_X} \log(N+1) + 1 && \text{(A.159)}
\end{aligned}$$

The inequality A.158 follows from the second part of the Lemma C.1 from [79].

Now combining all three cases together, we have:

$$\begin{aligned}
& \sum_{i=1}^{N} \left|\zeta_i(X) - \zeta_i(X_{\setminus j})\right| \\
\leq\ & 2(2d+1)\log N + 2(d+1) + 2k(2d+1)\gamma_{d_X}\log N + 2d\gamma_{d_X}\log(N+1) + 1 \\
\leq\ & 6k(2d+1)\gamma_{d_X}\log(N+1) && \text{(A.160)}
\end{aligned}$$

Thus:

$$\sup_{x^{(1)},\ldots,x^{(N)},x'^{(j)}} \left| \widehat{\mathbb{GDMM}}^{(N)}(X,\mathcal{G}) - \widehat{\mathbb{GDMM}}^{(N)}(X^{(j)},\mathcal{G}) \right| \leq \frac{12k(2d+1)\gamma_{d_X}\log(N+1)}{N} \quad \text{(A.161)}$$

And:

$$\begin{aligned}
& \operatorname{Var}\left[\widehat{\mathbb{GDMM}}^{(N)}(X,\mathcal{G})\right] && \text{(A.162)}\\
\leq\ & \frac{1}{2}\sum_{j=1}^{N}\mathbb{E}\left[\left(\widehat{\mathbb{GDMM}}^{(N)}(X,\mathcal{G}) - \widehat{\mathbb{GDMM}}^{(N)}(X^{(j)},\mathcal{G})\right)^2\right] && \text{(A.163)}\\
\leq\ & \frac{1}{2}\sum_{j=1}^{N}\sup_{x^{(1)},\ldots,x^{(N)},x'^{(j)}}\left(\widehat{\mathbb{GDMM}}^{(N)}(X,\mathcal{G}) - \widehat{\mathbb{GDMM}}^{(N)}(X^{(j)},\mathcal{G})\right)^2 && \text{(A.164)}\\
\leq\ & \frac{1}{2}\sum_{j=1}^{N}\left(\frac{12k(2d+1)\gamma_{d_X}\log(N+1)}{N}\right)^2 && \text{(A.165)}\\
=\ & \frac{72\gamma_{d_X}^2\left(k(2d+1)\log N\right)^2}{N} && \text{(A.166)}
\end{aligned}$$

Therefore if $(k_N \log N)^2/N \to 0$ as N goes to infinity, we have $\lim_{N\to\infty}\operatorname{Var}\left[\widehat{\mathbb{GDMM}}^{(N)}(X,\mathcal{G})\right] = 0$.

Appendix B

SUPPLEMENTARY MATERIAL FOR CHAPTER 3

B.1 Proof of Theorem 3

In this section, we provide a proof for Theorem 3. Lemma 21 suggests a simpler condition under which the RDI graph will be the same as the causality graph. The proof of Part (a) of Theorem 3 is identical to the proof of that lemma; and follows from observing that Property A is all that is needed for the proof to go through.

Before that, the Prop. B.1.1 is stated without proof, which will be used through the proof of 21.

Proposition B.1.1. *Let Σ_X be a covariance matrix of m zero-mean jointly Gaussian random variables denoted by $\underline{X}$. Then $\exists \quad \underline{u} \neq \underline{0} \quad \forall \underline{x} : \underline{u}^T(\underline{x} - \bar{\underline{x}}) = 0$ if and only if Σ_X is not positive definite.*

Lemma 21. *If the covariance matrix of the linear deterministic system Σ_x is positive definite, then $G_{RDI} = G_C$.*

Proof. In a linear deterministic system, every variable X_j at each time t can be described as $X_j(t) = \sum_u A_{j,u} X_u(t-1)$. Let's assume $G_C = (V, E)$ is the causality graph of the system. it can be observed that $(i, j) \in E$ if $A_{j,i} \neq 0$, and $(i, j) \notin E$ otherwise.

We will show that given Σ_X is positive definite, for each pair (i, j) if $A_{j,i} = 0$ then $RDI\left(X_i \to X_j | \{X_u\}_{u \in [m]-\{i,j\}}\right) = 0$. Similarly, $RDI(X_i \to X_j | \{X_u\}_{u \in [m]-\{i,j\}}) > 0$ if $A_{j,i} \neq 0$.

We can write:

$$\begin{aligned} & RDI\left(X_i \to X_j | \{X_u\}_{u\in[m]-\{i,j\}}\right) \\ = \; & H\left(X_j(t)|\{X_u(t-1)\}_{u\in[m]-\{i\}}\right) - \underbrace{H\left(X_j(t)|\{X_u(t-1)\}_{u\in[m]}\right)}_{0} \qquad \text{(B.1)} \\ = \; & H\left(A_{j,i}X_i(t-1)|\{X_u(t-1)\}_{u\in[m]-\{i\}}\right) \qquad \text{(B.2)} \end{aligned}$$

If $A_{j,i} = 0$, then $H\left(A_{j,i}X_i(t-1)|\{X_u(t-1)\}_{u\in[m]-\{i\}}\right) = 0$ which means $RDI\left(X_i \to X_j|\{X_u\}_{u\in[m]-\{i,j\}}\right) = 0$ yielding the lemma.

If $A_{j,i} \neq 0$, then assuming $H\left(A_{j,i}X_i(t-1)|\{X_u(t-1)\}_{u\in[m]-\{i\}}\right) = 0$ implies that $X_i(t-1)$ is a function of $\{X_u(t-1)\}_{u\in[m]-\{i\}}$, i.e. there exists a linear function l_i such that $\forall t : X_i(t) = l_i\left(\{X_u(t)\}_{u\in[m]-\{i\}}\right)$ which in turn implies $\exists u \neq 0 : u^T x(t-1) = 0$, so from Prop.B.1.1 we conclude that Σ_x is not positive definite which contradicts our assumption. Thus $RDI\left(X_i \to X_j|\{X_u\}_{u\in[m]-\{i,j\}}\right) > 0$ which yields the lemma. □

So we see for a deterministic linear system, if the Σ_x is positive definite, then RDI can return the correct causality graph. As pointed out earlier, only Property A is needed for the proof of this Lemma and that concludes the proof of part (a) of Theorem 3.

We now need to prove (b) of Theorem 3 which shows that if property A is not true, then there's no method that can infer the correct causality graph. $\exists \underline{u}, u_{i_0} \neq 0 : \underline{u}^T \underline{X} = 0$ can be written as $\forall t : X_{i_0}(t) = \sum_{k\in[m]-\{i_0\}}(-u_k)X_k(t)$. From the dynamics of the system and knowing that the edge $(i_0, j_0) \in E$, i.e. $A_{j_0,i_0} \neq 0$, we can write:

$$\begin{aligned} X_j(t) &= \sum_{k\in[m]} A_{j,k}X_k(t-1) \qquad \text{(B.3)} \\ &= \sum_{k\in[m]-\{i_0\}} A_{j,k}X_k(t-1) + A_{j,i_0}X_{i_0}(t-1) \qquad \text{(B.4)} \\ &= \sum_{k\in[m]-\{i_0\}} \left(A_{j,k} - u_k\right) X_k(t-1) \qquad \text{(B.5)} \\ &= \sum_{k\in[m]-\{i_0\}} \tilde{A}_{j,k}X_k(t-1) \qquad \text{(B.6)} \end{aligned}$$

So there's an alternative system $\tilde{A} \neq A$ which gives us the same dynamics for the system. So there's an ambiguity in the system and no method will be able to return the correct graph. This completes the proof of Theorem 3.

We note that in linear deterministic systems, Σ_X is equal to identity only when A is orthonormal. In general Σ_X is non-singular only if the system A has eigenvalues 1, which implies that the system is not necessarily ergodic. Thus in linear deterministic systems, there is no example of a system that is both ergodic and has an invertible covariance matrix.

B.2 Proof of Theorem 4

We first prove a version that states that Σ_X is positive definite implying that $G_{RDI} = G_C$ in Lemma 22. We observe that only Property A is needed for the proof of Lemma 22, which concludes the proof of this theorem.

Lemma 22. *Let Σ_X be the covariance matrix of the stationary distribution for a linear system with additive Gaussian noise. If Σ_X is positive definite, then $G_{RDI} = G_C$.*

Proof. We take the steps similar to the ones in the Theorem 3. In the linear system described above, every variable X_j at each time t can be described as $X_j(t) = \sum_u A_{j,u} X_u(t-1) + N_j(t)$. We will show that given Σ_X is positive definite, for each pair (i,j) if $A_{j,i} = 0$ then $RDI\left(X_i \to X_j | \{X_u\}_{u\in[m]-\{i,j\}}\right) = 0$. Similarly, $RDI\left(X_i \to X_j | \{X_u\}_{u\in[m]-\{i,j\}}\right) > 0$ if $A_{j,i} \neq 0$.

We can write:

$$\begin{aligned} & RDI\left(X_i \to X_j | \{X_u\}_{u\in[m]-\{i,j\}}\right) \\ = \; & H\left(X_j(t) | \{X_u(t-1)\}_{u\in[m]-\{i\}}\right) - H\left(X_j(t) | \{X_u(t-1)\}_{u\in[m]}\right) \qquad \text{(B.7)} \\ = \; & H\left(A_{j,i} X_i(t-1) + N_j(t) | \{X_u(t-1)\}_{u\in[m]-\{i\}}\right) - H\left(N_j(t)\right) \qquad \text{(B.8)} \end{aligned}$$

If $A_{j,i} = 0$, then the RHS of (B.8) is zero, which yields the theorem. Consider the case $A_{j,i} \neq 0$. Since the process is Gaussian, and $N_j(t)$ is independent of the process at

time $t-1$, $H\left(A_{j,i}X_i(t-1)+N_j(t)|\{X_u(t-1)\}_{u\in[m]-\{i\}}\right)$ is equal to $\frac{1}{2}\log(2\pi e(A_{j,i}^2\sigma_e^2+\sigma_N^2))$ where σ_e^2 is the mean-square error in estimating $X_i(t-1)$ from $X_u(t-1)_{u\in[m]-\{i\}}$. Note that $H\left(N_j(t)\right)=\frac{1}{2}\log(2\pi e\sigma_N^2)$. Now, equality holds in (B.8) if and only if $\sigma_e=0$ which implies that the covariance matrix is singular (indeed, there exists a $\underline{u}$ with $u_i\neq 0$ such that $\underline{u}^T\Sigma_X\underline{u}=0$). □

Next, we study that the stability of the dynamical system implies that $\Sigma_X=\sum_{t=0}^{\infty}A^t\Sigma_n(A^t)^T$ is well defined. In fact, it is also known that when the system is stable, the Gaussian process is ergodic as well [227].

Lemma 23. *Let A and Σ_n be $m\times m$ matrices. Then if the eigenvalues of A are smaller than 1, the limit $\sum_{t=0}^{\infty}A^t\Sigma_n(A^t)^T$ exists, i.e. the series $\sum_{t=0}^{\tau}A^t\Sigma_n(A^t)^T$ is absolutely convergent.*

Proof. Let $B(\tau)=\sum_{t=0}^{\tau}A^t\Sigma_n(A^t)^T$. We show that every element of B converges. For arbitrary $B_{i,j}(\tau)$:

$$\begin{aligned} B_{i,j}(\tau) &= e_i^TB(\tau)e_j && \text{(B.9)}\\ &= e_i^T(\sum_{t=0}^{\tau}A^t\Sigma_n(A^t)^T)e_j && \text{(B.10)}\\ &= \sum_{t=0}^{\tau}e_i^TA^t\Sigma_n(A^t)^Te_j && \text{(B.11)}\\ &= \sum_{t=0}^{\tau}e_i^TA^t\Sigma_n(e_j^TA^t)^T && \text{(B.12)} \end{aligned}$$

for each term $a_t=e_i^TA^t\Sigma_n(e_j^TA^t)^T$ we can write:

$$
\begin{aligned}
a_t &\leq \lambda_{max}(\Sigma_n) e_i^T A^t (e_j^T A^t)^T && \text{(B.13)}\\
&= \lambda_{max}(\Sigma_n) \langle e_i^T A^t, e_j^T A^t \rangle && \text{(B.14)}\\
&\leq \lambda_{max}(\Sigma_n) \|e_i^T A^t\| \|e_j^T A^t\| && \text{(B.15)}\\
&\leq \lambda_{max}(\Sigma_n) |e_i| \lambda_{max}(A^t) |e_j| \lambda_{max}(A^t) && \text{(B.16)}\\
&= \lambda_{max}(\Sigma_n) \lambda_{max}(A^t)^2 && \text{(B.17)}\\
&= \lambda_{max}(\Sigma_n) \lambda_{max}(A)^{2t} && \text{(B.18)}
\end{aligned}
$$

Let's put $b_t = \lambda_{max}(\Sigma_n)\lambda_{max}(A)^{2t}$. The series $\sum_{t=0}^{\tau} b_t$ is a geometric series and it will be absolutely convergent if $\lambda_{max}(A) < 1$. So according to the direct comparison test of series, $\sum_{t=0}^{\tau} a_t = \sum_{t=0}^{\infty} A^t \Sigma_n (A^t)^T$ is absolutely convergent too.

□

B.3 Proof of Theorem 6 and Corollary 6.1

For Theorem 6 we have:

$$
\begin{aligned}
\Sigma_x > 0 &\iff \forall u \neq 0 : u^T \Sigma_X u = \sum_{t=0}^{k-1} u^T A^t e_j e_j^T (A^t)^T u = \sum_{t=0}^{k-1} \|u^T A^t e_j\|^2 > 0 && \text{(B.19)}\\
&\iff \exists t \geq 0 : u^T A^t e_j \neq 0 && \text{(B.20)}\\
&\iff u^T [\ldots | A^{k-1} e_j | \ldots | A e_j | e_j] \neq 0 && \text{(B.21)}\\
&\iff rank\left([\ldots | A^{k-1} e_j | \ldots | A e_j | e_j]\right) = m && \text{(B.22)}
\end{aligned}
$$

For Corollary 6.1 we have:

$$\Sigma_x > 0 \iff \forall u \neq 0 : u^T \Sigma_x u = \sum_{t=0}^{k-1} u^T A^t \sum_{j \in S_m} r_j e_j e_j^T (A^t)^T u = \sum_{t=0}^{k-1} \sum_{j \in S_m} |r_j| \|u^T A^t e_j\|^2 > 0$$

$$\iff \exists t \geq 0, j \in S_m : u^T A^t e_j \neq 0 \tag{B.23}$$

$$\iff \exists j \in S_m : u^T [\dots |A^{k-1} e_j| \dots |A e_j| e_j] \neq 0 \tag{B.24}$$

$$\iff u^T [B_{j_1} | \dots | B_{j_{|S_m|}}] \neq 0 \tag{B.25}$$

$$\iff rank \left([B_{j_1} | \dots | B_{j_{|S_m|}}] \right) = m \tag{B.26}$$

B.4 Proof of Theorem 13

First, we prove (a).

$RDI\left(X_i \to X_j | \{X_u\}_{u \in [m]-\{i,j\}}\right)$ can be written as:

$$H\left(X_j(t) | \{X_u(t-1)\}_{u \in [m]-\{i\}}\right) - \underbrace{H\left(X_j(t) | \{X_u(t-1)\}_{u \in [m]}\right)}_{0} \tag{B.27}$$

$$= H\left(X_j(t) | \{X_u(t-1)\}_{u \in [m]-\{i\}}\right) \tag{B.28}$$

$$= H\left(X_j | \{X_u\}_{u \in [m]-\{i\}}\right) \tag{B.29}$$

where the last equation holds under the stationary distribution. If X_j is not a function of X_i then $H\left(X_j(t) | \{X_u(t-1)\}_{u \in [m]-\{i\}}\right) = 0$. So $RDI\left(X_i \to X_j | \{X_u\}_{u \in [m]-\{i,j\}}\right) = 0$ and RDI graph correctly doesn't include the (i, j) edge.

If $X_j(t)$ is a function of $X_i(t-1)$, it implies that (i, j) is in the edge set of the G_C. Given that property B is satisfied, $H\left(X_j(t) | \{X_u(t-1)\}_{u \in [m]-\{i\}}\right) > 0$ and (i, j) is included in G_{RDI} hence (a) of the theorem is proved.

Next, we prove (b). Assuming property B is not satisfied means that there exists an edge (i_0, j_0) in G_C for which $H\left(X_j(t) | \{X_u(t-1)\}_{u \in [m]-\{i_0\}}\right) = 0$. So $\exists h : X_{j_0}(t) = h\left(\{X_u(t-1)\}_{u \in [m]-\{i_0\}}\right)$.

However, (i_0, j_0) included in G_C means

$$\exists g_{j_0} : X_{j_0}(t) = g_{j_0}\left(\{X_u(t-1)\}_{u\in[m]-\{i_0\}}, X_{i_0}(t-1)\right) \tag{B.30}$$

So there as an alternate function $h \neq g_{j_0}$, which yields the same system dynamics. So there's an ambiguity in the system and no algorithm can infer the correct graph.

B.5 Proof of the Theorem 15

Here we'll try to introduce a proof for qCMI algorithm we devised. As we know, the conditional mutual information is defined as,

$$I(X;Y|Z) = \int f_{XYZ}(x,y,z)\log\left(\frac{f_{Y|XZ}(y|x,z)}{f_{Y|Z}(y|z)}\right)dxdydz \tag{B.31}$$

The qCMI is defined as the mutual information of X and Y given Z when the joint distribution of X and Z is replaced by a joint uniform distribution $q_{XZ}(x,z)$,

$$qCMI_{X\to Y|Z} = I^q(X;Y|Z) = \int f_{Y|XZ}(y|x,z)q_{XZ}(x,z)\log\left(\frac{f_{Y|XZ}(y|x,z)}{f^q_{Y|Z}(y|z)}\right)dxdydz \tag{B.32}$$

In which:

$$f^q_{Y|Z}(y|z) = \frac{f^q_{YZ}(y,z)}{f^q_Z(z)} \tag{B.33}$$

$$f^q_{YZ}(y,z) = \int f^q_{XYZ}(x,y,z)dx \tag{B.34}$$

$$f^q_Z(z) = \int f^q_{XYZ}(x,y,z)dxdy \tag{B.35}$$

from now on, the superscript q over each distribution function implies that the actual $f_{XZ}(x,z)$ is replaced by $q_{XZ}(x,z)$. Equivalently, qCMI can be written as,

$$I^q(X;Y|Z) = -h^q(X,Y,Z) + h^q(X,Z) + h^q(X,Y) - h^q(Z) \tag{B.36}$$

where,

$$h^q(X,Y,Z) \equiv -\int f^q_{XYZ}(x,y,z)\log f_{XYZ}(x,y,z)dxdydz \tag{B.37}$$

$$h^q(X,Z) \equiv -\int f^q_{XYZ}(x,y,z)\log f_{XZ}(x,z)dxdydz \tag{B.38}$$

$$h^q(Y,Z) \equiv -\int f^q_{XYZ}(x,y,z)\log f^q_{YZ}(y,z)dxdydz \tag{B.39}$$

$$h^q(Z) \equiv -\int f^q_{XYZ}(x,y,z)\log f^q_{Z}(z)dxdydz \tag{B.40}$$

Note that $-h^q(X,Y,Z) + h^q(X,Z) = -h^q(Y|X,Z) = \int f^q_{XYZ}(x,y,z)\log\left(f_{Y|XZ}(y|x,z)\right)dxdydz$. So the term inside the logarithm is independent of the distribution over (X,Z) and hence $\log f_{XYZ}(x,y,z)$ and $\log f_{XZ}(x,z)$ appear when defining $h^q(X,Y,Z)$ and $h^q(X,Z)$.

Here we introduce a qCMI estimator, based on the KSG estimator for *UMI*. Remember the KSG-type estimator for the conditional MI,

$$\hat{I}(X;Y|Z) = \frac{1}{N}\sum_{i=1}^{N}\left(\psi(k) - \log(n_{xz,i}) - \log(n_{yz,i}) + \log(n_{z,i})\right) + C(d_x,d_y,d_z) \tag{B.41}$$

where

$$C(d_x,d_y,d_z) := \log\left(\frac{c_{d_x+d_z}c_{d_y+d_z}}{c_{d_x+d_y+d_z}c_{d_z}}\right) \tag{B.42}$$

The estimator for the qCMI is as below,

$$\hat{I}^q_N(X;Y|Z) = \frac{1}{N}\sum_{i=1}^{N}\omega_i g_i(x_i,y_i,z_i) \tag{B.43}$$

where,

$$\omega_i \equiv \frac{q_{XZ}(x_i, z_i)}{\hat{f}_{XZ}(x_i, z_i)}. \tag{B.44}$$

$$g_i(x_i, y_i, z_i) \equiv \psi(k) - \log(n_{xz}) - \log(\sum_{\|(y_i - y_j, z_i - z_j)\| < \rho_{k,i}} \omega_j) + \log(\sum_{\|z_i - z_j\| < \rho_{k,i}} \omega_j) + C(d_x, d_y, d_z) \tag{B.45}$$

B.6 KSG estimator for qCMI: Proof of convergence

Similar to the [76] we define,

$$\omega'_i \equiv \frac{q_{XZ}(x_i, z_i)}{f_{XZ}(x_i, z_i)}. \tag{B.46}$$

$$n'_{yz,i} \equiv \sum_{j \in N_{yz}^{\epsilon(i)}} \omega'_j. \tag{B.47}$$

$$n'_{z,i} \equiv \sum_{j \in N_{z}^{\epsilon(i)}} \omega'_j. \tag{B.48}$$

$$g'(x_i, y_i, z_i) \equiv \psi(k) - \log(n_{xz}) - \log(n'_{yz,i}) + \log(n'_{z,i}) + C(d_x, d_y, d_z) \tag{B.49}$$

From the triangle inequality, we can write,

$$|\hat{I}_N^q(X;Y|Z) - I^q(X;Y|Z)| \tag{B.50}$$

$$\leq |\hat{I}_N^q(X;Y|Z) - \frac{1}{N}\sum_{i=1}^{N} \omega'_i g'(x_i, y_i, z_i)| \tag{B.51}$$

$$+ |\frac{1}{N}\sum_{i=1}^{N} \omega'_i g'(x_i, y_i, z_i) - I^q(X;Y|Z)| \tag{B.52}$$

To show the convergence of the KSG estimator for qCMI, we will show that (B.51) and (B.52) both converge to zero. The Lemma 24 proves that (B.51) converges to zero. The lemma will be proven through the Section B.7.

Lemma 24. *The term* (B.51) *converges to* 0 *as* $N \longrightarrow \infty$ *in probability.*

For (B.52) we will first write the left-hand side term as four entropy terms and show the convergence of each of the terms to their respective true entropy term. This will be done in the Lemmas 25 and 26. So we can write,

$$\frac{1}{N}\sum_{i=1}^{N}\omega_i' g'(x_i,y_i,z_i) = \hat{h}_N^q(Y,Z) + \hat{h}_N^q(X,Z) - \hat{h}_N^q(X,Y,Z) - \hat{h}_N^q(Z) - \sum_{i=1}^{N}\frac{\omega_i'}{N}\left(\log(N-1)+\psi(N)\right) \tag{B.53}$$

where

$$\hat{h}_N^q(X,Y,Z) \equiv \frac{1}{N}\sum_{i=1}^{N}\omega_i'\left(-\psi(k)+\psi(N)+\log c_{d_x+d_y+d_z}+(d_x+d_y+d_z)\log\rho_{k,i}\right) \tag{B.54}$$

$$\hat{h}_N^q(X,Z) \equiv \frac{1}{N}\sum_{i=1}^{N}\omega_i'\left(-\log(n_{xz,i})+\log(N-1)+\log c_{d_x+d_z}+(d_x+d_z)\log\rho_{k,i}\right) \tag{B.55}$$

$$\hat{h}_N^q(Y,Z) \equiv \frac{1}{N}\sum_{i=1}^{N}\omega_i'\left(-\log(n'_{yz,i})+\log(N-1)+\log c_{d_y+d_z}+(d_y+d_z)\log\rho_{k,i}\right) \tag{B.56}$$

$$\hat{h}_N^q(Z) \equiv \frac{1}{N}\sum_{i=1}^{N}\omega_i'\left(-\log(n'_{z,i})+\log(N-1)+\log c_{d_z}+d_z\log\rho_{k,i}\right) \tag{B.57}$$

The term $\sum_{i=1}^{N}\frac{\omega_i'}{N}\left(\log(N-1)+\psi(N)\right)$ converges to 0 as $N\longrightarrow\infty$. We will prove the convergence of the rest of the terms in the following lemmas, proving the Theorem 15.

Lemma 25. *Under the Assumption 2,* $\hat{h}_N^q(X,Y,Z)\xrightarrow{p}h^q(X,Y,Z)$ *as* $N\longrightarrow\infty$.

Proof. It directly follows from the Lemma 2 from [76]. We just need to let $\tilde{X}\equiv(X,Z)$ and then show that $\hat{h}_N^q(\tilde{X},Y)\longrightarrow h^q(\tilde{X},Y)$ directly using the Lemma 2 from [76]. □

Lemma 26. *For* $N\longrightarrow\infty$,

$$\hat{h}_N^q(X,Z)+\hat{h}_N^q(Y,Z)-\hat{h}_N^q(Z)\xrightarrow{p}h^q(X,Z)+h^q(Y,Z)-h^q(Z) \tag{B.58}$$

B.7 Proof of lemma 24

Th proof is analog to that of Lemma 1 in [76]. The term (B.51) is upper-bounded as,

$$
\begin{aligned}
& |\hat{I}^q(X;Y|Z) - \frac{1}{N}\sum_{i=1}^{N}\omega_i' g'(x_i,y_i,z_i)| && \text{(B.59)}\\
= \;& |\frac{1}{N}\sum_{i=1}^{N}\left(\omega_i g(x_i,y_i,z_i) - \omega_i' g'(x_i,y_i,z_i)\right)| && \text{(B.60)}\\
\leq \;& \frac{1}{N}\sum_{i=1}^{N}|\omega_i g(x_i,y_i,z_i) - \omega_i' g'(x_i,y_i,z_i)| && \text{(B.61)}\\
\leq \;& \frac{1}{N}\sum_{i=1}^{N}\left(|\omega_i-\omega_i'||g'(x_i,y_i,z_i)| - \omega_i|g(x_i,y_i,z_i) - g'(x_i,y_i,z_i)|\right) && \text{(B.62)}\\
= \;& \frac{1}{N}\sum_{i=1}^{N}\left(|\omega_i-\omega_i'||g'(x_i,y_i,z_i)| + \omega_i|\log(n_{yz,i}) - \log(n'_{yz,i})| + \omega_i|\log(n_{z,i}) - \log(n'_{z,i})|\right) && \text{(B.63)}\\
\leq \;& \frac{1}{N}\sum_{i=1}^{N}\Bigg(|\omega_i-\omega_i'||g'(x_i,y_i,z_i)| + \omega_i|n_{yz,i} - n'_{yz,i}|(\frac{1}{2n_{yz,i}} + \frac{1}{2n'_{yz,i}})\\
& \qquad +\omega_i|n_{z,i} - n'_{z,i}|(\frac{1}{2n_{z,i}} + \frac{1}{2n'_{z,i}})\Bigg) && \text{(B.64)}\\
\leq \;& \frac{1}{N}\sum_{i=1}^{N}|\omega_i-\omega_i'||g'(x_i,y_i,z_i)| + \sum_{i=1}^{N}\frac{\omega_i}{N}\left(\frac{\max_{1\leq j\leq N}|\omega_i'-\omega_i|}{\min_{1\leq j\leq N}\omega_i'} + \frac{\max_{1\leq j\leq N}|\omega_i'-\omega_i|}{\min_{1\leq j\leq N}\omega_i}\right) && \text{(B.65)}\\
\leq \;& \max_{1\leq i\leq N}|\omega_i-\omega_i'|\left(\max_{1\leq i\leq N}|g'(x_i,y_i,z_i)| + \frac{1}{\min_{1\leq j\leq N}\omega_i'} + \frac{1}{\min_{1\leq j\leq N}\omega_i}\right) && \text{(B.66)}
\end{aligned}
$$

The term $g'(x_i,y_i,z_i)$ can be easily lower-bounded and upper-bounded as,

$$-2\log N \leq g'(x_i,y_i,z_i) \leq 2\log N \tag{B.67}$$

Thus, for any $\epsilon > 0$ and sufficiently large N such that $\log N > \max\{C_2\epsilon/3, 3C_2\}$ and, if

$|\omega_i - \omega_i'| < \epsilon/(3\log N)$ for all i , we have,

$$\max_{1\le i\le N} |\omega_i - \omega_i'| \left(\max_{1\le i\le N} |g'(x_i, y_i, z_i)| + \frac{1}{\min_{1\le j\le N} \omega_i'} + \frac{1}{\min_{1\le j\le N} \omega_i} \right) \tag{B.68}$$

$$\le \frac{\epsilon}{3\log N} \left(2\log N + C_2 + \frac{1}{1/C_2 - \frac{\epsilon}{3\log N}} \right) \tag{B.69}$$

$$\le \frac{\epsilon}{3\log N} (2\log N + C_2 + 2C_2) \le \epsilon \tag{B.70}$$

So for any $\epsilon > 0$ and sufficiently large N:

$$\mathbb{P}\left(\frac{1}{N} \sum_{i=1}^{N} |\omega_i g(x_i, y_i, z_i) - \omega_i' g'(x_i, y_i, z_i)| > \epsilon \right) \le \mathbb{P}\left(\max_{1\le i\le N} |\omega_i - \omega_i'| > \frac{\epsilon}{3\log N} \right) \tag{B.71}$$

Following the proof of Lemma 1 in [76], the term $\mathbb{P}\left(\max_{1\le i\le N} |\omega_i - \omega_i'| > \frac{\epsilon}{3\log N}\right)$ converges to zero as $N \longrightarrow \infty$. So the desired convergence for the term (B.51) is obtained.

B.8 Proof of lemma 26

If we define,

$$\hat{f}_{XZ}(x_i, z_i) \equiv \frac{n_{xz,i}}{(N-1)c_{d_x+d_z}\rho_{k,i}^{d_x+d_z}} \tag{B.72}$$

$$\hat{f}^q_{YZ}(y_i, z_i) \equiv \frac{n'_{yz,i}}{(N-1)c_{d_y+d_z}\rho_{k,i}^{d_y+d_z}} \tag{B.73}$$

$$\hat{f}^q_{Z}(z_i) \equiv \frac{n'_{z,i}}{(N-1)c_{d_z}\rho_{k,i}^{d_z}} \tag{B.74}$$

$$\hat{a}_i \equiv \log \hat{f}_{XZ}(x_i, z_i) + \log \hat{f}^q_{YZ}(y_i, z_i) - \log \hat{f}^q_{Z}(z_i) \tag{B.75}$$

$$a_i \equiv \log f_{XZ}(x_i, z_i) + \log f^q_{YZ}(y_i, z_i) - \log f^q_{Z}(z_i) \tag{B.76}$$

Then,

$$\begin{aligned}
& \hat{h}^q_N(X,Z) + \hat{h}^q_N(Y,Z) - \hat{h}^q_N(Z) \\
= & -\sum_{i=1}^{N} \frac{\omega'_i}{N} \left(\log \hat{f}_{XZ}(x_i,z_i) + \log \hat{f}^q_{YZ}(y_i,z_i) - \log \hat{f}^q_Z(z_i) \right) && \text{(B.77)} \\
= & -\sum_{i=1}^{N} \frac{\omega'_i}{N} \hat{a}_i && \text{(B.78)}
\end{aligned}$$

Now we can write,

$$\begin{aligned}
& |\hat{h}^q_N(X,Z) + \hat{h}^q_N(Y,Z) - \hat{h}^q_N(Z) - (h^q(X,Z) + h^q(Y,Z) - h^q(Z))\,| && \text{(B.79)} \\
\leq & \; |h^q(X,Z) + h^q(Y,Z) - h^q(Z) - \sum_{i=1}^{N} -\frac{\omega'_i}{N} a_i| && \text{(B.80)} \\
& + \sum_{i=1}^{N} \frac{\omega'_i}{N} |a_i - \hat{a}_i| && \text{(B.81)}
\end{aligned}$$

For the term (B.80), since the terms $a_i = \omega'_i \left(\log f_{XZ}(x_i,z_i) + \log f^q_{YZ}(y_i,z_i) + \log f^q_Z(z_i) \right)$ are i.i.d random variables, given the Assumption 2, by the strong law of large numbers, we can write,

$$\begin{aligned}
\sum_{i=1}^{N} -\frac{\omega'_i}{N} a_i \quad = & \quad \sum_{i=1}^{N} -\frac{\omega'_i}{N} \left(\log f_{XZ}(x,z) + \log f^q_{YZ}(y,z) - \log f^q_Z(z) \right) && \text{(B.82)} \\
\longrightarrow & \quad E\left[-\frac{q_{XZ}(x,z)}{f_{XZ}(x,z)} \left(\log f_{XZ}(x,z) + \log f^q_{YZ}(y,z) - \log f^q_Z(z) \right) \right] && \text{(B.83)} \\
= & \quad -\int f_{Y|XZ}(y|x,z) q_{XZ}(x,z) \\
& \qquad \times \left(\log f_{XZ}(x,z) + \log f^q_{YZ}(y,z) - \log f^q_Z(z) \right) dxdydz && \text{(B.84)} \\
= & \quad h^q(X,Z) + h^q(Y,Z) - h^q(Z) && \text{(B.85)}
\end{aligned}$$

Therefore, the term (B.80) converges to 0 almost surely. For the term (B.81), let $T_i =$

(X_i, Y_i, Z_i), thus $t = (x, y, z)$, and $f_T(t) = f_{XYZ}(x, y, z)$. For any fixed $\epsilon > 0$, we can write,

$$\mathbb{P}\left(\sum_{i=1}^{N} \frac{\omega_i'}{N}|a_i - \hat{a}_i| > \epsilon\right) \tag{B.86}$$

$$\leq \mathbb{P}\left(\bigcup_{i=1}^{N}\{|a_i - \hat{a}_i| > \epsilon/2\}\right) + \mathbb{P}\left(\sum_{i=1}^{N}\omega_i' > 2N\right) \tag{B.87}$$

The second term converges to zero. The first term can be bounded as,

$$\mathbb{P}\left(\bigcup_{i=1}^{N}\{|a_i - \hat{a}_i| > \epsilon/2\}\right) \tag{B.88}$$

$$\leq N.\mathbb{P}\left(|a_i - \hat{a}_i| > \epsilon/2\right) \leq N\int \mathbb{P}\left(|a_i - \hat{a}_i| > \epsilon/2|T_i = t\right) f_T(t)dt \tag{B.89}$$

The term $\mathbb{P}\left(|a_i - \hat{a}_i| > \epsilon/2|T_i = t\right)$ can be upper-bounded by $I_1(t) + I_2(t) + I_3(t) + I_4(t) + I_5(t)$, where,

$$I_1(t) = \mathbb{P}\left(\rho_{k,i} > r_1|T_i = t\right) \tag{B.90}$$

$$I_2(t) = \mathbb{P}\left(\rho_{k,i} < r_2|T_i = t\right) \tag{B.91}$$

$$I_3(t) = \int_{r=r_2}^{r_1} \mathbb{P}(|\log f_{XZ}(x_i, z_i) - \log \hat{f}_{XZ}(x_i, z_i)| > \epsilon/6|\rho_{k,i} = r, T_i = t)f_\rho(r)dr \tag{B.92}$$

$$I_4(t) = \int_{r=r_2}^{r_1} \mathbb{P}(|\log f^q_{YZ}(y_i, z_i) - \log \hat{f}^q_{YZ}(y_i, z_i)| > \epsilon/6|\rho_{k,i} = r, T_i = t)f_\rho(r)dr \tag{B.93}$$

$$I_5(t) = \int_{r=r_2}^{r_1} \mathbb{P}(|\log f^q_{Z}(z_i) - \log \hat{f}^q_{Z}(z_i)| > \epsilon/6|\rho_{k,i} = r, T_i = t)f_\rho(r)dr \tag{B.94}$$

In which,

$$r_1 \equiv \log N(Nf_T(t)c_{d_x+d_y+d_z})^{\frac{-1}{d_x+d_y+d_z}} \tag{B.95}$$

$$r_2 \equiv \max\Big\{(\log N)^2(Nf_{XZ}(x, z)c_{d_x+d_z})^{\frac{-1}{d_x+d_z}}, (\log N)^2(Nf^q(y, z)c_{d_y+d_z})^{\frac{-1}{d_y+d_z}}, (\log N)^2(Nf^q_Z(z)c_{d_z})^{\frac{-1}{d_z}}\Big\} \tag{B.96}$$

The terms $I_1(t)$ and $I_2(t)$ represent the probability that the value of $\rho_{k.i}$ is too large or too small. We will show that both probabilities converge to 0, i.e. $\rho_{i,k}$ obtained lies within a reasonable range. The r_1 threshold is determined based on the fact that $\frac{k}{N} \approx f_T(t_i)c_{d_x+d_y+d_z}\rho_{k,i}^{d_x+d_y+d_z}$ and selecting $k = \log N$ is a reasonable choice. The r_2 threshold, on the other hand, implies that $\rho_{i,k}$ should lie in a range such that $\frac{k_s}{N} \approx f_S(s_i)c_{d_s}\rho_{k,i}^{d_s}$ for all subspaces $s \in \{(x,z),(y,z),z\}$.

The terms $I_3(t)$, $I_4(t)$ and $I_5(t)$ represent the estimation error probability for a reasonable $\rho_{k,i}$.

Considering each of the terms separately, We will prove that each term will go to zero as N goes to infinity.

Convergence of I_1

The term $I_1(t)$ can be upper-bounded in the same way as explained in [76] for $I_1(z)$. Then we will have,

$$I_1(t) \leq kN^{k-1}\exp\left\{-\frac{(\log N)^{d_x+d_y+d_z}}{4}\right\} \tag{B.97}$$

Convergence of I_2

For the convergence of $I_2(t)$, we are going to take the same steps as [76] for $I_2(z)$. First let $B_T(t,r) \equiv n : \|n-t\| < r$ be the $(d_x+d_y+d_z)$-dimensional ball centered at z with radius r. For $r_{2,1} \equiv (\log N)^2(Nf_{XZ}(x,z)c_{d_x+d_z})^{\frac{-1}{d_x+d_z}}$ and for sufficiently large N, the probability mass

within the $B_T(t, r_{2,1})$ is given by,

$$
\begin{aligned}
p_{2,1} &\equiv \mathbb{P}\left(u \in B_T(t, (\log N)^2 (N f_{XZ}(x,z) c_{d_x+d_z})^{\frac{-1}{d_x+d_z}}\right) && \text{(B.98)}\\
&\leq f_T(t) c_{d_x+d_y+d_z} \left((\log N)^2 (N f_{XZ}(x,z) c_{d_x+d_z})^{\frac{-1}{d_x+d_z}}\right)^{d_x+d_y+d_z} \\
&\quad \times \left(1 + C(\log N)^4 (N f_{XZ}(x,z) c_{d_x+d_z})^{\frac{-1}{d_x+d_z}})^2\right) && \text{(B.99)}\\
&\leq \frac{2 f_T(t) c_{d_x+d_y+d_z}}{(f_{XZ}(x,z) c_{d_x+d_z})^{\frac{d_x+d_y+d_z}{d_x+d_z}}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_x+d_z}} && \text{(B.100)}\\
&\leq 2 f_{Y|XZ}(y|x,z) \frac{c_{d_x+d_y+d_z}}{c_{d_x+d_z}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_x+d_z}} && \text{(B.101)}\\
&\leq 2C' \frac{c_{d_x+d_y+d_z}}{c_{d_x+d_z}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_x+d_z}} && \text{(B.102)}
\end{aligned}
$$

Where the last inequality comes from the assumption that $f_{Y|XZ}(y|x,z) < C'$. Similarly, for the second threshold $r_{2,2} = (\log N)^2 (N f^q(y,z) c_{d_y+d_z})^{\frac{-1}{d_y+d_z}}$,

$$
\begin{aligned}
p_{2,2} &\equiv \mathbb{P}\left(u \in B_T(t, (\log N)^2 (N f^q_{YZ}(y,z) c_{d_y+d_z})^{\frac{-1}{d_y+d_z}}\right) \\
&\leq f_T(t) c_{d_x+d_y+d_z} \left((\log N)^2 (N f^q_{YZ}(y,z) c_{d_y+d_z})^{\frac{-1}{d_y+d_z}}\right)^{d_x+d_y+d_z} \\
&\quad \times \left(1 + C(\log N)^4 (N f^q_{YZ}(y,z) c_{d_y+d_z})^{\frac{-1}{d_y+d_z}})^2\right) && \text{(B.103)}\\
&\leq \frac{2 f_T(t) c_{d_x+d_y+d_z}}{(f^q_{YZ}(y,z) c_{d_y+d_z})^{\frac{d_x+d_y+d_z}{d_y+d_z}}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_y+d_z}} && \text{(B.104)}\\
&\leq 2 \frac{f_T(t)}{f^q_{YZ}(y,z)} \frac{c_{d_x+d_y+d_z}}{c_{d_y+d_z}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_y+d_z}} && \text{(B.105)}\\
&\leq 2C_2 \frac{f^q_T(t)}{f^q_{YZ}(y,z)} \frac{c_{d_x+d_y+d_z}}{c_{d_x+d_z}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_y+d_z}} && \text{(B.106)}\\
&\leq 2C_2C' \frac{c_{d_x+d_y+d_z}}{c_{d_y+d_z}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_y+d_z}} && \text{(B.107)}
\end{aligned}
$$

The last two inequalities come from the bounds assumed on the distribution functions.

Similarly, for the second threshold $r_{2,2} = (\log N)^2 (N f_Z^q(z) c_{d_z})^{\frac{-1}{d_z}}$ we can write:

$$
\begin{aligned}
p_{2,3} &\equiv \mathbb{P}\left(u \in B_T(t, (\log N)^2 (N f_Z^q(z) c_{d_z})^{\frac{-1}{d_z}}\right) && \text{(B.108)}\\
&\leq f_T(t) c_{d_x+d_y+d_z} \left((\log N)^2 (N f_Z^q(z) c_{d_z})^{\frac{-1}{d_z}}\right)^{d_x+d_y+d_z} \\
&\quad \times \left(1 + C(\log N)^4 (N f_Z^q(z) c_{d_z})^{\frac{-1}{d_z}})^2\right) && \text{(B.109)}\\
&\leq \frac{2 f_T(t) c_{d_x+d_y+d_z}}{(f_Z^q(z) c_{d_z})^{\frac{d_x+d_y+d_z}{d_z}}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_z}} && \text{(B.110)}\\
&\leq 2 \frac{f_T(t)}{f_Z^q(z)} \frac{c_{d_x+d_y+d_z}}{c_{d_z}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_z}} && \text{(B.111)}\\
&\leq 2C_2 \frac{f_T^q(t)}{f_Z^q(z)} \frac{c_{d_x+d_y+d_z}}{c_{d_z}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_z}} && \text{(B.112)}\\
&\leq 2C_2 C' \frac{c_{d_x+d_y+d_z}}{c_{d_z}} (\log N)^{2(d_x+d_y+d_z)} N^{-\frac{d_x+d_y+d_z}{d_z}} && \text{(B.113)}
\end{aligned}
$$

The last two inequalities come from the bounds assumed on the distribution functions.

Similar to the procedure for $I_2(z)$ in the [76], $I_2(t)$ is the probability that at least k samples lie in $B_T(t, \max\{r_{2,1}, r_{2,2}, r_{2,3}\})$. Then we have,

$$
\begin{aligned}
I_2(t) &= \mathbb{P}\left(\rho_{k,i} < \max\{r_{2,1}, r_{2,2}, r_{2,3}\}\right) && \text{(B.114)}\\
&= \sum_{m=k}^{N-1} \begin{pmatrix} \text{N-1} \\ \text{m} \end{pmatrix} \max\{p_{2,1}, p_{2,2}, p_{2,3}\}^m \left(1 - \max\{p_{2,1}, p_{2,2}, p_{2,3}\}\right)^{N-1-m} && \text{(B.115)}\\
&\leq \sum_{m=k}^{N-1} N^m \max\{p_{2,1}, p_{2,2}, p_{2,3}\}^m && \text{(B.116)}\\
&\leq \sum_{m=k}^{N-1} \left(2CC' \frac{c_{d_x+d_y+d_z}}{\min\{c_{d_y+d_z}, c_{d_x+d_z}, c_{d_z}\}} (\log N)^{2(d_x+d_y+d_z)} \right. \\
&\qquad \left. \times N^{-\min\{\frac{d_x+d_y}{d_z}, \frac{d_z}{d_x+d_y}, \frac{d_y}{d_x+d_z}\}} \right)^m && \text{(B.117)}
\end{aligned}
$$

Similarly, for N sufficiently large and applying the sum of geometric series, we have,

$$I_2(t) \leq \left(4CC' \frac{c_{d_x+d_y+d_z}}{\min\{c_{d_y+d_z}, c_{d_x+d_z}, c_{d_z}\}}\right)^k (\log N)^{2k(d_x+d_y+d_z)} N^{-k\min\{\frac{d_x+d_y}{d_z}, \frac{d_z}{d_x+d_y}, \frac{d_y}{d_x+d_z}\}} \tag{B.118}$$

Convergence of I_3

Given $T_i = t = (x, y, z)$, and $\rho_{k,i} = r$ and $\hat{f}_{XZ}(x_i, z_i) = \frac{n_{xz,i}}{(N-1)c_{d_x+d_z}\rho_{k,i}^{d_x+d_z}}$, we have,

$$\begin{aligned}
& \mathbb{P}\left(|\log f_{XZ}(X_i, Z_i) - \log \hat{f}_{XZ}(X_i, Z_i)| > \epsilon/6 | \rho_{k,i} = r, T_i = t\right) && \text{(B.119)} \\
= \; & \mathbb{P}\left(n_{xz,i} > (N-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x, z) e^{\epsilon/6} | \rho_{k,i} = r, T_i = t\right) \\
& + \mathbb{P}\left(n_{xz,i} < (N-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x, z) e^{\epsilon/6} | \rho_{k,i} = r, T_i = t\right) && \text{(B.120)}
\end{aligned}$$

Given $T_i = t$, Lemma 27 gives the probability distribution of the $n_{xz,i}$.

Lemma 27. *Given $T_i = t = (x, y, z)$, and $\rho_{k,i} = r < r_N$ for some deterministic sequance of r_N such that $\lim_{N \longleftarrow \infty} r_N = 0$ and for any $\epsilon > 0$, the number of neighbors $n_{xz,i} - k$ is distributed as $\sum_{l=k+1}^{N-1} U_l$, where U_l are i.i.d Bernoulli random variables with mean $f_{XZ}(x, z) c_{d_x+d_z} r^{d_x+d_z}(1 - \epsilon/8) \leq E[U_l] \leq f_{XZ}(x, z) c_{d_x+d_z} r^{d_x+d_z}(1 - \epsilon/8)$ for sufficiently large N.*

Proof. See the proof of Lemma 5 in [76]. □

Based on Lemma 27,

$$\begin{aligned}
& \mathbb{P}\left(n_{xz,i} > (N-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x, z) e^{\epsilon/6} | \rho_{k,i} = r, T_i = t\right) && \text{(B.121)} \\
= \; & \mathbb{P}\left(\sum_{l=k+1}^{N-1} U_l > (N-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x, z) e^{\epsilon/6} - k\right) && \text{(B.122)} \\
= \; & \mathbb{P}\Bigg(\sum_{l=k+1}^{N-1} U_l - (N-1-k)E[U_l] \\
& \qquad > (N-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x, z) e^{\epsilon/6} - k - (N-1-k)E[U_l]\Bigg) && \text{(B.123)}
\end{aligned}$$

The right-hand side term inside the probability can be lower bounded as,

$$
\begin{aligned}
& (N-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x,z) e^{\epsilon/6} - k - (N-1-k)E[U_l] && \text{(B.124)}\\
\geq\ & (N-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x,z) e^{\epsilon/6} - k \\
& -(N-1-k) f_{XZ}(x,z) c_{d_x+d_z} r^{d_x+d_z} (1-\epsilon/8) && \text{(B.125)}\\
\geq\ & (N-k-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x,z) \left(e^{\epsilon/6} - 1 - \epsilon/8\right) - k && \text{(B.126)}\\
\geq\ & (N-k-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x,z) \frac{\epsilon}{48} && \text{(B.127)}
\end{aligned}
$$

for sufficiently large N.

Applying Bernstein's inequality, (B.123) can be upper bounded by $\exp\{-\frac{\epsilon^2}{2304(1+19\epsilon/144)}(N-k-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x,z)\}$. The tail distribution can also be upper-bounded by the same term. Thus,

$$
\begin{aligned}
& \mathbb{P}(|\log f_{XZ}(x_i,z_i) - \log \hat{f}_{XZ}(x_i,z_i)| > \epsilon/6 | \rho_{k,i} = r, T_i = t) \\
\leq\ & 2\exp\{-\frac{\epsilon^2}{2304(1+19\epsilon/144)}(N-k-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x,z)\} && \text{(B.128)}
\end{aligned}
$$

Therefore, the term $I_3(t)$ can be upper-bounded as,

$$
\begin{aligned}
I_3(t) &= \int_{r=r_2}^{r_1} \mathbb{P}(|\log f_{XZ}(x_i,z_i) - \log \hat{f}_{XZ}(x_i,z_i)| > \epsilon/6 | \rho_{k,i} = r, T_i = t) f_\rho(r) dr && \text{(B.129)}\\
&\leq \int_{r=(\log N)^2 (N f_{XZ}(x,z) c_{d_x+d_z})^{\frac{-1}{d_x+d_z}}}^{\log N (N f_T(t) c_{d_x+d_y+d_z})^{\frac{-1}{d_x+d_y+d_z}}} \\
&\quad \mathbb{P}(|\log f_{XZ}(x_i,z_i) - \log \hat{f}_{XZ}(x_i,z_i)| > \epsilon/6 | \rho_{k,i} = r, T_i = t) f_\rho(r) dr && \text{(B.130)}\\
&\leq \int_{r=(\log N)^2 (N f_{XZ}(x,z) c_{d_x+d_z})^{\frac{-1}{d_x+d_z}}}^{\log N (N f_T(t) c_{d_x+d_y+d_z})^{\frac{-1}{d_x+d_y+d_z}}} \\
&\quad 2\exp\left\{-\frac{\epsilon^2}{2304(1+19\epsilon/144)}(N-k-1)c_{d_x+d_z} r^{d_x+d_z} f_{XZ}(x,z)\right\} f_\rho(r) dr && \text{(B.131)}\\
&\leq 2\exp\left\{-\frac{\epsilon^2}{4608} N c_{d_x+d_z} f_{XZ}(x,z) \left((\log N)^2 (N f_{XZ}(x,z) c_{d_x+d_z})^{\frac{-1}{d_x+d_z}}\right)^{d_x+d_z}\right\} && \text{(B.132)}\\
&\leq 2\exp\left\{-\frac{\epsilon^2}{4608} (\log N)^{2(d_x+d_z)}\right\} && \text{(B.133)}
\end{aligned}
$$

For sufficiently large N.

Convergence of I_4

Given $T_i = t = (x, y, z)$, and $\rho_{k,i} = r$ and $\hat{f}^q_{YZ}(y_i, z_i) = \frac{n'_{yz,i}}{(N-1)c_{d_y+d_z}\rho_{k,i}^{d_y+d_z}}$, we have,

$$\mathbb{P}\left(|\log f^q_{YZ}(Y_i, Z_i) - \log \hat{f}^q_{YZ}(Y_i, Z_i)| > \epsilon/6 | \rho_{k,i} = r, T_i = t\right) \tag{B.134}$$

$$= \mathbb{P}\left(n_{yz,i} > (N-1)c_{d_y+d_z} r^{d_y+d_z} f^q_{YZ}(Y_i, Z_i) e^{\epsilon/6} | \rho_{k,i} = r, T_i = t\right) \tag{B.135}$$

$$+\mathbb{P}\left(n_{yz,i} < (N-1)c_{d_y+d_z} r^{d_y+d_z} f^q_{YZ}(Y_i, Z_i) e^{\epsilon/6} | \rho_{k,i} = r, T_i = t\right). \tag{B.136}$$

We can write $n'_{yz,i} = n^{(1)}_{yz,i} + n^{(2)}_{yz,i}$, where,

$$n^{(1)}_{yz,i} = \sum_{j:\|T_j - t\| < \rho_{i,k}} \frac{q_{XZ}(x_j, z_j)}{f_{XZ}(x_j, z_j)} \tag{B.137}$$

$$n^{(2)}_{yz,i} = \sum_{j:\|T_j - t\| > \rho_{i,k}} \frac{q_{XZ}(x_j, z_j)}{f_{XZ}(x_j, z_j)} I\{\|(Y_j - Y_i, Z_j - Z_i)\| < \rho_{k,i}\} \tag{B.138}$$

Given $T_i = t$, Lemma 28 gives the probability distribution of the $n'_{yz,i}$.

Lemma 28. *Given $T_i = t = (x, y, z)$, and $\rho_{k,i} = r < r_N$ for some deterministic sequence of r_N such that $\lim_{N \longleftarrow \infty} r_N = 0$ and for any $\epsilon > 0$, the distribution of $n^{(2)}_{yz,i}$ is $\sum_{l=k+1}^{N-1} V_l$, where V_l are i.i.d random variables with $V_l \in [0, 1/C_1]$ and mean $f_{XZ}(x, z)c_{d_x+d_z} r^{d_x+d_z}(1 - \epsilon/8) \leq E[U_l] \leq f_{XZ}(x, z)c_{d_x+d_z} r^{d_x+d_z}(1 - \epsilon/8)$ for sufficiently large N.*

Proof. See the proof of Lemma 6 in [76]. □

According to the Lemma 28 and following the same procedure as $I_3(t)$, the term $I_4(t)$ will also be bounded here in th same way. We have:

$$I_4(t) \leq 2\exp\{-\frac{C_1\epsilon^2}{4608}(\log N)^{2(d_y+d_z)}\} \tag{B.139}$$

Convergence of I_5

Similar to the case of $I_4(t)$, given $T_i = t = (x, y, z)$, and $\rho_{k,i} = r$ and $\hat{f}_Z^q(z_i) = \frac{n'_{z,i}}{(N-1)c_{d_z}\rho_{k,i}^{d_z}}$, we have,

$$\mathbb{P}\left(|\log f_Z^q(Z_i) - \log \hat{f}_Z^q(Z_i)| > \epsilon/6 | \rho_{k,i} = r, T_i = t\right) \tag{B.140}$$

$$= \mathbb{P}\left(n_{z,i} > (N-1)c_{d_z} r^{d_z} f_Z^q(Z_i) e^{\epsilon/6} | \rho_{k,i} = r, T_i = t\right) \tag{B.141}$$

$$+\mathbb{P}\left(n_{z,i} < (N-1)c_{d_z} r^{d_z} f_Z^q(Z_i) e^{\epsilon/6} | \rho_{k,i} = r, T_i = t\right) \tag{B.142}$$

We can write $n'_{z,i} = n_{z,i}^{(1)} + n_{z,i}^{(2)}$, where,

$$n_{z,i}^{(1)} = \sum_{j:\|T_j - t\| < \rho_{i,k}} \frac{q_{XZ}(x_j, z_j)}{f_{XZ}(x_j, z_j)} \tag{B.143}$$

$$n_{z,i}^{(2)} = \sum_{j:\|T_j - t\| > \rho_{i,k}} \frac{q_{XZ}(x_j, z_j)}{f_{XZ}(x_j, z_j)} I\{\|Z_j - Z_i\| < \rho_{k,i}\} \tag{B.144}$$

Following the same procedure as for $I_4(t)$, we will obtain the upper bound below for $I_5(t)$:

$$I_5(t) \leq 2\exp\{-\frac{C_1\epsilon^2}{4608}(\log N)^{2(d_z)}\} \tag{B.145}$$

Now, we can write,

$$\mathbb{P}\left(\sum_{i=1}^{N} \frac{\omega'_i}{N}|a_i - \hat{a}_i| > \epsilon\right) \tag{B.146}$$

$$\leq N\int \left(I_1(t) + I_2(t) + I_3(t) + I_4(t) + I_5(t)\right) f_T(t)dt \tag{B.147}$$

$$\leq kN^k \exp\{-\frac{(\log N)^{d_x+d_y+d_z}}{4}\}$$

$$+ \left(4CC' \frac{c_{d_x+d_y+d_z}}{\min\{c_{d_y+d_z}, c_{d_x+d_z}, c_{d_z}\}}\right)^k (\log N)^{2k(d_x+d_y+d_z)} N^{1-k\min\{\frac{d_x+d_y}{d_z}, \frac{d_z}{d_x+d_y}, \frac{d_y}{d_x+d_z}\}}$$

$$+ 2N\exp\{-\frac{\epsilon^2}{4608}(\log N)^{2(d_x+d_z)}\} + 4N\exp\{-\frac{C_1\epsilon^2}{4608}(\log N)^{2(d_z)}\} \tag{B.148}$$

If k is chosen large enough so that $1 - k \min\{\frac{d_x+d_y}{d_z}, \frac{d_z}{d_x+d_y}, \frac{d_y}{d_x+d_z}\} < 0$, then all the terms will converge to 0 as N goes to infinity, and the proof of Lemma 26 is complete.

Appendix C

SUPPLEMENTARY MATERIAL FOR CHAPTERS 5 AND 6

C.1 Uncoupled Time-Course Data Samples and Divergence Measures

In the Dynode Vectorfield package, we have implemented two divergence measures as *loss functions* for the population (uncoupled) time-course data: a) Wasserstein Distance, and b) Kullback-Leibler Divergence. In this section, we discuss the details of how the two measures are implemented as loss functions.

C.1.1 Wasserstein Distance

Wasserstein Distance is defined as:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,\hat{x})\sim\gamma} \|x - \hat{x}\| \tag{C.1}$$

where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ represents the set of all possible joint distributions over X and $\hat{X}$ where the marginals are equal to $\mathbb{P}_r$ and $\mathbb{P}_g$ respectively. As we see, it can be challenging, if possible at all, to implement such a metric directly as a loss function. Hence we need to seek alternative indirect ways to develop estimators for these metrics to be used as losses in our Dynode package.

The idea here would be to use the Kantarovich-Rubenstein dual form [126] to compute the distance:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|h\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r} [h(x)] - \mathbb{E}_{\hat{x}\sim\mathbb{P}_g} [h(\hat{x})] \tag{C.2}$$

This duality converts the search over the joint distribution space Π to finding a Lipschitz smooth function $h(.)$ that would satisfy the maximization problem. One could see that $h(.)$ in fact may be represented by a neural net $h_\theta(.)$ and then the supremum and the corresponding h_θ are found via training. To enforce the Lipschitz continuity over the neural net, we apply spectral normalization per each linear layer [87].

It should be noted that since the loss function itself is now modeled by a neural net which would require its own training (We typically train h_θ for 20 iterations per each iteration of the original dynamics network f_θ), it will be more computationally intensive than the regular losses such as Mean-Squared-Error or Mean-Absolute-Deviation (i.e. L1-loss).

C.1.2 Kullback-Leibler Divergence

The Kullback-Leibler Divergence is defined as:

$$KL(\mathbb{P}_r, \mathbb{P}_g) = \int_x \mathbb{P}_r(x) \log \frac{\mathbb{P}_r(x)}{\mathbb{P}_g(x)} dx \tag{C.3}$$

Assuming $\mathbb{P}_r$ and $\mathbb{P}_g$ satisfy some reasonable assumptions for the integral to exist. Same as Wasserstein Distance, it's not straightforward to calculate the measure directly. The KNN-based and other estimators for KL-divergence wouldn't be differentiable either, which is required for the neural net to be trainable.

So, similar to the Wasserstein Distance, we use a variational representation of the KL-divergence introduced by Donsker and Varadhan [53] defined as follows:

$$KL(\mathbb{P}_r, \mathbb{P}_g) = \sup_h \mathbb{E}_{x\sim\mathbb{P}_r}\left[h(x)\right] - \log \mathbb{E}_{\hat{x}\sim\mathbb{P}_g}\left[e^{h(\hat{x})}\right] \tag{C.4}$$

Quite similar to Wasserstein Distance, the scalar function $h(.)$ could be represented by a neural net as $h_\theta(.)$ which is trained to find the supremum. Hence, similar to the Wasserstein

Distance, it's of relatively high computational complexity compared to that of the conventional losses.

C.2 Technical Noise, Communication Channels, and Loss Function Design

In all of the experimental cases, the collected cell expression data is subject to a lot of technical noise such as dropouts and sequencing errors, modeling and handling of which is a serious challenge in many research studies [304]. The data provided for training Dynode Vectorfield is no exception: While the underlying cell differentiation dynamics describe the true expressions and velocity values (which we denote by X and U respectively), the data provided for learning the aforementioned dynamics is a noisy observation them (which we denote here by Y and V respectively). In other words, the imperfect observations are described by functions $C_1(.)$ and $C_2(.)$ such that $Y = C_1(X, \zeta_1)$ and $V = C_2(U, \zeta_2)$ where ζ_i is a random variable denoting the random observation noise. Such functions are called "communication channels" in information theory. The communication channels are fully described by their associated *conditional probability distributions* $P_{Y|X}$ and $P_{V|U}$ defining the statistical dependence of the channel outputs Y and V over the channel inputs X and U respectively.

The Dynode platform is able to incorporate these noisy channel models in the training if configured accordingly, i.e. it not only models the dynamics function as a trainable neural net $f_\theta(.)$ but also can directly model the true underlying cell expressions x and RNA-velocity values u as unknown trainable variables, which will be estimated via Maximum Likelihood for the observations y and v assuming that the technical noise model (or equivalently the "communication channel") is known.

Dynode tries to learn the dynamics f_θ for the underlying true values x and v, such that $u = f_\theta(x)$ as well as $x(t) = \int_0^t f_\theta\left(\tau, x(\tau)\right) d\tau$ given $x(0)$ in the time-course evolution tracking setting. To relate x,u values to the provided observational data y and v (either expression-velocity pairs or the time-course of expression), let's assume the channel model $P_{Y|X}$ and $P_{V|U}$. For the simplicity of notations, let's assume both distributions to be identical, denoted by P. Thus, to find all the unknowns in the system, we can solve a log-likelihood

maximization problem:

- For expression-velocity regression mode: $\max_{\theta,x} = \log P(y|x) + \log P(v|f_\theta(x))$
- For Neural ODE mode: $\max_{\theta,x(0)} = \log P(y|x) + \sum_i \log P(y(t_i)|\int_0^{t_i} f_\theta(\tau, x(\tau))\, d\tau)$

Therefore, solving the above problems is equivalent to a training procedure with an appropriate loss function derived from the given P distribution. Thus it's needed to design the loss function correctly to represent the corresponding channel model. We will discuss a few of the special channels briefly and will introduce their corresponding loss function.

C.2.1 Additive White Gaussian Noise (AWGN) Channel

AWGN channel is one of the simplest channel models, in which the input X is added by a Gaussian random variable $Z \sim N(0, \sigma^2)$, i.e the output is $Y = X + Z$. The conditional distribution $P(y|x)$ [1] is therefore can be written as:

$$P(y|x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y - x\|^2}{2\sigma^2}\right) \tag{C.5}$$

Note that here x and y represent expression values y and x as well as velocity v and u. Suppose $y^{(i)}$ are the samples observed at the channel's output Y, with their corresponding input unobserved samples $x_\Theta^{(i)}$, where Θ denotes the set of underlying generative parameters for X samples. In other words, Θ are the learnable parameters that fully determine the underlying $x^{(i)}$ and $u^{(i)}$ samples. Therefore, we have $\Theta = \left\{x^{(i)}, \theta\right\}$ for the expression-velocity regression setting, and $\Theta = \left\{x^{(i)}(0), \theta\right\}$ in the case of time-course data.

[1] Note that P represents both $P_{Y|X}$ and $P_{V|U}$ distributions.

The Maximum Likelihood estimate of Θ will then be determined as:

$$\begin{aligned}\hat{\Theta} &= \text{argmax}_{\Theta} \sum_{i=0}^{N} \log P\left(y^{(i)} \middle| x_{\Theta}^{(i)}\right) \\ &= \text{argmax}_{\Theta} \sum_{i=0}^{N} \left(\log \frac{1}{\sqrt{2\pi}\sigma} + \log \exp\left(-\frac{\|y^{(i)} - x_{\Theta}^{(i)}\|^2}{2\sigma^2}\right)\right) \\ &= \text{argmin}_{\Theta} \sum_{i=0}^{N} \|y^{(i)} - x_{\Theta}^{(i)}\|^2 \end{aligned} \tag{C.6}$$

As we can see, the AWGN channel model will simply correspond to the Mean-Squared-Error (MSE) loss. Therefore, if the observational noise is Gaussian, we simply train our network with MSE.

C.2.2 Laplace Channel

In a Laplace channel, Y is related to X via a Laplace distribution:

$$P(y|x) = \frac{1}{2b} \exp\left(-\frac{|y - x|}{b}\right) \tag{C.7}$$

where the parameter b is *the scale of the distribution.* Similar to the Gaussian Channel, we can see that:

$$\hat{\Theta} = \text{argmin}_{\Theta} \sum_{i=0}^{N} |y^{(i)} - x_{\Theta}^{(i)}| \tag{C.8}$$

Thus the channel corresponds to using the Mean-Absolute-Difference (MAD) loss in our training procedure.

C.2.3 Binomial Channel

The binomial channel describes the number of successful trails Y out of the total trails X if the probability of each single success is p. The input X and the output Y are then two

integer numbers that are related via:

$$P(y|x) = \binom{x}{y} p^y (1-p)^{x-y} \tag{C.9}$$

The binomial channel can be thought of as a *counting noise*: Suppose there are X total objects and the goal is to count them to find out what the number X is. But the counting is subject to errors, chances are you miss some of the X objects while counting and thus end up with a number Y which is less than or equal to X. If each of the objects is independently being counted with a probability of p (or therefore missed with a probability of $1-p$), the resulting total count Y would be a binomial random variable with parameter p. This is the most dominant error factor present in single-cell RNA sequencing experiments, which will be described in greater detail in Subsection C.2.5.

The Maximum Likelihood estimate of Θ will be given as:

$$\begin{aligned}
\hat{\Theta} &= \text{argmax}_\Theta \sum_{i=0}^{N} \log P\left(y^{(i)} \middle| x_\Theta^{(i)}\right) \\
&= \text{argmax}_\Theta \sum_{i=0}^{N} \left(\log \frac{x_\Theta^{(i)}!}{y^{(i)}!(x_\Theta^{(i)} - y^{(i)})!} + y^{(i)} \log p + (x_\Theta^{(i)} - y^{(i)}) \log(1-p) \right) \\
&= \text{argmax}_\Theta \sum_{i=0}^{N} \left(\log\left(x_\Theta^{(i)}!\right) - \log\left((x_\Theta^{(i)} - y^{(i)})!\right) + (x_\Theta^{(i)} - y^{(i)}) \log(1-p) \right) \\
&\approx \text{argmax}_\Theta \sum_{i=0}^{N} \left(x_\Theta^{(i)} \log(x_\Theta^{(i)}) - (x_\Theta^{(i)} - y^{(i)}) \log(x_\Theta^{(i)} - y^{(i)}) + (x_\Theta^{(i)} - y^{(i)}) \log(1-p) \right) \\
&= \text{argmin}_\Theta \sum_{i=0}^{N} \left(-x_\Theta^{(i)} \log(x_\Theta^{(i)}) + (x_\Theta^{(i)} - y^{(i)}) \log(x_\Theta^{(i)} - y^{(i)}) - (x_\Theta^{(i)} - y^{(i)}) \log(1-p) \right)
\end{aligned} \tag{C.10}$$

in which we used *Stirling's Approximation* for $\log(n!)$. Thus, if the noise model follows that of a binomial channel, one should use the loss obtained above as the training loss for the neural network.

C.2.4 Poisson Channel

In a Poisson channel, The input X is the *rate* to a Poisson distribution which in turn would output Y. The conditional distribution would be:

$$P(y|x) = \frac{e^{-x}x^y}{y!}. \tag{C.11}$$

Similar to the previous distributions discussed, the Maximum Likelihood estimate of Θ will be given as:

$$\begin{aligned}
\hat{\Theta} &= \text{argmax}_\Theta \sum_{i=0}^{N} \log P\left(y^{(i)} \middle| x_\Theta^{(i)}\right) \\
&= \text{argmax}_\Theta \sum_{i=0}^{N} \left(\log(y^{(i)}!) + y^{(i)} \log(x_\Theta^{(i)}) + \log\exp(-x_\Theta^{(i)})\right) \\
&= \text{argmax}_\Theta \sum_{i=0}^{N} \left(y^{(i)} \log(x_\Theta^{(i)}) - x_\Theta^{(i)}\right) \\
&= \text{argmin}_\Theta \sum_{i=0}^{N} \left(-y^{(i)} \log(x_\Theta^{(i)}) + x_\Theta^{(i)}\right)
\end{aligned} \tag{C.12}$$

C.2.5 Modeling the Technical Noise in Real Single-cell RNA Sequencing Data

In the previous subsections, we explained how we design various training loss functions to incorporate various technical noise models and talked about a few example distributions. Now one might wonder what the most appropriate noise profile (and loss function accordingly) is for the actual sc-RNA-seq data. In [304], Zhang et al presented a simulator that explicitly models the processes that give rise to the data observed in single-cell RNA-Seq experiments. The final step of this 3-step simulation process includes emulating the technical noise introduced by *mRNA capture*, *reverse transcription*, *PCR amplification*, *RNA fragmentation*, and *sequencing* in a realistic manner by explicitly simulating each major step.

Here we analyze the procedure mentioned above to obtain a proper channel model. We will focus on UMI-added experiments, such as *10x Chromium*. Consider a cell with d genes

and the corresponding true expression vector $x = (x_1, x_2, \ldots, x_d)$ where gene i's expression x_i naturally represents the number of the expressed mRNA molecules for the gene and hence is a non-negative integer. Let $x_{\text{tot}} = \sum_{i=1}^{d} x_i$ be the total number of RNA molecules in the cell.

1. Each mRNA molecule j first needs to be captured so it can be read. It's reasonable to assume each molecule can be captured independently with a probability $\hat{\alpha}$ or otherwise missed. Although $\hat{\alpha}$ is assumed to be identical for all the molecules of a single cell, it is taken from a normal distribution and hence varies across different cells. In practice, the variance of $\hat{\alpha}$'s distribution is usually very small, and $\hat{\alpha}$ values lie around 0.1 for all the cells.

2. Each captured molecule is then amplified and fragmented, resulting in k_j fragments for each molecule j. Thus the total number of fragments would be $k_{\text{tot}} = \sum_{j=1}^{x_{\text{tot}}} k_j$. Note that although k_j values are not the same for all the molecules due to the amplification and fragmentation biases, the values are close to each other in practice.

3. In the next step, a total number of N reads will be captured out of the k_{tot} fragments. The probability of capturing at least one fragment from molecule j is obtained by:

$$\epsilon_j = 1 - \frac{\binom{k_{\text{tot}} - k_j}{N}}{\binom{k_{\text{tot}}}{N}} \tag{C.13}$$

   We note that with appropriately large N, we can assure $\epsilon_j \approx 1$.

4. In the final step, in UMI experiments, the molecules for which at least one fragment is captured and sequenced are counted, and then the molecules from a single gene are summed up to make the observed expression vector $y = (y_1, y_2, \ldots, y_d)$.

As a result, the end-to-end capturing of each molecule is approximately a Bernoulli random variable with the probability $p = \hat{\alpha}\epsilon \approx \hat{\alpha}$. Therefore each y_i value can be interpreted as a Binomial random variable with parameters $(n = x_i, p = \hat{\alpha})$, and hence the binomial channel with parameter p is suitable for it.

C.3 Regularizers for Smoothness and Stability

Two optional regularizers are implemented in Dynode which can be used in velocity training mode. The two regularizers enforce the smoothness and stability of the learned dynamics via *Lipschitz Smoothness* and *Lyapunov Stability* notions respectively.

C.3.1 Lipschitz Smoothness

A function f_θ is called *Lipschitz Continuous* (or Lipschitz Smooth) if there exists a positive constant L called *Lipschitz Constant* such that $\|f_\theta(x) - f_\theta(y)\| \leq L\|x - y\|$ for any x and y sufficiently close to each other in f_θ's domain. [194]

In Dynode, the way the smoothness is optionally enforced is for each input sample $x^{(i)}$, a random sample $\tilde{x}^{(i)}$ is generated close to x by adding some controlled noise $n^{(i)}$ such that $\tilde{x}^{(i)} = x^{(i)} + n^{(i)}$. Then if $\|f_\theta(x^{(i)}) - f_\theta(\tilde{x}^{(i)})\| \geq L\|\tilde{x}^{(i)} - x^{(i)}\| = L\|n^{(i)}\|$, a penalty term equal to $\|f_\theta(x^{(i)}) - f_\theta(\tilde{x}^{(i)})\|$ is added to the training loss. The Lipschitz constant L can be configured by the user and its default value is 1.

C.3.2 Lyapunov Stability

The stability enforcement used in Dynode intuitively tries to keep the $x^{(i)}$ values bounded by enforcing the output velocity to be in the opposite direction. In other words, for any input sample $x^{(i)}$ to the neural net $f_\theta(.)$ and its corresponding output velocity $v^{(i)} = f_\theta(x^{(i)})$ we add a regularizer term $< x^{(i)}, v^{(i)} >$ which penalizes any velocity value with an amplifying effect over the x value and enforces a negative inner product for each pair. We won't dive deep into the mathematical details here, only briefly mentioning that this notion of stability is equivalent to considering a *Lyapunov function* equal to $V(x,t) = (x(t))^2$. You can find more details in [151].

C.4 C. Elegans Spatiotemporal and Lineage Data Preparation

In our C. Elegans system embryogenesis learning test in Section 6.2.2, the training dataset needs to represent at least one full C. Elegans individual with its complete gene expression [2] and spatial coordinates for every single cell at every single time point, alongside the properly constructed temporal and spatial graph for each time point. Here, we go through the details of how we pre-process the data from [162] and create the embryogenesis training and testing dataset.

C.4.1 Standard embryo Individual and its lineage tree

We started by picking a single C. Elegans individual representing a typical embryogenesis procedure. The individual animal is observed and tracked for a period of 350 minutes. At each time point (i.e. every minute), all the cells are completely tagged and labeled according to the C. Elegans cell atlas, and each cell's spatial coordinates are recorded at each time point. Finally, the lineage tree of C. Elegans was aligned with the so-called standard embryo's growth timeline according to the cell divisions observed.

One of the drawbacks of the current in-situ gene expression profiling technology is that it's impossible to track all the gene's expressions over time for any individual, hence this typical individual won't include any native gene expression data. The data will be provided from tracking a set of other C. Elegans animals, as we will see in a later subsection.

C.4.2 Spatial coordinates and Spatial graph construction

As mentioned, one piece of information provided from the typical sample mentioned above is the spatial (x, y, z) coordinates of each cell at any time point t. In addition to the cell migration prediction task where we try to predict each cell's spatial movements in the embryo, the coordinates recorded are the base upon which we construct the spatial graph at each time

[2] The expression in this dataset actually comes from profiling the proteins, which is directly proportional to gene expression levels.

point t, i.e. we build the graph based on the spatial proximity of the cells at any time point t. We do it by constructing the k nearest neighbors graph based on the recorded coordinates, with the default number of neighbors set to $k = 5$.

C.4.3 Lineage tree and temporal graph

The complete cell lineage and cell profiling of C. Elegans' embryogenesis is studied and is available to us and as mentioned above, it's aligned to the standard embryo above. This lineage tree which specifies cell division/growth/death events for every cell at every time point, in addition to providing the training and test data for the fate prediction task, makes the basis upon which we construct the temporal graph for each time point.

The process of building the temporal graph is as follows: At every time point t, for each cell $C_i(t)$, if the cell keeps growing (i.e. according to the standard embryo, it also exists at time $t+1$ as $C_i(t+1)$) we simply connect $C_i(t)$ to $C_i(t+1)$ with a directed edge representing the temporal message passing from its past state to its current state. On the other hand, if the cell $C_i(t)$ divides into two daughter cells $C_{i1}(t+1)$ and $C_{i2}(t+1)$ according to the standard embryo, we connect $C_i(t)$ to $C_{i1}(t+1)$ and $C_{i2}(t+1)$ to account for the message passing from the parent cell to its daughter cells. Therefore, the constructed temporal graph would be a directed bipartite graph with each target node having exactly one incoming edge, and each source node having one or two outgoing edges.

C.4.4 Gene expression alignment, re-sampling, and smoothing

As mentioned about the standard C. Elegans embryo, it's impossible to track an entire animal (with its entire cell expression for its growth period). However, the current live-imaging technology provides us with tracking a single gene's expression levels across all the cells and across all time points, in a single embryo [162]. Therefore, we may collect multiple gene expression datasets, each from a single embryo, and then align each of them to the standard embryo, to assemble a complete individual embryo (i.e. with gene expression, location, and fate available for every cell at every time point) for training and testing. Note

that since the gene expressions are collected from multiple animals, hence they don't truly represent the coupled interactions of all the genes, however, they still provide us with partial information on the overall trend of each gene through the embryogenesis process and the rough spatiotemporal dynamics.

We are provided with 291 C. Elegans genes' expression data collected by tracking 291 embryos. Since each embryo follows a slightly different growth path from the standard embryo (meaning that cell divisions could happen at slightly different time points, hence each cell being present at different time points and having different life spans), the gene expression for each cell over time needs to be aligned to that of the base individual. For example, consider the cell C to be born at $t = 10$ and divide at time $t = 20$ in the standard embryo. We observe that the same cell was born at $t = 13$ and divided at $t = 25$ in the embryo in which the gene *LIN13* was recorded. Therefore, we need to realign the gene expression *LIN13* recorded for this cell to map to the span between $t = 10$ and $t = 20$. To do so, we used the cubic-spline interpolation for each (cell, gene) pair, and then re-sampled each gene for each cell to match the life span of that cell in the standard. We then used the Savitzky-Golay smoothing filter [240] to smoothen each gene expression across the entire tracking span of the standard embryo.