Deep Learning for Causal Inference and Latent Dynamical Modeling in Biomedical Research

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Kai Andre Lagemann

aus
Osnabrück

Bonn 2024

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter/Betreuer:	Prof. Dr. Sach Mukherjee
Gutachter:	Prof. Dr. Reinhard Klein

Tag der Promotion: 30. April 2024
Erscheinungsjahr: 2024

Danksagung

Diese Arbeit entstand während meiner Zeit am Deutschen Zentrum für Neurodegenerative Erkrankungen und ist den unermüdlichen Quellen meiner Inspiration und Unterstützung auf meinem Weg gewidmet.

Ich bedanke mich bei meinem außergewöhnlichen Betreuer, Prof. Dr. Sach Mukherjee. Ihre Anleitung und Fachkenntnisse haben mich stets ermutigt neue Wege zu erkunden und sowohl mein akademisches Wachstum als auch meine persönliche Entwicklung gefördert. Ihr Glaube an mich war eine treibende Kraft und ich bin zutiefst dankbar für Ihr Mentorship.

Ebenso bedanke ich mich bei Herrn Prof. Dr. Reinhard Klein für die Übernahme des Koreferats. Herrn Prof. Dr. Christian Bauckhage danke ich für die Übernahme des Vorsitzes der Prüfungskommission und Herrn Prof. Dr. Alexander Radbruch danke ich für die Teilnahme als fachfremdes Mitglied an meiner Prüfungskommission.

Ich möchte allen Kollegen für ihre Hilfsbereitschaft und für die gute Arbeitsatmosphäre an unserem Institut danken. Eure Kameradschaft und unsere gemeinsame Suche nach Wissen haben dieses Unterfangen nicht nur erfüllend, sondern auch angenehm gemacht. Unsere zahllosen Diskussionen und gemeinsamen Anstrengungen haben mein Verständnis bereichert.

Ganz besonders möchte ich mich bei meinen Eltern, Claudia und Raimond, bedanken. Eure anhaltende Liebe, Opfer und unbeirrbare Ermutigung waren der Eckpfeiler meiner Leistungen. Euer unerschütterlicher Glaube an mein Potenzial hat meine Wünsche geprägt und mich vorangetrieben. Alles, was ich erreiche, ist ein Spiegelbild eurer grenzenlosen Unterstützung.

Ich bin unendlich dankbar für meinen wundervollen Zwillingsbruder Christian. Deine ständige Begleitung hat mir in herausfordernden Zeiten Stärke verliehen. Dein Glaube an meine Fähigkeiten war eine motivierende Kraft, die mich über meine Grenzen hinausgebracht hat. Egal wie schwierig die Herausforderungen auf unserem Weg sind, zusammen werden sie immer meistern. Du bist ein ganz besonderer Teil meines Lebens!

Zum Schluss möchte ich mich bei meiner geliebten Freundin Cornelia bedanken. Du bist ein Leuchtfeuer in meinem Leben, das Freude, Trost und Verständnis gebracht hat. Deine Geduld und Ermutigung haben mich durch die Höhen und Tiefen dieser Reise getragen. Du erinnerst mich ständig daran, was wirklich zählt. Ich danke dir von Herzen!

Danke, dass ihr meine Stützen der Stärke und Inspiration seid.

Abstract

Biological systems are ubiquitous, encompassing complex molecular networks governing single-cell organisms to expansive ecosystems profoundly impacting our planet's environment. In biology, the adoption of a systems approach seeks to achieve a comprehensive, quantitative understanding of living organisms comparable in some ways to the kind of understanding we have of systems in engineering and physics. In this context, a major challenge in scientific AI is causal learning. To address emerging biomedical questions, this work proposes a deep neural architecture that learns causal relationships between variables by combining high-dimensional data with prior causal knowledge. In particular a combination of convolutional and graph neural networks is utilized within a causal risk framework, specifically designed to handle the high dimensionality and typical sources of noise frequently occurring in large-scale biological data. In experimental evaluations, the proposed learner demonstrate its effectiveness in identifying novel causal relationships among thousands of variables. The results are based on extensive gold-standard simulations with known ground-truth. Additionally, real biological examples are considered, where the models are applied to high-dimensional molecular data and their output compared against entirely unseen validation experiments. These findings showcase the feasibility of using deep neural approaches to learn causal networks at a large scale.

Additionally, this work presents a novel method for learning dynamical systems from high-dimensional empirical data combining variational autoencoders and spatio-temporal attention within a framework that enforces scientifically-motivated invariances. The focus is set to scenarios in which data are available from multiple different instances of a system whose underlying dynamical model is entirely unknown at the outset. The presented approach builds upon a separation, dividing the encoding into instance-specific information and a universal latent dynamics model shared across all instances. This separation is achieved automatically and driven solely by empirical data. The results offer a promising new framework for efficiently learning dynamical models from heterogeneous data. This framework has the potential for applications in various fields, including physics, medicine, biology, and engineering.

In a different approach, this work explores interventional experiments to shed light on the causal structure within a system. Under the framework of instrumental variables, a new and mathematically sound cause-effect estimator is proposed to uncover sparse causal relations based on unpaired data regimes. The primary focus lies in predicting the outcomes of interventions that have not been performed before, based on data gathered from observed interventions with unknown characteristics. To illustrate, this framework addresses inquiries such as how hypothetical alterations through gene-level interventions could impact the growth rate of a cell. The efficacy of this method is studied on simulated benchmarks and semi-simulated test cases

incorporating human single cell measurements.

Last, this work intends to advance the prediction and comprehension of individual treatment effects in a longitudinal setting. Specifically, this work is investigating clinical records of patients afflicted with wet age-related macular degeneration which if untreated can lead to severe vision loss and legal blindness. To gain a comprehensive understanding of this disease progression, supervised end-to-end models are devised and evaluated to estimate drug responses based on highly irregular time-series data and forecast future treatment effects at individual patient level.

Zusammenfassung

Biologische Systeme sind allgegenwärtig und umfassen komplexe molekulare Netzwerke, die von einzelligen Organismen bis hin zu ausgedehnten Ökosystemen reichen, und die Umwelt unseres Planeten tiefgreifend beeinflussen. Derzeit wird in der Biologie ein umfassendes, quantitatives Verständnis lebender Organismen durch komplexe Systemansätze angestrebt, vergleichbar mit der Art und Weise wie technische Systeme abstrahiert und beschrieben werden. In diesem Zusammenhang ist das Erkennen von kausalen Zusammenhängen eine große Herausforderung für derzeitige Algorithmen der künstlichen Intelligenz. Diese Arbeit stellt ein neues Konzept auf Basis eines neuartigen neuronalen Netzes zur Integration von Kausalität in der Biomedizin vor, das kausale Beziehungen zwischen Variablen durch die Kombination von hochdimensionalen Daten und bereits bekannten kausalen Zusammenhängen identifizieren kann. Insbesondere wird eine Kombination von Netzwerken basierend auf Faltungs- und Graph-Operationen verwendet, um Herausforderungen, wie z.B. hochdimensionale Messungen und Störquellen, effektiv in biologischen Anwendungen zu adressieren. Die Effektivität zur Bestimmung von kausalen Beziehungen zwischen Tausenden von Variablen wird in umfangreichen Simulationen gezeigt. Darüber hinaus werden reale biologische Beispiele betrachtet, bei denen der präsentierte Ansatz auf hochdimensionale molekulare Daten angewandt und die Ergebnisse mit Validierungsexperimenten verglichen werden.

Desweiteren wird in dieser Arbeit eine neuartige Methode zum Lernen dynamischer Systeme aus hochdimensionalen empirischen Daten vorgestellt. Dieser neue Ansatz, basierend auf einem “Variational Autoencoder” mit räumlich und zeitlicher Gewichtung, ist in der Lage invariante Strukturen abzuleiten. Der Schwerpunkt liegt dabei auf Szenarien, in denen Daten von mehreren verschiedenen Realisierungen eines Systems, dessen zugrunde liegendes dynamisches Modell unbekannt ist, verfügbar sind. Der vorgestellte Ansatz basiert auf einer Trennung in instanzspezifische Informationen und ein universelles latentes Dynamikmodell, das alle Realisierungen beschreibt. Diese Aufteilung erfolgt implizit und wird ausschließlich durch empirische Daten gesteuert. Diese Herangehensweise bietet einen vielversprechenden neuen Rahmen für das effiziente Lernen dynamischer Modelle aus heterogenen Daten mit potentiellen Anwendungen in der Physik, Medizin, Biologie und Technik.

Ein anderer Ansatz dieser Arbeit untersucht die Möglichkeit kausale Strukturen anhand von Experimenten mit Interventionen zu identifizieren. Unter Zuhilfenahme von Instrument-Variablen wird ein neuer und mathematisch fundierter Algorithmus zur Bestimmung von Ursache und Wirkung vorgestellt. In diesem Zusammenhang wird die Indentifizierung von seltenen, kausalen Beziehungen auf Grundlage von nicht zusammenhängenden Datenpaaren untersucht. Das Hauptaugenmerk liegt hierbei auf der Vorhersage der Auswirkungen von unbeobachteten Interventionen auf der Grundlage von Daten, die bei Experimenten mit unbekannten Interventions-

charakteristiken gesammelt wurden. Dieser Forschungsansatz umfasst beispielsweise die Frage, wie sich Eingriffe an Genen auf die Wachstumsrate einer Zelle auswirken könnten. Die Wirksamkeit dieser Methode wird anhand von simulierten und teilweise simulierten Testfällen, die aus Gensequenzierungen von einzelnen menschlichen Zellen bestehen, näher beleuchtet.

Zuletzt befasst sich diese Arbeit mit dem Verständnis und einer verbesserten Vorhersage individueller Behandlungseffekte auf Basis von logitudinalen Daten. Konkret werden in dieser Arbeit klinische Aufzeichnungen von Patienten untersucht, die an neovaskulärer altersbedingten Makuladegeneration erkrankt sind. Unbehandelt kann diese Krankheit zu einem signifikantem Sehverlust und der vollständiegen Erblindung führen. Um ein umfassenderes Verständnis dieses Krankheitsverlaufes zu erlangen, werden neue Algorithmen abgeleitet, adptiert und untersucht. Im Fokus steht dabei die Vorhersage des individuellen Ansprechverhaltens von Patienten auf verabreichte Medikamente unter Berücksichtigung von zeitlich sehr unregelmäßigen Untersuchungen und die Vorhersage von zukünftigen Behandlungseffekten eines einzelnen Patienten.

Contents

Contributions

This thesis describes the author's own research. The research was carried out within the context of collaborative projects, with contributions from collaborators as detailed below:

1. **Chapter 2: Causal Learning in High Dimensions:** Chapter 2 contains partially published results [1]. Implementation was done by Kai Lagemann, supported by Christian Lagemann. Methods were developed by Kai Lagemann and Sach Mukherjee. Experiments were performed by Kai Lagemann, supported by Christian Lagemann. Bernd Taschler contributed to design and implementation of experiments using the baseline algorithms.

2. **Chapter 3: Learning Latent Dynamical Models:** Chapter 3 was partially published in [2]. Implementation and empirical evaluation was done by Kai Lagemann and Christian Lagemann. Methods were developed by Kai Lagemann, Christian Lagemann, and Sach Mukherjee.

3. **Chapter 4 Forecasting Responses in Unpaired Interventional Data using Sparse Causal Modeling:** Theoretical work was proven by Niklas Pfister, Jonas Peters and Sach Mukherjee. Algorithm design and implementation was done by Niklas Pfister and Kai Lagemann. Experimental evaluation was done by Kai Lagemann.

4. **Chapter 5: Estimating Treatment Effects using Deep Neural Networks:** Data preprocessing was carried out by Nastassya Horlava supported by Johannes Wahle and Kai Lagemann. Algorithmic design, implementation and empirical evaluation was done by Kai Lagemann. Sach Mukherjee and Fabian Theis supervised the research project of Chapter 5.

1 Introduction

Biological systems surround us almost everywhere, ranging from intricate molecular networks that dictate the functions of single-cell organisms to expansive ecosystems that profoundly impact our planet's environmental conditions. The primary objective of incorporating a systemic approach in the field of biology is to attain quantitative comprehension of living organisms. This level of understanding would mirror the meticulous manner in which engineering elucidates the engineered counterparts of physical systems. This endeavor would enable valuable predictions and insights into biological processes. In the past, machine learning methods have been shown to be efficient in finding correlations from data, but it remains challenging to derive causal – rather than predictive or correlative – insights via machine learning methods. This issue severely limits the applicability of traditional machine learning methods to provide a deeper understanding of the relationships between biological entities. The study of causality is of utmost importance across many scientific disciplines, particularly in public health, medicine, and related fields. Visionary figures like Wright, Rubin, and Pearl have paved the way for causality to become a mathematical concept with precise semantics and a well-founded logical foundation, addressing crucial questions: (1) why events unfold as they do, and (2) what factors contribute to their unfolding? This concept serves as the bedrock upon which scientific principles are built, shaping our understanding of natural phenomena, human behavior, and the interplay of societal systems.

Despite the considerable progress made, understanding the underlying dynamics of complex systems remains an ongoing challenge that continues to inspire researchers worldwide to expand our current knowledge boundaries. Before delving into the mathematical concepts presented later in this work, let's first explore the implications and various aspects of causality. To illustrate this, we first consider a few examples of common diseases and how their understanding and management is linked to causal understanding.

An essential and complex issue in disease studies lies in the identification and explanation of genotype-phenotype relationships. Cause-effect networks in cancer research, for instance, encompass a conceptual structure designed to depict the complex interplay of factors and occurrences contributing to the evolution and advancement of cancer. These networks offer way to grasp the interactions and mutual influences among diverse molecular and cellular constituents illuminating the onset and dissemination of malignant cells. In numerous instances of cancer, distinct biomarkers such as specific genetic mutations, proteins, or molecules with well-defined characteristics are available. These biomarkers can be exploited for early detection, diagnosis,

monitoring treatment responses as well as streamline research including the identification of cancer specific cause-effect networks. Cancer research draws advantage from a broad spectrum of firmly established cell lines and animal models that simulate various facets of the disease. These models empower researchers to analyze the progression of cancer, evaluate potential therapeutic approaches, and delve into the underlying mechanisms at play. The relatively swift advancement of the disease in cancer cases allows researchers to expedite the evaluation of potential treatments in terms of their effectiveness. However, it is important to note that only certain types of cancer can be attributed to specific gene mutations, whereas for others, the relationships between genetic modifications and resulting phenotypic traits are not strictly one-to-one. Distinct genetic abnormalities in various cancer patients can result in identical disease characteristics, creating a challenge in elucidating connections between genetic makeup and observable traits as well as response to therapy, which can differ between patients with seemingly similar cancers. One plausible reason for the diversity observed among cancer instances is that disparate genetic modifications can disrupt shared pathways, ultimately yielding similar disease manifestations. Approaching diseases from a network-oriented perspective aids in surmounting the complexities linked to genotype-phenotype associations and simplifies the task of identifying the genetic origins of diseases.

It is now widely accepted that the initiation of human diseases is driven by a multitude of factors, rather than being attributed to a single, distinct cause. In essence, the synergy of various elements contributes to the progression of a disease. As an illustration, the decline in cognitive and functional abilities among elderly individuals, as witnessed in the context of progressive neurodegenerative disorders like Alzheimer's disease, represents one of the most significant public health challenges of our time. Alzheimer's is not a single disorder but a spectrum of related conditions with varying presentations. This heterogeneity complicates efforts to find a single cause that applies uniformly across all cases. The brain changes associated with Alzheimer's are multifaceted, involving protein misfolding, accumulation of amyloid plaques and tau tangles, inflammation, and synaptic dysfunction. Understanding how these changes interact and cascade is challenging. While some rare cases of early-onset Alzheimer's can be linked to specific genetic mutations, the more common late-onset form has a more complex genetic component involving multiple genes with smaller effects. Identifying which genes contribute and how they interact with environmental, social and lifestyle factors is a complex puzzle, especially considering the vast amount of potential causes and the fact that our understanding of the intricate cellular and molecular processes of the brain is still incomplete.

The mechanical and electro-physiological properties of the heart are among the best understood of any organ in the human body. Yet we see heart diseases, e.g. supraventricular arrhythmia, for which we are not able to derive exact causes. Typically, the heart's electrical system generates impulses that coordinate the heart muscle's contraction and relaxation, resulting in a regular heartbeat. However, in cases of

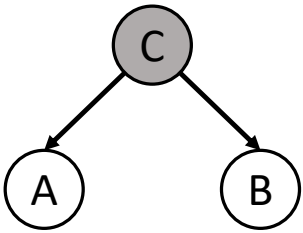


Figure 1.1: Example of a confounder: The settings shows the interconnected nodes A, B, and the hidden common cause C. The arrow directions signify potential influences. Observational data suggests a correlation between A and B, however, the relation between A and B is not a direct cause-and-effect relationship; rather, it is confounded by the variable C. This situation demonstrates how a confounding factor (in this case the variable C), can create a deceptive appearance of a causal relationship between A and B, when in reality, there is no causal influence of A on B or B on A, rather both are caused by C.

supraventricular arrhythmia, an extra electrical signal arises prematurely, causing an early heartbeat, or the electrical signals may follow an abnormal pathway, leading to a rapid and irregular rhythm.

Supraventricular arrhythmia can be triggered by various factors, such as stress, caffeine, alcohol, certain medications, electrolyte imbalances, and underlying heart conditions. Additionally, there is an association between supraventricular arrhythmia and a genetic defect called Wolff-Parkinson-White syndrome. Supraventricular extrasystoles may also arise from prior cardiac infections and cardiomyopathy. Viral and bacterial pathogens can invade atrial tissue, causing mutations that affect the cells' conductivity. These mutated loci within the atrium discharge additional electrical impulses, leading to irregular heart contractions. In such cases, supraventricular arrhythmia may result from a potentially resolved infection that still has lingering effects. Many cases of supraventricular extrasystoles go undetected or are detected at a later stage since long-term electrocardiograms are not routinely performed for all individuals. Determining the exact cause of supraventricular arrhythmia remains challenging for many patients, and reliable proof for diagnosis cannot be obtained through medical imaging or probes prior to surgical intervention. In this scenario, algorithmic assistance aimed at enhancing diagnosis could involve the identification of abnormal biomarkers associated with cardiac arrhythmia in blood samples, the prediction of a patient's future health condition, or the evaluation of the possible outcomes resulting from interventions or treatments like surgery or medication.

These examples and many others highlight the significant importance of causal learning in biomedical applications. It enables researchers to gain insights into the

underlying mechanisms of biological systems and their responses to external factors like environmental influences and therapeutic treatments. However, learning causal structures from data in such contexts remains challenging. These networks are now understood to be context-dependent and are believed to underlie disease heterogeneity and variations in response to molecular therapies. Yet, characterizing this heterogeneity faces obstacles due to the complex nature of learning causal structures at scale. The difficulties in understanding causality, combined with specific characteristics of extensive biological systems like numerous factors, intricate processes, scarce data, and varying levels of noise, contribute to this challenging situation. Confounders, in particular, have the tendency to create a false perception of cause-and-effect relationships between two variables, even when there is no actual link between them. To illustrate, consider the scenario involving three interconnected points, depicted in Figure 1.1. Any apparent correlation between node A and B is not the result of a direct causal connection between the observed nodes but rather arises from a hidden common cause. Such a confounding element can mislead medical professionals and researchers into making erroneous assumptions about the impacts of treatments, interventions, or exposures. Furthermore, confounders themselves might be subject to the influence of other unmeasured factors, leading to a complex sequence of confounding that is intricate to disentangle. To compound the challenge, experimental settings that hold promise to uncover confounding variables could either raise ethical concerns or become prohibitively expensive in practical terms.

Furthermore, describing underlying dynamics, even in simple systems, may appear complex. As a result, data-analytics tools are often relied upon to establish tractable and interpretable reduced-order models. Most commonly used approaches require prior knowledge of the underlying set of state variables before discovering new natural laws or low-level system descriptions. Unfortunately, in many research scenarios, scientists only have access to data that does not directly correspond to the variable space of the underlying system but represents its evolution. Therefore, the ability to extract knowledge based on longitudinal observations without prior knowledge would represent a paradigm-changing advancement for problems where access to the variable space of the underlying system is limited or restricted.

Motivated by these challenges, this thesis puts forward four novel approaches to tackle some of the most relevant shortcomings with regard to causal learning and dynamics identification in biomedical applications. These include

- **Deep Causal Structure Learning:** A novel end-to-end learning framework to shed light on causal relationships in large-scale molecular networks is introduced in Chapter 2. Despite remarkable progress in theory and methods, such causal structure learning problems remain challenging for large-scale real-world applications due to a number of factors, including unknown and complex data-generating processes and high-dimensionality. Hence, this thesis

exploits a novel deep architecture for causal structure learning in high dimensions. The presented approach is based on decision-theoretic/risk-based views of structure learning and is targeted at learning from a combination of empirical data and prior causal knowledge. The proposed learner can effectively identify causal relationships across thousands of variables, as verified in extensive (linear and nonlinear) simulations where ground-truth structure is known and can be directly compared against. Furthermore, using real-world biological data it is found that the learner is able to provide output – at genome scale, spanning thousands of variables – that agrees well with entirely unseen validation experiments.

- **Learning of Latent Dynamical Systems:** Chapter 3 proposes a new framework for learning latent dynamics from observed data combining variational autoencoders and spatio-temporal attention within a learning framework motivated by certain scientifically-observed invariances. These invariances are motivated by two findings concerning classical scientific models. First, it is essential for every output from a class of mechanistic systems to be explicable through a single model that possesses universality across all instances within this class, irrespective of their apparent differences or variations. Second, realization-specific factors (such as initial conditions or constants) are often the same at all times in a given realization and are in that sense time-invariant. Extensive empirical evaluation proved the effectiveness of the proposed learner for various spatio-temporal systems characterized by dynamics governed by ordinary or partial differential equations.

- **Intervention Response Prediction:** A sparse-effect model for unpaired data using instrumental variables is presented in Chapter 4. Intervention experiments emerge as a vital tool, playing a crucial role in advancing our comprehension of the underlying causal structure within a system. Moreover, these experiments empower us with the ability to predict the outcomes of unobserved interventions, further deepening our understanding of complex phenomena. In this context, a scientifically motivated scenario is examined consisting of a response variable and corresponding covariates. Some of these covariates act as causal predecessors, while others are linked through concealed confounding factors. The system is observed through interventions, where the targets of these interventions remain unknown. The ultimate objective is to forecast the outcome of the investigated response variable based on its covariates in the context of an unseen intervention. The effectiveness of the proposed sparse regression framework is demonstrated on a simulated benchmark and a semi-simulated test cases in which the data of the covariates stems from human single cell data.

- **Estimation of Indivual Treatment Effects:** The focus of Chapter 5 lies on predicting individual treatment effects for patients afflicted with wet age-related macular degeneration based on Optical Coherence Tomography (OCT) scans

taken at multiple time points. In this context, a sophisticated treatment effect estimator is proposed that predicts a continuous future treatment effect for individual patients given its past data trajectory. To investigate the complex dynamics of wet AMD and to predict individual patient responses to treatment, state-of-the-art statistical methodologies, including Bayesian linear models, Deep Gaussian Processes, and temporal attention networks are employed.

2 Causal Learning in High Dimensions

Causality is a fundamental concept that lies at the heart of scientific inquiry, enabling us to understand the relationships between variables and uncover the mechanisms that drive observed phenomena. It delves into the fundamental question of how changes in one variable directly influence changes in another. By unraveling cause and effect relationships, causality provides a deeper understanding of the world around us.

In our daily lives, we often observe associations between variables. However, it is important to differentiate between mere correlation and true causation. Correlation refers to a statistical association between two variables, indicating that they tend to vary together. However, correlation alone does not establish causation. To truly understand causality, we need to go beyond the surface-level observations and discern the causal links that drive the observed patterns. It enables us to untwist the intricate tapestry of cause and effect, shedding light on the underlying processes, hidden factors, and mechanisms that govern the phenomena we observe.

In the pursuit of causal inference, we employ a diverse array of sophisticated methodologies and techniques. Rigorous experimental designs, quasi-experimental approaches, natural experiments, and observational studies all contribute to the arsenal of tools at our disposal. From randomized controlled trials that hold sway in the medical field to instrumental variable methods that prevail in econometrics, each methodology serves as a valuable instrument in our quest for causal knowledge.

Causality is a complex task that involves rigorous investigation and careful consideration of various factors. One of the primary challenges in establishing causality is the presence of confounding variables. These are additional factors that are associated with both the cause and the effect, creating a misleading relationship. Confounding variables can obscure the true causal relationships, making it essential to disentangle their influence. Another challenge is selection bias, which arises when the process of selecting individuals or samples for a study is not random, leading to a non-representative sample. This bias can distort causal estimates and compromise the validity of causal inferences. Reverse causality is yet another challenge, where the perceived cause and effect are actually reversed. Differentiating between cause and effect is crucial to avoid drawing incorrect conclusions. Ethical and practical limitations also pose challenges in establishing causality. Conducting randomized controlled trials or interventions may not always be feasible due to ethical concerns or practical constraints. In such cases, researchers often rely on observational data, which can introduce additional complexities and biases.

Despite these challenges, establishing causality is of utmost importance. It forms the basis for evidence-based decision making in various fields, including public policy, healthcare, economics, and social sciences. By understanding causality, we can identify effective interventions, guide policy development, and improve outcomes for individuals and societies. Understanding the factors that causally contribute to diseases and health conditions is vital for devising successful preventive measures. Causal inference plays a key role in identifying these factors and enables us to determine how they influence an individual's response to treatments or interventions. This understanding empowers researchers to develop personalized medicine approaches that tailor treatments to specific patient characteristics, thereby optimizing health outcomes. Furthermore, causal inference methods offer a pathway to unravel the intricate pathways and mechanisms underlying disease progression. By identifying causal factors and their relationships, researchers can gain valuable insights into the fundamental biological processes at play. This knowledge holds the potential for developing new therapies or interventions that target the root causes of diseases.

In this Chapter, we will explore the complexities of causality and introduce a novel methodology for inferring causal structures. Our proposed deep architecture for causal learning is specifically motivated by the challenges posed by high-dimensional biomedical problems. By establishing causal relationships, we gain the ability to make better-informed decisions, design interventions that are more effective, and develop policies and treatments that have a meaningful impact.

2.1 Descriminative Causal Structure Learning

From a machine learning (ML) perspective, changes in causal regime (e.g. under intervention on a system) can lead to nontrivial changes in data distributions. Hence, causal learning involves a broader kind of generalization than in standard predictive ML tasks and the study of causality continues to be a significant and unresolved area within AI research [3, 4]. To better understand the limitations and sources of error in an overall causal inference process, it can be beneficial to make a clear distinction between three key components. First, statistical inference focuses on drawing conclusions about the generating distribution or its properties based on the available data. Second, causal discovery and causal structure learning involves extracting as much information as possible about the underlying causal structure using the statistical quantities, such as probability distributions or their characteristics. Lastly, causal inference involves determining quantitative causal effects by considering both the identified causal structure and the associated statistical quantities. It is important to note that these three inference steps are not always completely separable, and many approaches exist that combine them in interesting ways.

This work focuses on the crucial task of discerning causal relationships between variables, known as causal structure learning, which holds significant importance across various scientific domains [5]. The rich body of work in learning causal structures

includes, among other methods, PC [6], LiNGAM [7], IDA [8], GIES [9], RFCI [10], ICP [11] and MRCL [12]. Scaling causal structure learning to large problems has been facilitated by reformulation as a continuous optimization problem [13] and recent neural approaches, e.g. SDI [14], DCDI [15], DCD-FG [16], or ENCO [17] have demonstrated state-of-the-art performance. However, the process of learning causal structures from data remains challenging and inherently complex. It continues to present difficulties, especially when dealing with real-world problems that involve conditions such as high dimensionality, limited data sizes, and the presence of hidden variables.

In biomedicine, causal networks encoding interplay between entities such as genes or proteins, encoded as directed graphs or networks, play a central conceptual and practical role and carry causal semantics that are widely used by biomedical researchers to understand and reason about processes underlying health and disease. Such networks are increasingly understood to be context-dependent, in the sense that the causal architecture can differ between cell types or disease states manifesting as variation in disease heterogeneity and inhomogeneous response to molecular therapies (see, among others, [18, 19, 20, 21]).

Going beyond the available data to ask what would happen under a change to the system by an imposed intervention [3, 4] and understanding biomedical heterogeneity through causal networks motivates a need for efficient and effective approaches for causal structure learning [5]. However, a key bottleneck in realizing such a vision lies in the lack of effective AI workflows to learn causal structures from real-world data under conditions characterized by high dimensionality, limited data sizes, or the presence of hidden variables.

Under these settings, the ability to learn such networks from data would enable a paradigm shift by facilitating a comprehensive characterization of networks, e.g., across disease states. In high-dimensional biomedical settings, there are common problems stemming from methodological limitations, difficulties in scaling, and specific characteristics of large-scale biology. These characteristics include high dimensionality, intricate underlying events, the existence of hidden or unmeasured variables, limited data availability, varying levels of noise, and more. Thus, we propose a deep architecture for causal structure learning that is motivated in particular by high-dimensional biomedical problems. Specifically, we frame causal structure learning as a non-linear optimization problem in which distributional and graph structural information are combined to output information on causal relationships between variables. The approach we put forward operates within an emerging causal risk paradigm that allows us to leverage AI tools and scale to very high dimensional problems involving thousands of variables. The learners proposed allow for the integration of partial knowledge concerning a subset of causal relationships and then seek to generalize beyond what is initially known to learn relationships between all variables. This corresponds to a common scientific use-case, in which some prior

knowledge is available at the outset – from previous experiments or scientific background knowledge – but it is desired to go beyond what is known to learn a model spanning all available variables.

A large part of the causal structure learning literature involves learning models that allow explicit description of the relevant data-generating model (including both observational and interventional distributions) and are in that sense "generative" (see, e.g. [5] and references therein). Taking a different approach, a number of recent papers, including [22, 23, 12, 24], have considered learning discrete indicators of causal relationships between variables (without necessarily learning full details of the underlying data-generating models) and this can be viewed as related to notions of causal risk [25]. Such indicators may encode for example, whether, for a pair of variables A and B, A has a causal influence on B, B on A, or neither.

The approach we propose, called "Deep Discriminative Causal Learning" (D^2CL), is in the latter vein and inspired by these efforts. In very general terms, the idea is as follows. We consider a version of the causal structure learning problem in which the desired output consists of binary indicators of causal relationships between observed variables [12, 25], i.e. a directed graph with nodes identified with the variables. Available multivariate data X are transformed to provide inputs to a neural network (NN) whose outputs denote the causal indicators. The dimension of the problem is the number of variables between which causal relationships are to be inferred. As noted above, this can be large in biomedical problems, hence we focus on high dimensional problems with thousands of variables. D^2CL differs from classical causal structure learning approaches, both in terms of the underlying framework (based on discriminative causal risk rather than generative causal models) and in leveraging neural learning. The assumptions underlying the approach are also different in nature from those usually made in causal structure learning and concern higher-level regularities in the data-generating processes (see Sec. 2.7).

Thus, D^2CL represents a discriminative neural causal learning approach that is demonstrably effective in the high-dimensional, limited data regime characteristic of many real-world problems, including in biomedicine, spanning large numbers of variables. In summary, the proposed approach has the following main characteristics:

- D^2CL focuses on causal learning for real-world, high-dimensional problems with thousands of nodes but limited data availability and lack of gold-standard simulation engines, representing typical conditions in high-dimensional biomedical problems. Acyclicity (of the directed graphs to be learned) is not assumed, nor is availability of any standard factorization of the joint probability distribution.

- For D^2CL it is not required that samples in the data matrix X are drawn from a single distribution. Samples can be drawn from, e.g., a mix of observational, and interventional distributions and the causal characteristics of these regimes (e.g. which node(s) or latents were intervened upon) need not be known in

advance. This is a common setup for real-world data and in particular for emerging experimental designs in biology (see examples below).

- During the training phase, pattern and feature detectors are trained in a supervised fashion. These learned patterns are statistical representations of certain aspects of underlying causal processes (see Sec. 2.5) and possess causal semantics via the input labels, in a similar sense to the way learned feature maps in image processing and object recognition implicitly capture image semantics.

- For inference, the trained network can be used to output a directed causal relation between *any* pair of variables and thus, $\mathrm{D}^2\mathrm{CL}$ is able to provide a global graph (i.e. over all nodes) which could possibly include cycles. Thus, the model is trained on partial knowledge of some cause-effect relations (prior causal knowledge Π) but seeks to generalize to the complete problem.

- Inference of causal direction is achieved via a combination of causal knowledge encoded in supervision labels and the fact that the NNs used are *not* rotation-invariant (since if an image feature is rotated, in general this will not match the expected input of a filter trained on non-rotated input). Although lack of rotation invariance is not desirable in classic image processing, the proposed learner exploits this to break symmetries. Hence, the output adjacency matrices are *not* symmetric in general.

2.2 Preliminaries

Graphical models, in some form, are commonly utilized as the fundamental framework for both causal structure learning and the exploration of causal relationships. In this work, we set focus to *structural causal models* (SCMs) [6, 3], but other models might be more suitable under different settings. Let $\{X_1 \ldots, X_p\}$ be a set of p random variables whose relations are summarized in a *directed* graph G. We assume that one node of the vertex set $V_i \in V(G)$ is associated with the corresponding variable X_i. In line with [26], we define the domains of random variables as $\mathcal{X}_i = x_i \in \mathbb{R} : p(x_i) > 0$, $i \in V(G)$ and the term $p(X_i)$ is used to refer to the distribution or probability. Here, we only consider continuous random variables as they are dominant in our biomedical applications. The set of graphical parents of X_j is denoted by $Pa(X_j)$ and each edge $(i, j) \in E(G)$ represents a direct causal relation from variable X_i to X_j, if and only if $i \in Pa(X_j)$. Then, we can define a SCM as a set of functional assignments of the form

$$X_i{=}f_i(Pa_{G^*}(X_i), U_{X_i}) \quad \text{for } i \in \{1, \ldots, p\}, \tag{2.1}$$

where $Pa_{G^*}(X_i)$ denotes the set of parents in the ground truth graph G^* for node i and f_i is a node-specific function. Exogenous noise terms U_{X_i} are assumed jointly independent and distributed as $U_{X_i} {\sim} p_i$, where p_i is a node-specific density. We only

assume a factorization of observations according to the structure encoded in graph G. This entails a joint probability distribution that satisfies

$$p(X) = \prod_{i \in V(G)} p(X_i | Pa(X_i)). \tag{2.2}$$

We refer the set of distributions $p(X)$ that are factorized according to eq. 2.2 as exhibiting the *Markov factorization* property [3] in relation to graph G. It is important to note that the Markov factorization does - in general - not entail a unique identifiability of a graph structure based on its conditional independencies. It is quite often the case that the same set of conditional independencies holds for a number of different graph structures. These graphs are called *Markov equivalent* and form together a *Markov Equivalence Class* (MEC). In general, we can state that a graph G is only identifiable up to a Markov Equivalence Class $\mathcal{M}(G)$. Furthermore, many causal structure learning algorithms rely on the assumption that the set of conditional independencies in eq. 2.2 is directly reflected in the structure of the graph G. This assumption is known as causal *faithfulness*.

Performance evaluation in non-causal tasks relies on classical sampling theory, assuming that all present and future data follow the same probability model. However, causal models introduce challenges as they need to capture a range of distributions to overcome the limitation of learning Markov equivalent graph structures solely from observational data. This divergence from conventional machine learning and statistical tasks necessitates alternative evaluation methods for causal structure learning. To gain insights into the specific structures within a Markov equivalence class (MEC), researchers can employ the concept of interventions. In general, interventions refer to any modification made to the structural assignments in the equation 2.1. Intuitively, interventions can involve changing coefficients within the function f_i, altering the parent set $Pa(X_i)$, or modifying the noise term U_{X_i}. Consequently, the resulting *interventional* distribution, which represents the data distribution under a specific intervention regime $\mathcal{I}$, can differ significantly from the observational data distribution. In formal terms, an intervention upon a set of nodes $\{X_i : i \in \mathcal{I}\}$ means replacing the conditional distributions with new distributions. The joint probability distribution changes under an intervention to

$$\tilde{p}(X) = \prod_{i \notin \mathcal{I}} p(X_i | Pa(X_i)) \prod_{i \in \mathcal{I}} \hat{p}(X_i | Pa(X_i)) \tag{2.3}$$

where $\hat{p}(X_i|Pa(X_i))$ indicates the conditional distribution of node X_i in its general form. Similar to the setting with pure observational data, it is only possible to characterize the graph structure under interventions up to an *interventional Markov Equivalence Class* ($\mathcal{I}$-MEC), meaning that two different graphs with the same set of conditional independencies under interventions with regime $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_k\}$ are called $\mathcal{I}$-Markov equivalent. In the limit of interventional knowledge, it is generally possible to identify a unique graph structure from observational and interventional

data. The formulation of upper bounds on the number of experiments necessary to identify causal structure is subject to recent and ongoing research [27, 28].

2.3 Methods Summary

Here, we provide a very brief, high-level summary of the main ideas, which will be explored in more details later on. We propose an end-to-end neural approach to learn causal networks from a combination of empirical data X and prior causal knowledge Π. The general D^{2}CL workflow and its application to biomolecular problems are summarized in Figure 2.1. Here, we offer a concise overview of the key concepts that will be further investigated in greater depth at a later stage.

Suppose $X_1, \ldots, X_p$ is a set of variables whose mutual causal relationships are of interest. Let G^* denote an (unknown) graph whose directed edges encode these causal relationships. D^{2}CL seeks to learn G^* from two inputs: (1) empirical data X containing measurements on each of the variables of interest, and (2) prior causal knowledge Π concerning a subset of causal relationships. This corresponds to a common paradigm in real-world scientific settings, where some data is measured on variables of interest, but only limited knowledge about causal relationships is available at the outset (e.g. from prior scientific knowledge or specific experiments).

Hence, we formalize the learning task in the following way. For each ordered pair of variables with indices (i, j) whose causal status is *not* known via Π, our goal is to learn an indicator of whether or not X_i has a causal influence on X_j. D^{2}CL treats these causal indicators as “labels" in a machine learning sense, using the available inputs to learn a suitable mapping. The goal of the mapping is to minimize discrepancy with respect to the true, unknown causal status; this learning task can be viewed through the lens of causal risk [25]. In line with scientific settings of interest, we assume that the data X does *not* contain interventions that would allow an unknown edge to be directly estimated. Specifically, the learner never has access to data in which the parent node of an unknown edge was intervened upon. This makes learning challenging, as we require generalization to interventional regimes/distributions that are entirely unseen.

Learning is done using a flexible, neural model F_θ with a set of trainable parameters θ using the variable pairs $\mathcal{T}(\Pi)$ as a training set. The model is trained in a specific fashion that leverages the input information Π as a supervision/training signal to allow the model to learn representations suitable for generalization to novel causal relationships. The network F_θ combines a convolutional neural network (CNN) and a graph neural network (GNN) to resolve distributional and graph structural regularities. A high-level sketch of the architecture is given in Fig. 2.2 of Sec. 2.8.

In image processing, CNNs make use of certain properties, e.g. spatial invariance, that exploit the notion of an image as a function on the plane. Here, we leverage

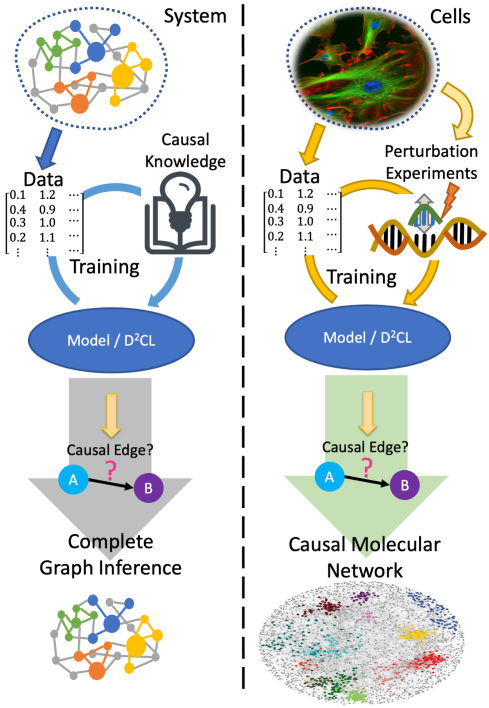


Figure 2.1: Conceptual overview of the proposed learning scheme (left) and its use for large-scale biological experiments (right). We developed a neural architecture for learning causal structures from a combination of empirical data and prior causal knowledge. Learning causal structures from data entails learning a graph G, whose nodes are system variables and whose directed edges encode causal relationships between the variables. Such graphs encode causal – not just correlational – information and their learning remains challenging. In an abstract workflow (left), the proposed learner combines empirical data, obtained from a specific system (with unknown underlying causal structure) with prior causal knowledge, to give an estimate of the unknown causal structure. In an instantiation of the general workflow for biological problems, data is obtained from a specific biological system and causal prior knowledge is derived either from known science or interventional experiments on the system. In the example shown, the problem is high-dimensional since the data matrix spans many different variables whose mutual causal relationships are of interest. The learner combines the sources of information to give an estimate of a graph spanning *all* nodes of interest, thereby leveraging the limited inputs to generalize to the entire system. During the training phase, the model learns features that help identify causal relationships between variables. During inference, the trained model then decides, for any pair of variables A and B, whether A has a causal influence on A, B on A, or neither.

the CNN toolkit to capture distributional information in data X represented as images. We create these visual representations for 2-tuples of nodes. Specifically, for a variable pair (i, j) we use the $n \times 2$ submatrix $X_{(\cdot,[ij])}$, to form a bivariate kernel density estimate $f_{ij} = \mathrm{KDE}(X_{(\cdot,[ij])})$ that is treated as an image input. The learned structures are intended to follow the causal semantics of the input labels, in a similar sense to the way learned feature maps in image processing and object recognition implicitly capture image semantics. Note that this is in general *asymmetric* in the sense that $f_{ij} \neq f_{ji}$. This is important since we want to learn ordered/directed relationships (symmetry here would imply inability to distinguish causal direction).

In addition, we use a graph neural network (GNN) approach to capture regularities in edge structure. GNNs extend key CNN ideas to non-Euclidean settings via operations on graphs. The GNN learns a state embedding h_j which contains the information of the neighborhood for each node j. Each node is associated with an initial feature vector containing the structural information of the neighborhood. The GNN tower requires a graph as input; we provide an initial input graph $\hat{G}_0$ via computationally lightweight routines which are solely based on the available data X (see Methods). For each pair of nodes (i, j), an enclosing subgraph is extracted from the input graph $\hat{G}_0$ and causally informative state embeddings are computed when the enclosing subgraph traverses the GNN. In a GCN layer, each node aggregates information from its neighbors by taking a weighted average of their feature vectors. These aggregated messages are multiplied by a learnable weight matrix to combine information from neighbors. Then, the transformed messages are merged through an aggregation layer capturing global causally informative patterns in the graph.

Finally, following training, the model F_θ – with parameters now fixed as a function of the inputs X and Π – can be used to assign causal status to *any* pair via an inference step. Note that the overall estimate depends solely on the data X and prior causal information Π. In applications, the global model output is tested systematically at large scale against either the true graph G^* (in simulations) or against entirely unseen interventional experiments (for real biological examples).

Our focus is on causal learning for real-world, high-dimensional problems with thousands of nodes and finite, limited data, motivated by large-scale biomedical problems. As outlined above, our model is trained end-to-end in a data-driven fashion under a causal risk paradigm [12, 25]. Within this paradigm, acyclicity (of the directed graphs to be learned) is not assumed, nor is availability of any standard factorization of the joint probability distribution. Furthermore, it is not required that samples in the data matrix X are drawn from a single distribution, rather samples can be drawn from, e.g., a mix of observational, and interventional distributions and the causal characteristics of these regimes (e.g. which node(s) or latents were intervened upon) need not be known in advance. Nor is it required that we have interventional data or prior information concerning all nodes. On the contrary, in all experiments the learner never has access to data in which the parent node of an unknown edge

was intervened upon nor prior information concerning the unknown edge. This is a common setup for real-world data and in particular for emerging experimental designs in biology (see examples below). During the training phase, pattern and feature detectors are trained in a supervised fashion. The inference step allows generalization to the complete problem (i.e. a global graph over all nodes); thus, the model is trained on partial knowledge of some cause-effect relations (encoded in prior causal knowledge Π) but seeks to generalize to the complete problem. We emphasize that the NNs used are *not* rotation-invariant and hence can break symmetries and allow inference of causal direction (experiments below include results concerning this specific aspect).

2.4 Related Work

A significant body of research in causal structure learning focuses on developing models that explicitly represent the data-generating process, encompassing both observational and interventional distributions. These models, referred to as "generative" models, aim to capture the underlying mechanisms that generate the data, accommodating various assumptions and offering different levels of representational power. Typically, these algorithms not only identify the causal directed acyclic graph (DAG) but also establish functional relationships between nodes in the graph and their estimated parent sets. In contrast, discriminative approaches diverge from the notion of inferring functional representations and instead focus on unveiling causal patterns directly from data. These approaches aim to discern causal structure by analyzing the data itself, without explicitly attempting to infer the underlying functional relationships between variables.

Causal structure learning algorithms can be broadly classified into a few main categories: (i) Constraint-based methods use statistical tests to identify conditional independence relationships between variables, which can then be used to infer the causal relationships between them. (ii) Score-based methods apply a scoring function to evaluate candidate causal models based on how well they fit the data. The goal is to find the model that maximizes the score. (iii) Hybrid methods combine elements of both constraint-based and score-based approaches to overcome the limitations of each. The first category of methods employ combinatorial optimization techniques to identify the optimal graph structure by leveraging conditional independencies present in the data. Under the assumptions of causal Markov property, causal faithfulness, and no confounding variables, [6] were the first to present an algorithm that estimated an asymptotically correct completed partial graph by iteratively checking the conditional independence relations of two adjacent nodes conditioned on all-size subsets of their neighbors. Further improvements suggest new versions that are order-independent [29] or can detect unknown confounding variables [30, 31]. Recently, score-based methods have garnered significant attention, and several

new promising approaches have been introduced. Traditionally, score-based learning seeks to optimize a discrete score $Q : \mathbb{D} \to \mathbb{R}$ over the space of DAGs $\mathbb{D}$

$$\min_{G} Q(G) \quad s.t. \quad G \in \mathbb{D}. \tag{2.4}$$

This formulation is equal to a NP-hard optimization problem due to the non-convex, combinatorial nature of the DAGness constraint growing super-exponentially with increasing number of variables p. Commonly used methods solve this optimization problem by performing some form of local search, which involves adding edges and parent sets of one node at a time. This approach is efficient when each node has only a few parents, but as the number of potential parents increases, local search quickly becomes impractical. Additionally, these methods often rely on strict structural assumptions such as bounded in-degree, bounded tree-width, or edge constraints.

The GES algorithm [32] commences with an empty graph and proceeds iteratively by adding and removing edges based on the optimization of a score function. During this equivalence search, it explores the DAG space by adding edges in the forward phase until the score stops increasing. Then, it repeats the process by deleting edges one at a time, maximizing score improvement. The algorithm stops when no more edges can be deleted. This concept was expanded to encompass situations involving multiple intervention experiments and their corresponding interventional distributions. This extension resulted in the development of the Greedy Interventional Equivalence Search (GIES) algorithm [9] and enables regularized maximum likelihood estimation within an interventional framework.

In their work, [13] propose a novel approach to the structure learning problem that eliminates the need for combinatorial constraints. Instead, they frame the problem as a continuous optimization task involving real-valued matrices. The combinatorial constraint $G \in \mathbb{D}$ is replaced with a smooth equality constraint $h(W) = 0$, where W represents the weighted adjacency matrix of the graph G. The formulation is as follows:

$$h(W) = tr(\exp(W \circ W) - d = 0 \tag{2.5}$$

$$\nabla h(W) = (\exp(W \circ W))^T \circ 2W \tag{2.6}$$

Here, $\circ$ denotes the Hadamard product, and exp refers to the matrix exponential. Importantly, the evaluation of the matrix exponential, which is crucial for both h and its gradient $\nabla h(W)$, is a well-established topic in numerical analysis, featuring an algorithm with a complexity of $\mathcal{O}(d^3)$. Consequently, the resulting problem can be efficiently solved using standard numerical algorithms, making implementation straightforward.

The introduction of the smooth equality constraint in [13] has inspired numerous subsequent approaches. One notable algorithm, SDI [14], addresses interventional

settings characterized by sparse interventions that typically impact a single random variable, even if the specific variable remains unknown. This assumption aligns with realistic scenarios where it is unlikely for a single agent to coordinate large-scale interventions across a broad range of causal mechanisms. The algorithm's effectiveness is also demonstrated in cases where the graph structure is partially provided but requires completion. In such scenarios, the focus shifts from complete graph recovery to partial graph recovery, leveraging prior information about the existence of specific cause-effect edges and non-edges. The training process comprises three phases, where both the structural representation of a DAG and the functional representation of independent causal mechanisms are jointly optimized until convergence. To address the interdependence between these parameters, the training alternates between different phases using block coordinate descent optimization. In phase 1, the functional parameters are trained to maximize the likelihood of observational data, leveraging randomly generated graphs that align with our current beliefs about the edge structure. Moving to phase 2, we sample multiple graph configurations based on the parameterized edge beliefs and assess their performance using data samples from the intervened black-box SCMs. Phase 3 involves aggregating scores from interventional data batches across various graph configurations to compute the gradient for the structural parameters. Through this iterative training procedure, the causal structure progressively refines until convergence is achieved.

The work presented in [15] introduces a novel differentiable approach to causal discovery utilizing interventional data and expressive density estimators. Unlike existing methods, their approach does not rely on strong assumptions about the functional form of causal mechanisms. By incorporating an unconstrained objective with a regularization term inspired by the smooth acyclicity constraint, they provide a theoretically-grounded framework for causal discovery. Importantly, the proposed method extends beyond the consideration of discrete random variables and also applies to continuous ones. While it does not explicitly account for latent confounders, the method demonstrates the validity of its theoretical results in the infinite-data regime. These results are established under the assumptions of causal sufficiency, independent and identically distributed samples, and an acyclic graph structure. In subsequent research, [16] introduced a novel concept known as factor directed acyclic graphs, which serves as a means to limit the exploration of non-linear low-rank causal interaction models. By incorporating this innovative structural assumption and leveraging recent advancements [13, 15] that connect causal discovery with continuous optimization, they were able to successfully perform causal discovery on a large scale, involving thousands of variables, primary in biomedical applications.

The work of [17] expands on this line of research by introducing ENCO, an innovative approach that addresses the problem of graph search in causal discovery. ENCO formulates the graph search as an optimization problem that focuses on independent edge likelihoods, with edge orientation treated as a separate parameter. Unlike traditional constrained optimization methods or approaches with acyclicity regularizers,

ENCO does not enforce acyclicity constraints or penalties. Instead, it utilizes unbiased low-variance gradient estimators, allowing for scalability to larger graphs while maintaining convergence guarantees. While interventions on all variables ensure convergence to the correct acyclic graph, ENCO also demonstrates robust performance even in scenarios with limited interventions or small sample sizes. This approach overcomes the limitations of scaling continuous optimization methods beyond linear settings.

An alternative research direction revolves around extracting causal structure directly from data by leveraging causally informative patterns. This line of inquiry acknowledges the necessity for a more flexible approach to causal inference that can learn pertinent causal footprints from data, reducing the dependence on predefined identifiability conditions. Studies conducted by [22, 23] tackle the challenge of distinguishing between A causing B and B causing A using exclusively observational data. Essentially, they explore methods to determine the direction of causality within a finite, independent, and identically distributed samples drawn from the joint distribution. Their approach frames causal inference as a classification problem involving probability measures for pairs of random variables with causal relationships. They propose employing kernel mean embeddings to nonparametrically represent cause-effect samples and extend these techniques to infer causal connections among multiple variables. Theoretical guarantees, including consistency and learning rates, are derived, and approximations are introduced to facilitate the scalability of the learning process for large-scale data.

In subsequent work, [12] build upon the aforementioned concept and introduce a novel approach that expands the idea of utilizing bivariate histograms within a manifold regularization framework to encompass entire graph structures. They specifically address the challenge of estimating edges in a graph that captures causal relationships among a predetermined set of vertices. However, their method diverges from conventional approaches by adopting a machine learning perspective, allowing for the integration of any available information pertaining to known cause-effect relationships. Consequently, the resulting graph generated by their approach may contain cycles, departing from the typical assumption of acyclicity. This innovative technique is firmly rooted in the manifold regularization framework, which serves as the underlying basis for incorporating regularization constraints aimed at promoting favorable characteristics in the inferred graph structure. By leveraging this framework, the method strives to enhance the accuracy and reliability of the learned graph, optimizing its performance and usefulness in capturing causal dependencies. [24] deviate from the manifold framework and instead adopt a generalized linear model approach. This shift in methodology is motivated by the desire to enhance scalability and improve the ease of training the model. By leveraging generalized linear models, the researchers aim to overcome potential limitations associated with the manifold framework, allowing for more efficient and effective learning of causal relationships from data.

Recently, Ke et al. [33] have introduced an innovative approach that transforms discriminative causal structure learning into a meta-learning problem. Their method, CSIvA, employs a specialized variant of a transformer neural network that takes observational and interventional samples as input to predict the structure of a causal Bayesian network. The model incorporates an attention mechanism to effectively identify relationships among variables across samples, and a decoder that generates the inferred network structure. CSIvA exhibits remarkable generalization capabilities, successfully applying learned knowledge from synthetic data to real-world causal Bayesian networks with novel structures. By utilizing meta-learning techniques, CSIvA overcomes the challenges associated with acquiring training data containing known causal structures from diverse real-world domains. It achieves this by leveraging synthetic data with diverse graph structures, ensuring robustness even when confronted with shifts between training and test data distributions. The model is trained using maximum likelihood estimation and does not enforce acyclicity in the graph structure.

2.4.1 Comparison to D^{2}CL

The proposed method D^{2}CL differs fundamentally from classical causal structure learning methods rooted in causal graphical models, as it focuses on "directly" learning the directed causal status of edges. This means it neither seeks to estimate a causal generative model nor is its output based on a model of this kind or associated with explicit conditional independence relationships. This is analogous to the difference between generative and discriminative learning in standard ML tasks. In particular, this means that D^{2}CL can effectively learn novel causal edges (as seen in the experiments reported) and scales well to larger problems, but cannot provide richer output such as full interventional distributions on its own.

The learning framework upon which D^{2}CL is based was introduced in MRCL [12]; the key difference is that while MRCL is based on a classical semi-supervised manifold learning scheme, D^{2}CL uses neural networks in a supervised fashion and can scale to much larger problems that are typically not solvable by MRCL. Similar to all discriminative approaches D^{2}CL leverages the idea to infer causal structure by unveiling causal patterns directly from data. [22] and [23] present very first results of this new line of causal structure learning based on bivariate observational data. MRCL [12] and SCL [24] include interventional samples but differ from D^{2}CL in the applied method. MRCL builds on a manifold regularization framework, while SCL relies on generalized linear models. In contrast, D^{2}CL uses two different types of neural networks, a convolutional neural network and graph neural network, to estimate graph encoding causal relationships.

CSIvA and D^{2}CL are similar algorithms in some respects. One commonality is that both approaches do not assume any factorization of observational or interventional data. They also do not require the data to be generated by a directed acyclic graph,

Table 2.1: Comparision of algorithms

Method	Causal Model	Method Type	Link Function	Latent Confounders	Demonstrated scalability	Variable Type	Sample size used	Intervention Type
SDI	generative	continuous constraint	linear non-linear	No	48	categorical		single variable intervention, unknown target, soft intervention
DCDI	generative	continuous constraint	linear non-linear	No	100	categorical continuous	1000000 samples, interventions on all nodes	regime interventions, soft stochastic interventions, hard stochastic interventions, known/unknown targets
DCD-FG	generative	continuous constraint	linear non-linear	No	1000	continuous	>50000 observational and interventional samples	regime interventions, hard stochastic interventions, known/unknown targets
ENCO	generative	continuous constraint	linear non-linear	No	1000 (categorical)	categorical continuous	100000 observational samples, 4096 interventional samples per node	regime interventions, soft stochastic interventions, hard stochastic interventions full intervention information
CSIvA	discriminative	meta-learning	linear non-linear	Possibly	80	categorical continuous	40000 graphs, 1500 interventional and observational samples each graph	soft and hard interventions, single variable interventions
D^2CL	discriminative	supervised learning	linear non-linear	Possibly	50000	continuous	>1000 observational and interventional samples	hard determenistic interventions, hard stochastic interventions, single variable interventions, unknown targets

and they do not enforce acyclicity in the predicted graph. However, there are notable distinctions between the two methods.

In terms of inference, both CSIvA and D^2CL generate a graph sequentially. However, CSIvA predicts the graph in an autoregressive manner, taking into account previously predicted rows, while D^2CL does not utilize such dependencies in its predictions.

The formulation of D^2CL is based on a supervised learning task, where it is trained and applied to a single system W, without aiming to generalize to different systems. This assumption aligns with the understanding that causal mechanisms can differ greatly even between seemingly similar systems, such as brain cells and skin cells. D^2CL assumes access to observational and interventional samples, as well as prior causal knowledge that helps define causal labels.

On the other hand, CSIvA is designed as a meta-learning task. During training, CSIvA assumes access to tens of thousands of causal systems from a common distribution and learns statistical features representative of causal relations within that distribution. It requires observational and interventional data for all graph samples, along with the underlying adjacency matrix. The generalization capability of CSIvA depends on the diversity of the distribution of causal graphs encountered during training. It may not be feasible for CSIvA, trained on systems like dynamical rigid body systems, to generalize effectively to biomedical applications like gene regulatory networks.

CSIvA and D^2CL also differ in their architectural design. CSIvA learns from the entire graph data, while D^2CL focuses on identifying causally informative patterns between two nodes at a time. CSIvA employs two transformer networks with self-attention mechanisms, whereas D^2CL uses a convolutional neural network to capture distributional information and a graph neural network to capture structural patterns from the bivariate input. Additionally, CSIvA assigns a node identifier to each node at the graph scale, while D^2CL uses node identifiers within shuffled subgraphs. A comparison to recent neural methods for causal learning appears in the Table 2.1.

2.5 Problem Statement

The following section presents a comprehensive formulation of the mathematical problem under investigation, laying the foundation for subsequent analysis and exploration.

2.5.1 Notation

Observed variables with index set $V = \{1, \ldots, p\}$ are denoted $X_1, \ldots, X_p$. The variables will be identified with vertices in a directed graph G whose vertex and edge sets are denoted $V(G), E(G)$, respectively. We occasionally overload G to refer also to the corresponding binary adjacency matrix, using G_{ij} to refer to the entry (i, j) of the adjacency matrix, as will be clear from context.

We use linear indexing of variable pairs to aid formulation as a machine learning problem. Specifically, an ordered pair $(i, j) \in V \times V$ has an associated linear index $k \in \mathcal{K} = \{1, \ldots, K\}$, where K is the total number of variable pairs of interest. Where useful we make the mapping explicit, denoting the linear index corresponding to a pair (i, j) as $k(i, j)$ and the variable pair corresponding to a linear index k as $(i(k), j(k))$. The linear indices of pairs whose causal relationships are *unknown* and of interest are $\mathcal{U} \subset \mathcal{K}$ and those pairs known in advance via input knowledge Π are $\mathcal{T}(\Pi) \subset \mathcal{K}$. In all experiments $\mathcal{T}(\Pi)$ and $\mathcal{U}$ are disjoint, i.e., no prior causal information is available on the pairs $\mathcal{U}$ of interest.

Our intuition suggests that specialized methods are needed to ensure the validity of this type of inference. In particular, such methods rely on assumptions that are somewhat different in nature and arguably stronger than those used in traditional statistical inference.

2.5.2 Problem Statement

We focus on the setting in which available inputs are

(I1) Empirical data: an $n \times p$ data matrix X whose columns correspond to variables $X_1, \ldots, X_p$.

(I2) Causal background knowledge Π providing information on a subset $\mathcal{T}(\Pi) \subset \mathcal{K}$ of causal relationships.

No particular assumption is made on (I1) but in all experiments we ensure that the data matrices never contain either data from test interventions, nor any observational data realizations that were used to define gold-standard labels. Note that for (I1), we assume that the data matrix X consists of observational and interventional samples as follows: First, n_0 observational samples are available from the system of interest under no intervention, i.e. we collect n_0 samples of the system in a non-intervened state. Additionally, we assume having access to the outcome of multiple interventions.

We overload $\mathcal{I}$ to represent the family of k interventions performed upon the system, $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_k\}$. For every $\mathcal{I}_j \in \mathcal{I}$, we have n_j i.i.d. samples generated according to eq. 2.3. The total sample size of the data matrix X accumulates to $n = n_0 + \sum_{i=1}^{k} n_i$.

For (I2), we assume that information is available concerning the causal status of a subset of variable pairs. That is, for some variable pairs (X_i, X_j) the correct binary indicator G^*_{ij}, e.g., presence/absence of an edge in the target graphical object, is provided as an input. In terms of linear indexing, these can be viewed as available "labels" of causal status for the pairs $\mathcal{T}(\Pi) \subset \mathcal{K}$. No specific assumption is made on the data X, but in line with our focus on generalizing to unseen causal relationships, it is assumed that it does *not* contain interventional data corresponding to the pairs in $\mathcal{U}$. Furthermore, in all experiments, not only are the sets $\mathcal{T}$ and $\mathcal{U}$ disjoint, but we enforce the stronger requirement that $u \in \mathcal{U} \implies \nexists j : k(i(u), j) \in \mathcal{T}$, i.e. all interventions on which models are tested are entirely novel, i.e. unrepresented in the inputs to the learner.

Thus, the learning task can be formulated as follows: Given the inputs (I1) and (I2), the goal is to estimate for each ordered pair of variables (X_i, X_j) with unknown causal relationship whether or not X_i has a causal influence on X_j.

2.6 Summary of learning scheme

With the notation above, our goal is to learn a graph whose nodes correspond to the variables $X_1, \ldots, X_p$ and edges represent causal relationships. Our framework is discriminative (in the sense above) and supervised; accordingly, what constitutes a "causal relationship" depends on the setting and input Π (in real data experiments below these are potentially indirect causal effects). To this end, we train a parameterized network F_θ, i.e., a nonlinear function F with a set of unknown, trainable parameters θ. This is possible since we know for each pair $k \in \mathcal{T}$ the causal status G^*_{ij} based on input information Π.

The architecture we use as F_θ is detailed below, but for now assume this has been specified. Then, given the data X and the training labels $Y_k = G^*_{i(k),j(k)}$ for all pairs $k \in \mathcal{T}(\Pi)$, we train the set of parameters $\hat{\theta}(X, \Pi)$ under a loss that is supervised by the (causal) labels Y_k.

At this stage, the trained network $F_{\hat{\theta}(X,\Pi)}$ allows assignment of causal status to *any* pair since it gives an estimate of the entire graph including those pairs whose causal status was unknown. The output is given by

$$\hat{G}_{ij}(X, \Pi) = \begin{cases} F_{\hat{\theta}(X,\Pi)}(i, j;\, X) & \text{if } k(i,j) \notin \mathcal{T}(\Pi) \\ Y_{k(i,j)}(\Pi) & \text{otherwise} \end{cases}. \tag{2.7}$$

where (i, j) are ordered variable pairs. Note that the overall estimate depends solely on the data X and causal information Π. By default, no change is made for pairs $\mathcal{T}$ whose status was known at the outset. Eigenmann et al. [25] studied causal notions of risk based on loss functions of the form $L(\hat{G}, G^*)$ that compare a graph estimate $\hat{G}$ with ground-truth G^*. Given a causal estimator $\hat{G} = F_G(X)$, the theoretical risk is

$$R(F_G) = \mathbb{E}\left[L(G^*, F_G(X))\right]. \tag{2.8}$$

There are many different types of loss functions L that can be used for risk estimation, depending on the specific application and the nature of the problem being solved. In our setting, we consider a classification-type loss on the variable pairs k, where the causal status of known pairs $\mathcal{T}(\Pi)$ provides the training "labels". Therefore, the general risk estimator takes the form of a cross-entropy loss over ordered variable pairs k. This penalizes the model for assigning high probabilities to incorrect edges and low probabilities to correct edges. We further augment the loss function by standard terms that, for instance, prevent exploding weights. In line with [25], the concept of causal risk is tailored to the specific problem at hand and measures how well the causal structure learning method performs in the particular context defined by the system W. This notion of risk takes into account the finite sample size and provides an estimate of the method's performance based on the available data. Furthermore, the problem-specific nature of the risk estimator acknowledges that a particular method may perform well in certain settings, but not in others, depending on the specific characteristics of the system being studied, in alignment with our system-specific approach.

2.7 Causal interpretation of the learning scheme

In this Section, we discuss some concepts underpinning the idea of discriminative causal structure learning, in particular addressing interpretation and specifying the conditions under which discriminative causal structure learning may be expected to be effective. However, note that the following arguments are not intended to constitute a rigorous theory at this stage but rather to help gain understanding of the conditions under which discriminative causal structure learning may be expected to be effective.

D^{2}CL outputs a directed graph: the discriminative nature of D^{2}CL means that the notion of causal influence encoded by the edges is rooted in the application setting and input information Π, since causal semantics are inherited via the problem setting rather than specified by a generative model (see [12] for related discussion). Indeed, in the experiments we showed that depending on the problem set-up D^{2}CL could be used to successfully learn either direct or indirect/ancestral causal relationships.

General causal framework. To provide a framework within which to discuss causal structure learning, we start with a general structural causal model and then introduce assumptions for D^{2}CL (*MGA* and *DCSI*, see below). Following [3, 34], we assume decomposition of the underlying system into modular and independent mechanisms:

Independent Causal Mechanisms (ICMs): The causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other.

For variables X_i assume a structural causal model with equations $X_i{=}f_i(Pa_{G^*}(X_i), U_{X_i})$ with $i \in \{1, \ldots, p\}$, where $Pa_{G^*}(X_i)$ denotes the set of parents in the ground truth graph G^* for node i and f_i is a node-specific function. Exogenous noise terms U_{X_i} are assumed jointly independent and distributed as $U_{X_i}{\sim}p_i$, where p_i is a node-specific density. The foregoing is a generic causal structural model. The next assumption is specific to our approach and explains when machine learning tools as applied here can be effective in learning causal structures.

Our approach treats the f_i's and p_i's as unknown but assumes they are related at a higher level. This can be formalized as a meta-generator assumption as follows:

Meta-Generator Assumption (MGA): For a specific system W, the functions f_i and noise distributions p_i are (independently) generated as $f_i{\sim}\mathcal{F}_W$ and $p_i{\sim}\mathcal{P}_W$, where $\mathcal{F}_W$ denotes a *function generator*, and $\mathcal{P}_W$ a *stochastic generator*, that are specific to the applied problem setting W.

MGA is motivated by the notion that in any particular real-world system, underlying (biological, physical, social, etc.) processes tend to share some functional and stochastic aspects, which impart some higher-level regularity. That is, MGA states that in a given applied context, functions f_i and (ICM-consistent) noise terms U_{X_i} while unknown, varied and potentially complex, are nonetheless related at a "meta"-level. The generators $\mathcal{F}_W, \mathcal{P}_W$ are random processes, representing respectively a "distribution over functions" and "distribution over distributions", whose role here is to capture the notion of relatedness among f_i's (respectively p_i's) in a given setting W. Note that $\mathcal{F}_W, \mathcal{P}_W$ are treated as unknown and never directly estimated (see below).

As stated in Sec. 2.5, we focus on the causal status of variable pairs (X_i, X_j) (rather than general tuples) which denotes the simplest possible case under MGA. Furthermore, in both our work and the majority of interventional studies in applications such as biology, single interventions (rather than joint interventions on multiple nodes) are the norm. This requires the following additional assumption:

Dominant cause under single interventions (DCSI): A sufficiently large change in one of potentially multiple causes leads to a change w.r.t. the effect. Therefore, sin-

gle interventions are sufficient to drive variation in the child distribution which is independent of the occurrence of other dominant causes.

From MGA and DCSI to discriminative causal structure learning. Consider an applied problem W with underlying causal graph G^*_W, treated as fixed but unknown. The associated functions and noise terms are also unknown but assumed to follow MGA. Then, under DCSI, we have that *all* pairs of the form (X_i, X_j) , have underlying relationships of the form $X_j{=}f_j(X_i, U_{X_j})$ with components following the MGA (i.e. drawn from generators $\mathcal{F}_W, \mathcal{P}_W$). This in turn suggests that within the setting W, identification of causal pairs can be treated as a classification problem, since all pairs share the *same* generators. In other words, MGA restricts the distribution over relations of variables in U and noise terms to system-specific distributions.

Note that no particular assumption is made on the individual functions f_j, only that they are mutually related on a higher level. Furthermore, the generators themselves need not to be known or are directly estimated, it is only important that they are shared across the applied setting W. Further note that a model learned for setting W will not in general be able to classify pairs in an entirely different applied setting W' (since the generators may then differ strongly), i.e. we do not seek to learn "universal" patterns that apply to *all* causal relations in any system whatsoever. The classification task of D^2CL aims at telling apart causal parent-child functions, drawn from the system-specific function generator F_W, from non-causal ones. Naturally under MGA, a covariate shift between training and test distribution could occur due to biased sampling from $\mathcal{F}_W, \mathcal{P}_W$ which we address by normalization. Independence in the classification sense (i.e. conditional independence given the causal label), relies upon the independence of the noise terms and also the assumption within MGA that the f_i's are independently generated via $\mathcal{F}_W$. Given a set of nodes $V = \{V_1, \ldots, V_n\}$, ICM states that it is possible to perform a localized intervention on the parent set $Pa(V_i)$ for node V_i without changing the conditional $P(V_i|Pa(V_i))$. Hence, $P(V_i|Pa(V_i))$ is assumed to be invariant and independent of all other conditionals $P(V_j|Pa(V_j))\forall j \neq i$. If DCSI holds, a randomly drawn function sample $f_i \sim \mathcal{F}_W$ can be decomposed into an additive set of independent parent-child relations $f_i(Pa(V_i)) = \sum_{j \in DCSI(Pa(V_i))} f_i(V_j)$. Note that classification is performed on the level of invariant function samples and not rows of the data matrix X. Therefore, the collection of random samples forming the input data matrix X can be composed of samples from observational and interventional experiments as long as the overall mixture distribution is not changed throughout training and inference.

We note that in real systems, f_i's may be coupled via constraints on global functionality, hence non-independent, however, the good performance seen in Sec. 2.12 empirically justifies the approach. Despite the initial theoretical ideas above, rigorous theory and theoretical properties of the kind of approach studied here remain to be understood. In particular, precise conditions on the underlying system needed to ensure that the classification-type approach can guarantee recovery of specific

causal structures remain incompletely understood. We emphasize also that in contrast to classical causal learning schemes, for example based on causal DAGs, we cannot make theoretical statements concerning underlying multivariate distributions and their link to edges estimated by our models at this stage. Our goal is good performance in an edge-wise sense (as detailed above) and the core assumptions concern a limited notion of classifiability. We note also that our models learn edges separately and do not impose any particular wider/global constraints (such as acyclicity or path constraints), although this could in principle be done within the supervised framework.

2.8 Architecture details

In this Section, we present the neural network architecture employed in our approach for causal structure learning. Figure 2.2 illustrates a flow chart of the architecture.

CNN Tower: To capture distributional information from empirical data X, a preprocessing step is required. In principle, this could be done via a variety of multidimensional transformations of X. We consider the simplest possible case, namely for a pair (i, j) to consider only the corresponding columns i and j in the data matrix X. Specifically, we use the $n \times 2$ submatrix $X_{(\cdot,[ij])}$ to form a bivariate kernel density estimate $f_{ij} = \mathrm{KDE}(X_{(\cdot,[ij])})$. Note that this is in general asymmetric in the sense that $f_{ij} \neq f_{ji}$, which is important since we want to learn ordered/directed relationships. In other words, this ensures that in general the CNN tower can output different probabilities for the edges $A \to B$ and $B \to A$ (for any two nodes A and B). Evaluations of the KDE at equally spaced gridpoints on the plane (i.e. numerical values from the induced density function) are treated as the input to the CNN. The KDE itself is a standard bivariate approach using automated bandwidth selection following [35, 36]. This provides an "image" of the data and allows us to leverage existing image analysis ideas. Furthermore, we concatenate the numerical KDE values on the regularly spaced grid with a positional encoding of the grid points channelwise.

The concrete network architecture of our CNN tower is inspired by a ResNet-54 architecture [37]. From a high level perspective, it consists of a stem, five stages with $[3, 4, 6, 3, 3]$ ResNet blocks and multiple fully connected layers that transform the high-level feature maps into a latent space that is merged with the output of the GNN tower. The first ResNet block at each stage downsamples the spatial dimensions of the output of the previous stage by a factor of two. To enhance the computational efficiency of the bottleneck layers in each ResBlock, channel down- and upsampling exploiting 1×1 convolutions is performed before and after each feature extraction CNN layer [38]. We replaced ReLU activations by its parametric counterpart PReLU [39] allowing to learn the slope of the negative part at negligible additional computational costs, which addresses the problem of dying neurons. Following [40], we chose a full pre-activation of the convolutional layers, i.e.

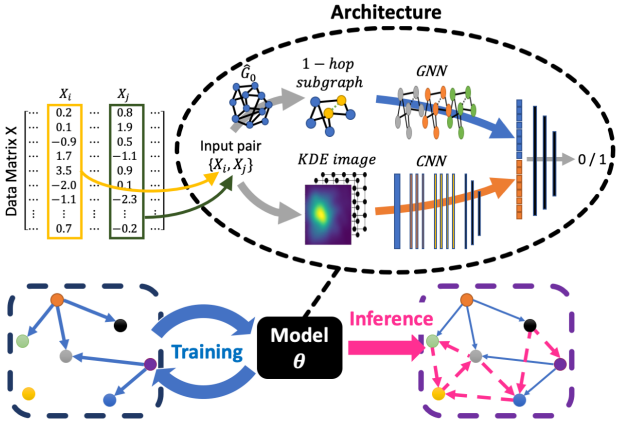


Figure 2.2: Overview of the D^{2}CL architecture, training and inference. D^{2}CL combines empirical data on a large number of variables with prior causal knowledge to learn causal relationships between variables. For any pair of variables X_i and X_j (corresponding to two columns of the input data matrix), D^{2}CL seeks to learn whether X_i has a causal influence on X_j, X_j on X_i, or neither. This is done using a neural architecture with two components: a CNN tower aimed at learning distributional features and a GNN tower that detects structural regularities. For an ordered pair (X_i, X_j), the CNN tower captures distributional information via a bivariate density estimate that traverses the tower to form an embedding. The GNN tower extracts a 1-hop subgraph from an initial graph $\hat{G}_0$ and computes an embedding containing structural information on the neighborhood of the nodes. The CNN and GNN embeddings are then merged through multiple layers which finally output the probability of a directed causal relationship. The input causal information is used to provide a training signal (see text for details). During inference the network generalizes beyond the initial inputs to provide an estimate of the global graph spanning all variables of interest.

normalization-activation-convolution.

GNN tower: Our GNN tower builds on the SEAL architecture of [41] and the resulting graph convolutional neural network (GCNN) for link prediction. The underlying notion is that a heuristic function predicts scores for the existence of a link. However, instead of employing predefined heuristics (such as the Katz coefficient or PageRank), an adaptive function is learned in an end-to-end fashion, which is formulated as a graph classification problem on enclosing subgraphs. [41] proved that a γ-decaying heuristic can indeed be approximated by an h-hop neighborhood while the approximation error is at least decreasing exponentially. These findings show that it is possible to learn high-order graph structure features from local enclosing subgraphs instead of the entire graph that can be exploited for link prediction. Consider the pair of nodes of interest (i, j), then, the GNN tower is intended to infer causally interesting node features and state embeddings based on a local 1-hop enclosing subgraph extracted from the approximated input graph $\hat{G}_0$. For node pair (i, j), we first extract a set of nodes $\mathcal{N}$ with all nodes that are connected to either node i or node j based on the adjacency matrix of the approximated input graph $\hat{G}_0$. Then, the edge structure within the subgraph $G_{i,j}$ is reconstructed by pulling out all edges from $\hat{G}_0$ for which the parent and child node are in $\mathcal{N}$. The order of the nodes is shuffled for each subgraph. The node features in every input subgraph consist of structural node labels that are assigned by a *Double-Radius Node Labeling* (DRNL) heuristic [41] and the individual data features. In a first step, the distances between node i and all other nodes of the local subgraph except node j are computed. The same is repeated for node j. A hashing function then transforms the two distance labels into a DRNL label that assigns the same label to nodes that are on the same "orbit" around the center nodes i and j. During the training process the DRNL label is transformed into a one-hot encoded vector and passed to the first graph convolutional layer. In contrast to traditional CNNs, GCNNs do not benefit strongly from very deep architecture design [42, 43]. Therefore, our GNN tower consists only of four sequentially stacked graph convolutional layers. The activation function is defined as the hyperbolic tangent. Since the number of nodes in the enclosing subgraph for each pair of variables (i, j) is different, a SortPooling layer [44] is applied to select the top k nodes according to their structural role within the graph. Afterwards, 1-dimensional convolutions extract features from the selected state embeddings.

Embedding Fusion: Each tower outputs a high-dimensional embedding of the individual features found. These embeddings are concatenated and further processed by multiple fully connected layers. Finally, the last layers output the log-likelihood of a directed edge from node i to node j.

Implementation Details All network architectures are implemented in the open source framework PyTorch [45]. The GNN is coded based on the deep graph library [46]. All modules are initialized from scratch using random weights. During training, we apply an Adam-Optimizer [47] starting at an initial learning rate $\epsilon_0 = 0.0001$. Furthermore, the learning rate is reduced by a factor of five once the evaluation metrics stopped improving for 15 consecutive epochs. The minimum learning rate is set to $\epsilon_{min} = 10^{-8}$. This learning rate scheduler shows the best results in the current study. The training predictions are supervised on the binary cross entropy loss between estimated and ground truth truth edge label. The evaluation metric is the area under the ROC-curve on the distinct test dataset. Every network architecture is trained for 100 epochs. All computations are run on multiple GPU nodes simultaneously each equipped with eight Nvidia Tesla V100.

2.9 Experimental Setup

In general terms, we use data X and causal information Π (in real problems, derived from interventional experiments) to train the learners and then test the output against entirely unseen interventional data. The data are strictly split in the sense that (i) in all experiments the pairs $\mathcal{U}$ on which the model output is tested are disjoint from those pairs $\mathcal{T}$ whose causal relationships are provided as training inputs and (ii) no data used to define the true causal relationships against which the model output is tested appear in inputs to the models.
At a high level, it is important to understand why evaluating causal structure learning methods empirically is a challenging task that differs from conventional non-causal tasks in machine learning and statistics. In non-causal tasks, performance measures based on classical sampling theory are appropriate, since the underlying assumption is that all data, present and future, are generated from the same probability model. However, in the case of causal models, the model captures a set of distributions that arise from various interventions on the system. This characteristic restricts the applicability of conventional sampling theory-based approaches to evaluating causal structure learning methods.

2.10 Overview of Datasets

2.10.1 Gold-standard simulated benchmark data

Structural Equation Model based on Directed Acyclic Graph: Given the set of variables $X_1, \ldots, X_p$, our chosen SEM is a set of p functions, each corresponding to one of the observed variables. Without loss of generality, the non-parametric form of the SEM comprises equations

$$X_i = f_{X_i}(Pa(X_i), U_{X_i}), \quad i = 1, \ldots, p,$$

where $Pa(X_i)$ is the set of parents for node i and the U_{X_i}'s are exogenous noise variables assumed jointly independent but arbitrarily distributed. The structure of

this SEM follows a directed acyclic graph (DAG) and a variety of synthetic datasets is generated for empirical investigation. The considered functions f are

1. linear: $f(X_i) = aPa(X_i) + b + U_{X_i}$
2. MLP-tanh: $f(X_i) = W_2 tanh(W_1 Pa(X_i) + b_1) + b_2 + U_{X_i}$
3. MLP-leaky ReLU: $f(X_i) = W_2 f_{l.ReLU}(W_1 Pa(X_i) + b_1) + b_2 + U_{X_i}$
4. tangent hyperbolic: $f(X_i) = tanh(norm(Pa(X_i)) + U_{X_i}$
5. leaky ReLU: $f(X_i) = f_{l.ReLU}(norm(Pa(X_i)) + U_{X_i}$
6. polynomial of order three: $f(X_i) = a_1 Pa(X_i) + a_2(Pa(X_i))^2 + a_3(Pa(X_i))^3 + b + U_{X_i}$

Additionally, we consider measurement noise at different signal-to-noise ratios varying from $SNR = 10$ to $SNR = 0.1$.

Definition of ground truth graph G^* In a first series of experiments, we sought to investigate *direct causal effects*. That is, for a pair of nodes (i, j), the corresponding edge $i \to j$ the ground truth adjacency matrix is set to one, if and only if node i is in the parent set of node j, i.e. $j \in Pa_{G^*}(i)$. In other words, $G_{ij} = 1$, if node j occurs directly on the right hand side in the SEM for node i.
The causal graph G^* in the above examples encodes direct causal effects. However, in many real-world examples, interest focuses also on indirect effects that may be mediated by other nodes. For example, if node A has a direct effect on B, and B on C, intervention on A may change C, even though A does not itself appear in the equation for C. To study the ability to identify such indirect effects, we additionally tested the various methods on the task of learning indirect edges. This was done in the same way as above, but with the inputs Π being indirect edges and output tested against a true, gold-standard indirect graph. That is, in this example, if A causes B and B causes C, the gold-standard graph (and, where relevant, causal inputs in Π) would have an additional edge $A \to C$ to capture the causal, but indirect, effect of A on C. Other aspects were as for the direct problems above. In a second series of experiments, we report performance metrics for causal relationships of this kind. This simulated dataset differs from the previous one in the sense that the ground truth graph is obtained as the transitive closure of the original DAG

$$G^* = \bigcup_{i=1}^{\infty} G_i^{*DAG}.$$

We used the same set of transition functions as before. This experimental setup is arguably closer to the real-world biological data, where effects may be indirect in this sense.

2.10.2 Yeast Gene Deletion Experiments

We use data from yeast gene deletion experiments [48], which have previously been employed for causal learning [11, 49, 12]. To define causal status, we follow the approach of [12], considering changes under interventions relative to the observational distribution. The data is a collection of $n^{int} = 1479$ interventional samples and $n^{obs} = 153$ observational data points each containing measured gene expression levels for a total of $p = 5535$ genes (after preprocessing). The interventional samples are from gene knockout experiments (each carried out for a specific target gene) while observational samples stem from experiments with no such intervention.

To ensure that models are tested on entirely unseen interventions, we split the data such that the set of interventions against which model output is tested is entirely disjoint from any model input. The number of interventions whose effects are available to the learner via Π is denoted m.

To define causal status, we follow the approach of [12], considering changes under intervention relative to the observational distribution for the same gene. Specifically, if c_{ij}^{int} is the expression level of gene j observed under intervention on gene i, for any variable pair $(i, j) \in V \times V$ we say that i has a causal effect on j if and only if $z_{ij} = |c_{ij}^{\text{int}} - m_j^{\text{obs}}|/s_j^{\text{obs}} > \tau$, where m_j^{obs} is the median level of gene j in the observational data; s_j^{obs} is the corresponding inter-quartile range (IQR) and τ is a threshold (set to 5 in the experiments). In other words, if G^* is the gold-standard $p \times p$ adjacency matrix, we have $G_{ij}^* = 1 \iff z_{ij} > \tau$. This approach focuses attention on strong causal effects and ensures that the notion of "change under intervention" is appropriate to the scale of each gene.

Note that in the yeast example causal effects may be indirect and our goal in the analysis is to learn a directed graph G with nodes corresponding to p observed genes and edges $(i, j) \in E(G)$ that represent (possibly indirect) causal influences, i.e. existence of a directed causal path from X_i to X_j (possibly via latent variables). Such edges are scientifically interesting as they are relatively amenable to experimental verification as noted in [50, 24]. Cycles can arise in systems biology (see e.g. [51]) and we do not enforce acyclicity in this example (see [52] and references therein, for discussion of cyclic causality). A fuller discussion of the causal interpretation of laboratory experiments is beyond the scope of this work, but relevant work includes [53, 52, 54] and we direct the interested reader to these references for further discussion.

We conducted the following experiments:

- **Varying amount of causal input.** In a first set of experiments, we seek to investigate the influence of varying the amount of causal input provided to the learners. More precisely, we vary the number of interventions m whose effects are available to the learner. (In terms of the notation above $m = |\{i : \exists j,\, k(i, j) \in \mathcal{T}(\Pi)\}|$). Results are shown in Figure 2.5a-c.

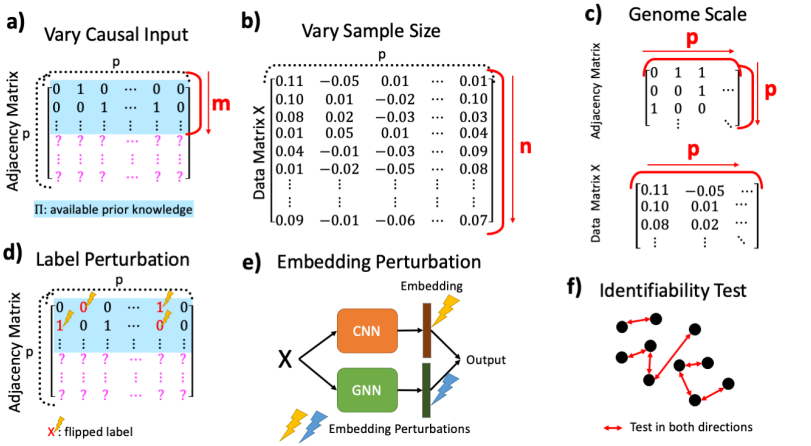


Figure 2.3: Visual illustration of changed configurations in evaluation of yeast gene deletion data (see Text for details): (a) Varying amount of causal input m, (b) Varying sample size n, (c) Scaling to full yeast genome, (d) Label perturbation, (e) Embedding perturbation, (f) Causal direction analysis

- **Varying sample size.** Here, we vary the sample size n of the data matrix X. The matrix X never contains interventions on variables V and its sample size can therefore be varied separately from the number m of training interventions or the dimension $p = |V|$. Results are given in Figure 2.5d-f.

- **Scaling to full yeast genome.** This experiment is a higher dimensional example with $p = 5535$ genes aimed at investigating performance in a larger problem. Since none of the baseline algorithms can scale to such a high-dimensional space, Figure 2.5g-k only contains evaluation results for D^2CL.

- **Label perturbation.** To study sensitivity, we introduced errors into Π; this was done by perturbing 10% of the training labels , i.e. labeling causal pairs as non-causal and vice versa at the outset. Results are presented in Figure 2.6a.

- **Embedding perturbation.** In this experiment, we intentionally perturb the independent embeddings of the two towers in the forward pass of D^2CL. In general, the embedding of either the CNN or the GNN tower is modified right before the fusion layer to test the impact of a failing tower on the overall performance. The experiment contains four different embedding modifications: (i) We set the complete embedding of one tower to zero and thereby, erase all information of this tower. In the other cases we apply strong zero-mean Gaussian noise with varying scale to the embedding , (ii) $\sigma = 1.0$, (iii) $\sigma = 2.0$,

and (iv) $\sigma = 5.0$. Subfigure 2.6b illustrates the results under embedding perturbations.

- **Causal direction analysis.** To empirically study the performance to recover directed and asymmetric causal relations, we constructed test datasets as follows: For each causal edge $k \to l$ of the test set, we also include the *reverse* direction $l \to k$. Intuitively, this means that any learner that predicts undirected causal links would achieve an AUC score of 0.5 since the prediction of a causal edge for $k \to l$ entails the prediction of edge $l \to k$, which, consequently, is a false positive. Table 2.7b summarizes the empirical results and Figure 2.7a presents a low-dimensional representation of the feature maps of the CNN tower right ahead of the embedding fusion.

In the first and second series of experiments, we limit the total number of variables to $p = 1000$ due to computational considerations for the baseline algorithms (not for D^{2}CL, which as we show further below can practically scale to very large problems). Since this is only a subset of the entire yeast genome (a total of 5535 genes were included in the full set of data of the third line of experiments), in these experiments (many) latent variables (genes not included in the dataset in addition to other biological variables, such as proteins, not measured in the data) are present by design.

2.10.3 CRISPR-based interventional data in human cells

Here, we use recent data due to [55] involving large-scale perturbation experiments in human cells to further test performance. In the experiments used here, a CRISPR-based gene editing protocol was combined with single-cell RNA sequencing to allow for efficient multiplexed interventions. This protocol gives mRNA readouts for thousands of genes under each of a large number of interventions in two cell types, a leukemia cell line (K562) and (non-cancer) retinal pigment epithelial (RPE) cells. For K562, the dataset includes $m = 2285$ detected genetic perturbations with a mean coverage of 148 cells per perturbation and a median coverage of 134 cells per perturbation. For RPE, $m = 2679$ genetic perturbations were detected with a mean coverage of 101 cells per perturbation and a median coverage of 79 cells per perturbation. After filtering, the causal graph of the K562 cell line contains $p = 8552$ nodes and the causal graph of the RPE cell line is of dimension $p = 8833$.

To ensure that models are tested on entirely unseen interventions, we split the data such that the set of interventions against which model output is tested is entirely disjoint from any model inputs.

To define causal status, we consider changes under intervention relative to the observational distribution. Treating the normalized gene expression levels as con-

tinuous random variables, we use an adapted version of the total variation between interventional and observational distributions

$$V_{i,j} = \frac{1}{2} \int |f_j^{int(i)}(x) - f_j^{obs}(x)| dx \ .$$

Specifically, if $V_{i,j} > \tau$ we consider the edge $i \to j$ causally relevant. In other words, if G^* is the gold-standard $p \times p$ adjacency matrix, we have $G^*_{ij} = 1 \iff V_{i,j} > \tau$. We chose $\tau = 0.3$ where the threshold τ is a problem specific hyperparameter. This approach focuses attention on strong causal effects and ensures that the observational distribution and the corresponding distribution under an intervention deviate sufficiently.

2.11 Baseline Comparisons

- **Simulated data:** We compare D^{2}CL against: (i) IDA [8], (ii)SCL [24], and (iii) Marginal correlation coefficients.

- **Yeast Gene Deletion Experiments:** We compare D^{2}CL against: (i) Classical causal structure learning approaches including IDA [8], LV-IDA [56], and the GIES algorithm [9]. (ii) A discriminative causal approach called SCL [24] and (iii) Marginal correlation coefficients (Pearson correlations in the figures, but Kendall and Spearman correlations gave similar results) as a simple baseline.

- **Human data:** We compare D^{2}CL against: (i) IDA [8], (ii)GES [57] and (iii) GIES [9], (iv) LiNGAM [58], (v) CAM [59], and (vi) RFCI [31].

In addition to the established methods and baselines listed above we also considered a range of recently developed neural network-based causal learning approaches including SDI [14], DCDI [15], DCD-FG [16], ENCO [17] and CSIvA [33]. For scalability reasons, it is not feasible to apply SDI or DCDI to the datasets explored during the empirical evaluation (scalability issues for SDI and DCDI are also reported in [16]). At the time of writing, there was no source code available for CSIvA [33]. We evaluated DCD-FG and ENCO on simulated benchmark test cases with known ground truth graphs. Despite hyper-parameter tuning, we found that results were poor (see Fig. 2.4b). This is likely due to the problem setting (including dataset type and sizes), in which $n = 6144$ observational and interventional training samples are available. For applications motivating our work this is realistic, but nevertheless much smaller than the datasets used in the original papers (DCD-FG was was applied to $p = 1000$ nodes using $n > 50000$ observational and interventional samples and ENCO to $p = 1000$ categorical variables using $n_{obs} = 10^5$ observational samples and $n_{int} = 4096$ interventional samples for each node). We emphasize that all of the above methods are conceptually exciting and very powerful in suitable settings. Our work is targeted at the high-dimensional, limited data regime – as relevant to

many current, real-world applications, including in biology – hence our experimental set-up is designed to test performance in this kind of regime.

More generally, we note that the comparison with existing causal approaches is not one-to-one, since in many cases methods differ in their expected inputs and outputs. For example, IDA is aimed at analysis of observational data, hence the comparison is unfair since our approach has access also to background causal information Π. GIES allows for interventional data, but requires different inputs. Due to these differences in input/output requirements, we emphasize that comparisons here are provided for completeness but with the caveat that the various methods are intended for different use-cases (and furthermore make assumptions that are likely not met in the real biological data).

2.12 Results

We assess the proposed approaches in comparison to a range of existing methods, using both simulated data and real biological data. In the case of the simulations, we have access to the true, underlying causal graph, and hence can assess results by direct comparison with the ground truth. For the real data examples, we test the model output against the outcome of entirely unseen interventional experiments. In all experiments, simulated or real, we test the model output with respect to causal relationships that are entirely unseen in the sense that (i) the variable pairs on which the model output is tested are disjoint from those pairs whose causal relationships are provided as inputs during training, and (ii) no data used to define the gold-standard causal relationships against which the model output is tested appear in inputs to the models.

2.12.1 Gold-standard simulated benchmark data

We first tested the proposed methods using linear and non-linear simulations. These involved generating data X (and obtaining prior knowledge Π) from a (linear or non-linear) SEM with noise, based on a known underlying causal graph G^*. The protocol is outlined in Figure 2.4a with further details in Sec. 2.10. Results were evaluated against the true, gold-standard causal structure G^* and hence tested in causal (and not correlational or predictive) terms.

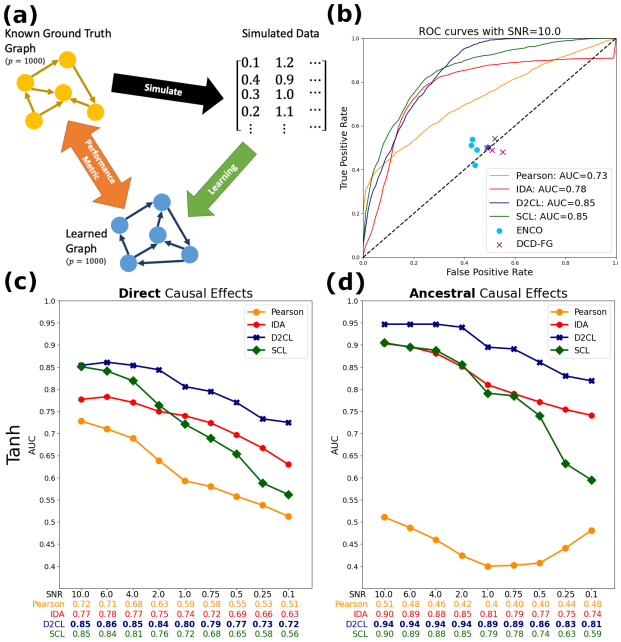


Figure 2.4: Results - large-scale simulated data. (a) Overview of experimental workflow. Data were simulated from known, gold-standard causal graphs and the output of the learners was compared with the true, underlying graph to quantify ability to recover causal structure. Finite-sample empirical data were generated using a directed causal graph of specified dimension p, specifically via linear and nonlinear structural equation models with noise. Functional forms used include simple linear functions, multi-layer perceptrons (MLPs) with tangent hyperbolic activations, MLPs with leaky ReLU activation, tangent hyperbolic, leaky ReLU and a polynomial of order three. (b) ROC curves for an illustrative nonlinear case (the tangent hyperbolic), at a signal-to-noise ratio $SNR = 10.0$, for direct causal relations in a graph with p=1500 nodes. D^{2}CL (blue) is compared against Pearson correlation coefficients (orange), IDA (red), SCL (green), ENCO (cyan) and DCD-FG (purple). The ROC curve and the area under the ROC curve (AUC) is given for algorithms providing continuous output (Pearson, IDA, SCL, D^{2}CL). The binary graph estimates of ENCO and DCD-FG are represented by single markers for five different runs. (c) Results for an illustrative nonlinear case (the tangent hyperbolic), at varying noise levels, for direct causal relationships. Causal area under the ROC-Curve is shown as a function of signal-to-noise ratio (SNR) for an experiment with p=1500 variables and a sample size of n=1024. Results for other linear and nonlinear functions appear in Table 2.2. D^{2}CL (blue) is compared with: Pearson correlations (yellow; this is a non-causal baseline); IDA (red); and SCL (green). (d) Results for indirect causal relationships, with other settings as in (c). Here, causal AUC is with respect to a graph encoding causal, but potentially indirect, relationships (see also Table 2.3). (Results shown are averages over five datasets at each specified SNR.)

In-system, out-of-distribution evaluation: We first study performance on *in-system* but *out-of-distribution* tests. In this setting, model training uses (limited) prior knowledge and data from a given system; assessment is with respect to unknown edges within the same system (test and training edges are always entirely disjoint). This is *out-of-distribution* since the learner never has access to samples from the test interventional distributions, but *in-system* since all data are from the same overall data-generating system. This corresponds to a common use-case in scientific applications where the goal is to learn a model for a given system of interest given available data on that system. Figure 2.4c shows results for a problem of dimension p=1500 using a nonlinear transition function (the tangent hyperbolic; other functions and configurations are shown in Table 2.2 (AUC) and A.1 (AUPRC) in the Appendix) and varying SNR. (For these first results, we restricted the dimension of the problem to facilitate comparison to existing approaches that are less scalable than D^{2}CL; higher dimensional examples appear below.) Note that pairwise correlations between the variables ("Pearson") are ineffective; this is expected due to the presence of latent variables in all experiments and the fundamental difference between correlational and causal relationships. Overall, D^{2}CL remains effective across a broad range of SNRs, as well as for a range of linear and nonlinear problems and problem sizes. These results support the notion that D^{2}CL can learn direct causal edges in systems spanning many variables. We also compared D^{2}CL to DCD-FG [16] and ENCO [17], two recently proposed, scalable neural-causal learners. Due to computational considerations, we restricted this comparison to a subset of the simulations. Exemplary results appear in Fig 2.4b. We find that neither approach is effective in this case. This is likely due to the limited nature of the data inputs and the presence of latent variables.

In addition, we test the effectiveness of D^{2}CL for additive and multiplicative Gaussian noise with varying SNRs under settings with hard deterministic and stochastic interventions. The test results (AUC and AUPRC values) are summarized in Table 2.4 and A.4 and support the notion that D^{2}CL is robust to different types of noise.

The graph G^* in the above examples encodes direct causal relationships since there is an edge from one node to another if the former appears in the equation for the latter. However, in many real-world examples, interest focuses also on indirect effects, that may be mediated by other nodes. For example, if node A has a direct effect on B, and B on C, intervention on A may change C, even though A does not itself appear in the equation for C. To study the ability to identify such indirect effects, we next tested the various methods on the task of learning indirect edges.

Table 2.2: AUC values for direct cause-effect relations for $p = |V| = 1500$.

	Linear				MLP(tanh)				MLP(leaky ReLU)				Tanh				Leaky ReLU				Polynom 3			
SNR	Pearson	IDA	D^{2}CL	SCL	Pearson	IDA	D^{2}CL	SCL	Pearson	IDA	D^{2}CL	SCL	Pearson	IDA	D^{2}CL	SCL	Pearson	IDA	D^{2}CL	SCL	Pearson	IDA	D^{2}CL	SCL
10.00	0.718	0.748	**0.789**	0.641	**0.691**	0.625	0.686	0.634	0.688	0.608	**0.693**	0.623	0.728	0.777	**0.854**	0.851	0.756	0.837	**0.843**	0.729	0.809	**0.848**	0.829	0.641
6.00	0.700	0.771	**0.795**	0.617	**0.670**	0.609	0.658	0.638	0.666	0.590	**0.683**	0.602	0.710	0.783	**0.861**	0.841	0.736	0.818	**0.839**	0.692	0.784	**0.824**	0.821	0.637
4.00	0.684	0.768	**0.784**	0.616	**0.648**	0.590	0.647	0.625	0.652	0.562	**0.667**	0.584	0.689	0.770	**0.854**	0.819	0.700	0.792	**0.831**	0.668	0.735	0.787	**0.812**	0.628
2.00	0.638	0.781	**0.802**	0.615	0.613	0.544	**0.639**	0.594	0.617	0.521	**0.661**	0.592	0.639	0.750	**0.844**	0.764	0.644	0.743	**0.815**	0.630	0.651	0.722	**0.777**	0.617
1.00	0.595	0.774	**0.796**	0.614	0.572	0.506	**0.622**	0.551	0.575	0.487	**0.642**	0.546	0.593	0.740	**0.806**	0.721	0.582	0.701	**0.787**	0.619	0.552	0.659	**0.743**	0.598
0.75	0.589	0.765	**0.793**	0.612	0.556	0.494	**0.619**	0.566	0.567	0.483	**0.638**	0.568	0.580	0.724	**0.795**	0.689	0.572	0.696	**0.783**	0.610	0.539	0.645	**0.734**	0.603
0.50	0.558	0.748	**0.787**	0.610	0.536	0.473	**0.631**	0.540	0.544	0.459	**0.641**	0.567	0.558	0.697	**0.770**	0.654	0.548	0.678	**0.771**	0.606	0.521	0.640	**0.717**	0.592
0.25	0.537	0.735	**0.784**	0.572	0.530	0.467	**0.617**	0.517	0.496	0.434	**0.624**	0.552	0.538	0.667	**0.733**	0.588	0.517	0.667	**0.748**	0.579	0.514	0.634	**0.694**	0.543
0.10	0.523	0.730	**0.774**	0.558	0.492	0.441	**0.618**	0.530	0.507	0.439	**0.616**	0.528	0.513	0.630	**0.725**	0.562	0.503	0.661	**0.743**	0.559	0.492	0.620	**0.691**	0.539

Table 2.3: AUC values for indirect cause-effect relations for $p = |V| = 1500$.

	Linear				MLP(tanh)				MLP(leaky ReLU)				Tanh				Leaky ReLU				Polynom 3			
SNR	Pearson	IDA	D²CL	SCL	Pearson	IDA	D²CL	SCL	Pearson	IDA	D²CL	SCL	Pearson	IDA	D²CL	SCL	Pearson	IDA	D²CL	SCL	Pearson	IDA	D²CL	SCL
10.00	0.553	0.907	**0.928**	0.708	0.548	0.522	**0.733**	0.700	0.563	0.483	**0.789**	0.738	0.511	0.903	**0.947**	0.905	0.502	0.857	**0.943**	0.839	0.610	0.822	**0.933**	0.761
6.00	0.540	0.905	**0.925**	0.700	0.537	0.502	**0.720**	0.658	0.552	0.458	**0.775**	0.735	0.487	0.896	**0.947**	0.895	0.501	0.852	**0.941**	0.808	0.598	0.815	**0.927**	0.751
4.00	0.530	0.905	**0.928**	0.677	0.533	0.490	**0.711**	0.675	0.548	0.447	**0.767**	0.727	0.460	0.881	**0.947**	0.888	0.504	0.848	**0.937**	0.782	0.581	0.803	**0.914**	0.732
2.00	0.506	0.897	**0.928**	0.619	0.523	0.461	**0.683**	0.656	0.532	0.420	**0.766**	0.695	0.424	0.851	**0.940**	0.856	0.501	0.834	**0.920**	0.736	0.543	0.775	**0.879**	0.704
1.00	0.507	0.888	**0.925**	0.609	0.513	0.433	**0.660**	0.626	0.518	0.393	**0.738**	0.672	0.400	0.810	**0.895**	0.791	0.502	0.824	**0.900**	0.676	0.520	0.753	**0.831**	0.658
0.75	0.507	0.882	**0.920**	0.631	0.513	0.431	**0.638**	0.610	0.514	0.385	**0.721**	0.637	0.402	0.790	**0.891**	0.785	0.501	0.822	**0.888**	0.667	0.516	0.747	**0.820**	0.641
0.50	0.506	0.877	**0.918**	0.505	0.510	0.422	**0.618**	0.577	0.513	0.387	**0.713**	0.655	0.407	0.771	**0.861**	0.740	0.502	0.821	**0.880**	0.651	0.510	0.742	**0.808**	0.635
0.25	0.513	0.873	**0.913**	0.516	0.511	0.423	**0.622**	0.567	0.509	0.385	**0.703**	0.639	0.441	0.754	**0.830**	0.632	0.498	0.816	**0.861**	0.624	0.502	0.736	**0.792**	0.589
0.10	0.506	0.867	**0.905**	0.542	0.501	0.417	**0.617**	0.571	0.503	0.385	**0.705**	0.613	0.481	0.741	**0.819**	0.595	0.500	0.817	**0.865**	0.598	0.503	0.734	**0.784**	0.537

Table 2.4: Gold-standard simulations: AUC values for *direct* cause-effect relations for $p = |V| = 1500$: additive and multiplicative noise

	deterministic hard interventions				stochastic hard interventions			
	additive Noise		multiplicative Noise		additive Noise		multiplicative Noise	
SNR	Linear	Tanh	Linear	Tanh	Linear	Tanh	Linear	Tanh
10.00	0.849	0.882	0.746	0.810	0.794	0.812	0.696	0.796
6.00	0.799	0.891	0.691	0.779	0.781	0.826	0.688	0.790
4.00	0.805	0.875	0.675	0.771	0.785	0.825	0.677	0.791
2.00	0.833	0.876	0.664	0.775	0.779	0.814	0.676	0.778
1.00	0.833	0.837	0.678	0.768	0.771	0.734	0.675	0.748
0.75	0.828	0.832	0.666	0.747	0.770	0.718	0.669	0.749
0.50	0.818	0.798	0.643	0.757	0.737	0.726	0.672	0.771
0.25	0.816	0.772	0.639	0.745	0.720	0.709	0.673	0.743
0.10	0.808	0.761	0.617	0.753	0.720	0.697	0.663	0.751

This was done in the same way as above, but with the inputs Π being indirect edges and output tested against the true indirect graph.

Results appear in Figure 2.4d. D^{2}CL outperforms existing methods across a range of SNRs and also in other linear/nonlinear problem configurations (see Table 2.3 and A.2 in the Appendix). IDA performs well in case of a linear SEM but not for functions based on nonlinear MLPs. D^{2}CL appears to be the most noise robust of the methods tested. These results show that D^{2}CL can learn indirect causal edges over many variables under conditions of noise and non-linearity.

Table 2.5: Out-of-distribution experiments: Cross validation experiments - AUC values

			Predicted class																
			Linear									leaky ReLU							
		SNR	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10
Trained class	Linear	10.00	0.787	0.791	0.793	0.793	0.786	0.781	0.772	0.755	0.728	0.752	0.756	0.756	0.746	0.730	0.725	0.718	0.705	0.701
		6.00	0.796	0.798	0.799	0.797	0.783	0.778	0.768	0.749	0.722	0.752	0.754	0.753	0.743	0.726	0.720	0.714	0.708	0.705
		4.00	0.792	0.797	0.800	0.803	0.791	0.784	0.774	0.756	0.730	0.725	0.731	0.735	0.738	0.730	0.726	0.719	0.710	0.703
		2.00	0.763	0.770	0.781	0.797	0.793	0.789	0.780	0.760	0.728	0.708	0.715	0.721	0.733	0.730	0.727	0.721	0.713	0.706
		1.00	0.732	0.738	0.752	0.781	0.791	0.792	0.790	0.776	0.746	0.692	0.699	0.704	0.724	0.732	0.730	0.728	0.718	0.709
		0.75	0.710	0.716	0.732	0.768	0.785	0.786	0.786	0.776	0.755	0.652	0.661	0.672	0.702	0.712	0.714	0.720	0.722	0.709
		0.50	0.709	0.715	0.727	0.763	0.782	0.785	0.788	0.782	0.769	0.670	0.676	0.683	0.700	0.710	0.715	0.718	0.721	0.716
		0.25	0.673	0.677	0.689	0.731	0.760	0.768	0.776	0.778	0.774	0.628	0.632	0.638	0.659	0.688	0.696	0.704	0.714	0.721
		0.10	0.662	0.667	0.678	0.712	0.736	0.742	0.751	0.763	0.776	0.674	0.678	0.682	0.687	0.688	0.687	0.687	0.694	0.714
	Tanh	10.00	0.654	0.656	0.656	0.656	0.644	0.636	0.627	0.619	0.608	0.788	0.786	0.778	0.747	0.697	0.685	0.666	0.632	0.619
		6.00	0.653	0.654	0.654	0.652	0.636	0.628	0.619	0.610	0.598	0.781	0.781	0.775	0.743	0.696	0.684	0.666	0.630	0.610
		4.00	0.649	0.651	0.651	0.651	0.639	0.630	0.622	0.614	0.604	0.777	0.777	0.773	0.744	0.691	0.678	0.659	0.622	0.611
		2.00	0.642	0.644	0.644	0.648	0.635	0.624	0.613	0.604	0.600	0.732	0.734	0.736	0.739	0.684	0.670	0.646	0.611	0.600
		1.00	0.672	0.673	0.674	0.682	0.681	0.673	0.659	0.645	0.637	0.711	0.716	0.721	0.736	0.739	0.726	0.705	0.673	0.650
		0.75	0.664	0.667	0.669	0.682	0.691	0.687	0.676	0.656	0.644	0.708	0.715	0.720	0.737	0.746	0.737	0.718	0.694	0.664
		0.50	0.673	0.676	0.678	0.689	0.694	0.690	0.677	0.657	0.644	0.683	0.692	0.703	0.723	0.732	0.733	0.720	0.697	0.663
		0.25	0.679	0.682	0.685	0.696	0.712	0.711	0.707	0.687	0.663	0.583	0.592	0.607	0.647	0.676	0.687	0.708	0.720	0.698
		0.10	0.674	0.676	0.678	0.686	0.702	0.708	0.712	0.713	0.694	0.501	0.504	0.502	0.505	0.549	0.588	0.638	0.697	0.718

			Predicted class																
			Polynom 3									Tanh							
		SNR	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10
Trained class	Linear	10.00	0.728	0.724	0.721	0.707	0.687	0.686	0.675	0.663	0.656	0.758	0.757	0.756	0.748	0.725	0.716	0.695	0.664	0.639
		6.00	0.739	0.735	0.729	0.712	0.689	0.685	0.674	0.667	0.665	0.768	0.766	0.764	0.752	0.723	0.712	0.692	0.669	0.648
		4.00	0.709	0.708	0.711	0.707	0.691	0.689	0.680	0.669	0.663	0.749	0.750	0.750	0.748	0.728	0.719	0.698	0.667	0.645
		2.00	0.690	0.686	0.687	0.692	0.687	0.688	0.680	0.673	0.668	0.742	0.742	0.743	0.742	0.723	0.717	0.697	0.672	0.648
		1.00	0.676	0.673	0.673	0.673	0.675	0.678	0.678	0.675	0.670	0.718	0.718	0.719	0.718	0.709	0.706	0.695	0.678	0.650
		0.75	0.607	0.611	0.625	0.655	0.668	0.672	0.674	0.677	0.671	0.694	0.696	0.700	0.705	0.699	0.698	0.690	0.682	0.653
		0.50	0.635	0.639	0.654	0.670	0.670	0.671	0.671	0.675	0.673	0.710	0.712	0.714	0.714	0.702	0.699	0.688	0.677	0.656
		0.25	0.563	0.562	0.569	0.608	0.645	0.655	0.661	0.668	0.674	0.644	0.648	0.656	0.670	0.679	0.680	0.675	0.671	0.660
		0.10	0.609	0.608	0.613	0.637	0.651	0.654	0.653	0.656	0.666	0.680	0.681	0.686	0.691	0.685	0.682	0.670	0.656	0.652
	Tanh	10.00	0.796	0.792	0.785	0.742	0.683	0.663	0.641	0.608	0.599	0.841	0.840	0.834	0.792	0.715	0.683	0.640	0.605	0.588
		6.00	0.792	0.791	0.785	0.750	0.694	0.672	0.650	0.612	0.592	0.840	0.841	0.839	0.801	0.720	0.686	0.644	0.609	0.582
		4.00	0.780	0.780	0.779	0.749	0.691	0.666	0.644	0.604	0.591	0.835	0.840	0.843	0.812	0.726	0.693	0.643	0.600	0.579
		2.00	0.738	0.739	0.740	0.749	0.702	0.672	0.641	0.596	0.581	0.797	0.805	0.815	0.832	0.754	0.721	0.659	0.596	0.569
		1.00	0.705	0.707	0.709	0.721	0.720	0.704	0.673	0.642	0.625	0.761	0.766	0.775	0.792	0.779	0.765	0.704	0.638	0.605
		0.75	0.699	0.702	0.703	0.715	0.714	0.710	0.690	0.655	0.633	0.750	0.753	0.758	0.768	0.762	0.758	0.721	0.660	0.611
		0.50	0.667	0.667	0.673	0.687	0.693	0.698	0.690	0.659	0.633	0.700	0.703	0.709	0.719	0.723	0.724	0.711	0.661	0.610
		0.25	0.555	0.561	0.564	0.587	0.625	0.638	0.662	0.681	0.660	0.591	0.596	0.603	0.621	0.639	0.644	0.666	0.688	0.643
		0.10	0.545	0.547	0.544	0.553	0.552	0.566	0.590	0.649	0.678	0.524	0.528	0.530	0.528	0.529	0.538	0.568	0.637	0.673

			Predicted class																
			MLP (leaky ReLU)									MLP (Tanh)							
		SNR	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10
Trained class	Linear	10.00	0.400	0.392	0.382	0.362	0.376	0.383	0.388	0.396	0.403	0.413	0.402	0.392	0.378	0.391	0.394	0.395	0.404	0.416
		6.00	0.394	0.386	0.384	0.368	0.378	0.384	0.388	0.397	0.405	0.407	0.397	0.393	0.381	0.392	0.396	0.396	0.407	0.418
		4.00	0.443	0.440	0.434	0.404	0.395	0.395	0.393	0.396	0.403	0.450	0.446	0.443	0.417	0.405	0.403	0.397	0.403	0.417
		2.00	0.434	0.430	0.419	0.388	0.388	0.390	0.390	0.398	0.404	0.437	0.434	0.424	0.403	0.401	0.397	0.395	0.408	0.415
		1.00	0.453	0.451	0.450	0.429	0.403	0.400	0.393	0.398	0.401	0.451	0.449	0.448	0.435	0.414	0.403	0.397	0.404	0.409
		0.75	0.467	0.465	0.460	0.423	0.398	0.394	0.391	0.396	0.398	0.469	0.466	0.463	0.435	0.408	0.401	0.396	0.403	0.407
		0.50	0.447	0.444	0.435	0.407	0.392	0.392	0.389	0.396	0.399	0.452	0.448	0.442	0.419	0.404	0.399	0.396	0.403	0.408
		0.25	0.455	0.454	0.453	0.432	0.414	0.409	0.398	0.398	0.397	0.464	0.463	0.461	0.448	0.425	0.415	0.405	0.404	0.406
		0.10	0.429	0.425	0.420	0.407	0.401	0.397	0.394	0.400	0.398	0.440	0.434	0.429	0.418	0.408	0.410	0.403	0.403	0.408
	Tanh	10.00	0.398	0.386	0.368	0.358	0.383	0.397	0.413	0.429	0.444	0.417	0.404	0.388	0.381	0.406	0.418	0.426	0.439	0.456
		6.00	0.379	0.371	0.364	0.362	0.382	0.395	0.412	0.432	0.425	0.400	0.390	0.385	0.381	0.400	0.412	0.424	0.441	0.442
		4.00	0.391	0.384	0.374	0.358	0.378	0.391	0.405	0.430	0.439	0.408	0.399	0.388	0.374	0.397	0.407	0.415	0.440	0.455
		2.00	0.400	0.399	0.401	0.382	0.379	0.390	0.395	0.427	0.451	0.407	0.409	0.416	0.400	0.395	0.401	0.406	0.441	0.464
		1.00	0.399	0.393	0.391	0.391	0.386	0.395	0.397	0.405	0.418	0.403	0.399	0.400	0.399	0.403	0.401	0.394	0.414	0.432
		0.75	0.442	0.435	0.431	0.403	0.405	0.404	0.398	0.402	0.409	0.446	0.445	0.439	0.419	0.415	0.409	0.403	0.409	0.419
		0.50	0.463	0.469	0.472	0.480	0.460	0.436	0.414	0.406	0.402	0.468	0.471	0.478	0.487	0.460	0.433	0.413	0.408	0.410
		0.25	0.503	0.504	0.506	0.515	0.495	0.487	0.473	0.446	0.412	0.493	0.496	0.497	0.503	0.490	0.482	0.478	0.444	0.413
		0.10	0.501	0.503	0.501	0.501	0.508	0.519	0.538	0.547	0.477	0.497	0.497	0.498	0.499	0.518	0.516	0.537	0.559	0.460

Out-of-system, out-of-distribution evaluation: D^2CL is intentionally designed to be trainable using (limited) data from a specific system (e.g. a specific biological system, such as cells of particular kind, or a disease state). However, it is interesting to see whether it is possible to generalize to *different* systems. We study such generalization by training D^2CL on a dataset from a certain system and cross-evaluate the trained model on data from another system (i.e. a different simulation regime). Results appear in Table 2.5 and A.5 in the Appendix. It is interesting to see that some generalization is possible, suggesting that D^2CL can extract patterns that are causally informative in a cross-system sense. However, the cross-evaluation performance is always worse compared to in-system training and evaluation. This is expected and we emphasize that we do not claim any general ability to achieve out-of-system generalization. Nevertheless, these results are interesting and support the use of large-scale meta-learning for causal structures [14].

Large-scale evaluation: Finally, to test the scalability of D^2CL to very high dimensional problems, we considered a problem with $p = 50,000$ variables (i.e. $p = 50,000$ nodes in the ground-truth graph). We consider learning of direct causal relationships; results appear in Table A.3 in the Appendix. Note that this is a setting that none of the compared methods can practically scale to. These results show that D^2CL can indeed scale to very high dimensional problems spanning many thousands of variables.

2.12.2 Large-scale biological data.

Next, we sought to study performance in the context of real biological data. To this end, we leveraged a large set of gene deletion experiments in yeast [48], which have previously been used for causal learning [11, 49, 12]. These data involve measuring gene expression in yeast cells under each of a large number of interventional (gene deletion) experiments.
In biological experiments, causal effects may be indirect (e.g. via latent variables) and our goal in the analysis is to learn a directed graph with nodes corresponding to p observed genes and edges representing (possibly indirect) causal influences. Such edges are scientifically interesting as they are relatively amenable to experimental verification as noted in [50, 24]. Cycles can arise in systems biology (see e.g. [51]) and we do not enforce acyclicity (see [52] and references therein, for discussion of cyclic causality). A fuller discussion of the causal interpretation of laboratory experiments is beyond the scope of this work, but relevant work includes [53, 52, 54] and we direct the interested reader to these references for further discussion.

Since causal background knowledge is an input to our approach, it is relevant to consider performance as a function of the amount of such input. To this end, we fixed the problem size to $p = 1000$ and varied the number of interventions m whose effects were available to the learner (see Sec. 2.10 for details). Since each experiment involves only a subset of the entire yeast genome, latent variables are present by de-

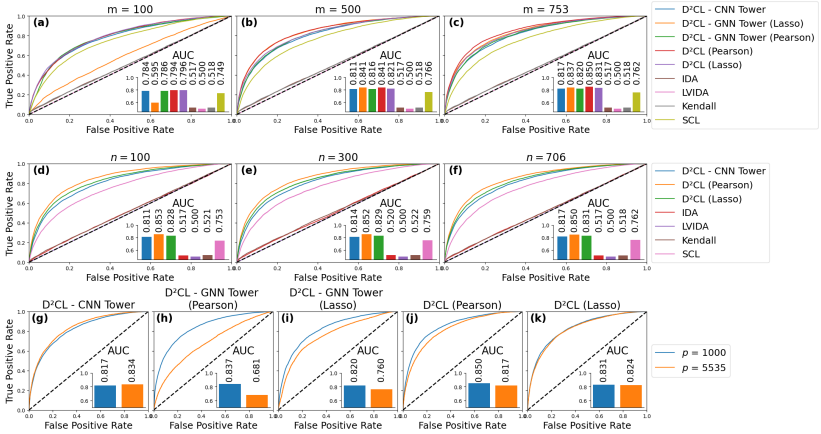


Figure 2.5: Results - yeast gene deletion experiments. Causal learning methods, including D^{2}CL, were applied to gene expression measurements from yeast cells. Performance was quantified using causal ROC curves (and the area under the curves, or AUC) computed with respect to a causal ground truth obtained from entirely unseen interventional experiments (see Text for details). Panels (a)–(c): the number of interventions whose effects are available to the learner is varied as shown (with problem dimension fixed to p=1000 and sample size to n=706). Panels (d)–(f): the sample size n of the data matrix X is varied as shown (with problem dimension fixed to p=1000 and number of available interventions fixed to to m=753). Panels (g)–(k): analogous results for a higher-dimensional setting covering all available genes (roughly the full yeast genome) with p=5535 (with n=706 and m=753). Here, only D^{2}CL variants are shown, as the other methods could not be run due to the computational burden in this higher dimensional case. Comparison with the corresponding p=1000 case demonstrates the scalability of D^{2}CL, with performance broadly maintained in the higher dimensional setting. [D^{2}CL variants shown include a CNN tower alone, GNN tower alone and the entire D^{2}CL architecture; methods compared against include IDA, LVIDA, Kendall correlations (as a non-causal baseline) and SCL (see text and SI for details and references). For D^{2}CL and its variants two different initial graph estimates were used based respectively on Pearson correlation coefficients ("Pearson") and on a lightweight regression ("Lasso"; see Text for details).]

sign. The input prior knowledge Π is derived from the causal status, but, as in all experiments, is strictly disjoint with respect to any test edges.

Results are shown in Figure 2.5a-c, including the area under the ROC curve (AUC; computed with respect to an experimentally-determined gold-standard). Overall, the proposed methods perform well, achieving good results in this high-dimensional, limited data regime. Interestingly, the two towers differ in some ways: the CNN tower degrades slowly with fewer causal inputs while the performance of the GNN tower degrades faster. GIES [9] was not effective in this setting (result not shown; findings are in line with [12] using the same data); however, we note that GIES requires different inputs to our approach and its assumptions are likely violated in this setting. Next, to shed light on data efficiency we varied the sample size n of the data matrix X. Results are shown in Figure 2.5d-f.

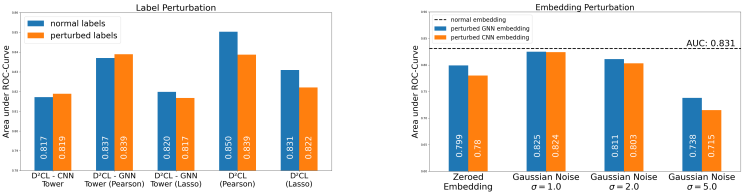
Finally, we tested performance in a higher dimensional example spanning all p=5535 available genes (cf. Figure 2.5g-k) and found that D^{2}CL remains effective at genome scale. Interestingly, while the CNN tower performs particularly well, the GNN tower degrades more. This may be because larger p leads to a larger number of variable pairs (which is helpful for the CNN), but also to a (rapid) increase in the number of nodes and edges in the GNN subgraphs and hence a harder GNN learning task in practice.

2.12.3 Performance under perturbations

D^{2}CL leverages prior causal knowledge; however, in practice, available causal inputs Π may be *incorrect*, e.g. due to flawed initial experiments or errors in the known science. To study sensitivity to flawed causal inputs we introduced errors into Π. This was done by perturbing 10% of the inputs (i.e. labeling causal pairs as non-causal and vice versa) at the outset. Figure 2.6a shows corresponding results; this confirms that the networks are reasonably robust in this sense. These experiments reveal also a benefit of the dual network variants: when one tower underperforms, the combined network still performs well, as it (automatically) adapts to rely on the effective tower, which is further investigated in Figure 2.6b. In general, the embedding of either tower is modified right before the fusion layer to test the impact of a failing tower on the overall performance. The experiment contains four different modifications: (i) We set the complete embedding of one tower to zero and hence effectively remove all information from this tower. In the other cases we apply Gaussian noise with magnitude (ii) $\sigma = 1.0$, (iii) $\sigma = 2.0$, and (iv) $\sigma = 5.0$. The results support the notion that even when one tower fails, the second can compensate so that D^{2}CL still provides useful output.

2.12.4 Identifying causal direction

Causal relations are in general directed and asymmetric, hence it is interesting to explore this behavior with respect to causal direction. Given an image representation,



(a) Label Perturbation **(b)** Embedding Perturbation

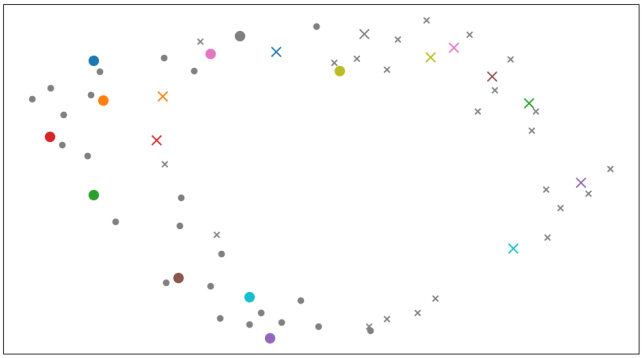
Figure 2.6: Sensitivity to incorrect causal inputs and additional results on causal direction. (a) Robustness to incorrect causal inputs. Sensitivity of D^{2}CL to errors in prior/input causal knowledge Π was studied by artificially introducing errors into Π, with 10% of inputs corrupted. Results quantified via causal AUC shown for several D^{2}CL variants. (b) An ablation-like study in which failures of either the CNN (orange) or the GNN (blue) tower within D^{2}CL are artificially introduced. The affected embedding is either set to zero or a zero-mean Gaussian noise with varying scale is applied. The unaffected case is given as dashed black line.

the CNN tower extracts feature maps that are unique for (ordered) node pairs. The two-dimensional convolutional operation $S(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$ that convolves image I with kernel K would produce the same feature map for two causal images $I_{k\to l}$ and $I_{l\to k}$ if and only if $I_{k\to l}$ and $I_{l\to k}$ were identical. In other words, unless the probability distribution $P(X_i, X_j)$ is perfectly symmetrical around the center of the causal image, the CNN tower extracts causal features that differ depending on direction.

To empirically study this behavior, we constructed additional test data as follows: for each truly causal edge $k \to l$ in the test set, we also included the *reverse* direction $l \to k$. This means that any learner estimating *undirected* links would have an AUC score of 0.5 (since the output $k \to l$ entails also $l \to k$, one of which is a false positive).Table 2.7b shows that D^{2}CL is indeed capable of accurately identifying causal direction. In addition, Figure 2.7a shows a low-dimensional representation of the feature maps of the converged CNN tower. These feature maps differ by causal direction ($k \to l$ vs. $l \to k$) throughout the forward pass, supporting the foregoing arguments.

2.12.5 High-dimensional human CRISPR-based data.

Finally, we used recent, single-cell CRISPR-based interventional experiments [55] to illustrate the use of the proposed approaches in very high-dimensional human data. The experimental protocol (see [55] for full details) includes interventions on a large number of gene targets for two different human cell lines: a cancer (leukemia) cell line (K562) and a (non-cancer) cell line (retinal pigment epithelial or RPE cells). The K562 and RPE experiments include gene expression levels for a total of, respectively,



(a) Low-dimensional representation of feature maps of CNN tower

SNR	Linear	MLP(tanh)	MLP (leaky ReLU)	Tanh	Leaky ReLU	Polynom 3
10.00	0.983	0.711	0.787	1.000	0.994	0.929
6.00	0.981	0.659	0.806	1.000	0.990	0.920
4.00	0.989	0.648	0.802	1.000	0.989	0.927
2.00	0.988	0.721	0.777	1.000	0.992	0.929
1.00	0.987	0.668	0.798	0.999	0.990	0.924
0.75	0.997	0.618	0.765	0.998	0.993	0.926
0.50	0.990	0.669	0.784	0.994	0.991	0.920
0.25	0.991	0.683	0.721	0.962	0.991	0.905
0.10	0.985	0.674	0.790	0.941	0.992	0.915

(b) Causal Directions: AUC values for test sets containing edges $A \to B$ and $B \to A$.

Figure 2.7: Causal direction analysis (see text for details). (a) Low-dimensional representations of latent feature maps of the converged CNN tower. Edges $A \to B$ shown as dots and *reverse* edges $B \to A$ as x-shaped markers. An edge and its corresponding reverse is indicated by the same color. For improved readability, only ten random pairs are highlighted in colors and bigger markers. We see that the embedding is not invariant with respect to causal direction and able to effectively identify the correct direction. [The different D^2CL variants include: a CNN tower alone; a GNN tower for two different initial graph estimates; and the complete network for the same two initial graph estimates. Initial graph estimates for the GNN and combined models were either based on Pearson correlation coefficients ("Pearson") or a lightweight regression ("Lasso"; see Text for details).] (b) AUC values of causal direction tests for different test sets and signal to noise ratios (SNR).

p=8552 and p=8833 genes (see Sec. 2.10 for details). This is a highly challenging setting, due to the complexity of regulatory events in these human cells and high levels of variability and noise in the recently developed single-cell CRISPR protocols. Results are shown in Figure 2.8. Results for K562 and RPE cells (each with more than 8000 genes in the data) show good performance for RPE, and slightly worse performance but still nontrivial consistency with the experimental gold-standard for K562.

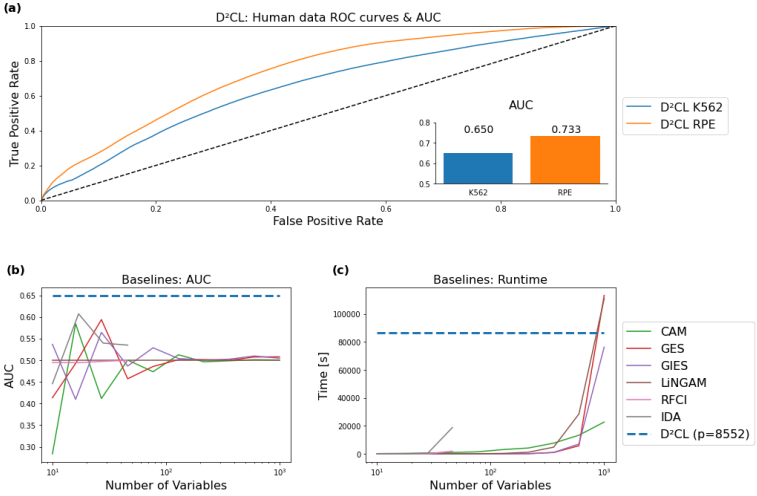


Figure 2.8: Results - high-dimensional human data. Single-cell CRISPR-based experiments (due to [55]) were used to illustrate the use of the proposed approaches in high-dimensional human data. Performance was quantified using causal ROC curves (and the area under the curve, or AUC) computed with respect to a causal ground truth obtained from entirely unseen interventional experiments. (a) Results from D^2CL applied to data obtained from retinal pigment epithelial cells (RPE) and a cancer cell line (K562) in problems spanning more than 8000 variables (other methods could not be practically run in this case due to computational burden). (b) Performance of existing causal learning approaches (on K562 data) as a function of problem dimension. The dashed line indicates D^2CL performance on the *full* problem (p=8552 variables). We see that for the existing methods tested, performance converges to the level of random guessing once more than ∼100 variables are considered. (c) Computational runtime as a function of problem dimension. (Runtime of D^2CL on the full problem with p=8552 including full training and inference is shown as a dashed line; shown only for completeness).

Additional plots in Figure 2.8 show performance and the runtime for a set of baseline algorithms (CAM, GES, GIES, LiNGAM, RFCI, and IDA). These results demonstrate two key points. First, that the runtime for these existing algorithms grows so rapidly with increasing number of variables as to render them unsuitable for problems at this scale. Second, in terms of performance, all the tested methods are considerably less effective than D^{2}CL and in fact, for these data, converge to the performance level of random guessing once more than $\sim$100 variables are considered.

2.13 Conclusions

Emerging experimental protocols, involving combinations of perturbations and high-dimensional readouts, are allowing for new, scalable ways to query molecular networks in a context-specific fashion. Combined with scalable causal learning tools, these approaches have the potential to strongly impact disease biology by allowing global networks, spanning thousands or tens of thousands of variables, to be investigated across many different contexts. We propose an innovative method for inferring causal graph structures. Our approach leverages supervised learning from limited data from a single system and prior causal knowledge to determine the causal status of the remaining edges. Our unique two-tower architecture builds on directly examining statistical characteristics within bivariate distributions and analyzing structural graph features through subgraph processing. During inference, it generates probability distributions for queried edges and estimates sequentially the entire graph. Through extensive experiments, we demonstrate that our approach achieves high accuracy, efficiency, robustness, and scalability when applied to both simulated and real-world datasets. Furthermore, we illustrate that our model generalizes well in out-of-distribution tests and consistently outperforms other comparison methods across various conditions.

Networks learned in this way could, in future, be leveraged to allow for prediction of disease phenotypes or drug response under novel perturbations (this is a different task from standard supervised learning since the test case involves an unseen perturbation to the system). Furthermore, in the area of personalized medicine, such an approach could even allow for rational optimization over potential therapeutic strategies, since the latter are often interventions targeted at molecular nodes, e.g., genes and proteins.

Our model leverages deep learning tools to learn causal relationships between variables at large scale. However, and in contrast to well established approaches based on causal graphical models, it provides only structural output rather than a probability model of the underlying system. It would therefore be interesting to consider coupling our approach, as a first learning step, with a graphical model based analysis in a second step. This would amount to using the flexible and scalable discriminative approach as a filter to render subsequent causal modeling more tractable. It is also interesting to contrast D^{2}CL with the recently proposed CSIvA [33]. Both approaches pursue in a sense a “direct" mapping of data inputs to graph outputs,

with a key difference being that CSIvA uses meta-learning and seeks to generalize across systems while D^2CL uses supervised learning to generalize to new interventions on a given system (for example a biological system of interest). An interesting direction for future work may be to combine both approaches, e.g. by using CSIvA to provide input to D^2CL's GNN tower: this would allow the combined learner to leverage both the general patterns discovered by meta-learning and the data efficient, system-specific approach of D^2CL.

At present, rigorous theory and theoretical properties of the kind of approach studied here remain to be understood. A key direction for future theoretical work will be to understand precise conditions on the underlying system needed to ensure that direct mapping or classification-type approaches can guarantee recovery of specific causal structures. An interesting observation is that the proposed approach may benefit from a "blessing of dimensionality", since the learning problem will typically enjoy a larger number of examples as the dimension p grows. Conversely, and in contrast to established statistical-causal models, our approach (at the current stage) *cannot* be used in the small-p regime, since then the number of examples will be too small for deep learning.

3 Learning Latent Dynamical Models

The world around us is filled with dynamic systems that evolve and change over time. From the weather patterns that shape our climate to the intricate dynamics of financial markets, understanding and predicting the behavior of such systems is a fundamental challenge and of utmost importance for numerous domains ranging from bio-medicine to traditional engineering applications. In recent years, learning latent dynamical models has emerged as a powerful tool for unraveling the hidden dynamics within these complex systems. To begin, let us explore what a latent dynamical model entails. In essence, it is a mathematical framework that aims to capture the hidden dynamics underlying a system's behavior. It provides us with a powerful tool to extract meaningful insights and predictions from complex, time-varying data. By uncovering the latent variables driving the observed phenomena, we gain a deeper understanding of the underlying processes at play. Nonetheless, when it comes to numerous research inquiries, scientists often find themselves limited to data that does not directly align with the variables of the actual system, merely reflecting its progression. Consider, for instance, a scenario where a camera observes a pendulum in motion. The recorded image data encompasses the temporal development of the dynamic system, albeit solely within the observational state realm. In the absence of prior knowledge or additional analysis, it becomes impossible to discern that the governing ordinary differential equation is actually defined by the pendulum's angle and angular velocity.

For these problems, the beauty of latent dynamical models lies in their ability to handle uncertainty and infer hidden states. They enable us to deal with incomplete or noisy data and make accurate predictions even in the face of sparse longitudinal snapshots. This is achieved through a combination of probabilistic modeling techniques, machine learning algorithms, and sophisticated inference methods. The latent variables in these models serve as a bridge between the observed data and the underlying dynamics. They encode valuable information about the system's behavior that may not be immediately apparent from the raw data. By extracting and leveraging this hidden knowledge, latent dynamical models have the potential to deepen our comprehension of the system and empower us to make well-informed choices, particularly in situations where theoretical principles and observations fail to align in terms of physical connections. In fact, the ability to extract knowledge solely from longitudinal observations, without any preexisting knowledge, presents a revolutionary advancement for such problems wherein our access to the underlying system's variable space is either limited or nonexistent. Consequently, it becomes crucial to possess automated scientific discovery tools that condense raw sensory per-

ceptions into a concise collection of state variables and their interdependencies.

The applications of latent dynamical models are vast and diverse. In the field of robotics, these models have proven invaluable for motion planning and control, allowing robots to navigate and interact with their environment effectively. In finance, they have been employed to forecast market trends, manage risk, and detect anomalies. In healthcare, latent dynamical models have helped in understanding disease progression, personalized medicine, and early diagnosis.

The key challenge of modeling multi-variate time sequences lies in estimating the hidden variables and their interactions from observed data. This necessitates the integration of optimization algorithms, statistical inference techniques, and often, domain expertise. The advent of deep learning has revolutionized this process, with neural network architectures enabling more powerful and flexible modeling capabilities. Novel techniques, such as variational inference, deep generative models, and recurrent neural networks, have pushed the boundaries of what is possible and helped discover unknown relations for traditionally challenging problems.
Hence, learning latent dynamical models has the potential to revolutionize our understanding of complex systems and enhance predictive capabilities. With its interdisciplinary nature, this approach surpasses disciplinary boundaries, finding practical applications across diverse domains. In this Chapter, we delve into the intricacies of learning latent dynamics and present a novel methodology for inferring latent dynamical models using explicit invariance decomposition. Through this approach, we aim to address the challenges associated with understanding and modeling complex systems, paving the way for more accurate predictions and deeper insights.

3.1 Neural Latent Dynamical Models via Invariance Decomposition

For many systems of interest in fields like biology, medicine and engineering, high-dimensional observations can reasonably be thought of as obtained via dynamics operating in a lower dimensional space and this assumption (related to the manifold hypothesis [60]) is a common one in many settings. Machine learning approaches for learning dynamical systems have been an important area of recent research, including in particular neural ordinary differential equations [42] and a wider class of related neural-dynamical models ([61, 62, 63, 64, 65, 66]). These models define layers as differential equations and in that sense incorporate an informative (and often appropriate) inductive bias for physical systems. However, the training of neural ordinary differential equations (NODEs) poses a challenging task, necessitating the application of diverse techniques and assumptions to ensure practical viability. Primarily, these methods suffer from the curse of length, wherein the complexity of the loss function escalates as the observed trajectory of the system increases. Surprisingly, even for moderately long sequences, the landscape of the loss function

can become enigmatically complex. Consequently, conventional gold-standard optimization techniques can readily deviate when employed in NODE-based approaches.

We propose a new framework for learning latent dynamics from observed data and our approach, named "Latent Dynamics via Invariant Decomposition" (LaDID), combines variational autoencoders and spatio-temporal attention within a learning framework motivated by certain scientifically-motivated invariances, but which does not require an explicit ODE formulation (details below). Although the focus of this paper is primarily on methodology, the research we present is driven by real-world applications, particularly in the fields of biomedicine and health. In these domains, it is often the case that explicit dynamical models are not initially accessible. However, we anticipate that the invariances outlined below are nonetheless expected to hold (see also Discussion).

The methods we propose build on two observations concerning classical scientific models:

- First, the notion that every output from a class of scientific systems should be explainable via a single model that operates across all *instances* within this class and is, in that sense, universal. For example, for a class of mechanical systems, the same equations can explain all specific instances (with different initial conditions or constants, say) even when the instances are very different in terms of their respective observations. This is interesting from an ML point of view because the induced distribution shifts between instances can be very large.

- Second, instance/realization-specific factors (such as initial conditions or constants) tend to remain unchanged throughout the entire duration of a given realization. In this regard, these factors exhibit time invariance, as they maintain their values consistently over time.

Our approach and the derived transformer-based architecture builds directly on these notions of invariance. We describe the network in detail below, but in brief the set-up is as follows. From an available trajectory – thought of as representing a specific instance/realization r of a more general model class $\mathcal{M}$ – we learn an encoding ψ^r of the realization-specific information. This is intended to implicitly capture information (such as initial conditions or constants) that are specific to the instance or realization r, but the information should remain valid for all times within a realization/instance; hence the encoding has a superscript indicating the realization but no time index. This encoding ψ^r is treated as an input to a "universal" model f to enable prediction of system output at any time t. The model f itself is learned across multiple realizations r of the system defined by the model class but the same function is always used for prediction (for the system $\mathcal{M}$). In other words, f is intended to be universal (across all queries concerning the system $\mathcal{M}$) with realization-specific information provided only by the input ψ^r. We argue that under certain conditions,

this decomposition into *realization-specific* (RS) and *realization-invariant* (RI) information allows for definition of a simple and convenient learning framework. We propose a deep neural architecture for this purpose, showing how learning of universal latent dynamics and realization-specific information can be done jointly, in an end-to-end manner. This enables prediction of system behavior at *any* continuous time t, for *any* realization r (for which minimal data are available).

We empirically validate our proposed method on spatio-temporal systems with dynamics governed by ordinary or partial differential equations. These systems are ubiquitous in nature and include physical phenomena in rigid body motion, fluid dynamics and turbulent flows, electromagnetism and molecular dynamics. Our work is related to a large body of previous work on neural learning of dynamical models, which we discuss in detail below. A key distinction of our approach is that by leveraging the framework outlined above, we do not require an explicit neural ODE at all; rather we can carry out learning within a simple, broadly supervised framework that, as we show, substantially outperforms existing neural-dynamical models over a range of challenging tasks relative to both, regular *and* irregular time grids.

Thus, our main contributions are:

- We present a novel framework, and associated transformer-based network for the separation of realization-specific information and (realization-invariant) latent dynamical systems.
- We systematically study performance on short- and longer-horizon prediction of a wide range of complex temporal and spatio-temporal problems, comparing against a range of state-of-the-art neural-dynamical baselines.
- We study the challenging case of transfer to data obtained under entirely novel system interventions via a few-shot learning (FSL) approach.

3.2 Related Works

We start by exploring the diverse range of methods that have been proposed for learning dynamical models from data, with a particular emphasis on approaches rooted in machine learning. Non-linearity, as present in a wide array of complex phenomena, poses a significant challenge in the study of dynamical systems. Unfortunately, a comprehensive mathematical framework that explicitly and universally describes nonlinear systems is currently missing. In contrast, linear systems can be fully characterized using spectral decomposition that enables the utilization of generic and computationally efficient algorithms for prediction, estimation, and control. One way to transfer these properties to *nonlinear* dynamical systems roots in the Koopman operator theory of dynamical systems which presents a promising alternative perspective [67, 68]. It suggests that even highly nonlinear dynamics may exhibit a form of linear superposition through the utilization of the infinite-dimensional, yet

linear, Koopman operator. Unlike traditional approaches, the Koopman operator, operates on measurement functions of the system. Its spectral decomposition provides a comprehensive understanding of the behavior of the nonlinear system, akin to the characterization afforded by eigenvalues and eigenvectors in linear systems. [69] introduces a novel approach to the problem of discovering dynamical systems, viewing it through the lens of sparse regression in combination with Koopman theory. This new perspective allows us to identify nonlinear models while striking a balance between model complexity and accuracy. The inherent convex optimization algorithms employed in this method ensure its applicability to large-scale problems. The key insight of this line of research lies in recognizing that, for many systems of interest, the dynamical model can often be expressed using only a small number of functions, rendering it sparse within the space of possible functions. To capture this sparsity, they construct an augmented library that comprises candidate nonlinear functions offering flexibility and various choices. This approach found wide-spread application in the community [70, 71, 72, 73] and was extended to an implicit formulation [74] and neural reformulation [75].

In certain cases, the modeling of dynamical systems necessitates meticulous discretization to accurately capture the underlying phenomenon. However, this reliance on fine discretization can often result in slow and inefficient traditional numerical solvers. To address this challenge, a recent line of research has proposed the use of neural networks to learn mesh-free, infinite-dimensional operators. The introduction of neural operators offers a solution to the dependency on specific meshes by generating a single set of network parameters that can be utilized across different discretizations [76, 77, 78, 79, 80]. This allows for the transfer of solutions between meshes, eliminating the need for retraining. Moreover, neural operators only require training once, and obtaining a new solution merely involves a forward pass through the network. Importantly, the neural operator does not require prior knowledge of the underlying partial differential equation, relying solely on data-driven insights. Tailored neural operators, e.g. Fourier Neural Operator [81, 82], target the resolution-invariant solution of the turbulent regime of the Navier-Stokes equations with applications in e.g. weather forecasting [83].

Within the existing body of research, a significant amount of work has been dedicated to addressing the computationally intensive nature of forward solutions by leveraging the assumption of a known parametric form of a differential equation [84, 85, 86]. This approach involves matching empirical gradients $\dot{x} = f(x)$, thereby circumventing the need for costly integration steps. In more recent investigations, the focus has shifted towards estimating an unknown, non-parametric differential equation using Gaussian processes [87, 88, 89, 90, 91]. This novel approach offers the potential to uncover the underlying dynamics without relying on pre-defined parameterizations, opening up new avenues for modeling and analysis.

In dynamical systems it is typically not feasible to have a reliable analytical model

of the underlying processes. In such cases, a more general approach is to learn and capture the latent dynamics of the data using an architecture that incorporates an appropriate inductive bias. Hamiltonian and Lagrangian mechanics offer distinct mathematical reformulations of Newton's equations of motion, specifically for energy-conservative dynamics. The conservation of energy in these systems enables predictions of the system's state over significantly longer time horizons, both forward and backward, compared to the training data. This property makes them attractive biases to incorporate into deep neural networks. Despite the different coordinate frames they employ (Hamiltonian using position q and momentum p, and Lagrangian using position q and velocity $\dot{q}$), both formalisms describe the same underlying dynamics, allowing for seamless translation between the two without any loss of generality. One advantage of these models is that they only need to infer the Hamiltonian or the Lagrangian, without the additional burden of learning a state representation. This also simplifies evaluation, as it requires calculating the distance between the ground truth states and the states predicted by the model along the trajectory. To tackle this challenge, several approaches have been proposed that augment physics-inspired models with encoder/decoder modules [92, 93, 94, 95, 96, 97]. These modules facilitate the inference of low-dimensional states from high-dimensional pixel observations bridging the gap between visual input and the underlying dynamics.

Adopting a continuous dynamical system approach facilitates the combination of ideas from machine learning and physical modeling. The underlying idea emanates from the following theoretical observation: Each residual block in a ResNet can be represented as a step in the forward Euler discretization of an ODE, suggesting a potential connection between discrete dynamic systems and deep networks featuring skip connections [98, 99, 100, 101]. In fact, such networks may be formalized as

$$h_{t+1} = h_t + f(h_t, \sigma_t), \tag{3.1}$$

with h_t denoting the hidden network state at time step t and $f(h_t, \sigma_t)$ is the learned function of the current hidden state information. Hence, deep neural networks can be seen as a discretized version of continuous dynamical systems. Mathematically, working with continuous dynamical systems is often more convenient and manageable. The continuous formulation offers greater flexibility and ease of analysis compared to discrete systems. The application of physics-inspired partial differential equation (PDE) models in image processing has resulted in significant advancements. Some notable contributions include optical flow models for motion estimation [102] or nonlinear diffusion models for image filtering [103] amongst others. In the realm of differential equation based data processing, the data is often treated as discretized representations of multivariate functions. Consequently, various operations on the data can be understood as discretizations of differential equation operators acting on these underlying functions. In this context, the governing idea of neural ODEs suggests to gradually decrease the step size Δt yielding a differential version of the above equation. In other words, the difference $h_{t+1} - h_t$ can be interpreted as a discretization of the derivative $\frac{d}{dt}h(t)$ with time step $\Delta t = 1$ [98, 99, 104]. Letting

$\Delta t \to 0$, eq. 3.1 can be rewritten as

$$lim_{\Delta t \to 0} \frac{h_{t+\Delta t} - h_t}{\Delta t} = \frac{dh(t)}{dt} = f(h(t), t). \tag{3.2}$$

Consequently, the hidden state can be represented by an ODE that maps a data point x to a set of features $\phi(x)$ by solving an Initial Value Problem (IVP) up to a certain time T

$$\frac{dh(t)}{dt} = f(h(t), t) \quad h(0) = x \ . \tag{3.3}$$

The hidden state at time T, i.e. $h(T)$, corresponds to the features learned by the model. That is, in neural ODEs the input x is mapped to an output y by solving an ODE starting from x. Then, the dynamics of the system (encoded by f) is adjusted such that the ODE transforms x to a y which is close to y_{true}.

When employing the Euler integration scheme, setting the step size to $m = 1$ precisely aligns with the update of sequence transformations in neural networks. Consequently, adding an infinite number of layers to the neural network defines the dynamics of the hidden state h as an ordinary differential equation:

$$\frac{d(h(t))}{dt} = f(h(t), t, \theta) \tag{3.4}$$

To address the memory consumption issue associated with solving an ODE in the forward pass, various existing ODE solvers such as Runge-Kutta [105, 106] or DOPRI [107] integration schemes can be employed. However, these ODE solutions come at the expense of high memory usage, typically of the order $\mathcal{O}(\tilde{L})$, where $\tilde{L}$ represents the number of function evaluations involved. In order to mitigate this challenge, [104] propose to utilized the adjoint sensitivity method [108]. The adjoint sensitivity method offers a general approach applicable to all ODE solvers and tackles the problem of memory costs during gradient computation. It achieves this by solving a second, augmented ODE in reverse time. By applying the adjoint sensitivity method, the time complexity scales linearly resulting in constant memory costs. This reduction in memory consumption is highly desirable and addresses a significant concern associated with solving ODEs in neural networks.

A well-known limitations of neural ODEs emanates from the curse of length: as training progresses and the flow becomes increasingly more complex, the number of steps required to solve the ODE increases [104, 109]. One approach to overcome these limitations is through the use of augmented neural ODEs [110]. ANODEs aim to address the challenges by augmenting the space in which the ODE is solved. This augmentation enables the model to leverage the additional dimensions, facilitating the learning of more complex functions while employing simpler flows. As a result, ANODEs offer a notable reduction in the computational cost of both the forward and backward passes of the model when compared to the original neural ODE counterpart.

Another issue of NODEs arises from their continuous modeling idea which does not allow to incorporate discrete events that abruptly change the latent vector. One solution to tackle this drawback are neural jump ODEs (NJ-ODEs) [111]. The authors use a latent vector $z(t)$ to encode the state of a system which flows continuously over time until an event happens at random causing an abrupt jump and change in its trajectory. To model such rare events, its conditional intensity and its influence are parameterized with neural networks as functions of $z(t)$ while the continuous flow is modeled via an original NODE. Concurrent work aiming to adjust the trajectory based on subsequent observations was proposed in [112] combining general characteristics of RNNs with neural ODEs to form neural controlled differential equations (NCDEs). Generally speaking, NCDEs are the continuous analogue of RNNs similar to NODEs being the continuous version of ResNets. Follow-up work in [113] proposed a NCDE extension based on the rough path theory called neural rough differential equations (NRDEs). Instead of directly embedding into the path space, the input signal is represented over small time intervals through its log-signature, which are statistics describing how the signal drives a CDE. The log-signature captures statistical information about how the signal influences a controlled differential equation (CDE). This log-ODE method enables the updating of the hidden state of a non-commutative differential equation (NCDE) over extended intervals, surpassing what would typically be achievable based solely on the sampling rate or data length. Consequently, the effective length of the time series is effectively reduced, as the log-signatures serve as a specific choice of summarization tailored to the CDE.

An alternative approach to address the challenge of long-term trajectory prediction involves applying the concept of multiple shooting, commonly used in differential equations, to neural networks. In this regard, [114] introduced a novel class of implicit neural models called Multiple Shooting Layers (MSLs). These MSLs leverage time-parallel methods for differential equations, enabling them to seek solutions to initial value problems using parallelizable root-finding algorithms [115, 116]. Differentiable MSLs focus on maximizing parallelization by utilizing the interplay between numerical methods for root finding and differential equations. This new class of neural models, MSLs, can often serve as drop-in replacements for Neural ODEs with the advantage of frequently requiring a smaller number of function evaluations. Multiple-shooting methods for differential equations involve a fundamental concept of transforming an initial value problem into a boundary value problem (BVP). The time interval [0, T] is divided into N sub-intervals $[t_n, t_{n+1}]$ with $0 = t_0 < t_1 < \cdots < t_N = T$, each associated with a left boundary subproblem characterized by the following condition:

$$z_n(t_n) = b_n \tag{3.5}$$

$$\frac{z_n(t)}{dt} = f_\theta(t, z_n(t)),\ t \in [t_n, t_{n+1}]\,, \tag{3.6}$$

where b_n represents shooting parameters. The solution of the second subproblem matches the solution of the first subproblem at each time $t \in [0, T]$ if and only if all shooting parameters are identical to $z(t_n)$, i.e. $b_n = \phi_\theta(z_0, t_0, t_n)$. A remarkable

feature of Multiple Shooting Layers is their ability to compute the solutions of all N initial value problems in parallel based on the shooting parameters in the boundary value problem. The research conducted by [117] presents an innovative perspective that expands upon conventional multiple shooting methods. Their approach introduces a probabilistic framework that integrates sparsity into the shooting variables, enabling the effective integration of irregularly sampled time grids and redundant shooting variables. This novel methodology effectively addresses the challenges associated with irregularities and redundancies in a systematic manner. Furthermore, the authors propose an attention-based encoder architecture specifically tailored for latent neural ordinary differential equations. This architecture seamlessly complements the sparse shooting formulation and demonstrates remarkable capabilities in handling high-dimensional data that is noisy and partially observed. Leveraging the power of Bayesian inference, the authors naturally incorporate a continuity prior.

Ordinary differential equations are mathematical equations that describe deterministic systems and are known to define a flow of diffeomorphisms [118]. These systems have predictable behavior, where the evolution of a variable is solely determined by its current state. ODEs involve derivatives with respect to a single independent variable, usually representing time and find extensive application in physics, engineering, and various scientific fields. This allows the modeling of phenomena like classical mechanics or chemical reactions, which exhibit consistent patterns. On the other hand, stochastic differential equations (SDEs) provide a generalization of ODEs by incorporating instantaneous noise into their dynamics. SDEs can be expressed in general by the following integral equation

$$\frac{dh(t)}{dt} = f(h(t), t) + \sigma h(t)\zeta(t), \tag{3.7}$$

where $\zeta(t)$ denotes a random component, commonly represented by a Wiener process or Brownian motion. SDEs are utilized to model dynamic systems affected by random forces or external sources of uncertainty. They are particularly useful when studying phenomena that involve noise, fluctuations, or random processes. The incorporation of randomness in SDEs enables them to capture systems with inherent uncertainty. Recent advancements have enabled efficient training of neural stochastic differential equations , similar to the concept of neural ordinary differential equations. To compute gradients through a neural SDE [119, 120] proposed to employ the pathway approach [121] to simulate the forward dynamics of an explicit Jacobian matrix while [122] presented a stochastic version of the adjoint sensitivity method which is computationally cheaper. SDEs and their neural equivalent find widespread usage in fields such as finance [123, 124, 125], biology [126], and physics [127, 128], where they are well-suited for modeling phenomena governed by small, unobserved interactions. However, as these models assume distinct characteristics such as deterministic transitions versus stochastic transitions we will confine our experimental comparisons to (neural) ODE-based methods.

Our approach is inspired by this body of work in that we also use neural networks to learn latent dynamical models. However, two key differences are as follows. First, our models are specifically designed to exploit certain invariances that are important in classical scientific models. From this point of view, we leverage a particular kind of scientifically-motivated inductive bias from the outset. Second, exploiting these invariances allows us to eschew explicit neural ODEs altogether, providing an arguably simpler, transformer-based scheme that can be trained in a straightforward fashion, but that, as we show, achieves excellent performance on unseen data from complex dynamical systems and that can even be leveraged for few-shot learning to generalize to nontrivial system interventions.

3.3 Method

Initially, we present a problem statement, followed by a discussion of several fundamental arguments. These arguments aim to demonstrate how a relatively straightforward learning framework can be applied within this particular scenario. While these arguments are simplified for the purpose of clarity, they are nonetheless pertinent in providing a conceptual foundation and explaining why learning is feasible within this context. We then put forward a specific architecture to permit learning in practice, i.e. a concrete learning framework that we implement and study empirically.

3.3.1 Problem statement

We focus on settings in which we capture observations of a system of interest at irregularly spaced time points $t \in T$ forming an individual realization trajectory. Here, we employ the term "realization" to emphasize that our framework is not confined to Newtonian or Hamiltonian mechanics. It also accommodates longitudinal observations of intricate systems, e.g. complex genotype-phenotype relationships or others.
Let $X \in \mathbb{R}^{T \times C \times H \times W}$ represent a high-dimensional trajectory in the observational space. Here, T indicates the number of time steps, and C, H, and W correspond to channels, frame height, and frame width respectively. Our objective is to predict future observations $\hat{X}_{t_q}$ at queried future time points $t_q > T$. For realizations that were not represented in the training data at all, we assume availability of some data specific to the realization at inference-time. With initial observations X at hand, we strive to develop a model g that captures observed dynamics in a lower-dimensional latent space. This model should enable accurate future predictions of the system at any continuous time t, i.e. $\hat{X}_{t_q} = g(X, t_q)$. The uniqueness of our model lies in its innovative design, which enforces a novel latent structure separating information into "realization-specific" and "realization-invariant" components.

Analogy to traditional ODE formulations.

To facilitate a more intuitive understanding of our framework, we would like to draw some analogies to standard ODE problems. Typical ODE solvers comprise a specifi-

cally formulated ODE function and some initial value (IV) which are evolved over time using well-known integration methods, e.g, Euler, Runge-Kutta, or DOPRI schemes. From a high-level perspective, the IVs relate to our RS representation while the ODE functions and its corresponding integration approach is our RI part. However, please note that this analogy only holds on a superficial level as two fundamental differences exist to our framework: First, our framework purely relies on *observations* of a specific system, for which underlying state variables required to apply standard ODE solver are not known (and in fact neither observed). Hence, our RS representation requires access to a collection of consecutive high-dimensional observations as a single sample does not provide sufficient information to derive a unique solution for its temporal evolution. Second, our RI function is continuous in time and therefore unites temporal integration and dynamics function on an abstract level. To do so, we condition our RI model on specific RS representations and only query time points from it to obtain a discretized latent trajectory.

In Section B.1, we demonstrate that, under reasonable assumptions, our model driven by invariances can generally yield accurate predictions. Importantly, this holds true even without prior knowledge of the actual dynamical system or its accurate latent variables. The underlying theoretical proof was done by Sach Mukherjee and is beyond the scope of this thesis, however, very important for a deeper understanding of our approach.

3.3.2 Neural architecture

Based on the initial thoughts and theoretical motivations in Section B.1, we now put forward a specific architecture leveraging an amortized variational inference model.

3.3.2.1 Model, inference and forecasting

Model. Overall, our implementation is composed of three primary components: the encoder $f_{\phi_{enc}}$, the invariant dynamical system $f_{\phi_{dyn}}$, and the decoder $f_{\phi_{dec}}$, all of which are characterized by parameters ϕ_{enc}, ϕ_{dyn}, and ϕ_{dec} respectively. The encoder is a collection of three NNs, i.e. a CNN processing spatial information in the observation space, a transformer utilizing temporal attention and a learnable mapping function. Since we want to predict future observations based on a few observations, we only use the first K data points in time and process these in a shared convolutional encoder (green trapezoid in Fig. 3.1 (ii)). We employ a shallow CNN that compresses the input to 1/16 of the initial input size using four ReLU activated and batch-normalized convolutional layers. The resulting tensors are then flattened and mapped linearly to a single vector. Next, we use a transformer on the K output vectors of the convolutional encoder, applying temporal attention to reweigh vectors. We tested two approaches [129, 117] with comparable performance which are discussed in the Appendix in more detail. For each of the $k \in K$ time aware representations ρ_k^{TA}, we sample a latent embedding using the reparameterization trick, i.e. $l_k^{emb} \sim \mathcal{N}(f_\mu(\rho_k^{TA}), f_\sigma(\rho_k^{TA}))$. The final trajectory representation ψ^r is the out-

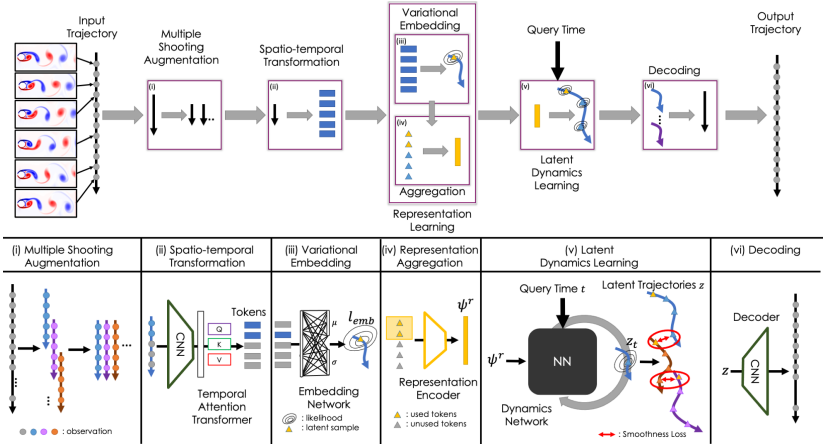


Figure 3.1: Architecture: Learning of Latent Dynamics via Invariant Decomposition (LaDID). Top half: general flow chart of the compute steps (i)-(v) of LaDID; lower half: more detailed representation of the compute steps. A set of high-dimensional snapshots of a system on a regular or irregular time-grid serves as the empirical input to LaDID. The trajectory is split into subpatches using Multiple Shooting Augmentation (i). The first time-points of each subpatch are used to compute a subtrajectory representation: features of the selected snapshots are re-weighted w.r.t. time and spatial location (ii), transformed to low-dimensional variational embedding (iii), and aggregated into one trajectory representation ψ_r (iv). During inference, the latent dynamical model is conditioned on the specific representation ψ_r. Prediction is possible at any continuous time by querying the latent state of any time point of interest (v). Latent subtrajectories are sewn together by a smoothness loss. Finally, the entire latent trajectory is decoded to the observation space (vi).

put of an aggregation over all K tokens. In our implementation, we choose a simple yet effective mean-aggregation which can be changed based on the task at hand.
The second important part of our proposed framework is the dynamical model $f_{\phi_{dyn}}$. We utilized a three layer MLP which can also be interchanged by other functions. To obtain a latent trajectory, we condition the latent dynamical model to our end-to-end learned trajectory representation ψ^r and roll-out the latent trajectory z based on the queried time points t_q represented through a time encoding which we choose as a set of different sine and cosine waves with different wave length.
Finally, we map all data points of our latent trajectory back to the original observation space. Our decoder module $f_{\phi_{dec}}$ is kept very simple consisting of four deconvolutional layers.

The key novelty of our approach lies in the unique structure of the latent space mimicking the interplay of realization-specific information in initial/boundary condi-

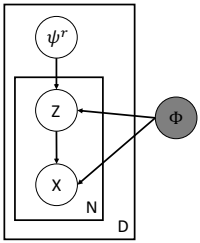


Figure 3.2: Graphical model: The graph consists of three random variables, the trajectory representation ψ^r, the latent states z and the observations x. Fixed parameters ϕ are represented by the gray node.

tions and a realization-invariant dynamical model similar to the frame of differential equations. However, we can significantly reduce computational costs as we are not forced to solve explicitly any differential equation since we rely on an unsupervised end-to-end learning scheme.

Generative model, inference and optimization. We now turn the descriptive technical context of our method to a probabilistic model. Our graphical model (see Fig. 3.2) consists of (trainable) parameters $\Phi = \phi_{enc} \cup \phi_{dyn} \cup \phi_{dec}$, a random variable ψ^r which additionally acts as global random variable at the level of latent states z_{t_q} and observations x_{t_q}. The index t_q refers to a specific queried time point within a trajectory. The joint distribution is given by

$$p(x, z, \psi^r) = p(x, z|\psi^r)p(\psi^r) = p(x|z)p(z|\psi^r)p(\psi^r). \tag{3.8}$$

Our graphical model assumes these independencies: (i) The dataset contains i.i.d. trajectories of varying length. (ii) The observation of trajectory $x^r_{t_q}$ at time t_q is conditionally independent of $x^r_{t_{q-1}}$ at time t_{q-1}, given latent states $z^r_{t_q}$ and trajectory representation ψ^r: $p(x_{t_q}|z_{t_q}, \psi^r) \perp\!\!\!\perp p(x_{t_{q-1}}|z_{t_{q-1}}, \psi^r)$. Analyzing data with this graphical model involves computing posterior distributions of hidden variables given observations

$$p(z, \psi^r|x) = \frac{p(x, z, \psi^r)}{\int p(x|z)p(z|\psi^r)p(\psi^r)dzd\psi^r}. \tag{3.9}$$

To effectively process long-horizon time series data, we apply a variant of *multiple shooting*. However, since our model does not rely on an explicit ODE formulation, we are not concerned with turning an initial value problem into a boundary value problem [114]. Instead, we incorporate a Bayesian continuity prior [91, 117] to extend the multiple-shooting framework from deterministic neural ODEs to a probabilistic context. Our approach dissects each realization $x^r_{t:T}$ into a series of N overlapping

subtrajectories and independently condenses each patch into a latent representation. Within this Bayesian multiple shooting framework, the smoothness prior connects the patches via

$$p(z|\psi^r) = \prod_{i=1}^{N} p(z_n|\psi_n^r)p(z_n|z_{n-1}, \psi_{n-1}^r) \tag{3.10}$$

to form a cohesive global trajectory. We leverage the independence of trajectory representations in subpatches i.e. $p(z_i|\psi_i^r) \perp\!\!\!\perp p(z_j|\psi_j^r)$. For the continuity prior, we follow [91] and place a Gaussian prior on the error between consecutive subtrajectories, i.e. $\Delta \sim \mathcal{N}(0, \sigma_\Delta)$ entailing exact overlapping if $\Delta \to 0$. This yields our continuity prior

$$p(z_n|z_{n-1}, \psi_{n-1}^r) = \mathcal{N}((z_n^{t_1}|z_{n-1}^{-t}, \psi_{n-1}^r), \sigma_\Delta), \tag{3.11}$$

where the time index $-t$ refers to the last time point of a subpatch. The prior trajectory representation is set to a Gaussian, i.e. $p(\psi^r) \sim \mathcal{N}(0, 1)$. With the priors introduced above, we get the following generative model (we drop the subpatch index n for improved readability):

$$p(l_K^{emb}|x) = \mathcal{N}(f_{\phi_{enc},\mu}(x_K), f_{\phi_{enc},\sigma}(x_K)) \tag{3.12}$$

$$p(\psi^r|x) = f_{agg}(l_K^{emb}) \tag{3.13}$$

$$p(z|\psi^r, x) = f_{\phi_{dyn}}(\psi^r, t_q) \tag{3.14}$$

$$p(x|z) = \mathcal{N}(f_{\phi_{dec}}(z), \sigma_{dec}) \tag{3.15}$$

In the context of inference, we opt for Gaussian distributions as our variational approximations and set $\sigma_{dec} = 10^{-2}$. We then work to minimize the Kullback-Leibler divergence $\mathrm{KL}[q(z, \psi^r)||p(z, \psi^r|x)]$ and derive the ensuing *evidence lower bound*

$$\ln p(x) = \ln \int p(x, z, \psi^r) \frac{q(z, \psi^r)}{q(z, \psi^r)} dz d\psi^r \tag{3.16}$$

$$\geq \int q(z, \psi^r) \ln \frac{p(x, z, \psi^r)}{q(z, \psi^r)} dz d\psi^r \tag{3.17}$$

$$= \int q(z, \psi^r) \ln \frac{p(x|z)p(z|\psi^r)p(\psi^r)}{q(z)q(\psi^r)} dz d\psi^r \tag{3.18}$$

$$= \int q(z, \psi^r) \ln \frac{p(x|z)p(\psi^r) \prod_{n=1}^{N} p(z_n|\psi_n^r)p(z_n|z_{n-1}, \psi_{n-1}^r)}{q(z)q(\psi^r)} dz d\psi^r \tag{3.19}$$

$$\begin{aligned} = &\int q(z, \psi^r) \ln p(x|z) \prod_{n=1}^{N} p(z_n|\psi_n^r) dz d\psi^r + \int q(z, \psi^r) \ln \frac{p(\psi^r)}{q(\psi^r)} dz d\psi^r \\ &+ \int q(z, \psi^r) \ln \frac{p(z_1|z_0, \psi_0^r) \prod_{n=2}^{N} p(z_n|z_{n-1}\psi_{n-1}^r)}{q(z_n)} dz d\psi^r \end{aligned} \tag{3.20}$$

This is equivalent to maximizing the following loss function with short term notation $p_n(\hat{x}_n) = p(x_n|z_n)p(z_n|\psi_n^r)$:

$$\begin{aligned} \max \quad & \underbrace{\mathbb{E}_{q(z,\psi^r)} \sum_{n=1}^{N} \ln p_n(\hat{x}_n)}_{\text{(i) likelihood}} - \underbrace{\sum_{n=1}^{N} \mathrm{KL}(q(\psi^r)||p(\psi^r|x))}_{\text{(ii) representation prior}} \\ & \underbrace{- \sum_{n=2}^{N} \mathbb{E}_{q(z,\psi^r)} \mathrm{KL}(q(z_n)||p(z_n|z_{n-1}, \psi_n^r))}_{\text{(iii) smoothness prior}} \end{aligned} \tag{3.21}$$

3.4 Datasets

To study the capabilities of LaDID relative to existing models, we consider a large range of physical systems ranging from rather simple ODE-based datasets to complex turbulence driven fluid flows. Specifically, we evaluate LaDID on high-dimensional observations (p=16384) of a nonlinear swinging pendulum, the chaotic motion of a swinging double pendulum, and realistic simulations of the two-dimensional wave equation, a lambda-omega reaction-diffusion system, the two-dimensional incompressible Navier-Stokes equations, and the fluid flow around a blunt body solved via the latticed Boltzmann equations. This extensive range of applications covers datasets which are frequently used in literature for dynamical modeling and therefore enable fair comparisons to state-of-the-art baselines and at the same time

study effectiveness in the context of complex datasets relevant to real-world use-cases. To this end, we evaluate LaDID on regular and irregular time grids and further transfer a learned model prior to a completely unknown setting obtained by *intervention* on the system. That is, we generate small datasets on intervened dynamical systems (either via modifying the underlying systems, for example by changing the gravitational constant or the mass of a pendulum, or via augmenting the realization-specific observation, e.g. by changing the length of the pendulum or the location of the simulated cylinder) and fine-tune a pre-trained model on a fraction of the initially seen training data.

3.4.1 Swinging pendulum

For the first dataset we consider synthetic videos of a nonlinear pendulum simulated in two spatial dimensions. Typically, a nonlinear swinging pendulum is governed by the following second order differential equation:

$$\frac{d^2 z}{dt^2} = -\sin z \tag{3.22}$$

with z denoting the angle of the pendulum. Overall, we simulated 500 trajectories with different initial conditions. For each trajectory, the initial angle z and its angular velocity $\frac{dz}{dt}$ is sampled uniformly from $z \sim \mathcal{U}(0, 2\pi)$ and $\frac{dz}{dt} \sim \mathcal{U}(-\pi/2, \pi/2)$. All trajectories are simulated for $t = 3$ seconds. The training, validation and test dataset is split into 400, 50 and 50 trajectories, respectively. The swinging pendulum is rendered in black/white image space over 128 pixels for each spatial dimension. Hence, each observation is a high-dimensional image representation (16384 dimensions - flatted $128 \times 128\,\mathrm{px}^2$ image) of an instantaneous state of the second-order ODE.

3.4.2 Swinging double pendulum

To increase the complexity of the second dataset, we selected the kinematics of a nonlinear double pendulum motion. The pendulums are treated as two point masses with the upper pendulum being denoted by the subscript "1" and the lower one by subscript "2". The kinematics of this nearly chaotic system is governed by the following set of ordinary differential equations:

$$\frac{d^2 z_1}{dt^2} = \frac{\left[\begin{array}{l} -g(2m_1 + m_2)\sin z_1 - m_2 g \sin(z_1 - 2z_2) \\ \quad -2\sin(z_1 - z_2) m_2 (\frac{dz_1}{dt}^2 L_2 + \frac{dz_1}{dt}^2 L_1 \cos(z_1 - z_2)) \end{array}\right]}{L_1(2m_1 + m_2 - m_2\cos(2z_1 - 2z_2))} \tag{3.23}$$

$$\frac{d^2 z_2}{dt^2} = \frac{\left[\begin{array}{l} 2\sin(z_1 - z_2)(\frac{dz_1}{dt}^2 L_1(m_1 + m_2) \\ \quad + g(m_1 + m_2)\cos z_1 + \frac{dz_2}{dt}^2 L_2 m_2 \cos(z_1 - z_2)) \end{array}\right]}{L_2(2m_1 + m_2 - m_2\cos(2z_1 - 2z_2))} \tag{3.24}$$

with m_i denoting the mass and the length of each pendulum respectively, and g is the gravitational constant. Again, we simulated 500 trajectories split in sets of 400, 50 and 50 samples for training, validation and testing. The initial condition for (z_1, z_2) and $(\frac{dz_1}{dt}, \frac{dz_2}{dt})$ are uniformly sampled in the range $\mathcal{U}(0, 2\pi)$ and $\mathcal{U}(-\pi/2, \pi/2)$. The double pendulum is rendered in a RGB color space over 128 pixels for each spatial dimension with the first pendulum colored in red and the second one in green. Hence, each observation is a high-dimensional image representation (16384×3 dimensions - flatted $128 \times 128\,\mathrm{px}^2$ RGB image) of an instantaneous double pendulum state.

3.4.3 Reaction-diffusion equation

Many real-world applications of interest originate from dynamics governed by partial differential equations with more complex interactions between spatial and temporal dynamics. One such set of PDEs we selected as test case is based on a lambda-omega reaction-diffusion system which is described by the following equations:

$$\frac{du}{dt} = (1 - (u^2 + v^2))u + \beta(u^2 + v^2)v + d_1(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}) \tag{3.25}$$

$$\frac{dv}{dt} = -\beta(u^2 + v^2)u + (1 - (u^2 + v^2))v + d_2(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}) \tag{3.26}$$

with $(d_1, d_2) = 0.1$ denoting diffusion constants and $\beta = 1$. This set of equations generates a spiral wave formation which can be approximated by two oscillating spiral modes. The system is simulated from a single initial condition from $t = 0$ to $t = 10$ in $\Delta t = 0.05$ time intervals for a total number of 10 000 samples. The initial conditions is defined as

$$u(x, y, 0) = \tanh\left(\sqrt{x^2 + y^2}\cos\left((x + iy) - \sqrt{x^2 - y^2}\right)\right) \tag{3.27}$$

$$v(x, y, 0) = \tanh\left(\sqrt{x^2 + y^2}\sin\left((x + iy) - \sqrt{x^2 - y^2}\right)\right). \tag{3.28}$$

The simulation is performed over a spatial domain of ($x \in [-10, 10]$ and $y \in [-10, 10]$ on grid with 128 points in each spatial dimension. We split this simulation into trajectories of 50 consecutive samples resulting in 200 independent realizations. We use 160 randomly sampled trajectories for training, 20 trajectories for validation and the remaining 20 trajectories for testing. Source code of the simulation can be found in [130].

3.4.4 Two-dimensional wave equation

A classical example of a hyperbolic PDE is the two-dimensional wave equation describing the temporal and spatial propagation of waves such as sound or water waves. Wave equations are important for a variety of fields including acoustics,

electromagnetics, and fluid dynamics. In two dimensions, the wave equation can be described as follows:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u, \tag{3.29}$$

with ∇^2 denoting the Laplacian operator in $\mathbb{R}^2$ and c is a constant speed of the wave propagation. The initial displacement u_0 is a Gaussian function

$$u_0 = a \exp\left(-\frac{(x-b)^2}{2r^2}\right), \tag{3.30}$$

where the amplitude of the peak displacement a, the location of the peak displacement b and the standard deviation r are uniformly sampled from $a \sim \mathcal{U}(2,4)$, $b \sim \mathcal{U}(-1,1)$, and $r \sim \mathcal{U}(0.25, 0.30)$, respectively. Similar to [131], the initial velocity gradient $\frac{\partial u}{\partial t}$ is set to zero. The wave simulations are performed over a spatial domain of ($x \in [-1,1]$ and $y \in [-1,1]$ on a grid with 128 points in each spatial dimension. Overall, 500 independent trajectories (individual initial conditions) are computed which are split in 400 randomly sampled trajectories for training, 50 trajectories for validation and the remaining 50 trajectories for testing.

3.4.5 Navier-Stokes equations

To ultimately test the performance of our model on complex real-world data, we simulated fluid flows governed by a complex set of partial differential equations called Navier-Stokes equations. Overall, two flow cases of different nature are considered, e.g., the temporal evolution of generic initial vorticity fields and the flow around an obstacle characterized by the formations of dominant vortex patterns also known as the von Kármán vortex street.
Due to the characteristics of the selected flow fields, we consider the incompressible two-dimensional Navier-Stokes equations given by

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u - \nu \nabla^2 u = -\frac{1}{\rho}\nabla p. \tag{3.31}$$

Here, u denotes the velocity in two dimensions, t and p are the time and pressure, and ν is the kinematic viscosity. For the generic test case, we solve this set of PDEs in its vorticity form and chose initial conditions as described in [132]. Simulations are performed over a spatial domain of ($x \in [-1,1]$ and $y \in [-1,1]$ on a grid with 128 points in each spatial dimension. Overall, 500 independent trajectories (individual initial vorticity fields) are computed which are split in 400 randomly sampled trajectories for training, 50 trajectories for validation and the remaining 50 trajectories for testing.

3.4.6 Flow around a blunt body

The second fluid flow case mimics an engineering inspired applications and captures the flow around a blunt cylinder body, also known as von Kármán vortex street. Von Kármán vortices manifest in a repeating pattern of swirling vortices caused by the unsteady flow separation around blunt bodies and occur when the inertial forces in a flow are significantly greater than the viscous forces. A large dynamic viscosity of a fluid suppresses vortices, whereas a higher density, velocity, and larger size of the flowed object provide for more dynamics and a less ordered flow pattern. If the factors that increase the inertial forces are put in relation to the viscosity, a dimensionless measure - the Reynolds number - is obtained that can be used to characterize a flow regime. If the Reynolds number is larger than $Re > 80$, the two vortices in the wake of the body become unstable until they finally detach periodically. The detached vortices remain stable for a while until they slowly dissociate again in the flow due to friction, and finally disappear. The incompressible vortex street is simulated using an open-source Lattice-Boltzmann solver due to computational efficiency. The governing equation is the Boltzmann equation with the simplified right-hand side (RHS) Bhatnagar-Gross-Krook (BGK) collision term [133]:

$$\frac{\partial f}{\partial t} + \zeta_k \frac{\partial f}{\partial x_k} = -\frac{1}{\tau}(f - f^{eq}) \tag{3.32}$$

These particle probability density functions (PPDFs) $f = f(\vec{x}, \vec{\zeta}, t)$ describe the probability to find a fluid particle around a location $\vec{x}$ with a particle velocity $\vec{\zeta}$ at time t [134]. The left-hand side (LHS) describes the evolution of fluid particles in space and time, while the RHS describes the collision of particles. The collision process is governed by the relaxation parameter $1/\tau$ with the relaxation time τ to reach the Maxwellian equilibrium state f^{eq}. The discretized form of equation 3.32 yield the lattice-BGK equation

$$f_k(\vec{x} + \zeta_k \Delta t, t + \Delta t) = f_k(\vec{x}, t) - \frac{1}{\tau}(f_k(\vec{x}, t) - f_k^{eq}(\vec{x}, t)). \tag{3.33}$$

The standard D_2Q_9 discretization scheme with nine PPDFs [135] is applied. The equilibrium PPDF is given by

$$f_k^{eq} = w_k \rho \left(1 + \frac{\zeta_k \vec{u}}{c_s^2} + \frac{(\zeta_k \vec{u})^2}{2c_s^4} - \frac{\vec{u}^2}{2c_s^2}\right) \tag{3.34}$$

where the quantities w_k are weighting factors for the D_2Q_9 scheme given by 4/9 for $k \in 0$, 1/9 for $k \in 1, \ldots, 4$, and 1/36 for $k \in 5, \ldots 9$, and $\vec{u}$ is the fluid velocity. c_s denotes the speed of sound. The macroscopic variables can be obtained from the moments of the PPDFs. Within the continuum limit, i.e., for small Knudsen numbers, the Navier-Stokes equations can directly be derived from the Boltzmann equation and the BGK model [136]. We simulated three different Reynolds numbers $Re = 100, 250, 500$ for 425 000 iterations with a mesh size of 128 point in vertical and

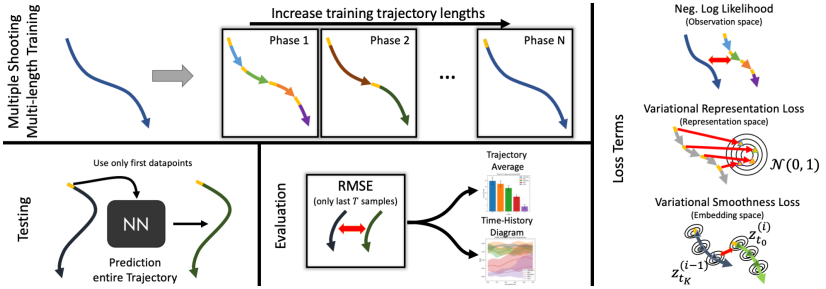


Figure 3.3: Training scheme with losses, and test/evaluation procedure. Top left: Multiple Shooting Multi-length Training. An input trajectory is split into subpatches. Subtrajectory length is increased in multiple phases to the length of the input trajectory. Bottom left: Testing: only the first few points are used to roll-out the latent trajectory and transformed to the observational space. Evaluation: Last T samples of the predicted trajectory are used to compute the evaluation metrics, the average of the summed normalized mean squared error and a time history diagram showing the error evolution. Right: Loss consisting of three parts: negative log likelihood loss to penalize reconstruction errors , representation loss to define a gradient field between representations, smoothness loss to penalize jumps between latent subpatches.

256 points in horizontal direction. We skipped the first 25 000 iterations to ensure a developed flow field and extracted velocity snapshot every 100 iterations. The simulation is performed over a spatial domain of ($x \in [-20, 20]$ and $y \in [-10, 10]$. We split this simulation into trajectories of 50 consecutive samples resulting in 200 in dependent realizations. We use 160 randomly sampled trajectories for training, 20 trajectories for validation and the remaining 20 trajectories for testing.

3.5 Experimental Setup

Training. We train all experiments in a multi-phase schedule w.r.t. the multiple shooting loss in eq. 3.21. In the different phases, we split the input trajectory into overlapping patches and start learning by predicting one step ahead. We double the number of prediction steps per patch every 3000 epochs meaning that learning is done on longer patches with decreased number of patches per trajectory. In cases, where the trajectory length is not dividable by the number of prediction steps, we drop the last patch and scale the loss accordingly. In the final phase, training is carried out on the entire trajectory. All network architectures are implemented in the open source framework PyTorch [45]. Training hyperparameters can be found in Appendix B.3.

Testing. We test the trained models on entirely unseen trajectories. During testing, we only provide the first k=10 trajectory points to the learned model. Based on these samples, we compute a trajectory representation ψ^r followed by rolling out the latent trajectory at the time points of interest. Last, we compare predictions and ground truth observations by evaluation metrics.

Evaluation metrics. Our experiments focus on two key metrics. Firstly, we calculate the mean squared error (MSE) for extrapolated trajectories, where we evaluate the model's performance over a total of $2T$ steps and measure the MSE over the last T time steps. This approach allows us to assess whether extrapolation over future time periods can better predict the model's ability to extrapolate further in time, compared to reconstruction MSE. The value of T is set to 60 for all our experiments, and we normalize the MSE value by dividing it with the average intensity of the ground truth observation, as recommended in [137, 138]. Additionally, we provide time history diagrams that plot the root mean square error (RMSE) against the normalized time, which maps the time interval $[T, 2T]$ to the interval $[0, 1]$. All evaluation metrics presented are averaged across all test trajectories and five runs, with mean and 75% inter-quantile range (IQR) reported on all metrics. Last, we also provide subsampled predictions and the pixelwise L_2 error of one trajectory for visual inspection but we emphasize that one trajectory might not be representative for the overall performance (the trajectory shown is chosen at random). Please see Figure 3.3 for visual intuition on the training and testing procedure and how evaluation metrics are applied.

Baselines. We compare our approach to recent models from the literature, e.g., ODE-RNN [139], NDP [140], ODE2VAE [141], and MSVI [117]. All the baseline methodologies share similarities with our proposed approach in terms of how they handle longitudinal observations. They encode these observations into latent spaces, simulate low-dimensional latent trajectories, and then decode these trajectories to predict future observations. Similar to our methodology, NDP [140] utilizes two latent variables for encoding (an "initial state" and a "global control of an ODE"). They employ MLPs or convolutions to model dynamics and use neural ODEs to integrate over time. The decoder generates predictions from a Bernoulli distribution.

From a broader perspective, MSVI [117] operates similarly, incorporating a modified encoder. In this case, a transformer module is introduced, while the dynamics function and decoder utilize Bayesian MLPs and CNNs, with their parameterization assumed to be Gaussian. Comparable to our work, MSVI utilizes Bayesian multiple shooting, relying on a smoothness prior. Consequently, training involves a loss function that integrates data, continuity, dynamics, and decoder priors.

Similarly, ODE2VAE [141] is grounded in a variational inference framework based on Bayesian Neural Networks. Like our approach, it encodes observations into a latent initial state, with explicit shaping through a physics-inspired prior. This prior

separates the latent space into velocity and position components. Subsequently, high-order dynamics are approximated using a BNN and evolve over time. The decoder setup is akin to MSVI, where both BNN priors are assumed to be Gaussian.

Echoing the other baseline methodologies, ODE-RNN represents a family of time series models with hidden state dynamics governed by Neural ODEs. As such, ODE-RNN models, when trained, rely on latent ODEs and can accommodate irregular time gaps between observations. To derive latent representations, our approach employs the same CNN encoding as ODE-RNN, ensuring a fair comparison. The dynamics function and decoder align with those of MSVI.

All models are subjected to training and testing following the default parameters and code provided in their original publications. Please note that for comparisons against these baseline methods, our options are restricted to regular time grid datasets, as ODE2VAE's encoder exclusively suits evenly spaced grids. For datasets with irregularly sampled observations, we present comparisons against MSVI.

3.6 Results

We report on a series of increasingly challenging experiments to test LaDID. First, we examine whether our model generalizes well on synthetic data for which the training and test data come from the same dynamical system. This body of experiments test whether the model can learn to map from a finite, empirical dataset to an effective latent dynamical model. Second, we examined few-shot generalization to data obtained from systems subject to nontrivial intervention (and in that sense strongly out-of-distribution). In particular, we train our model on a set of trajectories under interventions, i.e. interventions upon the mass or length of the pendulum, changes to the Reynolds number, or variations to the camera view on the observed system, and apply the learned inductive bias to new and unseen interventional regimes in a few-shot learning setting. This tests the hypothesis that the inductive bias of our learned latent dynamical models can be a useful proxy for dynamical systems exposed to a number of interventions.

3.6.1 Benchmark comparisons to state-of-the-art models for ODE and PDE problems

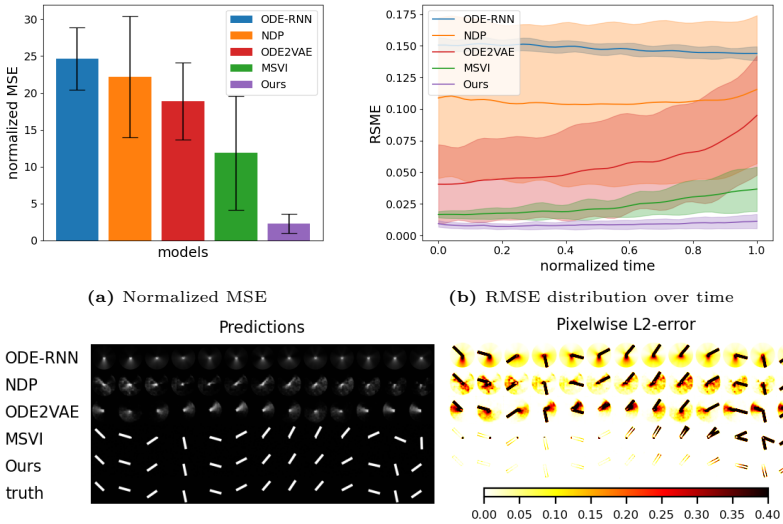
We begin by investigating whether our approach can learn latent dynamical models in the conventional case in which the training and test data come from the same system. We evaluate the performance of ODE-RNN, ODE2VAE, NODEP, MSVI and our model on the on data described in Sec. 3.4 with increasing order of difficulty, starting with the non-linear mechanical swing systems with underlying ODEs, before moving to non-linear cases based on PDEs (reaction-diffusion system, 2D wave equation, von Kármán vortex street at the transition from laminar to turbulent flows,

and Naiver-Stokes equations).

3.6.1.1 Applications to ODE-based systems.

Swinging pendulum. For visual inspection and intuition, Figure 3.4c provides predicted observations $\hat{x}_t^r$ of a few time points of one test trajectory of the single pendulum dataset for all tested algorithms, followed by the ground truth trajectory and the pixelwise L_2-error. In addition, Figure 3.4a presents the normalized MSE over entire trajectories averaged across the entire test dataset and the evolution of the RMSE over time for the second half of the predicted observations averaged over all test trajectories (see Sec. 3.5) is provided in Fig. 3.4b. First, we see that across all ODE-based datasets LaDID achieves consistently the lowest normalized MSE. Second, the time history diagram (see Fig. 3.4b) shows that LaDID predicts future time points with lower mean RMSE and lower standard deviation for long-horizon predictions relative to all other algorithms tested.

This can also seen by visual inspection in Figure 3.4c as for other approaches the predicted states at later time points deviate from the ground truth trajectory substantially while LaDID's predictions follow the ground truth. Considering only the baselines, one can observe that MSVI (a recent and sophisticated approach), achieves the best results and predicts accurately within a short-term horizon but fails on long-horizon predictions. The predictions generated by ODE2VAE exhibit blurry pendulum bars that do not align with the position of the ground truth trajectory. Similarly, NDP and ODE-RNN produce predictions characterized by indistinct dots centered within the image. These algorithms prioritize minimizing the loss function across all time steps, resulting in minimal variation in predictions over time. This lack of variation indicates a failure to learn the underlying dynamical model. We think that the high errors depicted in Figure 3.4a and 3.4b are a direct consequence of this limitation.



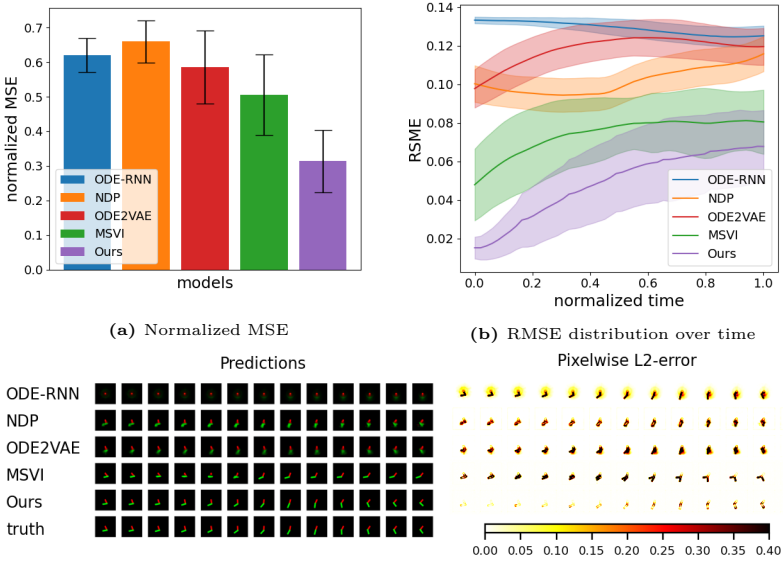
(a) Normalized MSE

(b) RMSE distribution over time

(c) Example trajectory

Figure 3.4: Test errors and exemplary test trajectories of different models for the single pendulum test case.

Swinging double pendulum. In line with the results of the rather simple single pendulum, the performance of LaDID remains also good for the swinging double pendulum test case governed by a set of complex ODEs. As shown in Fig. 3.5, LaDID achieves lowest errors relative to both the normalized MSE and the RMSE distribution plotted over time. For this test, however, the observed performance gap to the baseline is comparably moderate which may yield erroneous conclusions. In fact, if one only considers MSE and RMSE values, it may be concluded that LaDID performs only slightly better compared to the baseline methods. However, these methods do not learn any useful dynamics at all yielding trajectory predictions that are characterized by completely different motion patterns, e.g., the pendulum swings in a wrong direction or barely moves at all as shown in Fig. 3.5c. In contrast, LaDID predicts accurately the movement of the pendulum system with an increasing RMSE error over time. Indeed, the RMSE errors of MSVI and LaDID become comparable for long-horizon predictions although the dynamics of the trajectory predicted by MSVI differ substantially from the ground truth as visualized in Fig. 3.5c. This finding illustrates that the importance of an in-depth test case analysis and visualization as a pure comparison of error metrics might disguise relevant shortcomings of the learned dynamics model which at the same time creates the need



(a) Normalized MSE

(b) RMSE distribution over time

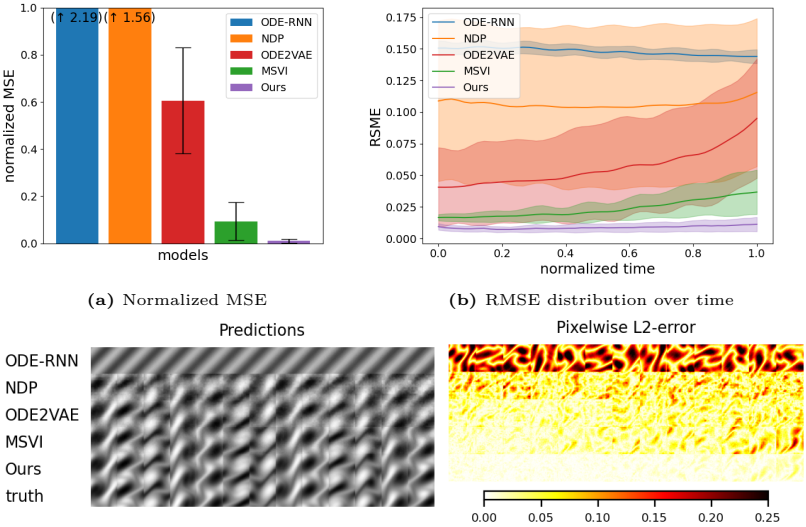
(c) Example trajectory

Figure 3.5: Test errors and exemplary test trajectories of different models for the double pendulum test case.

for well defined loss functions during training. For the present double pendulum test case, only LaDID is capable of learning a reliable approximation of the underlying system and demonstrates a convincing accuracy level when predicting long-term future trajectories. Note that none of the comparison baseline can accomplish this learning task for this complex system.

3.6.1.2 Applications to PDE-based processes.

Navier-Stokes equations. We additionally evaluated all baselines and our proposed method on PDE-based processes. We focus our analysis on the flow evolution characterized by the Navier-Stokes equation in the two dimensional case, which is of great importance in many engineering tasks, e.g. the analysis of internal air flow in a combustion engine [142, 143], drag reduction concepts in the transportation and energy sector [144, 145, 146], and many more. Results in Figure 3.6 show that LaDID clearly outperforms all considered competitors. The normalized MSE is the lowest and the averaged RMSE is also the lowest at any time. When comparing with the established baselines, it becomes evident that ODE-RNN struggles to capture



(a) Normalized MSE

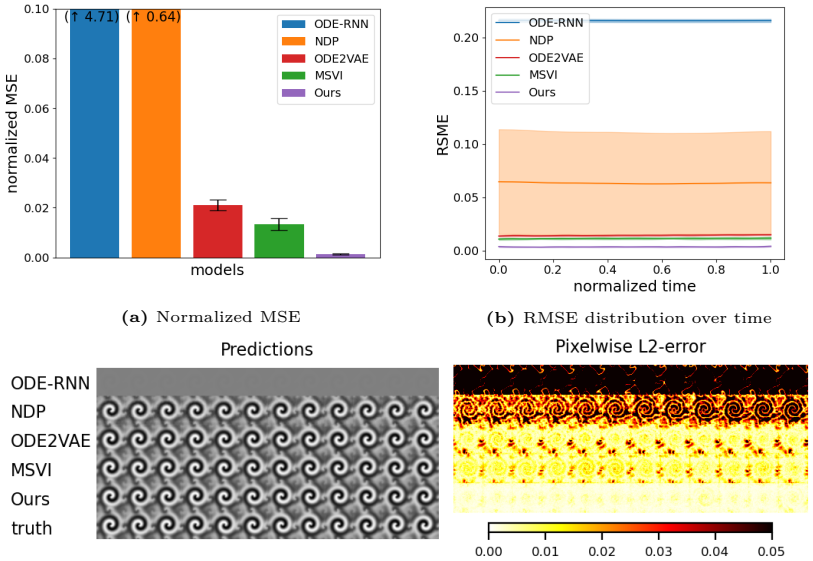
(b) RMSE distribution over time

(c) Example trajectory

Figure 3.6: Test errors and exemplary test trajectories of different models for the Navier-Stokes equations test case.

the underlying dynamics based on partial differential equations. This observation holds true across all other experiments involving PDE-based methods. On the other hand, in the case of the Bayesian approaches, ODE2VAE and MSVI, it is apparent that the error increases over time. However, this trend is not observed in the predictions made by LaDID, which consistently exhibits low error rates at all time points, approximately five times lower than the current gold-standard benchmark, MSVI. Moreover, when considering the interquartile range intervals, as indicated by the error bars in Figure 3.6a and 3.6b, it is evident that LaDID outperforms all baseline algorithms. This finding signifies that LaDID accurately forecasts dynamic behavior that closely aligns with the ground truth across various test trajectories.

Lambda-omega reaction-diffusion system. Prediction results and error distributions of the PDE based reaction-diffusion system are shown in Fig. 3.7 for all methods. Similar to the ODE system described above, the proposed approach demonstrates a convincing level of accuracy and precisely predicts future observational system states. In detail, LaDID can outperform all considered baseline by quite some margin while the RMSE error barely increases over time. Moreover, the IQR is much lower compared to the baseline methods. These findings evidence that LaDID effec-



(a) Normalized MSE

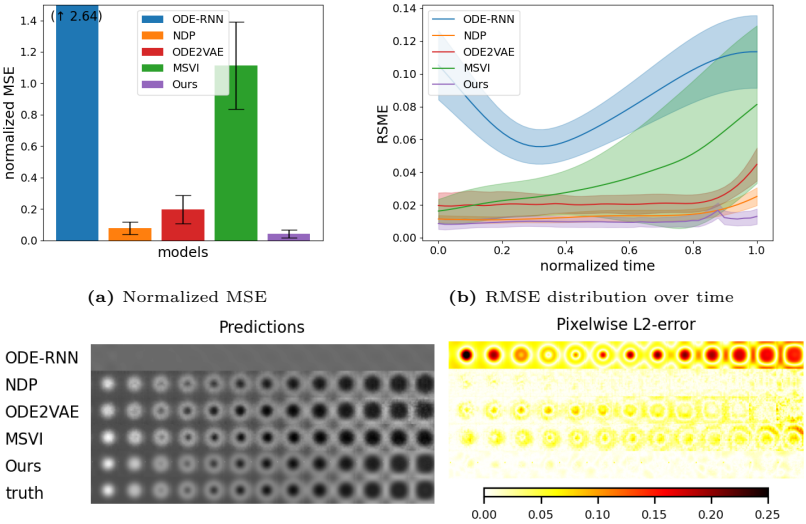
(b) RMSE distribution over time

(c) Example trajectory

Figure 3.7: Test errors and exemplary test trajectories of different models for the lambda-omega reaction-diffusion system.

tively learns the underlying PDE system and reliable predicts future states spaces of both near-term and long-horizon future predictions.

Two-dimensional wave equation. A detailed comparison of prediction results for the two dimensional wave equation can be found in Fig. 3.8. Interestingly, the general trend of baseline methods of this test case does not match the general trend of previous experiments. That is, NDP achieves much lower normalized MSE and RMSE error values compared to all other baseline which is nicely confirmed by the exemplary trajectory visualization shown in Fig. 3.8c. Note that the current gold-standard method MSVI only shows a modest performance for this test case, most likely based on its Bayesian prior. It appears that test cases characterized by spatially varying initial conditions (for the two dimensional wave equation system the initial condition of the simulation comprise the peak displacement of the wave and the initial wave location which may change over the entire spatial domain) constitute a major challenge for this Bayesian method since it derives a reconstruction distribution for every pixel over time individually. In cases with varying spatial initial conditions



(a) Normalized MSE

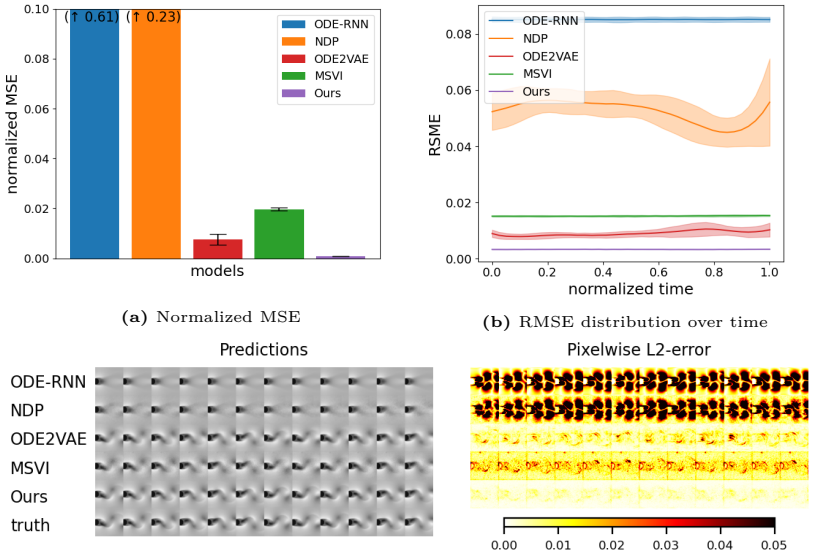
(b) RMSE distribution over time

(c) Example trajectory

Figure 3.8: Test errors and exemplary test trajectories of different models for the wave equation test case.

as present in the current test, this network approach most likely mixes realization-specific and dynamics information impeding the spatial and temporal reconstruction. To summarize, the two dimensional wave system demonstrates the system (and initial condition) dependent performance of the baseline method highlighting that different methods have different preferred sweet spots. Similar to the findings described above, the performance of our proposed LaDID approach remains convenient. Again, our approach can outperform all considered competitors and achieves lowest normalized MSE and RMSE values relative to both short-term prediction and long-horizon future states. The descent performance of LaDID is further evidenced in the trajectory visualization shown in Fig. 3.8c. We argue that our explicit decomposition in RS and RI learning parts helps the learner to extract correct dynamics even for cases characterized by spatially varying initial locations. This finding will be analyzed in more detail for transfer learning experiments under interventional distributions (see Section 3.6.4).

Turbulent flow around a blunt body. For the final test case, we analyze the performance of our proposed approach for a typical engineering application. That is, we seek to learn the time-dependent dynamics of a turbulent flow around a blunt cylin-



(a) Normalized MSE

(b) RMSE distribution over time

(c) Example trajectory

Figure 3.9: Test errors and exemplary test trajectories of different models for the von Kármán vortex street test case.

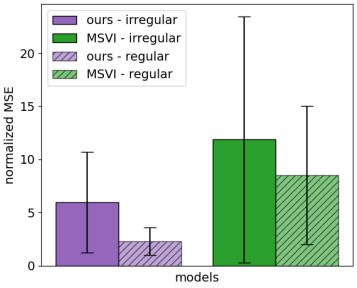
der body characterized by alternating vortex patterns. Results can be found in Fig. 3.9. In line with all experiments described above, the performance of LaDID remains good as it outperforms all considered baseline methods by quite some margin. In fact, our approach achieves lowest errors in combination with unmatched small uncertainties yielding very accurate predictions for short-term and long-horizon future sequences.

Overall, the results shown provide strong evidence that LaDID achieves state-of-the-art performance for ODE and PDE based systems.

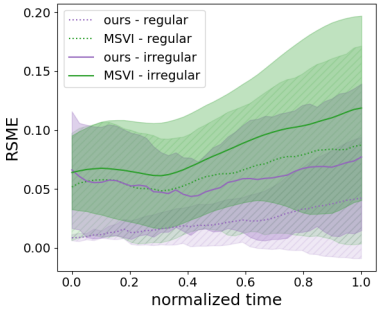
3.6.2 Performance on regular and irregular time grids.

Here, we study the performance of LaDID on regular and irregular time grids and compare it to other neural-dynamical models which are able to deal with irregular time series data. As shown in Fig. 3.10, the proposed LaDID performs very similarly on both types of the time grids relative to both ODE-based benchmark examples

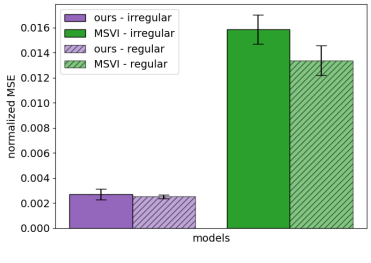
and challenging PDE-based real-world systems, outperforming existing methods and demonstrating strong and robust performance on irregularly sampled data. Moreover, it evidences that the proposed framework leveraging specific system invariances indeed extracts a realization-specific and a dynamics (realization-invariant) part as suggested.



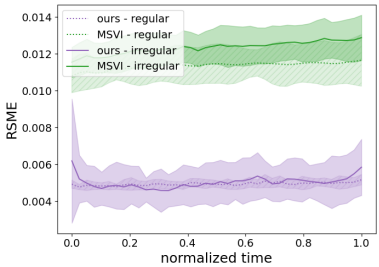
(a) Single pendulum: Normalized MSE



(b) Single pendulum: RMSE distribution over time



(c) Lambda-omega reaction-diffusion system: Normalized MSE



(d) Lambda-omega reaction-diffusion system-RMSE distribution over time

Figure 3.10: Test errors of different models for regular and irregular time grids: (a) Normalized MSE for single pendulum dataset, (b) RSME over time for single pendulum dataset, (c) Normalized MSE for lambda-omega reaction-diffusion system, (d) RSME over time for lambda-omega reaction-diffusion system.

Table 3.1: Errors for ablated LaDID models trained on the single pendulum test case.

loss heuristics			attention mechanism			representation encoding		
ablation	mean	IQR	ablation	mean	IQR	ablation	mean	IQR
reconstruction	2.66	1.04	no attention	7.79	6.84	w./o. encoding	2.83	2.02
reconstruction & representation	2.17	0.99	spatial attention	2.81	1.47	**w. encoding**	**2.02**	**0.88**
reconstruction & smoothness	2.04	0.92	temporal attention	2.41	0.92			
full loss	**2.02**	**0.88**	**spatio-temporal attention**	**2.02**	**0.88**			

Table 3.2: Comparison of training and inference time as well as the number of trainable model parameters for all methods applied to the single pendulum test case. Note that for MSVI (block size = 1 / 8), no inference times can be given since inference requires roll-out across the full trajectory. All tests are performed on a NVIDIA A100 40 Gb with an AMD EPYC 7742 processor.

	forward / backward pass [ms]	forward pass (inference) [ms]	trainable parameters [M params]
ODE-RNN	851.95	351.68	10.51
NDP	388.69	163.15	1.13
ODE2VAE	571.77	68.04	3.04
MSVI (block size = 60)	1192.86	48.37	1.51
MSVI (block size = 8)	285.79	(N/A)	1.51
MSVI (block size = 1)	60.08	(N/A)	1.51
LaDID (ours)	48.11	15.09	1.35

3.6.3 Effects of relevant network modules.

As discussed above, our model leverages three key features: a reconstruction embedding, a spatio-temporal attention module and a specifically designed loss heuristic to learn temporal dynamics from empirical data. Here, we show that each of these network modules is indeed relevant for the good performance of LaDID (see Table 3.1). First, we compare LaDID with counterparts trained on ablated loss heuristics, e.g. a pure reconstruction loss and loss combinations either using the described representation or smoothness loss. Overall, the proposed loss heuristic appears to stabilize training and yields the lowest MSE and IQR values. Second, we compare LaDID to counterparts trained on ablated attention modules. Table 3.1 highlights that the applied spatio-temporal attention helps to extract key dynamical patterns. Finally, Table 3.1 further shows the usefulness of the proposed representation encoding. This representation encoding can be thought of a learning-enhanced initial value stabilizing the temporal evolution of latent trajectory dynamics.

Table 3.2 displays the time required for various executions, including a training step involving both forward and backward passes, as well as a single forward pass conducted during inference. The figures prominently underscore the substantial speed advantage of LaDID, achieved with a notably smaller parameter count. This

advantage originates from our approach of not explicitly solving an ODE, but rather acquiring the ability to implicitly compute its solution through our end-to-end dynamics model.

3.6.4 Generalizing to novel systems via few-shot learning

Here, we assess LaDID's ability to generalize to a novel system obtained by nontrivial intervention on the system coefficients themselves (e.g., mass, length, Reynolds number). Such changes can induce large changes to data distributions and can be viewed through a causal lens. Consider a vectorized differential equation of the form

$$\frac{d}{dt}\boldsymbol{x} = f(\boldsymbol{x}, t) \tag{3.35}$$

with general initial conditions $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$. Using an appropriate discretization scheme allows us to foretell the future of the system based on its past states and we can directly read off the causal structure for such a system. This underlying causal structure remains the same when changing coefficients on the right hand side which is summarized by the principle of independent causal mechanisms [147, 148, 149]:

Independent Causal Mechanism: Let X be the outcome variable, and let $M_1, M_2, \ldots, M_n$ be n distinct mechanisms or factors that independently contribute to X. Then, we can write:

$$X = f(M_1, M_2, \ldots, M_n) \tag{3.36}$$

where f represents the functional relationship between the mechanisms and the outcome. In other words, the outcome variable X is determined by the independent contributions of each mechanism M_i, and these mechanisms operate independently of each other and do not inform each other.

For example, consider a system of ODEs that describes the dynamics of a population of predator and prey animals:

$$\frac{dx}{dt} = ax - bxy \tag{3.37}$$

$$\frac{dy}{dt} = -cy + dxy \tag{3.38}$$

where x represents the population of prey, y represents the population of predators, and a, b, c, d are parameters that describe the interactions between the two populations. In this system, x and y represent independent causal mechanisms that contribute to the dynamics of the population. The equations describe how the population of prey and predators changes over time as a result of their interactions. By solving the system of ODEs, we can study how changes in the parameters affect the long-term behavior of the system, and how interventions can be used to control the population dynamics. Mathematically, an intervention is typically represented

as a modification of the equations that describe the system, by setting the value of one or more variables to a fixed value or function. This modification represents the assumption that the variable(s) being intervened upon is no longer subject to external influences, and its value is determined by the intervention. In formal terms, an intervention upon a set of nodes $\mathcal{I}$ in the causal structure of a system $\{X_i : i \in \mathcal{I}\}$ means any manipulation of the system that alters its state or behavior, including changes in the initial conditions, modifying the parameters, or adding or removing variables or equations. When observing a latent process, which moves the deterministic setup of an ODE to a probabilistic case, this means that the conditional distribution when observing the state of a node given its parents $Pa(X_i)$ is replaced by a new, predefined distribution. Thus, the joint probability distribution of as system under an intervention changes to

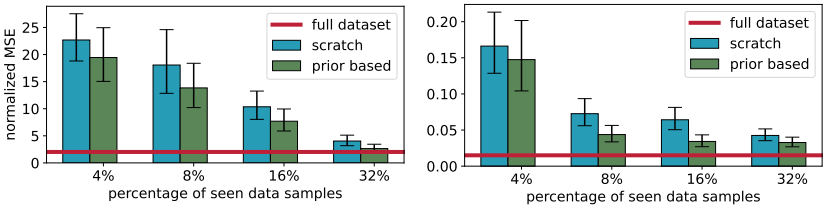
$$\tilde{p}(X) = \prod_{i \notin \mathcal{I}} p(X_i|Pa(X_i)) \prod_{i \in \mathcal{I}} \hat{p}(X_i|Pa(X_i)) \tag{3.39}$$

where $\hat{p}(X_i|Pa(X_i))$ indicates the conditional distribution of node X_i in its general form.

Assuming that the underlying dynamical model is an invariant causal mechanism, we aim to transfer the inductive bias learned from a set of training systems to a new set of systems with limited data availability in a few-shot learning setup. In particular, we train a dynamical model on a set of interventions and fine-tune it to new intervention regimes with only a few samples, finally evaluating performance on an entirely unseen dataset. We compare the performance of our prior-based few-shot learning model with a model trained solely on the fine-tuning dataset ("scratch-trained" model). In our first experiment, we use the single pendulum dataset and test the transferability hypothesis on fine-tuning datasets of varying sizes. In a few-shot learning setup, we train in a first stage a dynamical model on a set of interventions and then, fine-tune the the dynamical system to the new intervention regimes with only a few fine-tuning samples. Testing is carried out on an entirely unseen dataset and we compare performances between a prior based few-shot learning model and a model that is solely trained on the fine-tuning dataset. In our experiments, we use the single pendulum dataset and test prior transferability hypothesis on fine-tuning datasets that span 32%, 16%, 8%, and 4% of the training data size. The results are presented in Figure 3.11a and plot the normalized MSE against the fine-tuning dataset size. The results are averaged across five runs. Overall, one can observe that the normalized MSE is always lower for the model that fine-tunes the transferred inductive bias. At a fine-tuning dataset size of 32%, the prior based model reaches a comparable performance to a model that was trained from scratch on 100% data size. Decreasing the number of fine-tuning samples leads to an expected rise in the normalized MSE. By visual inspection, we could see that 8% dataset size is the absolute minimum which produces partially erroneous but still usable predictions which are only slightly worse compared to the predictions of MSVI trained on full data

availability.

In a second experiment, we are investigating effect of interventions upon the observation process. In other words, we leave the underlying dynamical system unchanged but introduce variation to e.g. the camera position and angle when capturing samples in the observation space. If the model learns a valid dynamical model, it should be transferable to new observation settings. We test this hypothesis on the von Kármán vortex street dataset by shifting the dataset to the camera to the left and right and up and down. Please note that since we learn a latent dynamical end-to-end, we do not assume a zero-shot transferability since encoder and decoder also need to adapt to the new input scenes. Thus, we evaluate this hypothesis in a few-shot learning problem as before. Figure 3.11b plots the normalized error against a fine-tuning dataset size of 32%, 16%, 8%, and 4% of the training data size. Again, we compare against model that is solely trained on the fine-tuning dataset. Overall, we can observe that prior based model show always lower normalized MSE values than models trained from scratch under limited data availability. Moreover, the results indicate that a fine-tuning a prior based model on a new observation scene with as little as 8% of the usual training set size leads to accurate and usable predictions under new observation conditions. This finding supports the hypothesis that our model extracts a general dynamical model.



(a) single pendulum **(b)** vortex street

Figure 3.11: Test errors for a set of transfer learning experiments.

3.7 Discussion and Conclusion

In this Section, we presented a novel approach called LaDID aimed at end-to-end learning of latent dynamical models from empirical data. LaDID uses a novel transformer-based architecture that leverages certain scientifically-motivated invariances to allow separation of a universal dynamics module and encoded realization-specific information. We demonstrated state-of-the-art performance on several new and challenging test cases and well-known benchmarks. Additionally, we showed that LaDID can generalize to systems under nontrivial intervention (when trained on the unintervened system) using few-shot learning, and in that sense provides a

useful prior for systems under novel interventions.

However, despite good performance relative to NODE-based methods, for complex models prediction over very long time horizons remains challenging and further modifications and possibly additional inductive bias will be needed for improved performance. At present, data sampled on an irregular *spatial* grid cannot be considered. Future work using graph-based approaches will address this point.

Our work was motivated by the broad range of examples in contemporary science and engineering in which data are available from different experiments/pipelines that are plausibly underpinned by a unified model but where the model itself cannot be directly specified from first principles or is too complex to work with. A key motivation for our work is problems in the biomedical and health domains, where we expect the invariances underpinning LaDID to hold, but where explicit dynamical models are often not available at the outset. In ongoing work we are exploring the use of LaDID and related schemes in these challenging settings, e.g. for the modeling of complex longitudinal observations. In such settings, an implicit yet universal model is useful as a way to capture system behavior and to generalize, in a data efficient manner, to new realizations/instances where only limited data may be available.

4 Forecasting Responses in Unpaired Interventional Data using Sparse Causal Modeling

In the realm of scientific exploration, the task to identify cause-effect relationships within high-dimensional systems stands as an enduring and intricate challenge. Interventions, in the form of controlled experiments, emerge as indispensable tools that play a pivotal role in enhancing our understanding of the underlying causal framework within a system. These interventions, although powerful for revealing causality, necessitate meticulous attention to study design, randomization, and the management of confounding factors to ensure accurate conclusions. The act of deliberately disrupting the natural sequence of events to observe the system's response yields a multifaceted range of insights. An immediate application of interventional experiments involves confirming expected causal relationships and pathways or identifying confounding structures as shown in Figure 1.1. Yet, interventions also extend their utility by enabling the anticipation of responses to subsequent interventions, i.e. knowledge acquired from one intervention can be extended to forecast outcomes in unobserved interventions. This comprehension of variable interactions during interventions and the recognition of recurring patterns facilitate the informed prediction of results in new intervention scenarios. In this Chapter, our goal is to develop a mathematical framework that enables the prediction of responses to interventions that have not been previously carried out. We achieve this by solely relying on information obtained from interventional experiments that have already been conducted. This model possesses considerable promise in predicting outcomes for interventions that are ethically unfeasible, financially burdensome, or resource-intensive and facilitates addressing counterfactual inquiries. For example, in the field of biomedicine, this model would allow us to anticipate the phenotypic responses to genetic interventions without requiring further experimental interventions.

In this context, we examine a scenario consisting of a response variable Y and covariates X. Some of these covariates act as causal predecessors to Y, while others are linked to Y through concealed confounding factors. Additionally, there are covariates that possess both causal relationships and connections via hidden confounders. The system is observed through interventions, where the targets of these interventions remain unknown. The ultimate objective in this setting is to forecast the outcome of Y based on X in the context of an unseen intervention. However, this ambitious endeavor is accompanied by a multitude of key challenges:

Confounding: Consider, for instance, the enigmatic interplay between the transcriptome of a cell and a phenotypic measurement, e.g. a disease state or cell growth. Unraveling the precise causal influence of the cell's transcriptome on the observed phenotypic outcome presents a complex puzzle. This is further compounded by the hidden relation between the high-dimensional covariates and the target variable which often leads to substantial confounding through unknown mechanisms. In many real-world applications, this confounding renders the stringent assumptions necessary for estimating cause-effect relationships from observational data unfeasible.

Distribution Generalization: While machine learning techniques have pioneered breakthroughs in fields such as statistics, econometrics, epidemiology, and related disciplines, they are often vulnerable to challenges of limited generalizability, instability, and inexplicability due to the presence of spurious relationships. These relationships entail associations between two or more events or variables without a causal connection. In the context of a response variable Y and the set of covariates X, existing methods frequently strive to identify a function that minimizes the worst-case risk within a small neighborhood of distributions. The selection of this neighborhood should be representative of the differences between the training and test datasets [150, 151]. In other words, these models will degrade performance when the test distribution undergoes uncontrolled and unknown distribution shifts [149], which commonly emerge as intervention distributions.

Unpaired Data: Recent technological advancements have empowered researchers to explore heterogeneous mixtures of cell populations at the single-cell level. Through techniques like single-cell RNA sequencing providing comprehensive transcription profiling across the entire genome, single-cell ATAC-seq allowing the identification of accessible chromatin regions, or single-cell bisulfite sequencing facilitating the measurement of DNA methylation patterns, it is possible to measure histone modifications or transcription factors at single-cell resolution. Despite these remarkable technological milestones, it remains technically infeasible to observe all genomic profiles and phenotypic traits simultaneously within the same single cell. As an alternative approach, researchers have sought to generate specific modalities of genomic data from certain cells and complement it with other modalities from other cells within the same heterogeneous population. This integration of multiple omics datasets has been instrumental in gaining a comprehensive understanding of cellular processes. A substantial amount of unpaired data, such as unpaired scRNA-seq and scATAC-seq datasets, has been generated [152, 153, 154], where the profiles are not derived from the same individual cells. This poses a significant challenge in establishing cause-effect relationships from this pool of unpaired data, necessitating the development of novel and robust analytical approaches to overcome this limitation.

Sparse Causal Effects: We specifically assume that the causal effect of the covariates X on Y is sparse, however, we do not make any assumption on the existence or strength of statistical correlations between nodes which, in fact, are likely to be

observed much more frequently than actual causal connections. The process of narrowing down the multitude of potential causes influencing a particular target variable to a more manageable subset of candidates holds significant practical importance. Hence, establishing an effective framework to tackle this challenge is crucial with potential applications ranging from validating biomarkers as causal risk factors to developing proxies for clinical trials. Such an efficient framework can pave the way for advancements in various domains and offer valuable insights for targeted interventions and improved decision-making processes.

In response to these challenges, we introduce a sparse-effect model designed for unpaired data. This model predicts the outcomes of interventions that have not been previously encountered, relying on insights from past experiments. We show the effectiveness of our framework on a simulated benchmark and semi-simulated test cases in which the data of the covariates stems from human single cell data. The rest of this Chapter is structured as follows. In Section 4.1 we review related work. Section 4.2 introduces the setup, followed by the introduction of our framework in Section 4.3. We evaluate our model in a set of diverse tests in Section 4.4 and 4.5. Finally, we draw conclusions and discuss limitations and improvements of our approach in Section 4.6.

4.1 Related Work

Machine learning techniques have made significant advancements in fields like statistics, econometrics, and epidemiology. However, they often suffer from limitations such as poor generalizability, instability, and lack of interpretability. This is primarily because these techniques can establish associations between variables but fail to identify causal relationships.

Inferring Causal Effects through Randomized Control Experiments
In various medical disciplines, the focus is centered on treatment effects, which pertain to understanding the influence of an intervention on the subjects under study. Essentially, we aim to assess how administering a drug to a patient affects their health condition compared to their state before receiving the initial medical dosage. To achieve this goal, Randomized Controlled Trials (RCTs) are considered the gold standard for estimating treatment effects. In an RCT, one or more causal variables are randomly assigned to different samples. By ensuring a sufficient number of participants, this approach can effectively control for confounding factors and provide valuable comparisons between interventions. However, conducting fully randomized controlled trials may not always be practical due to factors such as cost and ethical concerns [155]. Estimating treatment effects from observational data poses challenges in the presence of unmeasured confounders. Specifically, we can only observe the outcome when a specific treatment is applied, and we lack the ability to obtain counterfactual outcomes that would have arisen if a different treatment option had been assigned. Furthermore, non-random assignment of treatments and interven-

tions often results in notable variations in the covariate distribution across different treatment groups. Even when we account for all observed variables and adjust for confounding differences using observational covariates, the existence of unmeasured key variables can conceal the causal relationships within the observational data.

Prior approaches utilizing Instrumental Variables

A frequently used method which typically suffers from unmeasured confounders is the Ordinary Least Squares (OLS) analysis. In causal inference, the objective of regression analysis is to estimate the conditional expectation function, $\mathbb{E}[Y|do(X)]$, in order to recover the coefficient, β, from the scalar regression model. However, in real-world scenarios, there may be unobserved confounders that act as common causes for both the covariates X and the outcome Y. As a result, there exists a direct effect $X \to Y$ as well as an indirect effect through unknown confounders that influence X and, in turn, introduce an additional spurious correlation between the covariates and the outcome. In this situation, the OLS estimator becomes biased as it combines both of these effects within the regression analysis. To overcome the challenge of unmeasured confounders, researchers introduced instrumental variables [156, 157]. The underlying idea is to leverage additional information and utilize alternative sources providing independent variables to mitigate confounding effects. By assuming a linear causal relationship, instrumental variables provide a tool to uncover the causal impact of the treatment on the outcome. In situations where the independent variables are potentially endogenous, instrumental variables are employed as proxies to address this issue. These instruments are variables that are correlated with the endogenous independent variables but have no direct effect on the dependent variable. They are used to isolate the exogenous variation in the independent variables and estimate their true causal effect on the outcome variable. Two-stage least squares (2SLS) [158] involves conducting two separate regression analyses to estimate causal effects. In the first stage, a set of functions is fitted to establish the relationship between instrumental variables and each covariate. Then, in the second stage, a single function is learned to approximate the connections between the covariates and the outcome variable. The intuition behind 2SLS resembles the process of identifying a valid instrumental variable. Precisely, the goal of the first stage of 2SLS is to estimate the extent to which a specific covariate changes when the corresponding instrumental variable is modified. This step allows us to understand the impact of the instrument on the covariate. In the second stage, it is examined how the changes in the covariate, induced by the instrument, ultimately affect the outcome variable. By conducting these two stages, we can gain insights into the causal relationship between the instrument, the covariates, and the outcome. In order to accommodate non-linearity assumptions, researchers have developed specific identification assumptions tailored to different scenarios (see [159, 160, 161] amongst others). To extend the applicability of 2SLS regression to nonlinear settings, "Sieve NPIV" [162], provides a flexible and robust framework for nonlinear causal inference using instrumental variables. This method draws inspiration from non-parametric sieve regression estimators and involves a basis expansion technique in the two-stage

estimation process. More recent studies focus on modeling relations between instruments, covariates and outcome as nonlinear functions in reproducing kernel Hilbert spaces [163, 164] while other approaches build upon deep learning techniques [165, 166]. To ensure the effectiveness of conventional IV estimation methods, it is crucial to have valid instruments (refer to Section 4.2). For instance, if an instrument is confounded with the outcome, this immediately renders the IV invalid and thereby yields imprecise results in inference and limited applicability in real-world scenarios. One approach to enhance the precision of instrumental variables estimators is to employ multiple instruments or attempts to approximate the optimal instruments [167, 168].

Concurrent research focuses on synthesizing valid and robust instrumental variables from a pool of candidate instruments. In this context, LASSO is a popular method that serves as both a regression function estimator and a model selection tool. It offers a practical solution for incorporating optimal instruments into IV estimation while mitigating the challenges associated with a large number of instruments. The LASSO estimator selects instruments and estimates the coefficients of the first-stage regression using a shrinkage procedure [169, 170, 171, 172]. The Post-LASSO estimator, on the other hand, discards the LASSO coefficient estimates and employs the instrument set selected by LASSO to re-estimate the first-stage regression using ordinary least squares (OLS), thus reducing LASSO's shrinkage bias [173]. [174] propose a method for IV estimation with a large number of instruments using a shrinkage estimator that assumes a random coefficients structure for the first-stage coefficients. Similarly, [175] suggest to use ridge regression to estimate the first-stage regression in a homoscedastic framework, where the instruments can be ordered based on their relevance. [176] investigate underspecified instrumental variable settings by assuming sparsity between covariate and outcome. This allows to relax standard identifiability assumptions in the linear IV setting. [177] propose an estimator relying on distributionally robust variable selection. Their approach can be seen as an interpolation between OLS and 2SLS estimates.

Examining Distribution Shifts: Previous Research Overview
Another fundamental challenge is the potential mismatch between the distributions of the data used for training and the data on which the models are tested. This discrepancy has drawn significant attention in recent years due to its potential implications for the reliability and generalizability of scientific findings. The training-test distribution gap arises when the data used to train a model does not accurately represent the real-world scenarios that the model is expected to encounter during its deployment. This mismatch can occur due to various reasons, including differences in data collection methods, environmental conditions, experimental settings, or temporal factors. The consequences of the training-test distribution mismatch can be far-reaching, particularly in scientific domains where the robustness and generalizability of models are of utmost importance. For instance, in fields such as medicine, climate science, and particle physics, models trained on limited or biased datasets

can lead to inaccurate predictions, potentially compromising patient outcomes, environmental policy decisions, or fundamental understanding of physical phenomena. However, assuming an arbitrary test distribution is impractical. Thus, addressing the challenge of distributional differences between training and test distributions requires certain restrictions to enable generalization. To tackle this issue, a common approach is to define a neighborhood around the training distribution, using divergence measures such as the Kullback-Leibler divergence [178, 179] or the Wasserstein distance [151, 180], allowing for a controlled exploration of distributional differences. Adversarial attacks are frequently employed in empirical studies to tackle distributional shifts [181] but the theoretical understanding of these procedures is still evolving. In scenarios involving covariate shift, it is often assumed that the distributions of the covariates differ between training and test data, while the conditional distribution of the response given the covariates remains invariant [182, 183, 184]. Sometimes, it is additionally assumed that the support of the training distribution covers its counterpart of the test distribution [185]. Complementary research allows for the conditional distribution of the response given the covariates to vary between interventions due to the presence of a hidden confounder in settings where the test observations lie outside the range covered by the training data [186].

Our approach is based on the principles of linear instrument variables estimators [156, 157, 158]. To tackle the issue of many instruments, we employ instrument selection techniques that rely on approximate sparsity. Our work is heavily influenced by the research conducted by [176], where we extend their work to address the challenging scenario of unpaired covariate-outcome data. Specifically, we aim to identify sparse causal relationships from interventional data characterized by changing supports of observational and interventional distributions. To evaluate our model's performance, we conduct tests on simulated gold-standard benchmarks under controlled settings. Additionally, we also test our model on a semi-simulated scenario that incorporates covariates from real-world human data [55].

4.2 Modeling Causal Relation under Unpaired Interventional Data

Our primary goal is to estimate the causal impact of covariates $X \in \mathbb{R}^d$ on a scalar outcome $Y \in \mathbb{R}$. While the relationship between X and Y may be affected by unobserved variables $H \in \mathbb{R}^q$, we assume that we have access to valid instruments $I \in \mathbb{R}^m$. These exogenous instruments induce changes in the covariates X but have no direct influence on the outcome Y, i.e. mathematically we have $P(X|I) \neq P(I)$ (see Figure 4.1). Additionally, we require that there are no direct pathways from the instrumental variables I to the outcome Y, and the instruments I and covariates X are not confounded by common hidden factors H. It is important to note that we cannot directly observe or test for the presence of hidden confounders. Therefore, the assumption of independence between instruments and hidden variables is necessary to

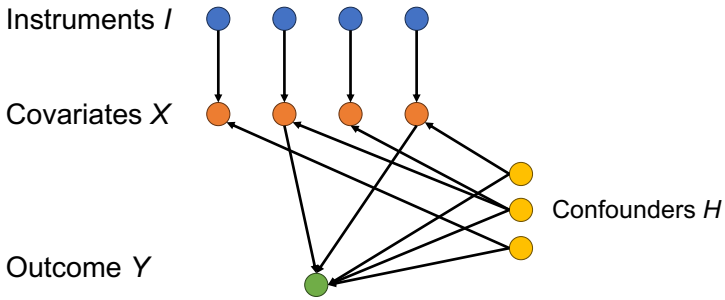


Figure 4.1: IV setup: We are concerned with finding sparse causal relations between between a set of covariates X and outcome Y. Potential confounders H might conceal direct cause-effect estimation and therefore, require additional intrumental variables I.

ensure the validity of the instruments. We consider the following structural equation model (SEM) describing the underlying data generating process:

$$\begin{aligned} X &:= BX + AI + h(H, \epsilon^X) \\ Y &:= X^\top \beta^* + g(H, \epsilon^Y) \end{aligned} \tag{4.1}$$

where B defines the adjacency matrix of a DAG describing inter-covariate dependencies, A expresses the relations between covariates X and instruments I, β^* is a sparse vector of coefficients and ϵ^X and ϵ^Y define noise terms. We assume that we observe data from different experiments/interventions and define for all $K = k \in \{1, \dots, m\}$, $I_k := e_k \in \mathbb{R}^m$ being the unit vector in $\mathbb{R}^m$ of the k-th dimension. That is, each column in the matrix A specifies a different experiment in which (a subset of) the covariates X is shifted by the corresponding column of A. Further, we assume that I_k, H, ϵ^X and ϵ^Y are jointly independent and $\mathrm{Id} - B$ is invertible, with Id denoting the identity matrix.

In our analysis, we consider data obtained from various interventions. We define the unit vector $I_k := e_k \in \mathbb{R}^m$ as the k-th column of matrix A, where A represents the experimental setup. Each column of A corresponds to a distinct experiment where a subset of the covariates X is shifted according to the amount specified in the corresponding column of A. Moreover, we make the assumption that the data of I_k, unobserved confounders H, and the error terms ϵ^X and ϵ^Y are mutually independent. Within an experiment $K = k$, we may have several repetitions which are split into two disjoint subsets of size $(n_k, \tilde{n}_k)$ and share the value of I_k . More explicitly, we write

$$X \coloneqq BX + AI_k + h(H, \epsilon^X) \qquad Y \coloneqq X^T\beta^* + g(H, \epsilon^Y)$$
$$\tilde{X} \coloneqq B\tilde{X} + AI_k + h(\tilde{H}, \tilde{\epsilon}^X) \qquad \tilde{Y} \coloneqq \tilde{X}^T\beta^* + g(\tilde{H}, \tilde{\epsilon}^Y)$$

where the variables H, $\tilde{H}$, ϵ^X, $\tilde{\epsilon}^X$, ϵ^Y, and $\tilde{\epsilon}^Y$ are jointly independent.
The experiment indicator k can be modeled in two ways: (i) as a uniform random variable $K \sim \mathcal{U}\{1, \dots, m\}$ as done above or (ii) as a non-random experiment indicator.

Our work develops methodology that only uses the realizations of $\tilde{X}$ and Y. We are therefore referring to an unpaired setting. Any identification of the dependence structure between the covariates $\tilde{X}$ and the response Y must come via the different experimental settings, i.e. within an experiment k, $\tilde{X}$ and Y are independent from each other.

4.3 Algorithm

Consider a data generating process of the form 4.1. It can be shown that -under mild conditions- the true parameter vector β^* is the unique solution to $\min_{\beta \in \mathcal{B}} \|\beta\|_0$ with $\mathcal{B} = \left\{\beta \in \mathbb{R}^d | Cov(\tilde{X}, X)\beta = Cov(\tilde{X}, Y)\right\}$. We refer the interested reader to Section C.1 and Theorem C.1.1 therein for a formal statement and proof of this statement. The underlying the theoretical work is a contribution from Niklas Pfister, Jonas Peters, and Sach Mukherjee which is not subject to this thesis but of great importance for the underlying methodology and the algorithm derived in the following.

The theoretical findings in Section C.1 of the Appendix emphasize that the causal coefficient β^* can be determined in scenarios with unpaired data. Now, we present a novel estimation approach that optimizes Eq. C.3 via coordinate descent and induce sparsity into the solution vector by adding a LASSO type penalty term. We perform a grid search over penalties to find a sparse yet effective solution. Recall that our work focuses on finding the true causal structure between covariates and outcome from unpaired data samples and estimates quantitatively the strength from cause on effect which goes beyond causal inference with typical IV approaches. In the following, we assume having access to observations of k experiments each defining an intervention upon an instrument. Further, we assume that the data consists of three data tuples $[(X, Y), (\tilde{X}, \tilde{Y}), (\tilde{\tilde{X}}, \tilde{\tilde{Y}})]$ that are generated from the same SEM in Eq. 4.1 but denote independent realizations of the same experiment given the identical set of instruments. In particular, we are interested in inferring the causal structure of the underlying SEM based on the unpaired three-tuple $(X, \tilde{X}, \tilde{\tilde{Y}})$. Following the idea of [176], we test the effectiveness of the sparse solution vectors with sparsity s=$\{1, \dots, d\}$ by the null hypothesis:

$$H_0(s): \quad \exists \beta \in \mathbb{R}^d \text{ with } \|\beta\|_0 = s \text{ s.t. } \beta \in \mathcal{B}. \tag{4.2}$$

We can test this hypothesis using an Anderson-Rubin test [150] as suggested in [176] with some modifications. Let $P_X \coloneqq \tilde{X}(\tilde{X}^\top \tilde{X})^{-1}\tilde{X}^\top$, then the Anderson-Rubin test is defined as

$$T(\beta) = \frac{(Y - X\beta)^\top P_X (Y - X\beta)}{(Y - X\beta)^\top (\mathrm{Id} - P_X)(Y - X\beta)} \frac{n-d}{d} \tag{4.3}$$

where Id denotes the identity matrix. Eq. 4.3 satisfies $T(\beta) \sim F(n-d, d),\ \forall \beta \in \mathcal{B}$. As argued in [176], the limited likelihood estimator (LIML) minimizes this test statistic. Following the same line of argumentation and assumptions in [176], we define a hypothesis test $\phi_s : \mathbb{R}^{n\times d} \times \mathbb{R}^{n\times d} \times \mathbb{R}^n \to \{0, 1\}$ by

$$\phi_s(X, \tilde{X}, Y) = \mathbb{1}(T(\hat{\beta}(s)) > F^{-1}_{n-d,d}(1-\alpha)). \tag{4.4}$$

Next, we discuss a novel algorithm that refers to the case with unpaired data. Our implementation provides a flexible and efficient way of unpaired LASSO regression using the coordinate descent algorithm, which can be applied to various problems that benefit from feature selection, regularization, and interpretability.

We determine the intensity of the penalty by conducting a grid search. For each penalty value λ, we obtain the estimated coefficient vector, denoted as $\hat{\beta}$, by

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \Big(\sum_{i=1}^{n} (Y_i - f(X_i, \hat{X}_i, \beta))^2 + \lambda \sum_{j=1}^{d} |\beta_j| \Big) \tag{4.5}$$

where f denotes our prediction function. To ensure that $\hat{\beta}$ belongs to the solution space $\mathcal{B}$, we perform a hypothesis test (see Eq. 4.4). The optimization problem in Equation 4.5 is solved using coordinate gradient descent. Our approach iteratively updates all regression coefficients in a coordinate-wise manner, applying a soft thresholding step to promote sparsity in the coefficient estimates. In each update step, we compute the influence of the j-th covariate on the residuals. This computation takes into account the difference between the measured target variables and the predicted values, as well as the contribution of the j-th covariate itself. Additionally, we leverage the relationship (or similarity) between the j-th covariate in X and the covariates in $\tilde{X}$. These two vectors are then utilized to update the j-th coefficient in the soft thresholding step, which drives the covariate selection process. In essence, our method captures the influence of the j-th covariate on the residuals and incorporates the relationships between the j-th covariate and the covariates in $\tilde{X}$. This information is crucial for estimating the importance of the j-th covariate and guiding the selection of relevant covariates within the unpaired LASSO regression framework. Further details of this approach are given in Algorithm 1 and 2.

Algorithm 1 Unpaired Causal Regression

Input: tuple of unpaired data $(X \in \mathbb{R}^{n\times d}, \tilde{X} \in \mathbb{R}^{n\times d}, \tilde{\tilde{Y}} \in \mathbb{R}^n)$, penalty grid $\Lambda \in \mathbb{R}^l$
Returns: estimators $\hat{B}$, test results Φ

```
Initialize estimator array B̂ ∈ ℝ^{l×d}
Initialize test results Φ ∈ ℝ^l
for i ∈ range(l) do
    λ ← Λ[i]
    Compute β̂ = argmin_β(Σ_{i=1}^n (Y_i − f(X, X̃, β))^2 + λ Σ_{j=1}^d |β_j|)  ▷ see Alg. 2
    Compute test statistic T(β̂) from Eq. 4.3
    Test whether H_0(s) can be rejected:
        φ_s(X, X̃, Y) = 𝟙(T(β̂(s)) > F^{-1}_{n−d,d}(1 − α))
    Set B̂[i] ← β̂,  Φ[i] ← φ_s
end for
```

Algorithm 2 Coordinate gradient descent with unpaired data

Input: tuple of unpaired data $(X \in \mathbb{R}^{n\times d}, \tilde{X} \in \mathbb{R}^{n\times d}, \tilde{\tilde{Y}} \in \mathbb{R}^n)$, penalty $\lambda \in \mathbb{R}$
Returns: estimator $\hat{\beta}$

```
Initialize estimator β̂ ∈ ℝ^d ← 0
for i in range(iters) do
    for j in range(d) do
        Compute influence of j-th covariate on residual:
            κ ← X̃^⊤(Ỹ − Xβ̂ + β̂[j] ∗ X[:, j])
        Compute similarity between independent covariates:
            ϑ ← X̃X[:, j]
        Aggregate:
            ρ ← (1/n) Σ κ ∗ ϑ
            ζ ← (1/n) Σ_i ϑ_i^2
        Perform soft thresholding:
        if ρ < −λ then
            β̂[j] ← (ρ + λ)/ζ
        else if ρ > λ then
            β̂[j] ← (ρ − λ)/ζ
        else
            β̂[j] ← 0
        end if
    end for
end for
```

4.4 Experimental Setup

To assess the capabilities of our approach compared to existing models, we examine two challenging datasets. Firstly, we evaluate the proposed method on a completely synthetic dataset, where we have full control over the entire data generating process including interventional setups. Secondly, we test the performance on a semi-simulated dataset in which only the outcome is simulated. The covariate data under interventions is derived from real-world measurements in the human genome [55]. This selection of applications covers datasets commonly used in literature while simultaneously studying effectiveness in the context of complex datasets relevant to real-world use-cases.

4.4.1 Gold-standard synthetic benchmark data

Structural Equation Model based on Directed Acyclic Graph: Our simulated benchmark data is generated in two stages, using a framework that resembles an inference setup with instrumental variables. In the first stage, we identify a set of independent instruments $I_1, \ldots, I_m$ that have no causal relationship with the outcome variable Y and do not share any common causes with it. We then create a DAG structure with nodes corresponding to the covariates $X_1, \ldots, X_d$, and add the instrument variables as independent root nodes connecting each instrument to every covariate. In the final stage, we randomly select a subset of covariates to serve as parents to the outcome variable Y. Our focus is on scenarios where the causal parents of Y are distributed sparsely among the covariates X. We utilize the above-mentioned DAG structure as the causal model underlying Eq. 4.1. In order to satisfy the linearity assumption of standard IV methods, we enforce linear node functions for all connections. To determine the strengths of the causal relations, we sample the edge strengths uniformly from $\mathcal{U}((-1.5, -0.5) \cup (0.5, 1.5))$. Given that the instruments are root nodes in the DAG, we assign distributions to them which might vary across our series of experiments (see below for details). In general, the intervention strength is related to the width of the instrument distribution. During simulation, we perform q interventions [187, 3] each targeting by default exclusively one instrument and collect data from n repetitions for the covariates and the outcome. The effect of the number of children per instrument variable is subject to our studies below. Importantly, an intervention on some of the variables does not change the assignment of any other variable. In particular, an intervention on I does not change the conditional distribution of Y, given X and H. This can be thought of as an instance of the invariance property [34, 3]. Our evaluation methodology involves stochastic soft interventions that adjust the distribution of the intervened instrument which is commonly seen in gene knock-down experiments and shift interventions [177], which maintain the confounding structure of the original assignment but shift it linearly.

This setup has been extensively researched in the case of *paired* data and numerous results haven been reported in literature [157, 158, 159, 176]. The first type of interventions is confounding removing, resulting in a unique solution for the underlying

causal function (as per Proposition 3.1 in [186]). Based on our assumption of linearity for all causal relationships, the shift interventions we employ suffice to yield unbounded variability in every direction of X's covariance matrix which is sufficient to obtain a unique solution for the underlying causal function (see Proposition 3.2 in [186]). Hence, using paired samples under the set of considered intervention types can generally guarantee causal identifiability which we also demonstrate in our evaluation. Therefore, we adopt this setup as our first benchmark case and extend it to the *unpaired* data scenario.

Specifically, we generate a paired dataset ($\bar{X} \in \mathbb{R}^{q \times n \times d}, \bar{Y} \in \mathbb{R}^{q \times n \times 1}$) based on the simulation following the structural equation model in Eq. 4.1. To transform this paired dataset into an unpaired setup, we divide the n observations for each intervention into three equally sized but entirely disjoint subsets, resulting in the following data partition (gray colored data not available in the unpaired setting):

$$(X \in \mathbb{R}^{q \times \frac{n}{3} n \times d}, Y \in \mathbb{R}^{q \times \frac{n}{3} \times 1})$$
$$(\tilde{X} \in \mathbb{R}^{q \times \frac{n}{3} n \times d}, \tilde{Y} \in \mathbb{R}^{q \times \frac{n}{3} \times 1})$$
$$(\tilde{\tilde{X}} \in \mathbb{R}^{q \times \frac{n}{3} n \times d}, \tilde{\tilde{Y}} \in \mathbb{R}^{q \times \frac{n}{3} \times 1})$$

Note that the samples themselves are always unpaired, but they are generated from the same data generating process. We conducted a series of experiments to evaluate our approach under different conditions which are described in the following:

(E1) *Intervention strength:* We varied the strength of the instruments with range $\sigma_I \in (0.5, 4.0)$ affecting the covariates to test the effect of intervention strength.

(E2) *DAG sparsity:* We adjusted the probability of sampling edges when generating the DAG to test the impact of DAG sparsity, i.e. $P(G^{DAG}_{i,j}) \in (0.1, 1.0)$.

(E3) *Number of affected covariates per intervention:* We varied the number of covariates affected per instrument $n_{children} \in \{1, \ldots, |X|\}$ to evaluate performance when instruments influence multiple covariates simultaneously.

(E4) *Number of available interventions:* We tested under- and overdetermined cases by adjusting the number of available instruments relative to the number of covariates, i.e. $n_{int} \in \{2^i\},\ i \in \{0, \ldots, 8\}$.

(E5) *Distribution of instruments:* We assigned different distributions to the instruments, including Gaussian, Laplace, uniform, Rayleigh, and a mix of the mentioned distributions, to assess the impact of the instrument distribution.

(E6) *Number of affected nodes and distribution shapes:* We combined experiments (E3) and (E5) to test the joint effect of the number and distribution of affected nodes.

Table 4.1: Dataset overview of the synthetic database: A series of of experiments under varying conditions is simulated including variation in intervention strength (E1), DAG sparsity (E2), number of affected covariates per intervention (E3), number of available interventions (E4), distribution of instruments (E5), number of affected nodes (E6), confounding strength (E7), confounding sparsity (E8), and number of samples (E9).

	Exp. E1	Exp. E2	Exp. E3	Exp. E4	Exp. E5	Exp. E6	Exp. E7	Exp. E8	Exp. E9
Intervention strength	{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0}	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
DAG edge probability	0.8	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8. 0.9, 1.0}	0.8	0.8	0.8	0.8	0.8	0.8	0.8
Num. effected covariates by IV	1	1	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}	1	1	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}	1	1	1
Distribution IV	Gaussian	Gaussian	Gaussian	Gaussian	{Gaussian, Laplace, Uniform, Rayleigh, Mix}	{Gaussian, Laplace, Uniform, Rayleigh, Mix}	Gaussian	Gaussian	Gaussian
Num. available interventions	50	50	50	{1, 2, 4 , 8, 16, 32, 64, 128, 256}	50	50	50	50	50
Confounder strength	0.5	0.5	0.5	0.5	0.5	0.5	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8. 0.9, 1.0}	0.5	0.5
Confounder edge prob.	0.8	0.8	0.8	0.8	0.8	0.8	0.8	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8. 0.9, 1.0}	0.8
Num. observed samples	1000	1000	1000	1000	1000	1000	1000	1000	{10, 17, 30, 54, 95, 167, 294, 517, 910, 1600}

(E7) *Confounding strength:* Similar to (E1), we introduced confounding of varying strengths $\sigma_{conf} \in (0.1, 1.0)$ to evaluate its impact.

(E8) *Confounding sparsity:* We adjusted the sparsity of confounding connections between X and Y, i.e. $P(G_{i,j}^{conf}) \in (0.1, 1.0)$ to test the effect of confounding sparsity.

(E9) *Number of samples:* Finally, we assessed the influence of sample size by varying the number of observed samples $n_{samples} \in \{10, \dots, 1600\}$.

Table 4.1 summarizes the different experimental setups and the corresponding variables at change.

4.4.2 Semi-simulated benchmark with human gene data

Recent large-scale perturbation experiments on human cells [55] contain a large database of gene expression measurements under genetic perturbations providing a database of genotypic causes and their effects on downstream mRNA levels. The dataset includes perturbations and expression measures for thousands of genes in two cell types - a leukemia line (K562) and retinal cells (RPE). There are 2285 and 2679 detected genetic interventions in the two cell types respectively, with median coverage of over 100 cells per intervention. This provides a database of genotypic causes (the perturbed genes) and their effects on the expression of many potential

downstream covariates. To complete the instrument variable framework, we introduce an outcome variable Y that depends linearly on a sparse subset of the covariate genes. This models a scenario where only some of the expression levels causally influence the outcome. The intervened genes provide exogenous variation in the full set of covariates, allowing us to uncover this small causal subset that are the parents of Y. This sets up a simulated phenotype Y that depends on a sparse set of covariate gene expression levels, which in turn depend on the larger set of genetically perturbed instrument variables. Hence, this dataset provides a database of instruments and their effects on potential covariates. By simulating Y as a sparse linear function of covariates, we can use the instruments to uncover the causal relationships from genotype to phenotype, with covariate gene expression levels as the intermediates between genetic instruments and outcome.

Technical details: In a first step, we follow the normalization routine described in [55, 188]. Then, we filter out genes that are insufficiently effected by the performed interventions. We discard interventions for which an insufficient number of cells was measured, i.e. we require at least $n = 60$ or more cells per intervention. We create two lines of experiments that use the interventional data differently:

(1) *Interventional data as intrumental variables:* The first experimental setup treats the interventional data as instruments. This setup is very similar to the setup used in Section 4.4.1 except that the data of the instruments is given by real-world measurements. We define a gold standard for causal relationships similar to [12]. Specifically, we calculated a robust z-score $\zeta_{i,j}$ that quantified the change in gene j under intervention on gene i, relative to the observational variation in gene j. That is, $\zeta_{i,j}=|\text{med}(I^{int}_{i,j})-\text{med}(I^{obs}_{i,j})|/\text{IQR}(I^{obs}_{i,j})$, where $\text{med}(I^{int}_{i,j})$ and $\text{med}(I^{obs}_{i,j})$ denote the median of the gene expression levels under intervention and in the truly observational case of the wildtype distribution respectively and $\text{IQR}(I^{obs}_{i,j})$ defines the interquartile range of the wildtype distributions. We concluded there is an experimentally verified causal relationship from gene i to gene j if and only if $\zeta_{i,j} > \tau$. That is, we inferred a sufficiently large causal effect of gene i on gene j when the intervention on gene i changed the expression of gene j by more than $\tau > 5$ interquartile ranges relative to the observational variation in gene j. This approach allows us to include sufficient variance in the instrumental variables. To compute the covariates X and the outcome Y, we follow the simulation framework as explained above.

(2) *Using interventional data as covariates:* In the second setup, we opt to employ interventional data as samples for the covariates, which seems intuitive at first. However, defining the confounding variables H becomes less clear in this arrangement. While the proposed method does not explicitly require the presence of confounding, evaluating it under such conditions is desirable, as confounding generally complicates predictions and certainly, exists in real-world applications. To preserve confounding, we adopt the following approach: the confounding variables H must belong to the subset of variables $\mathcal{I}$ that were

intervened upon during the experimentation. Additionally, the confounders need to be part of the intersection $\mathcal{D} = \mathcal{I} \cap \mathcal{O}$, where $\mathcal{I}$ represents the set of all intervened genes, and $\mathcal{O}$ denotes the set of observed variables. In other words, we can only consider those genes as confounding variables H for which we generally observe gene expression levels and additionally, are subject to interventions. We select covariates X based on two constraints: (i) there should be a substantial causal relationship in terms of the z-score from the confounders H to the covariates X and (ii), the instruments I need to act as causal predecessors of the covariates X, while not exerting any influence on the confounders H. Ideally, this arrangement ensures that the covariates are ancestors of the confounders and the instruments. However, it is essential to note that the relationships between confounders and covariates are not necessarily linear, and a causal connection is likely but not certain, as there could be unmeasured confounders that explain detected relations. Although our setup explicitly avoids direct connections between instrument variables and confounders, it is not possible to guarantee the absence of any causal influence from instrument to confounders through indirect paths or unknown or unmeasured confounders. Using the extracted covariates X and confounders Y, we proceed with the simulation protocol for generating Y, as defined in Equation 4.1 and detailed in Section 4.4.1.

To complete the causal setup, a small subset of the covariate genes is randomly selected to be the direct causal parents of the outcome variable Y. The goal is to examine scenarios where the genes that directly influence Y are sparse, meaning they make up a small fraction of the full set of covariate genes. This models a situation where only a limited number of genes out of the many observed expression levels have a direct causal effect on the phenotype Y. By using simulated outcome data while retaining the real instrument and covariate data, we can ensure there is no inherent confounding between the instruments and outcome that could undermine causal inference. However, the relationships between the instruments and covariates may still involve non-linearity or hidden confounding, as they reflect the complex real gene perturbation effects. Also, the true interventions done in the experiment do not perfectly meet the ideal intervention assumptions outlined in [187], as real gene perturbations can have off-target effects. In other words, the simulated outcome avoids built-in confounding, but complexities in the real instrument-covariate connections remain. The interventions are not ideal, but still provide useful variation in the covariates for inferring causal relationships.
By selecting appropriate subsets of perturbed genes as instruments and affected genes as covariates, the data can be used to uncover genotypic causal effects on a simulated phenotype drawn from the covariate gene expression levels.

4.5 Results

We conducted experiments to evaluate the generalization performance and robustness of our approach. First, using a fully synthetic dataset, we tested generalization by

individually varying simulation parameters and observing the effect on our estimator. This allowed us to isolate the impact of each parameter change.
Additionally, we evaluated our model on a semi-simulated dataset derived from human gene expression measurements across two cell lines. By using real gene expression data, this experiment tested whether our model can handle scenarios where experimental conditions are not fully controlled, such as noise, confounding, and non-linearity. Overall, these experiments aimed to assess the applicability of our model to both synthetic data where we control all parameters, as well as real-world data where experimental factors are imperfect.

4.5.1 Gold-standard synthetic data

We first assess the individual effects of simulation parameters on our proposed model's performance. To evaluate performance, we compare our model to three benchmark estimators: (1) an unpaired OLS estimator with covariance adjustment, (2) a paired OLS estimator using the true subset of nonzero coefficient covariates, and (3) a standard paired OLS estimator. For each simulation experiment, we report the root mean square error (RMSE) between the predicted and ground truth β-coefficients. We also report the RMSE of outcome Y predictions and present the learned causal structures using our model, including the signed distance to the ground truth. Together, these performance metrics allow us to thoroughly evaluate our proposed model under various simulation settings against relevant benchmark estimators.

(E1) Intervention strengths: In our initial experiment, we manipulate the intervention strength by increasing the variance of the distributions of the simulated instrumental variables. The underlying idea is that a wider distribution of an intervened instrument makes its impact on the corresponding covariate more detectable. By employing an algorithm that utilizes oracle-like data of the non-zero ground truth coefficient, we can deduce that a minimum intervention strength is necessary to enable causal inference in an unpaired setup. This is evident from the erroneous predictions for $\hat{\beta}$ and the decreasing root mean square error in Y since the intervention strength increases, see Figure 4.2. Figure 4.2b demonstrates that both unpaired ordinary least squares (Unp.-OLS) and our method are influenced by the intervention strength. Unp.-OLS assigns high values to all coefficients in $\hat{\beta}$, resulting in elevated RMSE values in Figure 4.2a. Conversely, our approach incorporates a penalization term, leading to a coefficient vector with small values across the board, lacking informative content. Consequently, the RMSE is reduced to a mere line in Figure 4.2a due to the minimal variation in the coefficient vector. As the intervention strength increases, both algorithms exhibit competitive performance and yield the actual causal structure.

(E2) DAG sparsity: This experiment demonstrates that the density of the adjacency matrix of the directed acyclic graph underlying the covariates is not a significant factor for any of the algorithms. The findings are visualized in Figure C.1 in the

Appendix.

(E3) Number of affected covariates per interventions: The findings of this experiment are depicted in Figure 4.3. As the number of ancestors per intervention in the covariates increases, the performance declines. The unpaired OLS estimator proves to be ineffective in scenarios where interventions impact nearly all covariates. Consequently, the predicted β-coefficients deviate significantly from the ground truth, resulting in poor predictions for the outcome variable Y. This observation holds true in general, however to a smaller extent, for both the penalized version and unpaired OLS on the oracle data subset. The experiments conducted with the oracle unpaired OLS suggest that recovering the causal structure becomes challenging when interventions on the instruments induce changes in the distributions of (almost) all covariates.

(E4) Number of available interventions: The performance in both underdetermined and overdetermined interventional setups is explored in Figure 4.4. Existing literature suggests that, in cases of abundant data, it is generally feasible to identify the causal structure when the number of interventions matches or exceeds the number of non-zero beta coefficients. However, our findings indicate that this assertion does not hold true for an unpaired experimental setup with limited data. The oracle unpaired ordinary least squares estimator demonstrates ineffectiveness when the number of interventions is less than $n = 8$. Consequently, the standard unpaired OLS estimator and our penalized counterpart are also not expected to perform well in such scenarios. The unpaired OLS estimator displays competitive performance when $n = 32$ interventions are available, while the penalized estimator requires a minimum of $n = 16$ interventions. This is evident from the decreasing root mean square error (RMSE) in Figure 4.4a for Unp.-OLS, as well as the erroneous predictions of β^* in 4.4b for both Unp.-OLS and the penalized version.

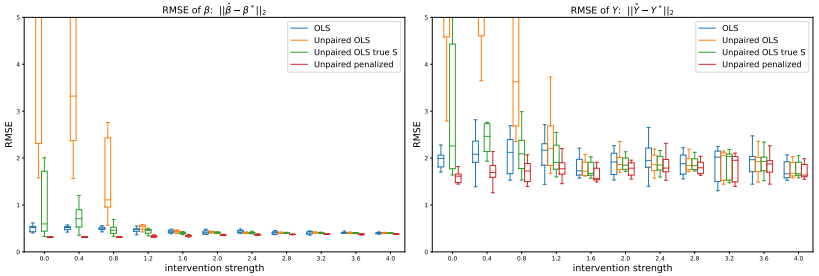
(E5) Distribution of instruments: The impact of different distributions of instrumental variables is illustrated in Figure 4.5. It is intriguing to observe that the oracle unpaired ordinary least squares (OLS) estimator is minimally affected by the choice of distribution in the instrumental variables. In contrast, the standard unpaired OLS algorithm proves ineffective when the instrumental variable (IV) data is distributed uniformly or according to a Rayleigh distribution. However, the setup utilizing a mixture of distributions appears to be less affected. Since the number of available interventions is set to $n = 50$, the valuable information obtained from the useful distributions is sufficient. Interestingly, the penalized version of the algorithm does not exhibit a comparable behavior. Our method consistently and reliably restores the causal relationship for all distributions, with only slightly higher errors observed for the Rayleigh distribution (see to Fig. 4.5b).

(E6) Number of affected nodes and distribution shape: As this experiment combines aspects from (E3) and (E5), we anticipate a similar pattern of behavior as observed in those experiments. This observation is validated in Figure C.2 of the Appendix.

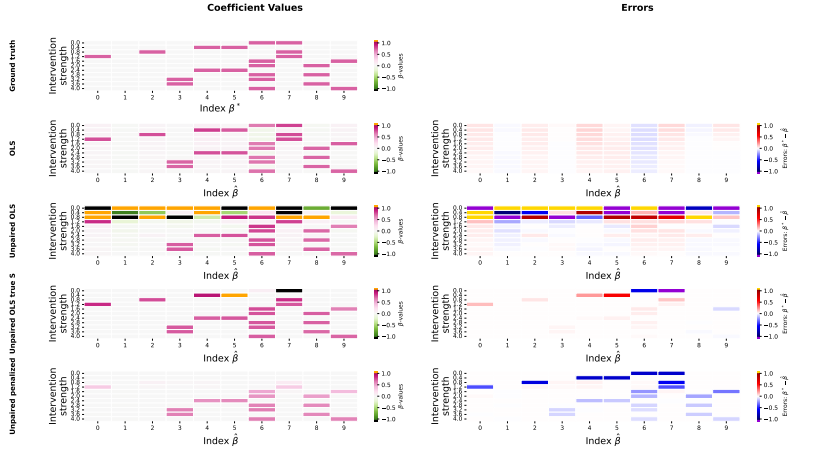
Once again, we can observe increasing errors for $\hat{\beta}$ and $\hat{Y}$ as the number of ancestors among the covariates increases. This effect is further amplified by the mixture of distributions employed in the unpaired OLS estimator.

The confounding strength investigated in experiment (E7) and the sparsity in confounder-outcome relations (experiment (E8)) have only a minor effect on the performance of all algorithms. The results are presented in Figure C.3 and C.4 of the Appendix.

(E9) Number of samples: The final experiment explores the impact of sample size on performance. The findings depicted in Figure 4.6 indicate that a minimum of at least $n = 17$ samples is necessary in the paired setup. Given that the unpaired estimators leverage covariance structures, we anticipate that a higher number of samples is required for the algorithms to operate effectively, and this expectation is validated. The oracle unpaired ordinary least squares (OLS) estimator demonstrates effectiveness at a sample size of $n = 54$, while both the standard unpaired OLS estimator and our proposed penalized method exhibit efficacy starting at a sample size of $n = 95$.

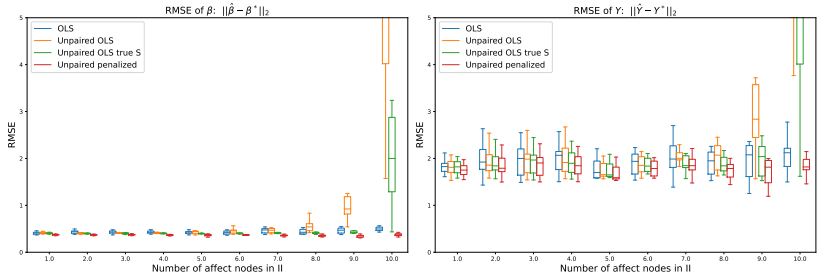


(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$

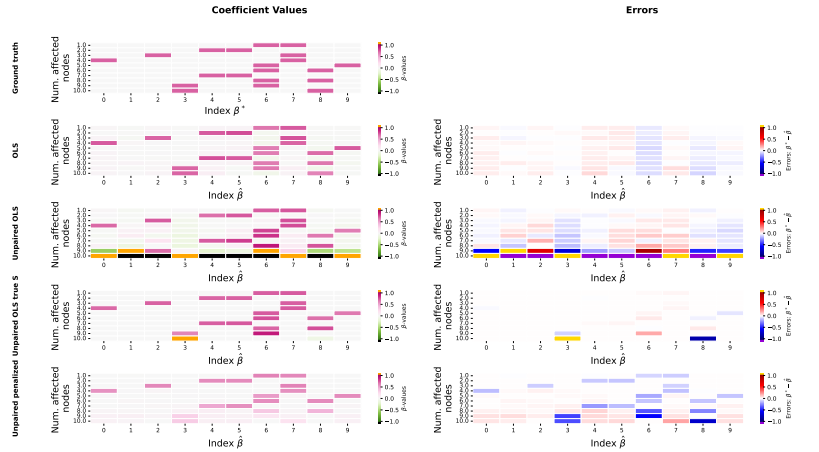


(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure 4.2: Experiment (E1): Intervention strength. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.

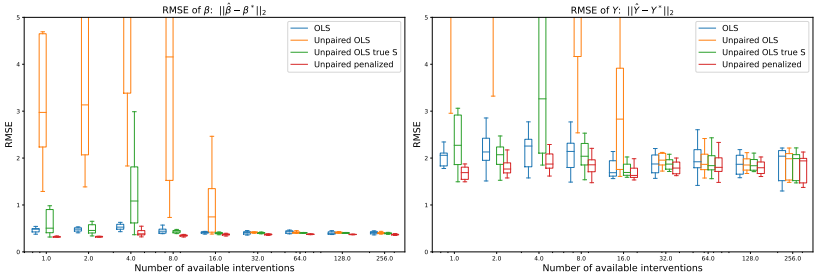


(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$

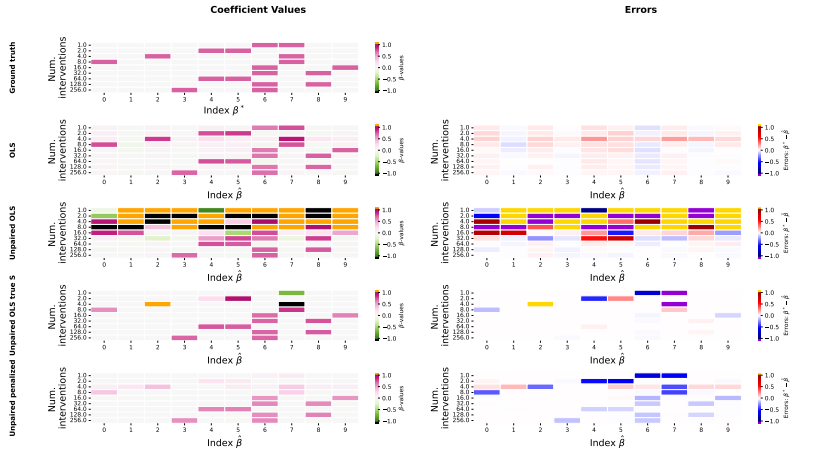


(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure 4.3: Experiment (E3): Number of affected covariates per intervention. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.

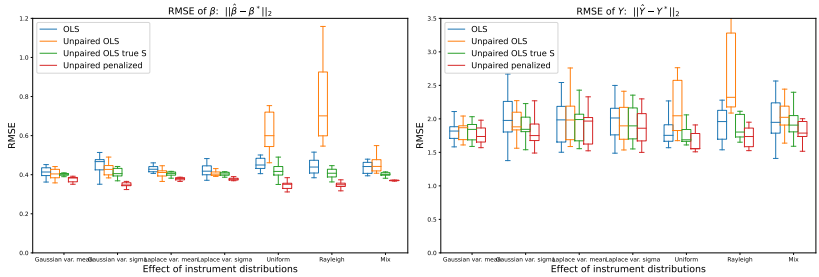


(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$

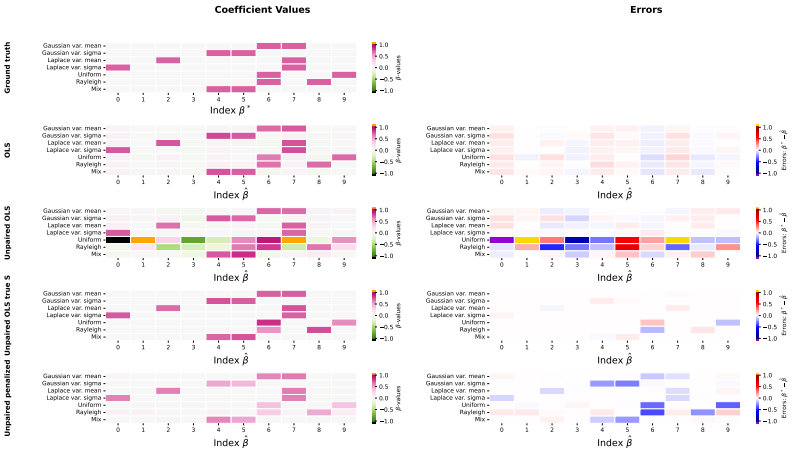


(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure 4.4: Experiment (E4): Number of available interventions. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.

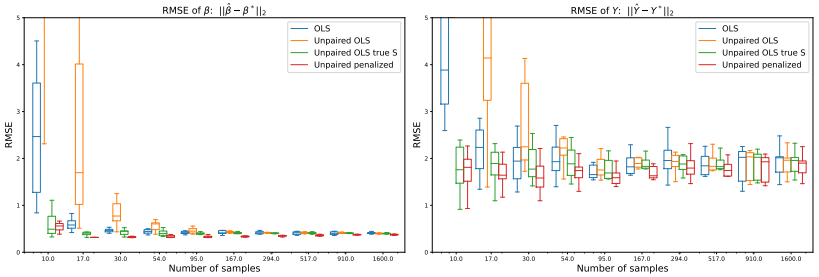


(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$

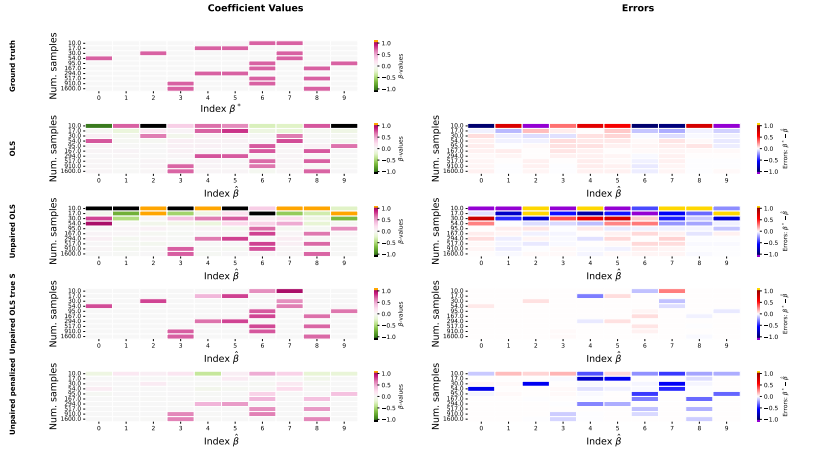


(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure 4.5: Experiment (E5): Distribution of instrumental variables. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.



(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$



(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure 4.6: Experiment (E9): Number of samples. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.

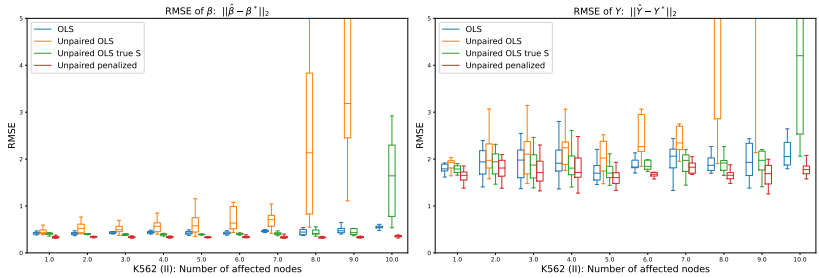
4.5.1.1 Semi-simulated human gene data

In addition to the synthetic benchmark cases, we conduct a series of experiments utilizing interventional data obtained from human gene knock-down measurements. For two cell-lines, namely a cancerous blood cell-line (K562) and retinal pigment epithelium tissue (RPE), we apply the same metrics and benchmark algorithms as presented in Section 4.5.1. This allows us to evaluate our proposed method in a real-world setting. We explore two different experimental setups, each varying in the utilization of interventional measurements (refer to Sec. 4.4.2 for detailed information).

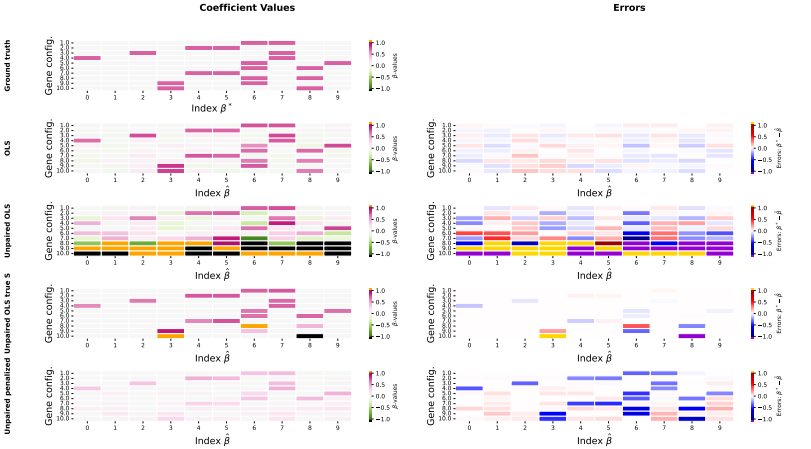
Using Interventional Data as Instrumental Variables: In this setup, we utilize interventional gene expression levels as instrumental variables to simulate the covariates X and the outcome Y. This approach allows us to account for noise in the covariates and address confounding between X and Y by employing gene distributions as instrumental variables. The results, depicted in Figure 4.7 for K562 and Figure 4.8 for RPE, present the outcomes. We conduct the tests using ten different gene configurations, whereby each repetition involves a distinct set of genes and interventions as instruments. These gene configurations are selected to fulfill the assumptions of the IV framework. As we increase the gene configuration index, we intentionally introduce more violations of assumptions, such as less expressive interventions or noisier interventions. This allows us to evaluate the algorithm performance on gene configurations that closely resemble synthetic cases as well as those that are less well-defined due to measurement noise, unmeasured confounding, and other sources of noise. We repeat each gene configuration experiment ten times with different random seeds and present the averaged results. For both cell lines, we observe that the OLS method applied to paired data successfully identifies the correct causal structure, although the estimates exhibit slightly higher errors compared to the fully synthetic test case. This indicates the presence of a detectable regression signal in the data. When examining the unpaired OLS estimator with the oracle data setup, we find that the predicted quantitative results are significantly more erroneous compared to the simulation benchmarks. This observation also holds for the standard unpaired OLS estimator. Figures 4.7b and 4.8b illustrate that the error in non-zero β-coefficients is higher, with an increasing trend for more challenging gene configurations. Specifically, the predictions for gene configurations 8, 9, and 10 lose their usefulness and informativeness. In contrast, we note that our proposed method exhibits higher quantitative errors compared to the simulation benchmark, but it reliably restores the causal structure even for the more challenging gene configurations.

Using Interventional Data as Covariates: In this setup, we employ interventional gene expression levels as actual covariates and solely simulate the outcome Y. Finding gene configuration setups characterized by strong confounding between X and Y requires highly stringent data requirements. To identify suitable candidate configurations, we had to relax the z-score threshold to $\tau > 3$. Moreover, the presence of confounding on the covariates is very sparse. We identified five gene configurations and averaged the

results over ten runs with different random seeds, as done previously. The outcomes are presented in Figure C.5 and C.6 in the Appendix. Within this setup, none of the considered algorithms demonstrate effectiveness, most likely due to the presence of noise. The complexity of the current problem is exacerbated by the vast number of potential genetic and environmental causes in large biomedical studies, rendering exhaustive interventional experiments implausible. Additionally, it is generally impossible to eliminate confounding bias caused by unmeasured latent variables that influence the associations between biomarkers and outcomes. Furthermore, when dealing with mRNA transcript levels as biomarkers, the measurements are known to be considerably noisy. In this context, it is important to note that we cannot guarantee the absence of unmeasured confounding between instruments and the outcome Y through connections involving the selected genes acting as confounders H. In quantitative genetics, such applications of instrumental variable methods are referred to as Mendelian randomization [189]. As per requirements of classic instrumental variable methods, it is assumed that the effects of the genetic instrument on a covariate are unconfounded, and the effects of the instrument on the outcome are solely mediated through the covariate [190]. However, the assumption of no hidden pleiotropy significantly limits the applicability of this approach, as most genotypic effects on complex traits lack sufficient understanding to exclude pleiotropy as a possible explanation for an association. In other words, a pleiotropic gene exhibits multiple phenotypic expressions, meaning that a mutation, such as CRISPR/Cas gene knock-down, in a pleiotropic gene may simultaneously affect several traits due to the gene coding for a product utilized by numerous cells or different targets with the same signaling function.

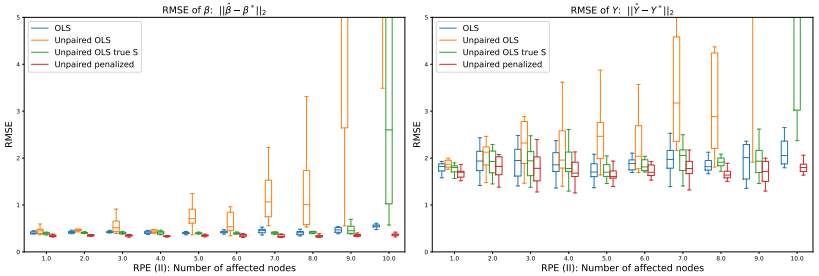


(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$

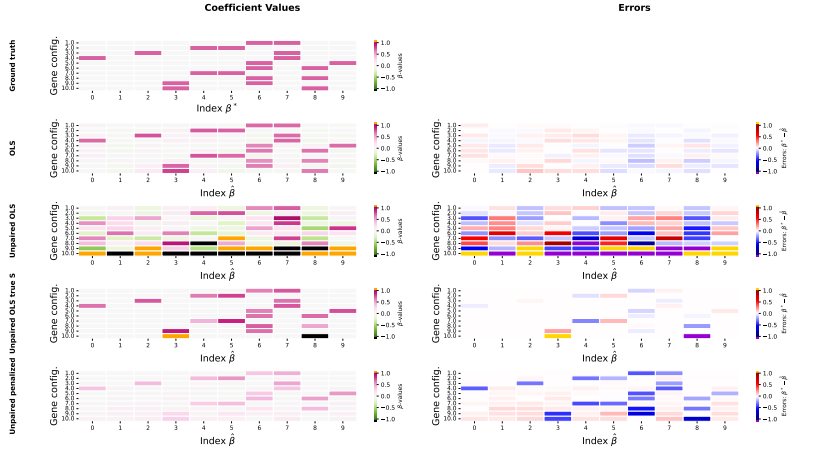


(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure 4.7: K562 gene expression levels as instrument variables. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.



(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$



(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure 4.8: RPE gene expression levels as instrument variables. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.

4.6 Conclusion and Discussion

Across the nine experiments of our synthetic benchmark test cases, we systematically evaluate the performance of our proposed regularized regression method in comparison to three benchmark algorithms and investigated various aspects of causal inference in an unpaired setup. Different intervention strengths, sparsity of the underlying graph, number of affected covariates per intervention, distribution of instruments, confounding strength and sparsity, and sample sizes were examined. The results showed that intervention strength affects the performance of the algorithms, with a minimum strength required for accurate causal inference. The density of the underlying graph did not significantly impact the algorithms. Increasing the number of affected covariates per intervention led to decreasing performance, particularly for the unpaired estimator. The number of available interventions was crucial, with a minimum requirement for accurate results. The choice of distribution for instrumental variables affected the algorithms differently. The penalized method consistently restored the causal relationship, while the standard unpaired OLS algorithm struggled with certain distributions. Lastly, the experiments highlighted the importance of sample size. The oracle unpaired OLS estimator demonstrated effectiveness with fewer samples, while the other algorithms required larger sample sizes for reliable performance. In summary, the experiments provided insights into the factors influencing causal inference in unpaired setups, emphasizing the significance of intervention strength, number of affected covariates, distribution of instruments, and sample size. Furthermore, our experiments demonstrate that our method consistently yields sparse coefficient estimates, while OLS produces dense solutions. This aligns with the regularization in our approach, which encodes a bias for sparsity. Moreover, our method proves effective at identifying the true causal relations even with limited samples and interventions. By contrast, standard OLS struggles to recover sparse structures without substantial data. Overall, the empirical results highlight the advantages of our proposed framework in terms of sparsity and sample efficiency.

Moreover, our experiments involve the use of interventional data in two different setups. In one setup, interventional gene expression levels were used as instrumental variables to simulate the covariates and the outcome. The results show that the proposed method performs well in restoring the causal structure, even for challenging gene configurations. In another setup, interventional gene expression levels were used as actual covariates while only simulating the outcome. However, none of the algorithms yield effective results in this case, most likely due to the presence of noise and sparse confounding on the covariates. The complexity of this problem is attributed to the large number of potential genetic and environmental causes in biomedical studies. The assumption of no hidden pleiotropy, which refers to the effects of a gene on multiple traits, posed limitations to the application of instrumental variable methods. Most genotypic effects on complex traits cannot be sufficiently understood to exclude pleiotropy as a possible explanation for associations. Overall, the study demonstrated the challenges and limitations in utilizing interventional data

and instrumental variable methods, particularly in the context of genetic research and complex traits.

5 Estimating Treatment Effects using Deep Neural Networks

5.1 Introduction and Motivation

In the clinical environment, decision makers such as physicians face several crucial questions that must be addressed carefully before taking any action. Typically, these questions encompass whether to administer a treatment, which specific treatment to choose, and the optimal timing for initiating the treatment. However, answering these questions is not a trivial task and requires a reliable estimation of the anticipated effects of a treatment. In this context, clinical trials traditionally serve as the gold standard for addressing these questions and often report an average treatment effect (ATE). However, it is crucial to recognize that the same treatment can have diverse impacts on different individuals, as evidenced by experience and real-world observations. Estimating these heterogeneous treatment effects (HTEs) is still a state-of-the-art challenge and requires the development of novel, sophisticated algorithms that are able to predict the individualized future effectiveness of a specific treatment plan. In essence, these models offer insights into diverse individual responses to various treatments. This aids physicians in making better-informed choices about treatment options for patients, customizing plans to suit individual requirements. Our aim revolves around foreseeing the future responses of patients based on their ongoing medication regimen. Specifically, we are interested in forecasting future state e.g. to enable early identification of inadequate response to ongoing therapy. This enables the identification of insufficient individual treatment plans and timely transitions to more promising medication strategies. In this Chapter, we primarily focus on a systemic disease, i.e. wet age-related macular degeneration (wet AMD), which is characterized by macroscopic symptoms like the growth of abnormal blood vessels beneath the retina. Its onset is triggered by the breakdown of the retinal pigment epithelium (RPE) and is associated with inflammation and angiogenesis. It is useful to contrast this with cancer research, which while a very different area of medicine, offers useful contrasts and similarities with respect to causal interplay. In the case of cancer, the data tends to be at the molecular level, but in terms of modelling and theory there are similarities in the need to go beyond ATEs towards understanding personalised response to therapy. Cancer involves the unchecked proliferation and division of abnormal cells capable of infiltrating nearby tissues and disseminating to distant body regions. This uncontrolled growth primarily stems from genetic mutations that disrupt the conventional regulatory pathways governing cell division and growth. Consequently, the scope of observations for these distinct disease categories spans a wide spectrum. For wet AMD, it encompasses medical

images, assessments of visual acuity, and diverse clinical outcomes. In contrast, for various cancer types, it encompasses mainly cellular and molecular readouts such as cell proliferation rates and gene expression assays.

The overwhelming majority of current models employed in this challenging setting are based on novel machine learning algorithms and leverage large diverse data sources to provide more profound HTE estimations. "High-Throughput Screening" has revolutionized the drug discovery process by providing a rapid and systematic way to identify potential drug candidates. Its combination of automation, speed, and large-scale testing has significantly increased the efficiency of finding compounds with therapeutic potential, leading to improved results in drug screening experiments. In addition, novel technological approaches at large scale including single-cell RNA sequencing or single-cell mass cytometry have enabled the profiling of molecular and functional attributes of individual cells within a complex mixture to uncover rare cell types and study cell-to-cell variations [191]. This has led to the development of large public datasets with thousands of experiments testing different drugs on different types of cell lines. These datasets are exceptionally valuable and enable scientists to investigate the relationship between the molecular profile of a cell (for example its DNA and other related biological information) and its phenotypic response to a specific medical treatment [192, 193, 194, 195]. However, the use of these datasets is still in its early stages and the primary objective of analyzing these datasets has been to construct a predictive model for drug response, at least from a data science perspective. These models should be designed to take in longitudinal image or probe data and forecast the anticipated outcome when employing a candidate drug for treatment. Consequently, the higher the accuracy of such a predictor, the greater the level of trust it instills, as it faithfully emulates wet lab results.

To summarize, understanding the effects of treatments over time is crucial for the implementation of complex treatment plans and personalized healthcare in real-world scenarios. That is, only in the longitudinal setting it is possible to gain insights into how diseases progress under different treatment plans, how individual patients respond to treatment over time, and the optimal timing for administering treatments. However, estimating counterfactual outcomes in the longitudinal setting presents additional challenges, with the most significant being the potential dependency of observed treatment assignments on time-varying confounding variables (time-dependent confounding, [196]). For instance, let us consider the treatment of leukemia patients. Not all cancer patients are equally likely to receive the same chemotherapy regimen, and their past treatment responses and history influence future treatment choices [197]. Consequently, this introduces a bias in causal effects and increases the variance in counterfactual estimation due to systematic differences in the distribution of confounding variables between different treatment sets over time. In other words, the primary challenge of causal inference over time lies in addressing a time-dependent confounding and distribution shift, which is not encountered in standard time-series analysis. Therefore, conventional time-series models are not suitable for this setting

[198].

Previous work in causal inference has attempted to address this confounding bias, as demonstrated in [199, 200, 201]. However, their approaches rely on strong assumptions that do not align with most real-world observational data. Specifically, these methods assume that the data is regularly sampled at fixed, evenly spaced time intervals, and that the sampling times perfectly align across different individuals. In contrast, observational data is typically not sampled at regular intervals due to practical reasons such as simple scheduling issues, e.g. patients missing an appointment, or more complex considerations, such as more frequent observations for severe cases or variations in monitoring requirements for different treatments. Therefore, such assumptions often do not hold in practice, significantly limiting the applicability of these approaches.

Motivated by these challenges, we have devised and evaluated supervised end-to-end models to estimate drug responses based on highly irregular time-series data, e.g., clinical patient records. Specifically, our focus lies on predicting individual treatment effects for patients afflicted with wet age-related macular degeneration (wet AMD). This condition is a leading cause of vision loss and affects the macula, a critical part of the retina responsible for sharp central vision. Detecting and intervening early are crucial for a successful treatment as the early stages may not present noticeable symptoms. Hence, regular eye exams are necessary for timely intervention, as untreated wet AMD can lead to severe vision loss and legal blindness. To gain a comprehensive understanding of this disease progression and treatment response, the database under consideration incorporates Optical Coherence Tomography (OCT) scans taken at multiple time points. Moreover, each patient's longitudinal trajectory is further enriched by accompanying information, including demographic characteristics, medical history, administered treatment drugs, visual acuity measurements, and other clinical outcomes.

Leveraging this multi-variate set of information, our overall goal is to design a sophisticated treatment effect estimator that predicts a continuous *future* treatment effect for individual patients given its past data trajectory. In contrast to other work reported in literature, here we are not interested in predicting individual treatment effects for a set of available drugs to substantiate the choice of medication, but focus on modeling the treatment effects of an existing medication plan to estimate its future (continuous) efficiency. In this setting, major challenges arise from the fact that the underlying dynamics of the highly irregularly sampled record data are learned inherently and aggregated to estimate the effectiveness of the designed medication plan. That is, relevant system interventions are assumed, but remain unknown to the learner relative to both, the exact interventional method as well as its precise timing. Moreover, our models have to account for the inherent uncertainty in the investigated (noisy) clinical record collection, considering measurement errors, natural retina variation, and drug response heterogeneity. To tackle these exceptional chal-

lenges, we capitalize on recent breakthroughs in machine learning and investigate the prediction accuracy of three fundamentally different approaches, e.g., sparsity promoting linear regression, deep Gaussian processes and attention-based deep neural networks. We test the performance of the methods empirically and demonstrate the effectiveness of our approach via comparing the results to expert evaluations given by two long-term ophthalmologists.

The remainder of this Chapter is organized as follows. In Section 5.2, we present a review of related literature. Section 5.3 introduces the notation and defines the problem, followed by the presentation of the considered methods in Section 5.4. In Section 5.5 and 5.6, we conduct a comprehensive evaluation of the considered models through a series of diverse tests. Finally, in Section 5.7, we draw conclusions, discuss limitations, and propose potential areas for improvement.

5.2 Related Work

Since this work primarily engages with traditional and recent approaches for averaged and heterogeneous treatment effect estimation, a short literature overview will be provided in the following. We start with available datasets used for drug response studies to characterize current requirements for our envisioned learner, and continue with existing machine learning methods reported in literature. Here, we specifically focus on work with time-varying covariates, treatments, and outcomes, but also draw on insights from causality in dynamical systems and recent work on modeling controlled differential equations. As outlined above, we explicitly note the difference between causal inference over time and conventional time series modeling and hence do not focus on recent advances in time series models.

Large-scale drug screening datasets for HTE estimation
The high-throughput screening of compounds is an important step in the drug discovery process. Its underlying purpose is to determine which medication or series of medications will potentially result in effective treatments. In recent years, several large-scale anti-cancer drug screens have been performed and were made publicly available. Well-known projects such as Genomics of Drug Sensitivity in Cancer (GDSC) [192], Cancer Cell Line Encyclopedia (CCLE) [193], Cancer Therapeutics Response Portal (CTRP) [202, 203] and NCI-60 [204] provide access to drug sensitivity profiles for a wide variety of cancer cell lines. In addition to the dose-response data gathered from the high-throughput screening experiments, these databases also provide access to omics data characterizing the cancer cell lines that the compounds were screened against. Precisely, all three projects provide genomic, transcriptomic and epigenomic data. Additionally, proteomics [205, 206] and metabolomics [207] data are available for the NCI-60 and CCLE cell lines.
Other studies, e.g., the Connectivity Map (CMap) [208, 209] or the Library of Integrated Network-Based Cellular Signatures (LINCS) [210, 211] projects, similarly provide data regarding the response of cells to treatment, but focus on small

molecules at the proteomic and epigenomic levels [212]. These cellular response signatures can complement the data from other drug screening initiatives and may be a very valuable source of information when building drug response prediction models. Complementary studies in this field focused on large pan-cancer drug combination screening datasets which also have been made available to the public (see for example [213, 214, 215]). These datasets can serve similarly as the basis for the development of deep learning based drug response prediction models.

Another important area for predictive modeling of drug-induced gene expressions beyond cancer cell research is individualized drug repurposing for Alzheimer's disease (AD). However, conventional target-based compound screening that follows the one-drug-one-gene drug discovery paradigm has low success rates for AD due to its multi-genic systemic effect. Hence, a comprehensive systems pharmacology strategy is required that targets whole gene regulatory networks. To enable such phenotypic screening, it is critical to utilize a mechanistic phenotype readout to link drug responses in a model system to drug toxicity and efficacy. Frequently used datasets in this research topic include data from the ROSMAP project [216], the MSBB project [217], and the MayoRNAseq [218] project. Within the ROSMAP project, multi-level omics, neurobiologic traits, and structural and functional neuroimaging are collected for approximately 3400 participants. Complementary work within the MSSB project generated whole genome RNA-sequencing and proteome profiling data from multiple regions of 364 postmortem traits, and collected rich clinical and pathophysiological data. Researchers of the MayoRNAseq project pursued a similar approach collecting whole genome genotype, microarray-based whole transcriptome and RNA-sequencing data from the Mayo Clinic eGWAS study [219].

Further pathologies that have been studied in the context of individualized drug response prediction involve detecting and treating first-episode drug-naive (FEDN) schizophrenia. Recently reported studies [220, 221] leverage schizophrenia biomarkers to establish diagnosis and make individualized predictions of future treatment responses to antipsychotics. Based on mutual information and the correlations of brain activities measured by functional MRI, dysconnectivity between cortical regions in patients were discovered. Other studies used electroencephalography signals to monitor and diagnose schizophrenia disease [222, 223, 224] and published their extensive clinic records.

Ongoing long-horizon studies focus on the collection of a more comprehensive dataset combining various state-of-the-art techniques. These endeavors are not exclusively concentrated on estimating heterogeneous treatment effects, but encompass abundant data that also holds significance for causal modeling. One promising work in this active research area is the Rhineland study [225] examining their more than 3000 participants repeatedly every 3-4 years over the course of their lives. The examinations include visual acuity, optical coherence tomography (OCT), magnetic resonance imaging, physical activity intensity data, genomic, transcriptomic and epigenomic se-

quencing, neuropsychological test data, blood samples and EEG data [226, 227, 228, 229, 230, 231]. This extensive set of diverse data sources opens the door for completely new approaches to characterize neurodegenerative and age-related diseases and will help to understand the complex interrelationships in HTEs of such diseases.

Learning based approaches for treatment effect estimation
Leveraging these diverse dataset sources, HTE estimation has been studied in great detail in the recent machine learning literature. Early work built mainly on tree-based methods, but many other methods, such as Gaussian processes and neural networks, have been adapted to estimate HTEs recently.
Mentionable work w.r.t. tree-based approaches for individual drug response estimation leverages a Bayesian modeling procedure called Bayesian Additive Regression Trees (BART) [232]. The authors contend that BART possesses the ability to detect interactions and non-linearities in the response surface, thereby making it well-suited to identify HTEs more effectively. In contrast to methods like propensity score matching and subclassification, BART naturally generates coherent posterior intervals.
Complementary work builds on regression tree methods, which is modified to optimize for goodness of fit in treatment effects and to account for honest estimation [233]. By partitioning the data into subpopulations that differ in the magnitude of their treatment effects, the proposed method enables the construction of valid confidence intervals for treatment effects, even with many covariates relative to the sample size. In follow-up work, Breiman's widely used random forest algorithm is applied for estimating HTEs [234]. In their framework, it is suggested that causal forests are pointwise consistent for the true treatment effect and have an asymptotically Gaussian and centered sampling distribution. The presented experiments find causal forests to be substantially more powerful than classical methods based on nearest-neighbor matching, especially in the presence of irrelevant covariates. Subsequent work evidences relevant generalization and performance gains when using an adaptive weighting function designed to express heterogeneity in the specified quantity of interest [235].

Leveraging a fundamentally different approach, Alaa et al. [236] developed a Bayesian method for learning the treatment effects using a multi-task Gaussian process (GP). The authors suggest to use a linear coregionalization kernel as prior to compute individualized measures of confidence in inferred estimates via pointwise credible intervals, which they argue are crucial for realizing the full potential of precision medicine. The impact of selection bias is alleviated via a risk-based empirical Bayesian method for adapting the multi-task GP prior, which jointly minimizes the empirical error in factual outcomes and the uncertainty in (unobserved) counterfactual outcomes. Based on this intial work, principled guidelines for building estimators of treatment effects are derived by characterizing the fundamental limits of estimating HTEs, and establishing conditions under which these limits can be achieved [237].
Motivated by the idea that the uncertainty in the counterfactual distributions can

be learned by a neural network, Yoon et al. [238] developed a Generative Adversarial Network (GANs) framework to estimate individualized treatment effects (ITEs). The proposed method, termed Generative Adversarial Nets for inference of Individualized Treatment Effects (GANITE), generates proxies of the counterfactual outcomes using a counterfactual generator, and forwards these proxies as training inputs to an ITE generator. By modeling both, the authors argue to infer correct ITEs based on the factual data, while still accounting for the unseen counterfactuals.

Arguably the largest stream of current work builds on neural networks, due to their flexibility and ease of manipulating loss functions, which allows for easy incorporation of balanced representation learning. Relevant initial work addresses counterfactual inference as a type of domain adaptation problem, and derive a novel way of learning representations [239]. The proposed models which are based on fully connected layers rely on regularized representations which have similar distributions among the treated and untreated cohorts. In more detail, the first hidden layers are used to learn a representation of the input. The output of the following layer is then used to calculate the discrepancy between the factual and counterfactual distribution. Finally, the last layer takes as additional input the treatment assignment and predicts the patient's response.
In a similar fashion, Shalit et al. [240] suggest to learn a "balanced" representation such that the induced treated and control distributions have a similar shape and statistical characteristics. In their work, a novel and intuitive generalization-error bound is derived showing that the expected ITE estimation error of a representation is bounded by the sum of the standard generalization-error and the distance between the treated and control distributions. However, some underlying assumptions, e.g. a well-specified model or prior knowledge about the policy that gave rise to the observed data, do not resemble realistic settings. To tackle this problem, [241] introduces a new bound on the generalization error incorporating both representation learning and sample re-weighting. Based on this bound, their novel algorithmic framework outperforms existing method on synthetic benchmarks under more realistic assumptions.
Contrarily, Hassanpour et al. [242] argued that not all factors in the observed covariates might contribute to the procedure of selecting treatment, or more importantly, determining their outcomes. They introduced a new algorithm called Disentangled Representations for CounterFactual Regression (DR-CFR), that can identify disentangled representations of the underlying data generating process and leverage this knowledge to reduce and account for the negative impact of selection bias on estimating the treatment effects from observational data.

More recently, [243] showed that the use of balancing weights complements representation learning in mitigating the covariate imbalance. Specifically, they link balance to the quality of propensity estimation, emphasize the importance of identifying a proper target population, and highlight the complementary roles of feature balancing and weight adjustments. Their claims are supported with theoretical results and

evaluations on synthetic datasets and realistic test benchmarks, reporting competitive performance throughout.

However, one major shortcoming of the methods described above stems from their strong observation sampling assumptions as they exclusively consider regular, discrete-time intervals between observations and treatment decisions and hence, are unable to naturally model the de facto standard of irregularly sampled data in practice. In the following, we will review further neural approaches that are specifically designed to handle arbitrary observation patterns. From a high-level perspective, further extensions have been proposed that interpret the data as samples from an underlying continuous-time process and propose to model its latent trajectory explicitly. These methods also differ by how they adjust for confounding and for differences in covariate distributions in different treatment regimes.
First work in this topic engages Marginal Structural Models (MSMs) [199]. MSMs are linear in treatment and covariate effect, and create a pseudo-population using inverse probability of treatment weighting, as the probabilities of treatments are independent w.r.t. time-varying confounders. Thus, MSMs can effectively control for confounding bias. With regard to deep learning based methods, Lim et al. [200] proposed a semi-parametric alternative to MSMs using recurrent neural networks (R-MSN). The model is subdivided into two parts, a set of propensity networks to accurately compute the inverse probability of treatment weightings, and a sequence-to-sequence architecture to predict responses using only a planned sequence of future actions. Based on this framework, R-MSN demonstrated performance improvements over traditional methods for joint treatment response prediction over multiple future time steps on synthetic datasets.
The Counterfactual Recurrent Network (CRN) [201] uses a similar architecture but instead leverages adversarial training to balance differences in covariate distributions in different treatment regimes.

Recent fundamentally different approaches leverage the outstanding capabilities of neural ordinary differential equations (NODEs) [104] and their numerous extensions, e.g., neural controlled differential equations (NCDEs) [112, 244], to model irregular time series data. However, neural ODE based methods are conventional time series models, which do not account for issues such as time-dependent confounding. To tackle this shortcoming, Seedat et al. [245] proposed Treatment Effect Neural Controlled Differential Equation (TE-CDE) that allows the potential outcomes to be evaluated at any time point. In the context of intervention modeling, Gwak et al. [246] proposed to use separate ODEs for interventions and outcome processes Their work applies to systems with deterministic dynamics in the absence of time-varying confounders.

All approaches summarized above share one common characteristics as they study the problem of inferring HTEs for binary or continuous outcomes. Recent work extends these frameworks via leveraging time-to-event data to estimate both the effects

of treatments on instantaneous risk and survival probabilities. Pioneer work in the context of machine learning methods was reported by Curth et al. [247] learning individual treatment estimation and discrete-time treatment-specific conditional hazard functions from time-to-event data. Instead of modeling event times directly, the authors adapt neural networks for the estimation of treatment-specific hazard functions. Subsequently, this learner is used to directly compute survival functions, i.e. mean survival time and hazard ratios. Follow-up work [248] studies the problem of inferring HTEs from time-to-event data in the presence of competing events. It is shown that inclusion of competing events not only leads to multiple definitions of effects but also to multiple sources of covariate shifts. As a result, future work will be required to model these new dependencies sufficiently.

5.3 Problem statement and notation

In this work, we explicitly focus on the task of predicting individual treatment effects for patients afflicted with wet age-related macular degeneration (wet AMD). Wet AMD is characterized by the gradual breakdown of light-sensitive cells in the macula and the growth of abnormal blood vessels beneath the retina leading to severe vision loss and legal blindness in final disease states. Our primary objective targets the detection of insufficient therapy after diagnosis to allow a timely transition to an alternative treatment plan at early stages.
Framed under these challenging conditions,we designed a sophisticated end-to-end trainable treatment effect estimator that is able to predict reliably a continuous *future* treatment effect for individual patients given its past data trajectory. Specifically, we focus on the setting in which available inputs are

(I1) preprocessed longitudinal OCT scans that provide statistical features, e.g. the thickness of the different layers of the retina or estimated disease-related fluid volumes, and

(I2) individual examination protocols providing relevant information on demographic characteristics and medical history only available at a highly irregular time grid.

For (I1), we assume that the raw OCT scans cover all relevant eye sections with a sufficient signal-to-noise ratio. Data preprocessing and quality assessment is carried out by a semantic segmentation approach under the supervision of medical experts which is not part of this work. For (I2), we assume precise patient records to transform absolute medical trail recordings to relative time-series data.

Suppose all input requirements are fulfilled, the underlying learning task of our approach can be formulated as follows: Unlike other, our focus is not on predicting heterogeneous treatment effects for a set of available drugs to guide medication selection. Instead, we concentrate on modeling the treatment effects of an existing

medication plan to estimate its future (continuous) efficiency concerning individual patients. In this context, significant challenges arise due to the complex dynamics underlying the highly irregularly sampled record data, which are learned and aggregated to estimate the effectiveness of the designed medication plan. Further challenges arise from the fact that relevant system interventions are assumed (and indeed performed), but their exact interventional method and precise timing remain unknown to the learner.

In the following, we introduce a notation to formalize the investigated problem setup. Let X be the input of size $(N \times T \times D)$. Entries are denoted $X_{n,t,d}$ with n identifying the patient and t being the snapshot index in time of available longitudinal patient features of dimension D. We further assume that for each of N different patient records, we have access to an observed trajectory of T time points. We need to know the sampling times themselves, but these need not be evenly spaced neither do we assume that each patient's trajectory comprises the exact same number of T OCT scans. In fact, such an assumption would significantly limit the applicability of the designed learner since clinical patient record characteristics, such as timing and number of performed examinations, are typically highly individualized.

With the notation above, our goal is to learn a continuous treatment effect estimator predicting future patient responses treated under an unknown but reasonable medication plan. To this end, we train a parameterized learner F_θ, i.e. a linear or non-linear function F with a set of trainable parameters θ. This is possible since training is performed in a supervised fashion assuming a carefully designed dataset with known ground-truth labels. The ground-truth is based on expert domain knowledge and is described in Section 5.5.1. Note that training is performed prior and independently to inference. Hence, we require our learner F_θ to generalize well to *unknown* presumably out-of-distribution patient records in an out-of-the-box fashion during inference. The architectures we use for F_θ are detailed below, but for now assume that this has been specified. Then, given the training data X_{train} (which might be independent in feature distribution from our inference data X_{test}), we train the set of parameters θ under a supervision loss using the labels Y_{train}. In our setting, we consider the negative log likelihood (NLL) between ground truth and predictions

$$\mathrm{NLL}(\theta) = -\sum_{i=1}^{N} \log P(Y_i|X_i;\theta) \tag{5.1}$$

where $P(Y_i|X_i;\theta)$ represents the probability of the correct label Y_i given the input X_i and model parameters θ. Assuming a suitable training dataset with corresponding labels, the optimization of this loss is carried out for multiple different learners and enables sufficient generalization capabilities as demonstrated empirically in Section 5.6.

5.4 Architectural details

In this work, we investigate the performance of three fundamentally different approaches with rising complexity. These include a Bayesian linear regression method augmented by a sparsity-promoting horseshoe prior, deep Gaussian processes leveraging a stochastic variational inference algorithm, and a temporal attention based deep neural network. Individual details are presented in the following.

5.4.1 Linear regression via a sparsity-promoting horseshoe prior

Regression analysis, while simple, is of central focus in statistics, data analysis and machine learning research as it provides useful yet easily interpretable relationships in many areas involving, amongst others, finance, healthcare, physics, and engineering. In supervised learning, the linear regression problem can be cast as the problem of estimating a set of coefficients that determine a strictly linear relationship between a set of inputs X and a target variable Y

$$Y = F_\theta(X) = Xw + b, \tag{5.2}$$

with w denoting the learnable weighting coefficients (slopes) and b is the linear bias (intercept). Due to this rather simple learning scheme, linear regression analysis tends to overfit in high-dimensional problems. That is, in order to avoid overly complex and barely interpretable models, some form of dimensionality reduction is required. This entails finding sparse solutions, where some elements of the learnable weighting coefficients w remain zero. In this work, we adopt a well-known shrinkage approach which is based on the horseshoe prior [249]. The horseshoe prior is a member of the family of multivariate scale mixtures of normals, and is therefore closely related to widely used sparsity-promoting methods for sparse Bayesian learning such as LASSO [250] or Student-t priors [251]. The horseshoe prior assumes that each entry in w is conditionally independent with density $\pi_{HS}(w_i|\tau)$, where π_{HS} can be formalized as a scale mixture of normals:

$$(w_i|\lambda_i, \tau) \sim N(0, \lambda_i^2\tau^2) \tag{5.3}$$

$$\lambda_i \sim C^+(0, 1) \tag{5.4}$$

Here, C^+ denotes a half-Cauchy distribution for standard deviation λ_i. In this context, the λ_i's refer to the local shrinkage parameters while τ characterizes the global shrinkage.

We intentionally choose this prior distribution since it provides two useful features in the context of sparsity promotion. First, its flat, Cauchy-like tails allow strong signals to remain large a-posteriori while its infinitely tall spike at the origin provides relevant shrinkage for the zero elements in w [249]. As a result, the horseshoe prior is robust for handling unknown sparsity and large outlying signals at the same time, which are typical challenges in treatment effect estimations.

5.4.2 Gaussian processes

The second approach which is used within this work leverages Gaussian Processes (GPs) to model the desired treatment effect estimation. Employing GPs in challenging applications, e.g. treatment effect estimation with continuous outcomes, proved to be effective in literature and provides several advantages [198]. First, GPs exhibit adaptability by increasing their complexity to approximate the data closely even in non-linear cases. Second, they demonstrate robustness against overfitting and provide accurate uncertainty estimates. Third, GPs possess the capability to model a wide range of functions using only a few hyperparameters, making them highly flexible and efficient in handling irregularly sampled datasets. Nevertheless, the expressiveness of the kernel/covariance function in single-layer GP models is limited, as shown in various studies [252, 253, 254, 255]. In contrast, a Deep Gaussian Process (Deep GP) overcomes the limitations of standard GPs while retaining their advantages by introducing an ordered stack of GPs. Deep GPs are more expressive models compared to standard GPs, in the same way how deep neural networks show a higher expressiveness in comparison to generalized linear models [256].
Given a set of input points X and corresponding function values Y, a GP generates a joint distribution over the function values as a multivariate Gaussian distribution. The mean function $\mu(X)$ and covariance function $\sigma(X, X')$ are used to define the parameters of this Gaussian distribution. We denote any covariance function hyperparameters as θ. In the light of a potentially large dataset, we employ a set of inducing points $Z = (z_1, \ldots, z_K)^T$ [257, 258] to summarize the behavior of the GP over the entire input space. These inducing points act as representatives of the entire dataset and allow us to make predictions based on a reduced set of points rather than considering all data points. Let $P(Y|F_\theta)$ define the likelihood and F being normally distributed, we can write the joint density as

$$P(Y, F_\theta(X), F_\theta(Z)) = \underbrace{p(F_\theta(X)|F_\theta(Z), X, Z)p(F_\theta(Z)|Z)}_{\text{GP prior}} \underbrace{\prod_{i=1}^{N} P(Y_i|F_\theta(X_i))}_{\text{likelihood}}. \quad (5.5)$$

Deep Gaussian Processes

A Deep Gaussian Process is a GP extension that integrates the principles of deep learning. In traditional GPs, the kernel function represents assumptions about the smoothness and structure of the underlying function which, in some cases, might be too complex for what a single kernel can express. Deep GPs overcome this limitation by introducing a hierarchical architecture with multiple layers, reminiscent of neural networks. In a Deep GP, each layer is a GP on its own, i.e. the first layer GP takes the raw input data, and each subsequent layer GP operates on the outputs of the previous layer GP. This arrangement enables each layer to have its own kernel function, which can either be based on prior assumptions about the data or learned from the data itself. Similar to [256], we define a prior recursively over different

stochastic functions $F_\theta^1, \ldots, F_\theta^L$. Note that the prior on each function F_θ^L is an independent GP in each dimension. Then, the resulting joint density reads

$$P(Y, F_\theta(X^l), F_\theta(Z^l)_{l=1}^L) = \underbrace{\prod_{i=1}^N P(Y_i|F_{\theta,i}^L)}_{\text{likelihood}} \underbrace{\prod_{l=1}^L P(F_\theta(X^l)|F_\theta(Z^l), F_\theta(X^{l-1}), Z^{l-1}) P(F_\theta(Z^l)|Z^{l-1})}_{\text{Deep GP prior}}. \tag{5.6}$$

During optimization of this Deep GP, the posterior retains the full conditional structure of the true model. This, however, comes at the cost of losing the analytical tractability, but due to the sparse posterior within each layer we can sample the bound using univariate Gaussians.

In the present work, we employ a Deep GP consisting of three single-layer GPs. To extract temporal patterns in the longitudinal input data, a temporal feature extractor is applied which comprises one-dimensional convolutions and multiple stacked Multi-Layer-Perceptrons (MLPs). These time-series features are then forwarded to the Deep GP which finally outputs the desired continuous treatment effect.

5.4.3 Temporal attention based deep neural networks

Motivated by recent successes based on transformer networks, we introduce an attention-based [259] encoder architecture for modeling treatment effects based on longitudinal patient records that can handle noisy and partially observed high-dimensional data. To this end, we employ a transformer-based encoder with a time-aware attention [117] and relative positional encodings, which efficiently handles data on an arbitrary time grid. These modifications provide useful inductive biases and allow the encoder to effectively operate on input sequences with a temporal component. Similar to [260, 117] the encoder computes

$$h_{\theta_{enc}}(X) = h_{read}(h_{agg}(h_{comp}(X))), \tag{5.7}$$

where

1. h_{comp} compresses observations into a low-dimensional sequence.
2. h_{agg} aggregates information of the low-dimensional features into a time-aware representation
3. h_{read} linearly transforms the extracted time-aware feature space into the required output domain.

Transformations h_{comp} and h_{read} may be any suitable differentiable functions, but here we employ stacked MLPs. The transformation h_{agg} is a transformer encoder

which is a sequence-to-sequence mapping represented by a stack of L layers. Each layer $l \in 1, \ldots, L$ contains a component called attention sub-layer, which - in the standard case - comprises a combination of the dot-product (C_{ij}^{DP}), softmax (C_{ij}), and attention score (β_i) and reads

$$C_{ij}^{DP} = \frac{\langle W_Q h_i, W_K h_j \rangle}{\sqrt{D_{low}}}, \quad C_{ij} = \frac{\exp(C_{ij}^{DP})}{\sum_{k=1}^{N} \exp(C_{ik}^{DP})}, \quad \beta_i = \sum_{j=1}^{N} C_{ij}(W_V h_j), \tag{5.8}$$

where W_Q, W_K, W_V are learnable layer-specific parameter matrices, and C is the attention matrix. However, this standard formulation of self-attention [259] works poorly on irregularly sampled data since it operates upon discrete and equidistant steps rather than a continuous scale as required in our time dependent weighting scheme. To reweigh the individual β_i's based on the distance in time, i.e. a time-aware feature representation, we follow [117] and augment the original dot-product attention with a temporal attention module C_{ij}^{TA}. The redefined attention matrix [117] reads

$$C_{ij}^{TA} = \ln(\epsilon) \left(\frac{|t_j - t_i|}{\delta_r} \right)^p, \quad C_{ij} = \frac{\exp(C_{ij}^{DP} + C_{ij}^{TA})}{\sum_{k=1}^{N} \exp(C_{ik}^{DP} + C_{ik}^{TA})}, \tag{5.9}$$

where $\epsilon \in (0, 1]$, $p \in \mathbb{N}$ and $\delta_r \in \mathbb{R}_{>0}$ are constants. Hence, the larger the distance $|t_j - t_i|$ grows, the stronger the time-aware attention is reduced [261]. The parameter δ_r determines the distance threshold beyond which the scaling of C_{ij}^{DP} occurs by at least ϵ. Moreover, parameter p governs the shape of the scaling curve.

5.5 Experimental Setup

We study the performance of the presented algorithms on a new longitudinal dataset of wet AMD patients. Age-related macular degeneration is a leading cause of vision loss among individuals. It affects the macula, a small yet critical part of the retina responsible for sharp central vision, which is essential for reading, driving, and recognizing faces. In particular, wet AMD is characterized by the gradual breakdown of light-sensitive cells in the macula and marked by the growth of abnormal blood vessels beneath the retina. These vessels are fragile and tend to leak fluid and blood into the surrounding tissue, causing rapid and severe damage to the macula. The early stages of wet AMD may not present noticeable symptoms, making regular eye exams essential for early detection and timely intervention. If left untreated, wet AMD can lead to severe vision loss and even legal blindness. Optical Coherence Tomography (OCT) is a cutting-edge imaging technique that has revolutionized the diagnosis and management of wet AMD. It provides detailed cross-sectional images of the retina, allowing eye care specialists to visualize the extent of the damage caused by the abnormal blood vessels. OCT scans enable early detection of wet AMD, facilitate accurate monitoring of disease progression, and guide treatment de-

cisions.

5.5.1 Datasets

The dataset used in this study comprises clinical records from a large cohort of patients diagnosed with wet AMD. All data preprocessing steps including normalization, imputation, and transformations were carried out by Nastassy Horlava. Johannes Wahle contributed the LMM preprocessing routine (see below).

The records were collected over a period of more than 10 years, providing a longitudinal perspective on the patients' medical histories and treatment interventions. Each patient's data includes diverse information, such as demographic characteristics, medical history, treatment drugs administered, visual acuity measurements, and other relevant clinical outcomes. To obtain a comprehensive understanding of the disease progression and treatment response, the dataset incorporates OCT scans taken at multiple time points. These scans are preprocessed using a neural network specialized in semantic segmentation, enabling the quantification of retinal layer thickness and the identification of disease-induced fluids between these layers. The primary focus of this study is to predict the individual patient responses to treatment using longitudinal data with multiple time points available. The starting point for each patient is defined as t=0, representing the date of the first treatment dose. The goal is to predict the individual quantitative treatment response after 90 days, i.e., at t=90. The dataset is divided into two subsets for evaluation:
(i) The first subset includes data from patients with two visits, n_{visits}=2. In addition to the initial measurements at t=0, there are additional measurements taken at approximately t=30 days.
(ii) The second subset uses data from three time points, $t \in \{0, 30, 60\}$, to predict the treatment response after 90 days.

The evaluation consists of two distinct tasks, each aiming to assess the predictive performance of the models on the different datasets:

1. *Regression task:* We predict the relative change of the disease related fluid volume $V(t)$ in relation to the volume measured at t=0, i.e.

$$\Delta V = \frac{V(t{=}90) - V(t{=}0)}{V(t{=}0)}. \tag{5.10}$$

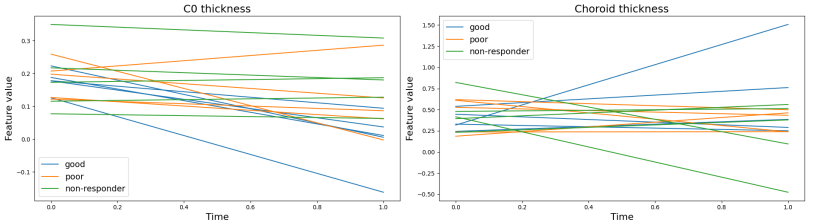
2. *Classification task:* We employ a thresholding function to assign the class labels - *good responder*, *poor responder*, *non-responder* - to each patient based

on the relative change. The labels are defined as follows:

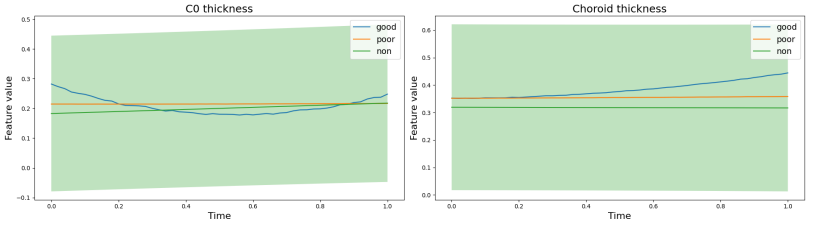
$$\tau(V(t)) = \begin{cases} \text{good responder} & \Delta V \leq -0.25 \\ \text{poor responder} & -0.25 < \Delta V \leq 0.01 \\ \text{non-responder} & \Delta V > 0.01 \end{cases} . \tag{5.11}$$

Note that we study the performance of our proposed algorithms based on two additional variants of this dataset. That is, further preprocessing and filtering is applied to extract valuable and informative data patterns more easily and to enhance the quality of subsequent analyses. One such preprocessing pipeline is the Linear Mixed Model (LMM), which offers a comprehensive and robust framework to handle the intricacies and challenges associated with longitudinal data analysis. By employing a LMM as a preprocessing step, we can effectively model the inherent correlation structure in the longitudinal data by including random effects for individual patients. This random effect captures the variability between patients and accounts for the within-subject correlation, providing a more accurate representation of the true relationships between variables over time. Furthermore, a LMM can handle unbalanced data, allowing for the inclusion of patients with different numbers of visits while appropriately handling missing data. This aspect is crucial in longitudinal studies like wet AMD, where patients may leave the study or miss visits for various reasons. In fact, this methods further provides the advantage to sample a patients trajectory at a finer time scale. Exemplary LMM-resampled patient trajectories are shown in Figure 5.1a.

Motivated by the fact that more sophisticated statistical techniques are well-equipped to capture complex interactions between variables, the second preprocessing scheme at use targets the inherent non-linearities in the wet AMD data. Here, we choose a non-parametric Gaussian Process. By defining a prior distribution over functions and updating it based on observed data, GP regression can effectively capture complex and non-linear relationships between variables, making it particularly well-suited for longitudinal analyses of wet AMD patient records. One of the key benefits of using a GP in this context is its ability to accommodate irregularly spaced and unbalanced data. Unlike parametric models that require fixed time points for each patient, GP regression seamlessly handles missing data and varying visitation patterns, ensuring that no valuable information is discarded due to data incompleteness. Our implemented GP preprocessing pipeline trains an overall Gaussian process per class which is then conditioned to each longitudinal patient trajectory in turn. For all test data points, we decide which GP to use based on the highest likelihood given the longitudinal data of the test trajectories. Similar to the LMM, we also sample data points on a finer time grid. The trained class priors are illustrated in Figure 5.1b.



(a) Linear mixed model preprocessing for two features.



(b) Gaussian process preprocessing for two features.

Figure 5.1: Additional preprocessing approaches: Two exemplary features are shown, the retina thickness of subsection C0 and the thickness of the choroid layer surrounding the retina. (a): Linear Mixed Model (LMM), (b): Gaussian Process. Subpopulations are indicated by color, i.e., good responders in blue, poor responders in orange, and non-responders in green.

5.5.2 Training and evaluation setup

We partitioned the dataset into two distinct subsets denoted as $\{\tilde{\mathcal{D}}, \mathcal{D}_{test}\}$. The subset $\tilde{\mathcal{D}}$ was utilized for training and validation purposes, employing a 5-fold cross-validation approach, while the test dataset $\mathcal{D}_{test}$ remained completely separate and was never used during any training or validation stages.

Training: To train the Bayesian linear model, we employed Pyro [262] with stochastic variational inference. A low-rank multivariate normal distribution was utilized as a guide, and an Adam optimizer [47] with a learning rate of $lr = 1e-3$ and an exponential learning rate scheduler with $\gamma = 0.995$ was utilized. Each model underwent 2000 epochs of training. We implemented the Deep GP model and the temporal attention network using PyTorch [45] with the same learning rate setup. The Deep GP model and the temporal attention network were trained for 1000 epochs and 300 epochs, respectively. All models were optimized using the negative log-likelihood as the reconstruction loss.

Test Setup and Evaluation: To evaluate the models, we performed testing on the

held-out subset $\mathcal{D}_{test}$, which consisted of 93 longitudinal data trajectories. For the regression task, we computed the overall and class-wise Root Mean Squared Error (RMSE) and included scatter plots to visualize the combination of ground truth and predicted values for further analysis. For the classification task, we used the confusion matrix and three class-wise binary metrics (F1 score, recall, and precision) to assess model performance. We averaged the performance metrics over all models trained in the 5-fold setup. Additionally, we extended the evaluation to the entire patient population by concatenating all prediction results from all validation folds, allowing us to present the overall performance on all data points.

5.6 Results

In this Section, we present the outcomes of our comprehensive analysis based on the three derived datasets (initial dataset, LMM preprocessed, and GP preprocessed) comprising clinical records of wet age-related macular degeneration (AMD) patients. To investigate the complex dynamics of wet AMD and predict individual patient responses to treatment, we employed state-of-the-art statistical methodologies, including Bayesian linear models, Deep Gaussian Processes, and temporal attention networks. We present and discuss the results obtained from each model and evaluation metric, shedding light on the efficacy of our methodologies in modeling wet AMD dynamics and predicting treatment responses.

Figure 5.2 and Figure 5.5 display the classification and regression performance of the considered methods on the initial dataset with $n_{visits} = 2$ and $n_{visits} = 3$, respectively. In the first row of these figures, the F1 score, precision, and recall values for the test dataset are presented. The second row shows the same metrics for the different patient subpopulations, i.e. good responders, poor responders, and non-responders. Additionally, the subfigures include the prediction baselines of two human ophthalmologists who were asked to estimate the class label based on the patient's OCT scans. Upon analyzing the overall dataset in the first rows of the figures, it becomes evident that the Deep GP exhibits strong performance, as indicated by its F1 score, precision, and recall, which either match or surpass human classification performance. However, this apparent superiority is deceiving and stems from the dataset's inherent class imbalance. A more thorough examination of the regression performance in Figure 5.2 unveils a noteworthy issue as it assigns a singular value to all input data points. This disparity is visualized in the scatter plots in Figure 5.2.

The effectiveness of the Bayesian linear regression model is found to be limited. In Figure 5.2, it can be observed that the Bayesian linear regression approach assigns a wide range of predicted values to a relatively small range of ground truth values, as evidenced by the vertical line of dots. We attribute this behavior to two possible reasons. First, the assumption of linearity between X and Y does not hold in this case. Second, after optimization, the distribution over non-zero parameters might become excessively broad, resulting in a wider range of potential values in the pre-

diction space for slightly varying input values.

The temporal attention neural network demonstrates the strongest performance among the considered models. The correlation between its predicted responses and the ground truth is notably high, with a RMSE three times lower compared to the Deep GP and ten times lower than the Bayesian linear regression. In terms of classification performance, the temporal attention neural network exhibits also strong results. The class-wise F1 score matches the performance of the two experts for good and non-responders, and it even outperforms them for the class of poor responders. This trend remains consistent for precision as well. Moreover, the precision and recall for poor-responders are notably better compared to the human experts, indicating that the predicted poor-responding subpopulation is labeled correctly with high probability. On the other hand, identifying the second rare subpopulation, non-responders, proves to be challenging for all competitors, whether human or machine algorithms. The corresponding precision in Figures 5.2a and 5.5a is low for non-responders, implying that predicted non-responders are unlikely to be correct. However, the recall of the human experts for the non-responding subpopulation is higher compared to the machine counterparts, suggesting that the trained ophthalmologists can, to some extent, identify non-responders at an early stage.

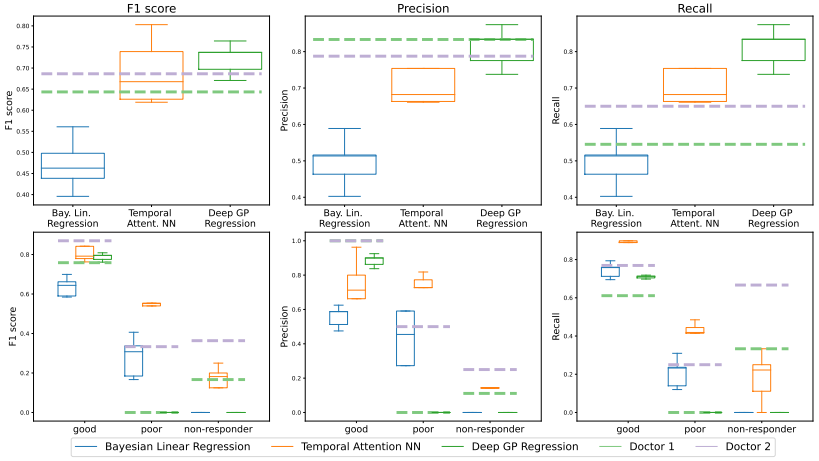
When comparing Figure 5.2 and Figure 5.5, we can observe that an increased number of longitudinal data points is advantageous for the temporal attention neural network. The correlation between predictions and ground truth values is notably improved, and the RMSE decreases by more than 10%. These findings strongly suggest that the temporal attention network is highly effective in this context. The classification performance also experiences a slight improvement with the increased number of longitudinal data points. F1 score, precision, and recall all show enhancements for good and poor responders, while the performance on the non-responding subpopulation remains relatively unchanged when utilizing more longitudinal visits. One possible reason for this might be an insufficient definition of the thresholding function $\tau(V(t))$, which could favor the over-expressing subpopulation of good responders. Since class computation is performed in a relative space, i.e. classes are computed relative to the fluid volume at t=0, small errors in the regression analysis do not transfer directly to small errors in the relative space underlying the classification performance. In extreme cases, this might yield a false class prediction despite a highly accurate regression prediction.

Figures 5.3 and 5.6 showcase the outcomes based on the LMM preprocessed dataset for $n_{visits} = 2$ and $n_{visits} = 3$, respectively. In this context, both Bayesian linear regression and Deep GP regression prove to be ineffective, displaying the same issues as observed in the initial dataset. Interestingly, the performance of the temporal attention neural network using the LMM preprocessed dataset is weaker compared to the initial dataset. The RMSE is 2-3 times higher when using the LMM preprocessed data. Regarding the classification task, it becomes apparent that the F1 score,

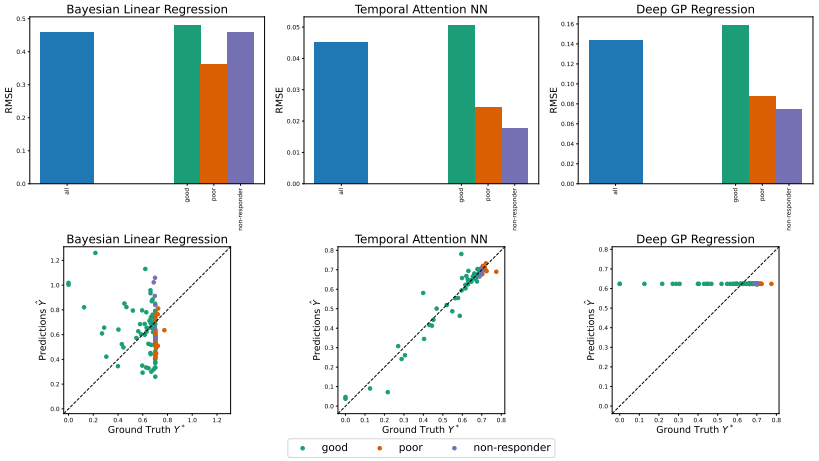
precision, and recall remain high for the good responding subpopulation. However, for the two underrepresented subpopulations of poor and non-responders, none of the algorithms are effective, as they rarely predict these classes. One possible explanation for this observation is that the LMM filters out subtle non-linear features within the patient trajectories, leading to erroneous regression predictions that lie within the support of the distribution of the good responders. Consequently, the class transformations overpredict the good responding subpopulation, thereby hindering accurate predictions for poor and non-responders.

Figures 5.4 and 5.7 depict the performance metrics for the dataset using a class-wise GP preprocessing. Upon examining the temporal attention network, we notice a slight increase in RMSE with this dataset; however, the correlation between ground truth and predictions remains high, indicating the trained model's effectiveness in the regression task. In terms of classification performance, the results for good and poor responders are notably improved, with higher F1 scores, precision, and recall values compared to those achieved on the initial dataset. In fact, the temporal attention network's prediction performance matches or even surpasses both ophthalmologists, particularly for the poor responding subpopulation, where the best models reach classification metrics close to 1. However, the GP trained on the non-responding subpopulation appears to have no beneficial effect on the downstream classification task. This may be attributed to the fact that the trained preprocessing GP for non-responders exhibits high variation at all times, indicating a lack of common trends within the non-responding subpopulation (see Fig. 5.1b).

The unaveraged results for the overall population, obtained through predictions on different validation runs within the 5-fold cross-validation procedure, are provided in Section D of the Appendix. These results validate the previously mentioned findings.

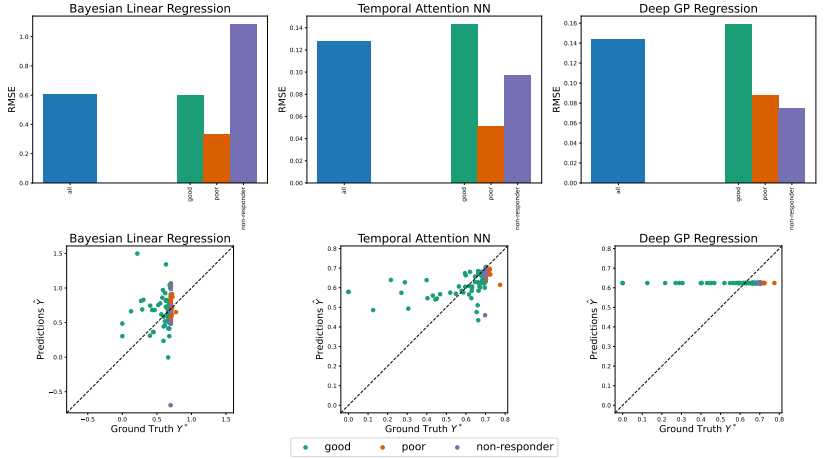


(a) Classification Performance: Initial data: $n_{visits} = 2$

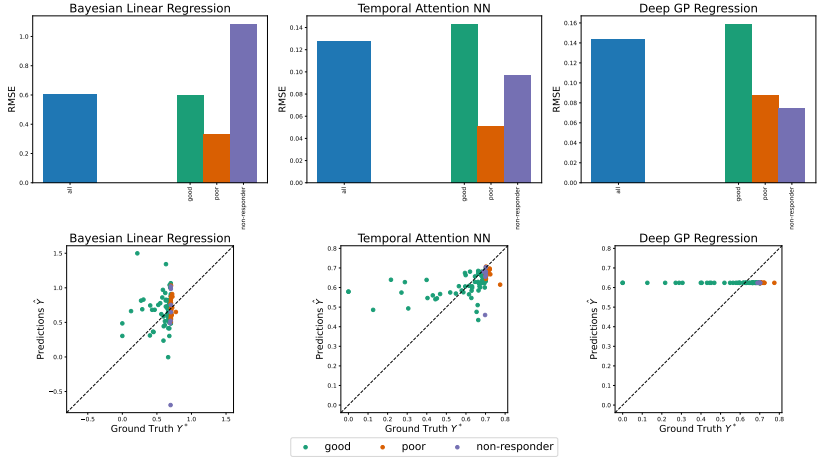


(b) Regression Performance: Initial data: $n_{visits} = 2$

Figure 5.2: Results of the initial dataset with $n_{visits} = 2$: (a) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (b) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.

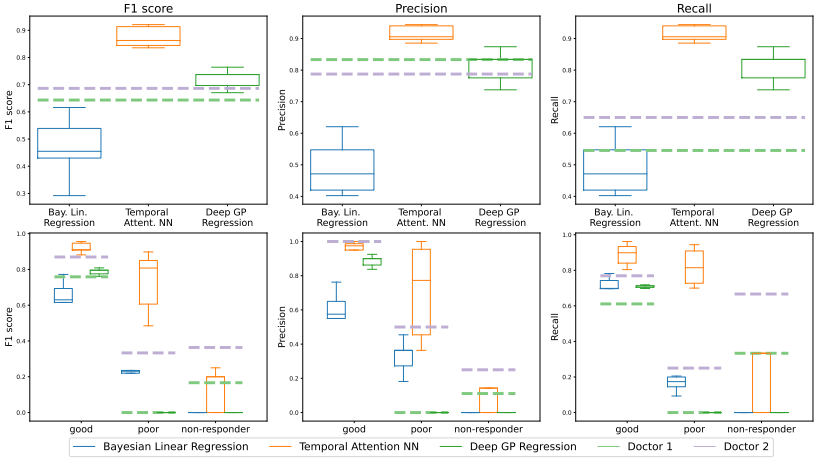


(a) Classification Performance: LMM preprocessed data: $n_{visits} = 2$

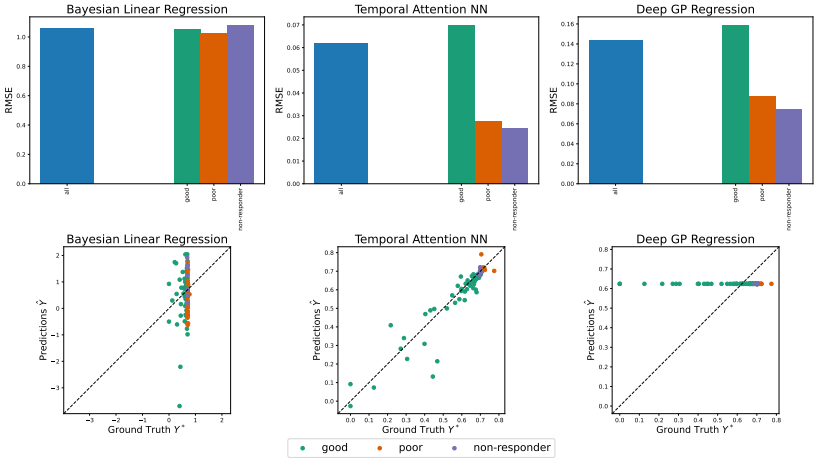


(b) Regression Performance: LMM preprocessed data: $n_{visits} = 2$

Figure 5.3: Results of the LMM preprocessed dataset with $n_{visits} = 2$: (a) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (b) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.

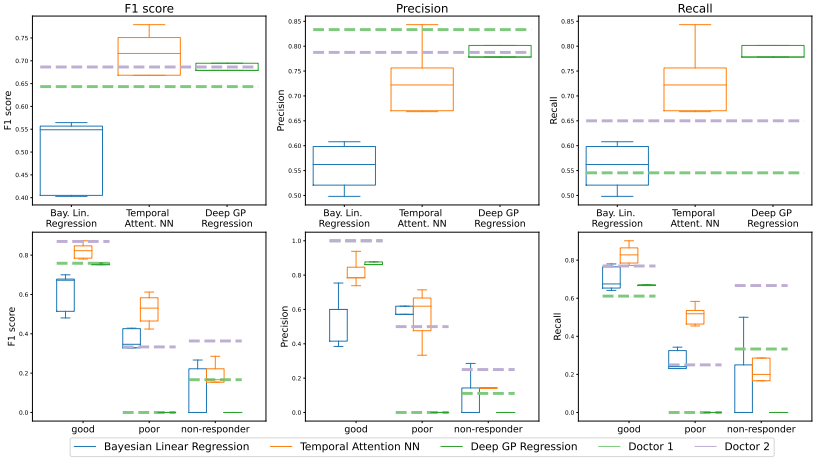


(a) Classification Performance: GP preprocessed data: $n_{visits} = 2$

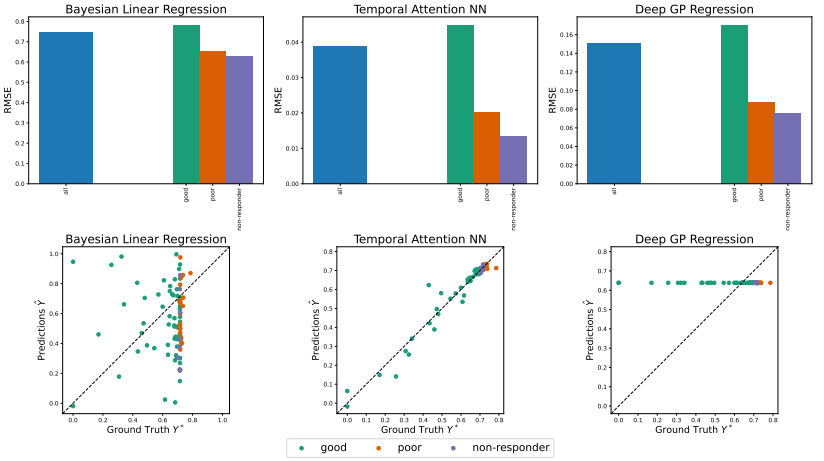


(b) Regression Performance: GP preprocessed data: $n_{visits} = 2$

Figure 5.4: Results of the GP preprocessed dataset with $n_{visits} = 2$: (a) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (b) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.

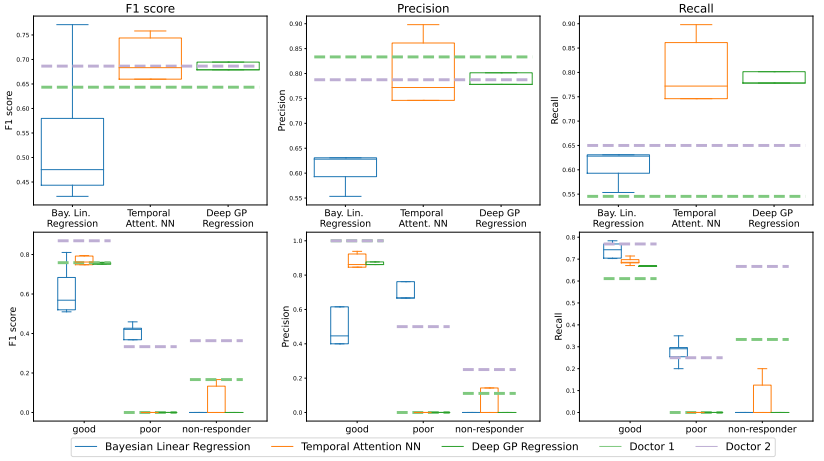


(a) Classification Performance: Initial data: $n_{visits} = 3$

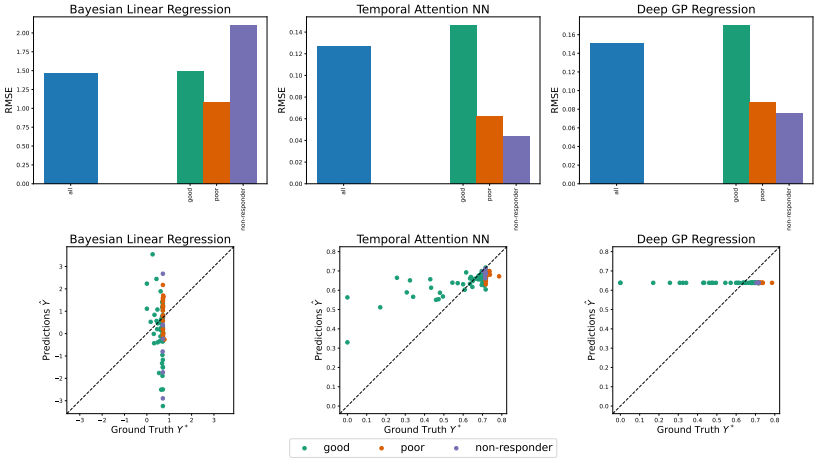


(b) Regression Performance: Initial data: $n_{visits} = 3$

Figure 5.5: Results of the initial dataset with $n_{visits} = 3$: (a) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (b) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.

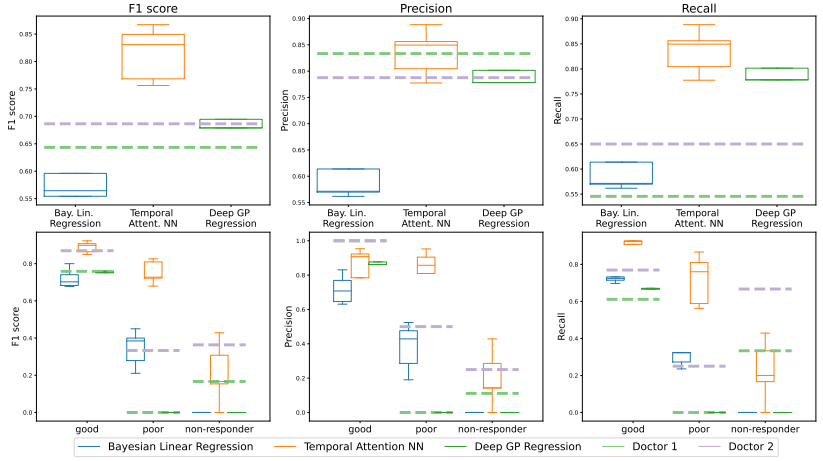


(a) Classification Performance: LMM preprocessed data: $n_{visits} = 3$

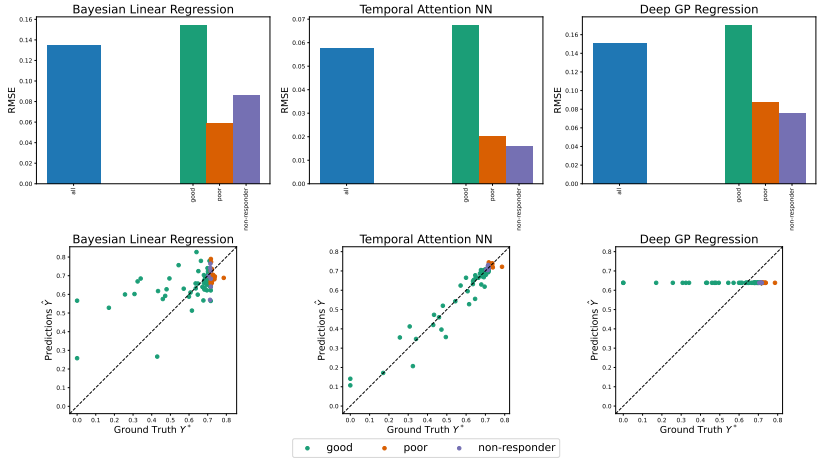


(b) Regression Performance: LMM preprocessed data: $n_{visits} = 3$

Figure 5.6: Results of the LMM preprocessed dataset with $n_{visits} = 3$: (a) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (b) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.



(a) Classification Performance: GP preprocessed data: $n_{visits} = 2$



(b) Regression Performance: GP preprocessed data: $n_{visits} = 2$

Figure 5.7: Results of the GP preprocessed dataset with $n_{visits} = 3$: (a) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (b) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.

5.7 Discussion and Conclusion

In this study, we evaluated the performance of different machine learning models on clinical records from a large cohort of patients diagnosed with wet AMD, specifically focusing on patients' responses to a treatment. We analyzed the results based on two preprocessing methods and the number of visits for each patient. For the initial dataset, the temporal attention neural network emerged as the most effective model. It demonstrated a strong correlation between predicted responses and ground truth values, with significantly lower RMSE values compared to the Bayesian linear regression and the Deep GP. Moreover, its classification performance, especially for good and poor responders, rivaled those of human experts, showcasing its potential in clinical applications.

When using the LMM preprocessed dataset, the effectiveness of the models decreased. Both, Bayesian linear regression and Deep GP, remained ineffective, while the temporal attention neural network's performance was reduced, exhibiting higher RMSE values. Additionally, the classification task for poor and non-responders proved difficult for all algorithms, likely due to the filtering of subtle non-linear features by the linear mixed model.

In contrast, employing class-wise GP preprocessing improved the classification performance for good and poor responders. The temporal attention neural network continued to display impressive results, outperforming human experts in several cases. However, the GP preprocessing for non-responders did not contribute positively to the classification task, most likely due to high variations within the non-responding subpopulation.

In conclusion, the temporal attention neural network showcased remarkable capabilities, particularly when class-wise GP preprocessing was applied. Its strong performance in both regression and classification tasks highlights its potential as a valuable tool in medical settings. However, the choice of preprocessing method and dataset characteristics can significantly impact model effectiveness. This study underscores the importance of selecting appropriate models and preprocessing techniques to achieve accurate and reliable predictions in medical applications, especially in the presence of under-represented subpopulations. One of the most prominent challenges faced in this study was the class imbalance within the dataset. This imbalance, specifically the poor and non-responding subpopulations, posed significant difficulties for all models. As a result, accurate predictions for these under-represented classes were elusive, and the models tended to favor the dominant class, leading to imbalanced classification results. Addressing class imbalance through techniques like resampling or using appropriate evaluation metrics is crucial for further improvement. Future research addressing critical points like class imbalance, appropriate preprocessing, model interpretability, and clinical relevance are essential to harnessing the full potential of these models and ensure their responsible and effective integration into healthcare practices.

Bibliography

[1] K. Lagemann, C. Lagemann, B. Taschler & S. Mukherjee. "Deep learning of causal structures in high dimensions under data limitations". In: *Nature Machine Intelligence* 5.11 (2023), pp. 1306–1316.

[2] K. Lagemann, C. Lagemann & S. Mukherjee. "Invariance-based Learning of Latent Dynamics". In: *The Twelfth International Conference on Learning Representations*. 2024.

[3] J. Peters, D. Janzing & B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Cambridge, MA, USA: MIT Press, 2017.

[4] M. Arjovsky, L. Bottou, I. Gulrajani & D. Lopez-Paz. "Invariant Risk Minimization". In: *arXiv preprint arXiv:1907.02893* (2019).

[5] C. Heinze-Deml, M. H. Maathuis & N. Meinshausen. "Causal Structure Learning". In: *Annual Review of Statistics and Its Application* 5 (2018), pp. 371–391.

[6] P. Spirtes, C. Glymour & R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, 2000.

[7] S. Shimizu, P. O. Hoyer, A. Hyvärinen & A. Kerminen. "A linear non-Gaussian acyclic model for causal discovery". In: *The Journal of Machine Learning Research* 7 (2006), pp. 2003–2030.

[8] M. H. Maathuis, M. Kalisch & P. Bühlmann. "Estimating high-dimensional intervention effects from observational data". In: *The Annals of Statistics* 37 (2009), pp. 3133–3164.

[9] A. Hauser & P. Bühlmann. "Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs". In: *The Journal of Machine Learning Research* 13 (2012), pp. 2409–2464.

[10] D. Colombo, M. H. Maathuis, M. Kalisch & T. S. Richardson. "Learning high-dimensional directed acyclic graphs with latent and selection variables". In: *The Annals of Statistics* 40 (2012), pp. 294–321.

[11] J. Peters, P. Bühlmann & N. Meinshausen. "Causal inference using invariant prediction: identification and confidence intervals". In: *Journal of the Royal Statistical Society* 78 (2016), pp. 947–1012.

[12] S. M. Hill, C. J. Oates, D. A. Blythe & S. Mukherjee. "Causal Learning via Manifold Regularization". In: *The Journal of Machine Learning Research* 20 (2019), pp. 1–32.

[13] X. Zheng, B. Aragam, P. K. Ravikumar & E. P. Xing. "Dags with no tears: Continuous optimization for structure learning". In: *Advances in neural information processing systems* 31 (2018).

[14] N. R. Ke, O. Bilaniuk, A. Goyal, S. Bauer, H. Larochelle, B. Schölkopf, M. C. Mozer, C. Pal & Y. Bengio. "Learning neural causal models from unknown interventions". In: *arXiv preprint arXiv:1910.01075* (2019).

[15] P. Brouillard, S. Lachapelle, A. Lacoste, S. Lacoste-Julien & A. Drouin. "Differentiable causal discovery from interventional data". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21865–21877.

[16] R. Lopez, J.-C. Hütter, J. Pritchard & A. Regev. "Large-scale differentiable causal discovery of factor graphs". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 19290–19303.

[17] P. Lippe, T. Cohen & E. Gavves. "Efficient Neural Causal Discovery without Acyclicity Constraints". In: *International Conference on Learning Representations*. 2022.

[18] T. Ideker & N. J. Krogan. "Differential network biology". In: *Molecular systems biology* 8.1 (2012), p. 565.

[19] S. M. Hill, L. Heiser, T. Cokelaer & et al. "Inferring causal molecular networks: Empirical assessment through a community-based effort". In: *Nature Methods* 13 (2016), pp. 310–318.

[20] S. M. Hill, N. K. Nesser, K. Johnson-Camacho, M. Jeffress, A. Johnson, C. Boniface, S. E. Spencer, Y. Lu, L. M. Heiser, Y. Lawrence, et al. "Context specificity in causal signaling networks revealed by phosphoprotein profiling". In: *Cell Systems* 4.1 (2017), pp. 73–83.

[21] B. M. Kuenzi & T. Ideker. "A census of pathway maps in cancer systems biology". In: *Nature Reviews Cancer* 20.4 (2020), pp. 233–246.

[22] D. Lopez-Paz, K. Muandet, B. Schölkopf & I. Tolstikhin. "Towards a learning theory of cause-effect inference". In: *International Conference on Machine Learning*. 2015.

[23] J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler & B. Schölkopf. "Distinguishing cause from effect using observational data: Methods and benchmarks". In: *The Journal of Machine Learning Research* 17 (2016), pp. 1–102.

[24] U. Noè, B. Taschler, J. Täger, P. Heutink & S. Mukherjee. "Ancestral causal learning in high dimensions with a human genome-wide application". In: *arXiv preprint arXiv:1905.11506* (2019).

[25] M. Eigenmann, S. Mukherjee & M. Maathuis. "Evaluation of Causal Structure Learning Algorithms via Risk Estimation". In: *Proceedings of Uncertainty in Artificial Intelligence 2020*. 2020.

[26] V. Didelez. "Causal Concepts and Graphical Models". In: *Handbook of Graphical Models*. CRC Press, Inc., 2018.

[27] F. Eberhardt, C. Glymour & R. Scheines. "N-1 experiments suffice to determine the causal relations among n variables". In: *Innovations in machine learning* 194 (2006), pp. 97–112.

[28] F. Eberhardt, C. Glymour & R. Scheines. "On the Number of Experiments Sufficient and in the Worst Case Necessary to Identify All Causal Relations Among N Variables". In: *Conference on Uncertainty in Artificial Intelligence*. 2005.

[29] D. Colombo & M. H. Maathuis. "Order-Independent Constraint-Based Causal Structure Learning". In: *Journal of Machine Learning Research* 15.116 (2014), pp. 3921–3962.

[30] P. Spirtes, C. Glymour & R. Scheines. *Causation, Prediction, and Search*. The MIT Press, 2001.

[31] D. Colombo, M. H. Maathuis, M. Kalisch & T. S. Richardson. "Learning high-dimensional directed acyclic graphs with latent and selection variables". In: *The Annals of Statistics* (2012).

[32] D. M. Chickering. "Optimal Structure Identification with Greedy Search". In: *The Journal of Machine Learning Research* (2003).

[33] N. R. Ke, S. Chiappa, J. Wang, J. Bornschein, T. Weber, A. Goyal, M. Botvinic, M. Mozer & D. J. Rezende. "Learning to induce causal structure". In: *arXiv preprint arXiv:2204.04875* (2022).

[34] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang & J. Mooij. "On Causal and Anticausal Learning". In: *Proceedings of the 29th International Conference on Machine Learning, ICML 2012* 2 (2012).

[35] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[36] B. Turlach. "Bandwidth Selection in Kernel Density Estimation: A Review". In: *Technical Report* (1999).

[37] K. He, X. Zhang, S. Ren & J. Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke & A. Rabinovich. "Going deeper with convolutions". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[39] K. He, X. Zhang, S. Ren & J. Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1026–1034.

[40] S. Xie, R. Girshick, P. Dollár, Z. Tu & K. He. "Aggregated Residual Transformations for Deep Neural Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[41] M. Zhang & Y. Chen. "Link prediction based on graph neural networks". In: *Advances in Neural Information Processing Systems 2018, NeurIPS 2018*. 2018.

[42] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou & X. Sun. "Measuring and Relieving the Over-smoothing Problem for Graph Neural Networks from the Topological View". In: *Computing Research Repository (CoRR)* (2019).

[43] Q. Li, Z. Han & X.-M. Wu. "Deeper insights into graph convolutional networks for semi-supervised learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 2018.

[44] M. Zhang, Z. Cui, M. Neumann & Y. Chen. "An end-to-end deep learning architecture for graph classification". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 2018.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai & S. Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[46] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li & Z. Zhang. "Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks". In: *arXiv preprint arXiv:1909.01315* (2019).

[47] D. P. Kingma & J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio & Y. LeCun. 2015.

[48] P. Kemmeren, K. Sameith, L. A. van de Pasch, J. J. Benschop, T. L. Lenstra, T. Margaritis, E. O Duibhir, E. Apweiler, S. van Wageningen, C. W. Ko, et al. "Large-scale genetic perturbations reveal regulatory networks and an abundance of gene-specific repressors". In: *Cell* 157.3 (2014), pp. 740–752.

[49] N. Meinshausen, A. Hauser, J. M. Mooij, J. Peters, P. Versteeg & P. Bühlmann. "Methods for causal inference from gene perturbation experiments and validation". In: *Proceedings of the National Academy of Sciences of the United States of America* 113.27 (2016), pp. 7361–7368.

[50] J. Zhang. "Causal reasoning with ancestral graphs". In: *The Journal of Machine Learning Research* 9 (2008), pp. 1437–1474.

[51] U. Alon. *An introduction to systems biology: design principles of biological circuits*. CRC press, 2019.

[52] A. Hyttinen, F. Eberhardt & P. O. Hoyer. "Learning linear cyclic causal models with latent variables". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 3387–3439.

[53] F. Eberhardt & R. Scheines. “Interventions and causal inference”. In: *Philosophy of Science* 74.5 (2007), pp. 981–995.

[54] M. Kocaoglu, K. Shanmugam & E. Bareinboim. “Experimental design for learning causal graphs with latent variables”. In: *Advances in Neural Information Processing Systems 30, NIPS*. 2017.

[55] J. M. Replogle, R. A. Saunders, A. N. Pogson, J. A. Hussmann, A. Lenail, A. Guna, L. Mascibroda, E. J. Wagner, K. Adelman, G. Lithwick-Yanai, N. Iremadze, F. Oberstrass, D. Lipson, J. L. Bonnar, M. Jost, T. M. Norman & J. S. Weissman. “Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq”. In: *Cell* (2022).

[56] D. Malinsky & P. Spirtes. “Estimating bounds on causal effects in high-dimensional and possibly confounded systems”. In: *International Journal of Approximate Reasoning* 88 (2017), pp. 371–384.

[57] D. M. Chickering. “Optimal Structure Identification With Greedy Search”. In: *The Journal of Machine Learning Research* 3 (2002), pp. 507–554.

[58] S. Shimizu, P. O. Hoyer, A. Hyvärinen & A. Kerminen. “A Linear Non-Gaussian Acyclic Model for Causal Discovery”. In: *Journal of Machine Learning Research* (2006).

[59] P. Bühlmann, J. Peters & J. Ernest. “CAM: Causal Additive Models, high-dimensional order search and penalized regression”. In: *The Annals of Statistics* (2014).

[60] C. Fefferman, S. Mitter & H. Narayanan. “Testing the manifold hypothesis”. In: *Journal of the American Mathematical Society* 29.4 (2016), pp. 983–1049.

[61] W. Zhi, T. Lai, L. Ott, E. V. Bonilla & F. Ramos. “Learning Efficient and Robust Ordinary Differential Equations via Invertible Neural Networks”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 27060–27074.

[62] C. Finlay, J.-H. Jacobsen, L. Nurbekyan & A. Oberman. “How to train your neural ODE: the world of Jacobian and kinetic regularization”. In: *International conference on machine learning*. PMLR. 2020, pp. 3154–3164.

[63] T. Duong & N. Atanasov. “Hamiltonian-based Neural ODE Networks on the SE(3) Manifold For Dynamics Learning and Control”. In: *Proceedings of Robotics: Science and Systems*. 2021.

[64] M. Choi, D. Flam-Shepherd, T. H. Kyaw & A. Aspuru-Guzik. “Learning quantum dynamics with latent neural ordinary differential equations”. In: *Physical Review A* 105.4 (2022), p. 042403.

[65] R. T. Q. Chen, B. Amos & M. Nickel. “Learning Neural Event Functions for Ordinary Differential Equations”. In: *International Conference on Learning Representations* (2021).

[66] T. D. Kim, T. Z. Luo, J. W. Pillow & C. D. Brody. "Inferring latent dynamics underlying neural population activity via neural differential equations". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5551–5561.

[67] S. L. Brunton, M. Budišić, E. Kaiser & J. N. Kutz. "Modern Koopman Theory for Dynamical Systems". In: *SIAM Review* 64.2 (2022), pp. 229–340.

[68] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser & J. N. Kutz. "Chaos as an intermittently forced linear system". In: *Nature Communications* 8.1 (2017), p. 19.

[69] S. L. Brunton, J. L. Proctor & J. N. Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937.

[70] H. Schaeffer. "Learning partial differential equations via data discovery and sparse optimization". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2197 (2017), p. 20160446.

[71] M. Hoffmann, C. Fröhner & F. Noé. "Reactive SINDy: Discovering governing reactions from concentration data". In: *The Journal of Chemical Physics* 150.2 (2019).

[72] M. Sorokina, S. Sygletos & S. Turitsyn. "Sparse identification for nonlinear optical communication systems: SINO method". In: *Opt. Express* 24.26 (2016), pp. 30433–30443.

[73] A. Narasingam & J. Sang-Il Kwon. "Data-driven identification of interpretable reduced-order models using sparse regression". In: *Computers and Chemical Engineering* 119 (2018), pp. 101–111.

[74] N. M. Mangan, S. L. Brunton, J. L. Proctor & J. N. Kutz. "Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics". In: *IEEE Transactions on Molecular, Biological and Multi-Scale Communications* 2.1 (2016), pp. 52–63.

[75] K. Kaheman, J. N. Kutz & S. L. Brunton. "SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 476.2242 (2020).

[76] L. Lu, P. Jin, G. Pang, Z. Zhang & G. E. Karniadakis. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators". In: *Nature Machine Intelligence* 3.3 (2021), pp. 218–229.

[77] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart & A. Anandkumar. "Multipole Graph Neural Operator for Parametric Partial Differential Equations". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Curran Associates Inc., 2020.

[78] R. G. Patel, N. A. Trask, M. A. Wood & E. C. Cyr. "A physics-informed operator regression framework for extracting data-driven continuum models". In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113500.

[79] N. H. Nelsen & A. M. Stuart. "The Random Feature Model for Input-Output Maps between Banach Spaces". In: *SIAM Journal on Scientific Computing* 43.5 (2021), A3212–A3243.

[80] K. Bhattacharya, B. Hosseini, N. B. Kovachki & A. M. Stuart. "Model Reduction And Neural Networks For Parametric PDEs". In: *The SMAI Journal of computational mathematics* 7 (2021), pp. 121–157.

[81] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart & A. Anandkumar. "Fourier Neural Operator for Parametric Partial Differential Equations". In: *International Conference on Learning Representations*. 2021.

[82] J. Guibas, M. Mardani, Z. Li, A. Tao, A. Anandkumar & B. Catanzaro. "Efficient Token Mixing for Transformers via Adaptive Fourier Neural Operators". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. 2022.

[83] T. Kurth, S. Subramanian, P. Harrington, J. Pathak, M. Mardani, D. Hall, A. Miele, K. Kashinath & A. Anandkumar. "FourCastNet: Accelerating Global High-Resolution Weather Forecasting Using Adaptive Fourier Neural Operators". In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '23. Association for Computing Machinery, 2023.

[84] J. M. Varah. "A Spline Least Squares Method for Numerical Parameter Estimation in Differential Equations". In: *SIAM Journal on Scientific and Statistical Computing* 3.1 (1982), pp. 28–46.

[85] S. P. Ellner, Y. Seifu & R. H. Smith. "Fitting Population Dynamic Models to Time-Series Data by Gradient Matching". In: *Ecology* 83.8 (2002), pp. 2256–2270.

[86] J. O. Ramsay, G. Hooker, D. Campbell & J. Cao. "Parameter estimation for differential equations: a generalized smoothing approach". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.5 (2007), pp. 741–796.

[87] T. Äijö & H. Lähdesmäki. "Learning gene regulatory networks from gene expression measurements using non-parametric molecular kinetics". In: *Bioinformatics* 25.22 (2009), pp. 2937–2944.

[88] T. Äijö, K. Granberg & H. Lähdesmäki. "Sorad: a systems biology approach to predict and modulate dynamic signaling pathway response from phosphoproteome time-course measurements". In: *Bioinformatics* 29.10 (2013), pp. 1283–1291.

[89] M. Heinonen, Ç. Yildiz, H. Mannerström, J. Intosalmi & H. Lähdesmäki. "Learning unknown ODE models with Gaussian processes". In: *Proceedings of the 35th International Conference on Machine Learning, ICML*. PMLR, 2018, pp. 1964–1973.

[90] Ç. Yıldız, M. Kandemir & B. Rakitsch. "Learning interacting dynamical systems with latent Gaussian process ODEs". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 9188–9200.

[91] P. Hegde, Ç. Yıldız, H. Lähdesmäki, S. Kaski & M. Heinonen. "Variational multiple shooting for Bayesian ODEs with Gaussian processes". In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. PMLR, 2022, pp. 790–799.

[92] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel & S. Ho. "Lagrangian Neural Networks". In: *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*. 2019.

[93] M. Lutter, C. Ritter & J. Peters. "Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning". In: *International Conference on Learning Representations*. 2019.

[94] S. Greydanus, M. Dzamba & J. Yosinski. "Hamiltonian Neural Networks". In: *Advances in Neural Information Processing Systems*. 2019.

[95] M. Finzi, K. A. Wang & A. G. Wilson. "Simplifying Hamiltonian and Lagrangian Neural Networks via Explicit Constraints". In: *Advances in Neural Information Processing Systems*. 2020, pp. 13880–13889.

[96] Y. D. Zhong, B. Dey & A. Chakraborty. "Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control". In: *International Conference on Learning Representations*. 2020.

[97] S. Bai, J. Z. Kolter & V. Koltun. "Deep Equilibrium Models". In: *Advances in Neural Information Processing Systems*. 2019.

[98] W. E. "A Proposal on Machine Learning via Dynamical Systems". In: *Communications in Mathematics and Statistics* 5.1 (2017), pp. 1–11.

[99] Y. Lu, A. Zhong, Q. Li & B. Dong. "Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy & A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 3276–3285.

[100] L. Ruthotto & E. Haber. "Deep Neural Networks Motivated by Partial Differential Equations". In: *Journal of Mathematical Imaging and Vision* 62.3 (2020), pp. 352–364.

[101] J. Richter-Powell, Y. Lipman & R. T. Q. Chen. "Neural Conservation Laws: A Divergence-Free Perspective". In: *Advances in Neural Information Processing Systems*. 2022, pp. 38075–38088.

[102] B. K. Horn & B. G. Schunck. "Determining optical flow". In: *Artificial Intelligence* 17.1 (1981), pp. 185–203.

[103] P. Perona & J. Malik. "Scale-space and edge detection using anisotropic diffusion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7 (1990), pp. 629–639.

[104] R. T. Q. Chen, Y. Rubanova, J. Bettencourt & D. K. Duvenaud. "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi & R. Garnett. Vol. 31. Curran Associates, Inc., 2018.

[105] C. Runge. *Über die numerische Auflösung von Differentialgleichungen*. 1895.

[106] W. Kutta. "Beitrag zur näherungsweisen Integration totaler Differentialgleichungen". In: *Zeit. Math. Phys.* 46 (1901), pp. 435–53.

[107] J. Dormand & P. Prince. "A family of embedded Runge-Kutta formulae". In: *Journal of Computational and Applied Mathematics* 6.1 (1980), pp. 19–26.

[108] L. S. Pontryagin. *Mathematical Theory of Optimal Processes*. 1962.

[109] W. Grathwohl, R. T. Q. Chen, J. Bettencourt & D. Duvenaud. "Scalable Reversible Generative Models with Free-form Continuous Dynamics". In: *International Conference on Learning Representations*. 2019.

[110] E. Dupont, A. Doucet & Y. W. Teh. "Augmented Neural ODEs". In: *Advances in Neural Information Processing Systems*. 2019.

[111] J. Jia & A. R. Benson. "Neural Jump Stochastic Differential Equations". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019.

[112] P. Kidger, J. Morrill, J. Foster & T. Lyons. "Neural Controlled Differential Equations for Irregular Time Series". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020, pp. 6696–6707.

[113] J. Morrill, C. Salvi, P. Kidger & J. Foster. "Neural Rough Differential Equations for Long Time Series". In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021, pp. 7829–7838.

[114] S. Massaroli, M. Poli, S. Sonoda, T. Suzuki, J. Park, A. Yamashita & H. Asama. "Differentiable Multiple Shooting Layers". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021, pp. 16532–16544.

[115] H. Bock & K. Plitt. "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems*". In: *IFAC Proceedings Volumes* 17.2 (1984), pp. 1603–1608.

[116] M. Diehl, H. Bock, H. Diedam & P.-B. Wieber. "Fast Direct Multiple Shooting Algorithms for Optimal Robot Control". In: *Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control*. Ed. by M. Diehl & K. Mombaur. Springer Berlin Heidelberg, 2006, pp. 65–93.

[117] V. Iakovlev, C. Yildiz, M. Heinonen & H. Lähdesmäki. "Latent Neural ODEs with Sparse Bayesian Multiple Shooting". In: *The Eleventh International Conference on Learning Representations*. 2023.

[118] V. Arnold & R. Silverman. *Ordinary Differential Equations*. Mit Press, 1978.

[119] B. Tzen & M. Raginsky. "Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit". In: *arXiv preprint* (2019).

[120] X. Liu, T. Xiao, S. Si, Q. Cao, S. Kumar & C. Hsieh. "Neural SDE: Stabilizing Neural ODE Networks with Stochastic Noise". In: *CoRR* (2019).

[121] E. Gobet & R. Munos. "Sensitivity Analysis Using Itô–Malliavin Calculus and Martingales, and Application to Stochastic Optimal Control". In: *SIAM Journal on Control and Optimization* 43.5 (2005), pp. 1676–1713.

[122] X. Li, T.-K. L. Wong, R. T. Q. Chen & D. K. Duvenaud. "Scalable Gradients and Variational Inference for Stochastic Differential Equations". In: *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*. PMLR, 2020, pp. 1–28.

[123] F. Black & M. Scholes. "The Pricing of Options and Corporate Liabilities". In: *Journal of Political Economy* 81.3 (1973), pp. 637–654.

[124] J. C. Cox, J. E. Ingersoll & S. A. Ross. "A Theory of the Term Structure of Interest Rates". In: *Econometrica* 53.2 (1985), pp. 385–407.

[125] D. Brigo & F. Mercurio. *Interest rate models : theory and practice*. Springer finance. Springer, 2001.

[126] T. Huillet. "On Wright–Fisher diffusion and its relatives". In: *Journal of Statistical Mechanics: Theory and Experiment* 2007.11 (2007), P11006.

[127] W. Coffey, Y. Kalmykov & J. Waldron. *The Langevin Equation: With Applications to Stochastic Problems in Physics, Chemistry and Electrical Engineering*. 2004.

[128] G. A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*. 2014.

[129] A. Bulat, J. M. Perez Rua, S. Sudhakaran, B. Martinez & G. Tzimiropoulos. "Space-time Mixing Attention for Video Transformer". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021, pp. 19594–19607.

[130] K. Champion, B. Lusch, J. N. Kutz & S. L. Brunton. "Data-driven discovery of coordinates and governing equations". In: *Proceedings of the National Academy of Sciences* 116.45 (2019), pp. 22445–22451.

[131] Y. Yin, M. Kirchmeyer, J.-Y. Franceschi, A. Rakotomamonjy & P. Gallinari. "Continuous PDE Dynamics Forecasting with Implicit Neural Representations". In: *International Conference on Learning Representations*. 2023.

[132] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart & A. Anandkumar. "Fourier Neural Operator for Parametric Partial Differential Equations". In: *International Conference on Learning Representations*. 2021.

[133] P. L. Bhatnagar, E. P. Gross & M. Krook. "A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems". In: *Physical review* 94.3 (1954), p. 511.

[134] R. Benzi, S. Succi & M. Vergassola. "The lattice Boltzmann equation: theory and applications". In: *Physics Reports* 222.3 (1992), pp. 145–197.

[135] Y.-H. Qian, D. d'Humières & P. Lallemand. "Lattice BGK models for Navier-Stokes equation". In: *Europhysics letters* 17.6 (1992), p. 479.

[136] D. Hänel. *Molekulare Gasdynamik: Einführung in die kinetische Theorie der Gase und Lattice-Boltzmann-Methoden*. Springer-Verlag, 2006.

[137] Y. D. Zhong, B. Dey & A. Chakraborty. "Benchmarking Energy-Conserving Neural Networks for Learning Dynamics from Data". In: *Proceedings of the 3rd Conference on Learning for Dynamics and Control*. PMLR, 2021, pp. 1218–1229.

[138] A. Botev, A. Jaegle, P. Wirnsberger, D. Hennes & I. Higgins. "Which priors matter? Benchmarking models for learning latent dynamics". In: *Advances in Neural Information Processing Systems* 34 (2021).

[139] Y. Rubanova, R. T. Q. Chen & D. K. Duvenaud. "Latent Ordinary Differential Equations for Irregularly-Sampled Time Series". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[140] A. Norcliffe, C. Bodnar, B. Day, J. Moss & P. Liò. "Neural ODE Processes". In: *International Conference on Learning Representations*. 2021.

[141] C. Yildiz, M. Heinonen & H. Lahdesmaki. "ODE^{2}VAE: Deep generative second order ODEs with Bayesian neural networks". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[142] M. Braun, W. Schröder & M. Klaas. "High-speed tomographic PIV measurements in a DISI engine". In: *Experiments in Fluids* 60.9 (2019), p. 146.

[143] C. Lagemann, K. Lagemann, S. Mukherjee & W. Schroeder. "Generalization of deep recurrent optical flow estimation for particle-image velocimetry data". In: *Measurement Science and Technology* (2022).

[144] E. R. Gowree, C. Jagadeesh, E. Talboys, C. Lagemann & C. Brücker. "Vortices enable the complex aerobatics of peregrine falcons". In: *Communications biology* 1.1 (2018), p. 27.

[145] E. Mäteling & W. Schröder. "Analysis of spatiotemporal inner-outer large-scale interactions in turbulent channel flow by multivariate empirical mode decomposition". In: *Physical Review Fluids* 7.3 (2022), p. 034603.

[146] E. Mäteling, M. Albers & W. Schröder. "How spanwise travelling transversal surface waves change the near-wall flow". In: *Journal of Fluid Mechanics* 957 (2023), A30.

[147] J. Peters, D. Janzing & B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[148] B. Schoelkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang & J. Mooij. "On Causal and Anticausal Learning". In: *International Conference on Machine Learning*. PMLR. 2012.

[149] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal & Y. Bengio. "Toward Causal Representation Learning". In: *Proceedings of the IEEE* 109.5 (2021), pp. 612–634.

[150] T. W. Anderson & H. Rubin. "Estimation of the Parameters of a Single Equation in a Complete System of Stochastic Equations". In: *The Annals of Mathematical Statistics* 20.1 (1949), pp. 46–63.

[151] A. Sinha, H. Namkoong & J. Duchi. "Certifiable Distributional Robustness with Principled Adversarial Training". In: *International Conference on Learning Representations*. 2018.

[152] J. Cao, D. R. O'Day, H. A. Pliner, P. D. Kingsley, M. Deng, R. M. Daza, M. A. Zager, K. A. Aldinger, R. Blecher-Gonen, F. Zhang, et al. "A human cell atlas of fetal gene expression". In: *Science* 370.6518 (2020), eaba7721.

[153] S. Domcke, A. J. Hill, R. M. Daza, J. Cao, D. R. O'Day, H. A. Pliner, K. A. Aldinger, D. Pokholok, F. Zhang, J. H. Milbank, et al. "A human cell atlas of fetal chromatin accessibility". In: *Science* 370.6518 (2020), eaba7612.

[154] K. Zhang, J. D. Hocker, M. Miller, X. Hou, J. Chiou, O. B. Poirion, Y. Qiu, Y. E. Li, K. J. Gaulton, A. Wang, et al. "A single-cell atlas of chromatin accessibility in the human genome". In: *Cell* 184.24 (2021), pp. 5985–6001.

[155] L. Bottou, J. Peters, J. Quiñonero-Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard & E. Snelson. "Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising". In: *Journal of Machine Learning Research* 14.101 (2013), pp. 3207–3260.

[156] P. G. Wright & B. Institution. *The tariff on animal and vegetable oils, by Philip G. Wright*. English. The Macmillan company New York, 1928, xviii, 347 p.

[157] O. Reiersøl. "Identifiability of a Linear Relation between Variables Which Are Subject to Error". In: *Econometrica* 18.4 (1950), pp. 375–389.

[158] J. D. Angrist & G. W. Imbens. "Two-Stage Least Squares Estimation of Average Causal Effects in Models with Variable Treatment Intensity". In: *Journal of the American Statistical Association* 90.430 (1995), pp. 431–442.

[159] J. D. Angrist, G. W. Imbens & D. B. Rubin. "Identification of Causal Effects Using Instrumental Variables". In: *Journal of the American Statistical Association* 91.434 (1996), pp. 444–455.

[160] J. Pearl. *Causality: Models, Reasoning, and Inference*. Second. Cambridge University Press, Cambridge, 2009.

[161] G. W. Imbens & J. D. Angrist. "Identification and Estimation of Local Average Treatment Effects". In: *Econometrica* 62.2 (1994), pp. 467–475.

[162] W. K. Newey & J. L. Powell. "Instrumental Variable Estimation of Nonparametric Models". In: *Econometrica* 71.5 (2003), pp. 1565–1578.

[163] R. Singh, M. Sahani & A. Gretton. "Kernel Instrumental Variable Regression". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[164] K. Muandet, A. Mehrjou, S. K. Lee & A. Raj. "Dual Instrumental Variable Regression". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 2710–2721.

[165] J. Hartford, G. Lewis, K. Leyton-Brown & M. Taddy. "Deep IV: A Flexible Approach for Counterfactual Prediction". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. PMLR, 2017, pp. 1414–1423.

[166] A. Bennett, N. Kallus & T. Schnabel. "Deep Generalized Method of Moments for Instrumental Variable Analysis". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[167] T. Amemiya. "The nonlinear two-stage least-squares estimator". In: *Journal of Econometrics* 2.2 (1974), pp. 105–110.

[168] W. K. Newey. "Efficient Instrumental Variables Estimation of Nonlinear Models". In: *Econometrica* 58.4 (1990), pp. 809–837.

[169] J. Bai & S. Ng. "Forecasting economic time series using targeted predictors". In: *Journal of Econometrics* 146.2 (2008), pp. 304–317.

[170] P. J. Bickel, Y. Ritov & A. B. Tsybakov. "Simultaneous Analysis of Lasso and Dantzig Selector". In: *The Annals of Statistics* 37.4 (2009), pp. 1705–1732.

[171] F. Bunea, A. Tsybakov & M. Wegkamp. "Sparsity Oracle Inequalities for the Lasso". In: *Electronic Journal of Statistics* 1 (2007).

[172] N. Meinshausen & B. Yu. "Lasso-Type Recovery of Sparse Representations for High-Dimensional Data". In: *The Annals of Statistics* 37.1 (2009), pp. 246–270.

[173] A. Belloni & V. Chernozhukov. "Least squares after model selection in high-dimensional sparse models". In: *Bernoulli* 19.2 (2013), pp. 521–547.

[174] G. Chamberlain & G. Imbens. "Random Effects Estimators with Many Instrumental Variables". In: *Econometrica* 72.1 (2004), pp. 295–306.

[175] R. Okui. “Instrumental variable estimation in the presence of many moment conditions”. In: *Journal of Econometrics* 165.1 (2011). Moment Restriction-Based Econometric Methods, pp. 70–86.

[176] N. Pfister & J. Peters. “Identifiability of sparse causal effects using instrumental variables”. In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Vol. 180. PMLR, 2022, pp. 1613–1622.

[177] D. Rothenhäusler, N. Meinshausen, P. Bühlmann & J. Peters. “Anchor Regression: Heterogeneous Data Meet Causality”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 83.2 (2021), pp. 215–246.

[178] Z. Hu & J. L. Hong. “Kullback-Leibler divergence constrained distributionally robust optimization”. In: *Optimization Online* (2012).

[179] H. Lam. “Recovering Best Statistical Guarantees via the Empirical Divergence-Based Distributionally Robust Optimization”. In: *Operations Research* 67.4 (2019), pp. 1090–1105.

[180] P. Mohajerin Esfahani & D. Kuhn. “Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations”. In: *Mathematical Programming* 171.1 (2018), pp. 115–166.

[181] I. J. Goodfellow, J. Shlens & C. Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *3rd International Conference on Learning Representations, ICLR*. 2015.

[182] H. Daumé & D. Marcu. “Domain Adaptation for Statistical Classifiers”. In: *Journal. Artificial Intelligence Research* 26 (2006), pp. 101–126.

[183] S. Bickel, M. Brückner & T. Scheffer. “Discriminative Learning Under Covariate Shift”. In: *Journal of Machine Learning Research* 10.75 (2009), pp. 2137–2155.

[184] K. Muandet, D. Balduzzi & B. Schölkopf. “Domain Generalization via Invariant Feature Representation”. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning*. Vol. 28. JMLR.org, 2013.

[185] H. Shimodaira. “Improving predictive inference under covariate shift by weighting the log-likelihood function”. In: *Journal of Statistical Planning and Inference* 90 (2000), pp. 227–244.

[186] R. Christiansen, N. Pfister, M. Jakobsen, N. Gnecco & J. Peters. “A Causal Framework for Distribution Generalization”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 44.10 (2022), pp. 6614–6630.

[187] J. Pearl. *Causality*. 2nd ed. Cambridge University Press, 2009.

[188] B. Adamson, T. M. Norman, M. Jost, M. Y. Cho, J. K. Nuñez, Y. Chen, J. E. Villalta, L. A. Gilbert, M. A. Horlbeck, M. Y. Hein, R. A. Pak, A. N. Gray, C. A. Gross, A. Dixit, O. Parnas, A. Regev & J. S. Weissman. "A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response". In: *Cell* 167.7 (2016).

[189] G. Davey Smith & S. Ebrahim. "'Mendelian randomization': can genetic epidemiology contribute to understanding environmental determinants of disease?" In: *International Journal of Epidemiology* 32.1 (2003), pp. 1–22.

[190] D. A. Lawlor, R. M. Harbord, J. A. C. Sterne, N. Timpson & G. Davey Smith. "Mendelian randomization: Using genes as instruments for making causal inferences in epidemiology". In: *Statistics in Medicine* 27.8 (2008), pp. 1133–1163.

[191] P. Muir, S. Li, S. Lou, D. Wang, D. J. Spakowicz, L. Salichos, J. Zhang, G. M. Weinstock, F. Isaacs, J. Rozowsky, et al. "The real cost of sequencing: scaling computation to keep pace with data generation". In: *Genome biology* 17.1 (2016), pp. 1–9.

[192] W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes, N. Bindal, D. Beare, J. A. Smith, I. R. Thompson, et al. "Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells". In: *Nucleic acids research* 41.D1 (2012), pp. D955–D961.

[193] J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin, et al. "The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity". In: *Nature* 483.7391 (2012), pp. 603–607.

[194] F. Iorio, T. A. Knijnenburg, D. J. Vis, G. R. Bignell, M. P. Menden, M. Schubert, N. Aben, E. Gonçalves, S. Barthorpe, H. Lightfoot, et al. "A landscape of pharmacogenomic interactions in cancer". In: *Cell* 166.3 (2016), pp. 740–754.

[195] P. M. Haverty, E. Lin, J. Tan, Y. Yu, B. Lam, S. Lianoglou, R. M. Neve, S. Martin, J. Settleman, R. L. Yauch, et al. "Reproducible pharmacogenomic profiling of cancer cell line panels". In: *Nature* 533.7603 (2016), pp. 333–337.

[196] R. W. Platt, E. F. Schisterman & S. R. Cole. "Time-modified confounding". In: *American journal of epidemiology* 170.6 (2009), pp. 687–694.

[197] I. Bica, A. M. Alaa, C. Lambert & M. Van Der Schaar. "From real-world patient data to individualized treatment effects using machine learning: current and future methods to address underlying challenges". In: *Clinical Pharmacology & Therapeutics* 109.1 (2021), pp. 87–100.

[198] P. Schulam & S. Saria. "Reliable decision support using counterfactual models". In: *Advances in neural information processing systems* 30 (2017).

[199] J. M. Robins, M. A. Hernan & B. Brumback. "Marginal structural models and causal inference in epidemiology". In: *Epidemiology* (2000), pp. 550–560.

[200] B. Lim. "Forecasting treatment responses over time using recurrent marginal structural networks". In: *Advances in neural information processing systems* 31 (2018).

[201] I. Bica, A. Alaa & M. Van Der Schaar. "Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 884–895.

[202] A. Basu, N. E. Bodycombe, J. H. Cheah, E. V. Price, K. Liu, G. I. Schaefer, R. Y. Ebright, M. L. Stewart, D. Ito, S. Wang, et al. "An interactive resource to identify cancer genetic and lineage dependencies targeted by small molecules". In: *Cell* 154.5 (2013), pp. 1151–1161.

[203] B. Seashore-Ludlow, M. G. Rees, J. H. Cheah, M. Cokol, E. V. Price, M. E. Coletti, V. Jones, N. E. Bodycombe, C. K. Soule, J. Gould, et al. "Harnessing connectivity in a large-scale small-molecule sensitivity dataset". In: *Cancer discovery* 5.11 (2015), pp. 1210–1223.

[204] R. H. Shoemaker. "The NCI60 human tumour cell line anticancer drug screen". In: *Nature Reviews Cancer* 6.10 (2006), pp. 813–823.

[205] M. Ghandi, F. W. Huang, J. Jané-Valbuena, G. V. Kryukov, C. C. Lo, E. R. McDonald III, J. Barretina, E. T. Gelfand, C. M. Bielski, H. Li, et al. "Next-generation characterization of the cancer cell line encyclopedia". In: *Nature* 569.7757 (2019), pp. 503–508.

[206] A. M. Gholami, H. Hahne, Z. Wu, F. J. Auer, C. Meng, M. Wilhelm & B. Kuster. "Global proteome analysis of the NCI-60 cell line panel". In: *Cell reports* 4.3 (2013), pp. 609–620.

[207] H. Li, S. Ning, M. Ghandi, G. V. Kryukov, S. Gopal, A. Deik, A. Souza, K. Pierce, P. Keskula, D. Hernandez, et al. "The landscape of cancer cell line metabolism". In: *Nature medicine* 25.5 (2019), pp. 850–860.

[208] J. Lamb, E. D. Crawford, D. Peck, J. W. Modell, I. C. Blat, M. J. Wrobel, J. Lerner, J.-P. Brunet, A. Subramanian, K. N. Ross, et al. "The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease". In: *science* 313.5795 (2006), pp. 1929–1935.

[209] A. Subramanian, R. Narayan, S. M. Corsello, D. D. Peck, T. E. Natoli, X. Lu, J. Gould, J. F. Davis, A. A. Tubelli, J. K. Asiedu, et al. "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles". In: *Cell* 171.6 (2017), pp. 1437–1452.

[210] A. B. Keenan, S. L. Jenkins, K. M. Jagodnik, S. Koplev, E. He, D. Torre, Z. Wang, A. B. Dohlman, M. C. Silverstein, A. Lachmann, et al. "The library of integrated network-based cellular signatures NIH program: system-level cataloging of human cells response to perturbations". In: *Cell systems* 6.1 (2018), pp. 13–24.

[211] A. Koleti, R. Terryn, V. Stathias, C. Chung, D. J. Cooper, J. P. Turner, D. Vidović, M. Forlin, T. T. Kelley, A. D'Urso, et al. "Data Portal for the Library of Integrated Network-based Cellular Signatures (LINCS) program: integrated access to diverse large-scale cellular perturbation response data". In: *Nucleic acids research* 46.D1 (2018), pp. D558–D566.

[212] L. Litichevskiy, R. Peckner, J. G. Abelin, J. K. Asiedu, A. L. Creech, J. F. Davis, D. Davison, C. M. Dunning, J. D. Egertson, S. Egri, et al. "A library of phosphoproteomic and chromatin signatures for characterizing cellular responses to drug perturbations". In: *Cell systems* 6.4 (2018), pp. 424–443.

[213] J. O'Neil, Y. Benita, I. Feldman, M. Chenard, B. Roberts, Y. Liu, J. Li, A. Kral, S. Lejnine, A. Loboda, et al. "An unbiased oncology compound screen to identify novel combination strategies". In: *Molecular cancer therapeutics* 15.6 (2016), pp. 1155–1162.

[214] S. L. Holbeck, R. Camalier, J. A. Crowell, J. P. Govindharajulu, M. Hollingshead, L. W. Anderson, E. Polley, L. Rubinstein, A. Srivastava, D. Wilsker, et al. "The National Cancer Institute ALMANAC: a comprehensive screening resource for the detection of anticancer drug pairs with enhanced therapeutic activity". In: *Cancer research* 77.13 (2017), pp. 3564–3576.

[215] M. P. Menden, D. Wang, M. J. Mason, B. Szalai, K. C. Bulusu, Y. Guan, T. Yu, J. Kang, M. Jeon, R. Wolfinger, et al. "Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen". In: *Nature communications* 10.1 (2019), p. 2674.

[216] D. A. Bennett, A. S. Buchman, P. A. Boyle, L. L. Barnes, R. S. Wilson & J. A. Schneider. "Religious orders study and rush memory and aging project". In: *Journal of Alzheimer's disease* 64.s1 (2018), S161–S189.

[217] M. Wang, N. D. Beckmann, P. Roussos, E. Wang, X. Zhou, Q. Wang, C. Ming, R. Neff, W. Ma, J. F. Fullard, et al. "The Mount Sinai cohort of large-scale genomic, transcriptomic and proteomic data in Alzheimer's disease". In: *Scientific data* 5.1 (2018), pp. 1–16.

[218] M. Allen, M. M. Carrasquillo, C. Funk, B. D. Heavner, F. Zou, C. S. Younkin, J. D. Burgess, H.-S. Chai, J. Crook, J. A. Eddy, et al. "Human whole genome genotype and transcriptome data for Alzheimer's and other neurodegenerative diseases". In: *Scientific data* 3.1 (2016), pp. 1–10.

[219] F. Zou, H. S. Chai, C. S. Younkin, M. Allen, J. Crook, V. S. Pankratz, M. M. Carrasquillo, C. N. Rowley, A. A. Nair, S. Middha, et al. "Brain expression genome-wide association study (eGWAS) identifies human disease-associated variants". In: *PLoS genetics* 8.6 (2012), e1002707.

[220] B. Cao, R. Y. Cho, D. Chen, M. Xiu, L. Wang, J. C. Soares & X. Y. Zhang. "Treatment response prediction and individualized identification of first-episode drug-naive schizophrenia using brain functional connectivity". In: *Molecular psychiatry* 25.4 (2020), pp. 906–913.

[221] Z. S. Chen, I. R. Galatzer-Levy, B. Bigio, C. Nasca, Y. Zhang, et al. “Modern views of machine learning for precision psychiatry”. In: *Patterns* 3.11 (2022).

[222] C.-R. Phang, C.-M. Ting, S. B. Samdin & H. C. Ombao. “Classification of EEG-based Effective Brain Connectivity in Schizophrenia using Deep Neural Networks”. In: *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)* (2019), pp. 401–406.

[223] S. L. Oh, J. Vicnesh, E. J. Ciaccio, R. Yuvaraj & U. R. Acharya. “Deep convolutional neural network model for automated diagnosis of schizophrenia using EEG signals”. In: *Applied Sciences* 9.14 (2019), p. 2870.

[224] Z. Aslan & M. Akin. “Automatic Detection of Schizophrenia by Applying Deep Learning over Spectrogram Images of EEG Signals.” In: *Traitement du Signal* 37.2 (2020).

[225] M. M. Mauschitz, F. G. Holz, R. P. Finger & M. M. Breteler. “Determinants of macular layers and optic disc characteristics on SD-OCT: The Rhineland study”. In: *Translational vision science & technology* 8.3 (2019), pp. 34–34.

[226] D. Garzone, R. P. Finger, M. M. Mauschitz, A. Koch, M. Reuter, M. M. Breteler & N. A. Aziz. “Visual impairment and retinal and brain neurodegeneration: A population-based study”. In: *Human Brain Mapping* 44.7 (2023), pp. 2701–2711.

[227] V. Lohner, R. Lu, S. J. Enkirch, T. Stöcker, E. Hattingen & M. M. Breteler. “Incidental findings on 3 T neuroimaging: cross-sectional observations from the population-based Rhineland Study”. In: *Neuroradiology* (2022), pp. 1–10.

[228] T. Ballarini, D. M. van Lent, J. Brunner, A. Schröder, S. Wolfsgruber, S. Altenstein, F. Brosseron, K. Buerger, P. Dechent, L. Dobisch, et al. “Mediterranean diet, Alzheimer disease biomarkers, and brain atrophy in old age”. In: *Neurology* 96.24 (2021), e2920–e2932.

[229] N. A. Aziz, V. M. Corman, A. K. Echterhoff, M. A. Müller, A. Richter, A. Schmandke, M. L. Schmidt, T. H. Schmidt, F. M. de Vries, C. Drosten, et al. “Seroprevalence and correlates of SARS-CoV-2 neutralizing antibodies from a population-based study in Bonn, Germany”. In: *Nature communications* 12.1 (2021), p. 2117.

[230] R. Lu, N. A. Aziz, K. Diers, T. Stöcker, M. Reuter & M. M. Breteler. “Insulin resistance accounts for metabolic syndrome-related alterations in brain structure”. In: *Human brain mapping* 42.8 (2021), pp. 2434–2444.

[231] A. C. Aschenbrenner, M. Mouktaroudi, B. Krämer, M. Oestreich, N. Antonakos, M. Nuesch-Germano, K. Gkizeli, L. Bonaguro, N. Reusch, K. Baßler, et al. “Disease severity-specific neutrophil signatures in blood transcriptomes stratify COVID-19 patients”. In: *Genome medicine* 13.1 (2021), pp. 1–25.

[232] J. L. Hill. “Bayesian nonparametric modeling for causal inference”. In: *Journal of Computational and Graphical Statistics* 20.1 (2011), pp. 217–240.

[233] S. Athey & G. Imbens. "Recursive partitioning for heterogeneous causal effects". In: *Proceedings of the National Academy of Sciences* 113.27 (2016), pp. 7353–7360.

[234] S. Wager & S. Athey. "Estimation and inference of heterogeneous treatment effects using random forests". In: *Journal of the American Statistical Association* 113.523 (2018), pp. 1228–1242.

[235] S. Athey, J. Tibshirani & S. Wager. "Generalized random forests". In: *The Annals of Statistics* 47.2 (2019), pp. 1148–1178.

[236] A. M. Alaa & M. Van Der Schaar. "Bayesian inference of individualized treatment effects using multi-task gaussian processes". In: *Advances in neural information processing systems* 30 (2017).

[237] A. Alaa & M. Schaar. "Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 129–138.

[238] J. Yoon, J. Jordon & M. Van Der Schaar. "GANITE: Estimation of individualized treatment effects using generative adversarial nets". In: *International conference on learning representations*. 2018.

[239] F. Johansson, U. Shalit & D. Sontag. "Learning representations for counterfactual inference". In: *International conference on machine learning*. PMLR. 2016, pp. 3020–3029.

[240] U. Shalit, F. D. Johansson & D. Sontag. "Estimating individual treatment effect: generalization bounds and algorithms". In: *International conference on machine learning*. PMLR. 2017, pp. 3076–3085.

[241] F. D. Johansson, N. Kallus, U. Shalit & D. Sontag. "Learning weighted representations for generalization across designs". In: *arXiv preprint arXiv:1802.08598* (2018).

[242] N. Hassanpour & R. Greiner. "Learning disentangled representations for counterfactual regression". In: *International Conference on Learning Representations*. 2019.

[243] S. Assaad, S. Zeng, C. Tao, S. Datta, N. Mehta, R. Henao, F. Li & L. Carin. "Counterfactual representation learning with balancing weights". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 1972–1980.

[244] J. Morrill, P. Kidger, L. Yang & T. Lyons. "Neural Controlled Differential Equations for Online Prediction Tasks". In: *arXiv preprint* (2021).

[245] N. Seedat, F. Imrie, A. Bellot, Z. Qian & M. van der Schaar. "Continuous-Time Modeling of Counterfactual Outcomes Using Neural Controlled Differential Equations". In: *International Conference on Machine Learning*. 2022.

[246] D. Gwak, G. Sim, M. Poli, S. Massaroli, J. Choo & E. Choi. "Neural ordinary differential equations for intervention modeling". In: *arXiv preprint arXiv:2010.08304* (2020).

[247] A. Curth, C. Lee & M. van der Schaar. "SurvITE: Learning Heterogeneous Treatment Effects from Time-to-Event Data". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 26740–26753.

[248] A. Curth & M. van der Schaar. "Understanding the Impact of Competing Events on Heterogeneous Treatment Effect Estimation from Time-to-Event Data". In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Vol. 206. PMLR, 2023, pp. 7961–7980.

[249] C. M. Carvalho, N. G. Polson & J. G. Scott. "The horseshoe estimator for sparse signals". In: *Biometrika* 97.2 (2010), pp. 465–480.

[250] R. Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1 (1996), pp. 267–288.

[251] M. Evans, N. Hastings, B. Peacock & C. Forbes. *Statistical distributions*. John Wiley & Sons, 2011.

[252] D. J. MacKay. "Comparison of approximate methods for handling hyperparameters". In: *Neural computation* 11.5 (1999), pp. 1035–1068.

[253] D. Duvenaud, J. Lloyd, R. Grosse, J. Tenenbaum & G. Zoubin. "Structure discovery in nonparametric regression through compositional kernel search". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 1166–1174.

[254] R. Calandra, J. Peters, C. E. Rasmussen & M. P. Deisenroth. "Manifold Gaussian processes for regression". In: *2016 International joint conference on neural networks (IJCNN)*. IEEE. 2016, pp. 3338–3345.

[255] A. G. Wilson, Z. Hu, R. Salakhutdinov & E. P. Xing. "Deep kernel learning". In: *Artificial intelligence and statistics*. PMLR. 2016, pp. 370–378.

[256] H. Salimbeni & M. Deisenroth. "Doubly stochastic variational inference for deep Gaussian processes". In: *Advances in neural information processing systems* 30 (2017).

[257] J. Hensman, N. Fusi & N. D. Lawrence. "Gaussian Processes for Big Data". In: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2013, pp. 282–290.

[258] J. Hensman, A. G. de G. Matthews & Z. Ghahramani. "Scalable Variational Gaussian Process Classification." In: *AISTATS*. Vol. 38. JMLR.org, 2015.

[259] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser & I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[260] S. N. Shukla & B. Marlin. "Multi-Time Attention Networks for Irregularly Sampled Time Series". In: *International Conference on Learning Representations*. 2021.

[261] Q. Zhang, A. Lipani, O. Kirnap & E. Yilmaz. "Self-attentive Hawkes process". In: *International conference on machine learning*. PMLR. 2020, pp. 11183–11193.

[262] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall & N. D. Goodman. "Pyro: Deep Universal Probabilistic Programming". In: *Journal of Machine Learning Research* (2018).

[263] D. P. Kingma & M. Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[264] I. Loshchilov & F. Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019.

A Supplementary Information: Descriminative Causal Learning (D^{2}CL)

A.1 Gold-standard simulated benchmark data: Direct causal effects

Table A.1: Problem Class I: AUPRC values for *direct* cause-effect relations for $p = |V| = 1500$.

	Linear				MLP(tanh)				MLP(leaky ReLU)				Tanh				Leaky ReLU				Polynom 3			
SNR	Pearson	IDA	DDCL	SCL	Pearson	IDA	DDCL	SCL	Pearson	IDA	SCL	DDCL	Pearson	IDA	DDCL	SCL	Pearson	IDA	DDCL	SCL	Pearson	IDA	DDCL	SCL
10.00	0.729	0.733	0.833	0.719	0.696	0.549	0.795	0.656	0.716	0.496	0.786	0.599	0.733	0.770	0.853	0.882	0.776	0.834	0.850	0.816	0.822	0.819	0.821	0.742
6.00	0.698	0.745	0.754	0.711	0.663	0.521	0.784	0.631	0.683	0.493	0.783	0.530	0.707	0.766	0.879	0.860	0.756	0.796	0.844	0.782	0.792	0.771	0.821	0.723
4.00	0.663	0.737	0.778	0.709	0.643	0.518	0.765	0.674	0.657	0.466	0.766	0.499	0.681	0.729	0.843	0.846	0.721	0.796	0.833	0.780	0.750	0.744	0.820	0.712
2.00	0.605	0.731	0.806	0.700	0.589	0.484	0.739	0.567	0.613	0.445	0.784	0.621	0.622	0.676	0.848	0.829	0.661	0.704	0.799	0.709	0.644	0.630	0.771	0.668
1.00	0.568	0.703	0.802	0.694	0.549	0.446	0.766	0.5	0.575	0.417	0.761	0.5	0.563	0.638	0.802	0.791	0.607	0.648	0.763	0.673	0.552	0.564	0.677	0.620
0.75	0.571	0.698	0.798	0.694	0.537	0.441	0.719	0.634	0.558	0.421	0.767	0.674	0.539	0.603	0.801	0.755	0.583	0.634	0.764	0.665	0.541	0.544	0.675	0.609
0.50	0.545	0.688	0.769	0.685	0.532	0.423	0.758	0.5	0.540	0.408	0.768	0.647	0.520	0.596	0.790	0.720	0.551	0.622	0.746	0.651	0.536	0.551	0.645	0.588
0.25	0.521	0.676	0.784	0.5	0.517	0.417	0.754	0.5	0.479	0.394	0.763	0.594	0.509	0.592	0.758	0.644	0.508	0.599	0.726	0.643	0.518	0.548	0.617	0.5
0.10	0.534	0.677	0.784	0.679	0.485	0.407	0.728	0.613	0.516	0.404	0.751	0.573	0.502	0.548	0.737	0.579	0.523	0.608	0.761	0.632	0.485	0.517	0.636	0.543

A.2 Gold-standard simulated benchmark data: Total causal effects

Table A.2: Problem Class II: AUPRC values for *ancestral* cause-effect relations for $p = |V| = 1500$.

	Linear				MLP(tanh)				MLP(leaky ReLU)				Tanh				Leaky ReLU				Polynom 3			
SNR	Pearson	IDA	DDCL	SCL	Pearson	IDA	DDCL	SCL	Pearson	IDA	SCL	DDCL	Pearson	IDA	DDCL	SCL	Pearson	IDA	DDCL	SCL	Pearson	IDA	DDCL	SCL
10.00	0.581	0.878	0.892	0.632	0.545	0.547	0.760	0.680	0.545	0.490	0.727	0.809	0.594	0.899	0.926	0.883	0.526	0.829	0.916	0.816	0.639	0.794	0.904	0.690
6.00	0.557	0.878	0.886	0.604	0.531	0.525	0.748	0.623	0.531	0.490	0.697	0.806	0.573	0.872	0.924	0.864	0.517	0.816	0.910	0.783	0.629	0.786	0.896	0.664
4.00	0.543	0.879	0.890	0.581	0.523	0.521	0.731	0.624	0.525	0.479	0.714	0.785	0.554	0.872	0.922	0.854	0.513v	0.806	0.900	0.708	0.604	0.761	0.866	0.639
2.00	0.512	0.853	0.878	0.519	0.517	0.484	0.701	0.635	0.513	0.457	0.700	0.790	0.522	0.818	0.905	0.795	0.505	0.790	0.834	0.640	0.565	0.715	0.816	0.601
1.00	0.510	0.847	0.879	0.479	0.514	0.470	0.695	0.650	0.503	0.436	0.680	0.769	0.492	0.768	0.821	0.718	0.503	0.781	0.825	0.581	0.533	0.690	0.741	0.588
0.75	0.517	0.839	0.861	0.554	0.517	0.483	0.669	0.636	0.502	0.442	0.676	0.759	0.483	0.747	0.822	0.700	0.504	0.788	0.811	0.590	0.523	0.692	0.732	0.577
0.50	0.506	0.833	0.864	0.534	0.511	0.468	0.629	0.618	0.506	0.435	0.673	0.774	0.467	0.727	0.789	0.652	0.495	0.771	0.811	0.528	0.512	0.684	0.674	0.565
0.25	0.523	0.831	0.870	0.572	0.518	0.467	0.577	0.522	0.506	0.436	0.649	0.712	0.479	0.714	0.751	0.593	0.500	0.776	0.814	0.537	0.506	0.669	0.694	0.565
0.10	0.515	0.827	0.865	0.531	0.495	0.453	0.673	0.524	0.502	0.431	0.574	0.751	0.474	0.680	0.711	0.548	0.502	0.774	0.810	0.545	0.507	0.678	0.706	0.545

A.3 Gold-standard simulated benchmark data: Scalability experiment with *p=50,000*

Table A.3: D^2CL: AUC values for direct cause-effect relations for $p = 50,000$.

SNR	Linear	Tanh	Leaky ReLU	Polynom 3
6.0	0.765	0.861	0.823	0.748
2.0	0.759	0.857	0.829	0.747
1.0	0.773	0.852	0.825	0.740
0.5	0.757	0.840	0.826	0.745
0.1	0.764	0.748	0.827	0.713

A.4 Gold-standard simulated benchmark data: Direct causal effect with different noise types

Table A.4: Gold-standard simulations: AUPRC values for *direct* cause-effect relations for $p = |V| = 1500$: additive and multiplicative noise

	deterministic hard interventions				stochastic hard interventions			
	additive Noise		multiplicative Noise		additive Noise		multiplicative Noise	
SNR	Linear	Tanh	Linear	Tanh	Linear	Tanh	Linear	Tanh
10.00	0.834	0.853	0.691	0.776	0.768	0.795	0.692	0.784
6.00	0.755	0.879	0.645	0.745	0.767	0.787	0.655	0.757
4.00	0.779	0.843	0.631	0.739	0.758	0.791	0.639	0.765
2.00	0.807	0.848	0.609	0.743	0.753	0.780	0.621	0.742
1.00	0.803	0.803	0.626	0.732	0.753	0.689	0.615	0.704
0.75	0.798	0.801	0.627	0.714	0.734	0.668	0.634	0.699
0.50	0.770	0.791	0.617	0.717	0.748	0.680	0.625	0.735
0.25	0.785	0.758	0.611	0.718	0.732	0.676	0.628	0.705
0.10	0.784	0.737	0.599	0.718	0.739	0.664	0.607	0.715

A.5 Gold-standard simulated benchmark data: Out-of-distribution testing

Table A.5: Out-of-distribution experiments: Cross validation experiments - AUPRC values

			Predicted class																
			Linear								LeakyReLU								
		SNR	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10
Trained class	Linear	10.00	0.757	0.762	0.763	0.765	0.758	0.752	0.743	0.726	0.703	0.715	0.719	0.718	0.708	0.680	0.675	0.666	0.651	0.647
		6.00	0.767	0.769	0.768	0.767	0.752	0.746	0.736	0.718	0.691	0.713	0.713	0.709	0.691	0.666	0.661	0.656	0.648	0.647
		4.00	0.755	0.761	0.764	0.769	0.759	0.752	0.742	0.722	0.699	0.678	0.683	0.683	0.679	0.664	0.663	0.658	0.653	0.647
		2.00	0.742	0.749	0.757	0.770	0.765	0.760	0.748	0.727	0.692	0.669	0.674	0.677	0.688	0.679	0.677	0.668	0.656	0.649
		1.00	0.707	0.714	0.723	0.744	0.753	0.755	0.753	0.739	0.704	0.652	0.658	0.660	0.679	0.679	0.676	0.670	0.658	0.650
		0.75	0.685	0.695	0.706	0.736	0.753	0.753	0.751	0.738	0.718	0.585	0.593	0.605	0.643	0.650	0.654	0.659	0.660	0.650
		0.50	0.679	0.687	0.697	0.729	0.746	0.750	0.753	0.745	0.730	0.606	0.609	0.617	0.636	0.651	0.656	0.656	0.658	0.655
		0.25	0.646	0.650	0.658	0.693	0.721	0.730	0.740	0.741	0.733	0.576	0.579	0.584	0.605	0.631	0.637	0.641	0.650	0.656
		0.10	0.629	0.633	0.642	0.670	0.692	0.700	0.711	0.725	0.736	0.614	0.617	0.621	0.630	0.626	0.625	0.623	0.629	0.649
	Tanh	10.00	0.685	0.689	0.690	0.689	0.667	0.656	0.642	0.625	0.603	0.765	0.764	0.753	0.720	0.675	0.667	0.655	0.622	0.610
		6.00	0.683	0.686	0.687	0.685	0.663	0.652	0.638	0.618	0.595	0.759	0.761	0.754	0.717	0.670	0.662	0.651	0.620	0.606
		4.00	0.683	0.687	0.688	0.688	0.664	0.652	0.640	0.623	0.599	0.751	0.752	0.746	0.713	0.661	0.658	0.646	0.616	0.607
		2.00	0.674	0.677	0.677	0.687	0.667	0.652	0.633	0.615	0.599	0.707	0.709	0.713	0.712	0.651	0.648	0.633	0.605	0.596
		1.00	0.686	0.688	0.688	0.699	0.701	0.691	0.672	0.652	0.640	0.684	0.690	0.691	0.700	0.695	0.688	0.666	0.641	0.633
		0.75	0.675	0.679	0.681	0.694	0.703	0.698	0.687	0.658	0.647	0.675	0.682	0.683	0.697	0.699	0.691	0.670	0.651	0.632
		0.50	0.681	0.685	0.688	0.700	0.703	0.699	0.685	0.663	0.650	0.638	0.645	0.656	0.674	0.687	0.686	0.673	0.652	0.631
		0.25	0.645	0.649	0.654	0.672	0.696	0.700	0.703	0.688	0.667	0.549	0.551	0.564	0.597	0.617	0.626	0.645	0.659	0.650
		0.10	0.630	0.633	0.633	0.641	0.658	0.670	0.683	0.696	0.689	0.483	0.485	0.481	0.483	0.514	0.543	0.585	0.638	0.661

			Predicted class																
			Polynom 3									Tanh								
		SNR	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10
Trained class	Linear	10.00	0.678	0.675	0.675	0.666	0.645	0.643	0.633	0.626	0.616	0.707	0.708	0.708	0.704	0.677	0.667	0.645	0.617	0.595
		6.00	0.701	0.698	0.694	0.674	0.650	0.644	0.632	0.623	0.619	0.723	0.722	0.721	0.709	0.677	0.665	0.644	0.619	0.598
		4.00	0.656	0.654	0.656	0.657	0.641	0.640	0.632	0.627	0.619	0.690	0.691	0.694	0.695	0.672	0.665	0.643	0.614	0.598
		2.00	0.650	0.648	0.649	0.652	0.646	0.645	0.636	0.629	0.621	0.693	0.695	0.697	0.698	0.678	0.669	0.648	0.620	0.597
		1.00	0.633	0.631	0.632	0.635	0.631	0.631	0.627	0.627	0.623	0.671	0.672	0.673	0.672	0.660	0.656	0.644	0.623	0.600
		0.75	0.549	0.554	0.571	0.606	0.622	0.626	0.626	0.631	0.625	0.626	0.631	0.638	0.653	0.651	0.650	0.643	0.626	0.599
		0.50	0.583	0.590	0.608	0.627	0.626	0.628	0.625	0.630	0.627	0.653	0.657	0.661	0.669	0.659	0.655	0.643	0.623	0.600
		0.25	0.530	0.530	0.537	0.570	0.598	0.608	0.612	0.622	0.627	0.592	0.596	0.604	0.620	0.633	0.636	0.630	0.620	0.603
		0.10	0.571	0.572	0.577	0.598	0.604	0.606	0.603	0.608	0.620	0.627	0.629	0.634	0.642	0.638	0.637	0.626	0.612	0.601
	Tanh	10.00	0.752	0.746	0.737	0.701	0.656	0.637	0.619	0.590	0.586	0.806	0.803	0.797	0.757	0.684	0.653	0.610	0.590	0.575
		6.00	0.752	0.750	0.744	0.709	0.662	0.640	0.620	0.591	0.585	0.805	0.808	0.805	0.765	0.682	0.649	0.608	0.591	0.574
		4.00	0.735	0.737	0.736	0.707	0.662	0.637	0.617	0.588	0.584	0.798	0.803	0.808	0.778	0.690	0.657	0.607	0.586	0.573
		2.00	0.695	0.696	0.694	0.695	0.662	0.634	0.610	0.578	0.571	0.749	0.757	0.771	0.794	0.729	0.690	0.629	0.580	0.560
		1.00	0.667	0.668	0.671	0.676	0.673	0.656	0.627	0.606	0.599	0.718	0.720	0.727	0.744	0.735	0.720	0.661	0.598	0.587
		0.75	0.660	0.660	0.662	0.669	0.663	0.660	0.640	0.616	0.595	0.702	0.705	0.710	0.718	0.706	0.699	0.667	0.609	0.577
		0.50	0.620	0.621	0.625	0.644	0.651	0.653	0.641	0.621	0.600	0.653	0.656	0.663	0.673	0.678	0.674	0.656	0.612	0.576
		0.25	0.539	0.543	0.543	0.560	0.586	0.591	0.616	0.627	0.620	0.562	0.565	0.568	0.582	0.593	0.596	0.619	0.634	0.596
		0.10	0.520	0.520	0.518	0.523	0.526	0.537	0.555	0.610	0.628	0.503	0.505	0.507	0.508	0.512	0.517	0.540	0.598	0.617

			Predicted class																
			MLP (leaky ReLU)									MLP (Tanh)								
		SNR	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.1	10.00	6.00	4.00	2.00	1.00	0.75	0.50	0.25	0.10
Trained class	Linear	10.00	0.459	0.452	0.447	0.427	0.433	0.435	0.437	0.441	0.447	0.466	0.455	0.447	0.434	0.442	0.444	0.446	0.455	0.459
		6.00	0.454	0.451	0.450	0.431	0.435	0.437	0.438	0.443	0.448	0.459	0.454	0.449	0.436	0.443	0.446	0.447	0.456	0.459
		4.00	0.492	0.487	0.480	0.457	0.448	0.445	0.444	0.444	0.449	0.494	0.487	0.477	0.461	0.450	0.451	0.448	0.453	0.461
		2.00	0.488	0.481	0.467	0.440	0.438	0.440	0.441	0.444	0.448	0.490	0.483	0.469	0.450	0.446	0.445	0.444	0.456	0.457
		1.00	0.503	0.500	0.495	0.464	0.447	0.445	0.443	0.444	0.449	0.497	0.493	0.489	0.471	0.455	0.450	0.446	0.453	0.458
		0.75	0.514	0.510	0.501	0.457	0.443	0.442	0.440	0.442	0.445	0.520	0.514	0.504	0.468	0.451	0.449	0.446	0.452	0.456
		0.50	0.489	0.485	0.474	0.446	0.439	0.440	0.439	0.441	0.446	0.496	0.487	0.478	0.457	0.449	0.447	0.446	0.454	0.457
		0.25	0.496	0.495	0.492	0.466	0.452	0.449	0.443	0.444	0.444	0.504	0.500	0.497	0.479	0.461	0.455	0.450	0.453	0.456
		0.10	0.468	0.467	0.465	0.453	0.447	0.444	0.441	0.444	0.443	0.477	0.472	0.469	0.458	0.449	0.453	0.450	0.453	0.457
	Tanh	10.00	0.450	0.443	0.431	0.427	0.438	0.442	0.448	0.457	0.464	0.457	0.450	0.436	0.437	0.450	0.452	0.456	0.467	0.477
		6.00	0.436	0.434	0.432	0.432	0.441	0.444	0.449	0.458	0.458	0.444	0.439	0.438	0.440	0.448	0.452	0.458	0.468	0.473
		4.00	0.448	0.443	0.435	0.429	0.437	0.440	0.446	0.455	0.461	0.451	0.443	0.438	0.435	0.446	0.450	0.452	0.468	0.476
		2.00	0.450	0.449	0.448	0.436	0.437	0.441	0.444	0.455	0.464	0.451	0.452	0.454	0.444	0.445	0.448	0.450	0.467	0.478
		1.00	0.447	0.442	0.441	0.440	0.438	0.443	0.447	0.450	0.454	0.447	0.444	0.443	0.442	0.447	0.447	0.445	0.459	0.466
		0.75	0.474	0.474	0.470	0.450	0.451	0.451	0.445	0.450	0.452	0.477	0.479	0.475	0.459	0.458	0.455	0.453	0.457	0.462
		0.50	0.486	0.493	0.497	0.500	0.485	0.467	0.456	0.450	0.451	0.490	0.493	0.500	0.506	0.481	0.467	0.456	0.453	0.457
		0.25	0.524	0.528	0.529	0.536	0.519	0.514	0.503	0.477	0.459	0.516	0.517	0.519	0.523	0.514	0.509	0.507	0.480	0.460
		0.10	0.519	0.525	0.524	0.521	0.529	0.540	0.556	0.572	0.500	0.517	0.519	0.520	0.518	0.534	0.532	0.555	0.580	0.485

A.5.1 Yeast Gene Deletion Experiments

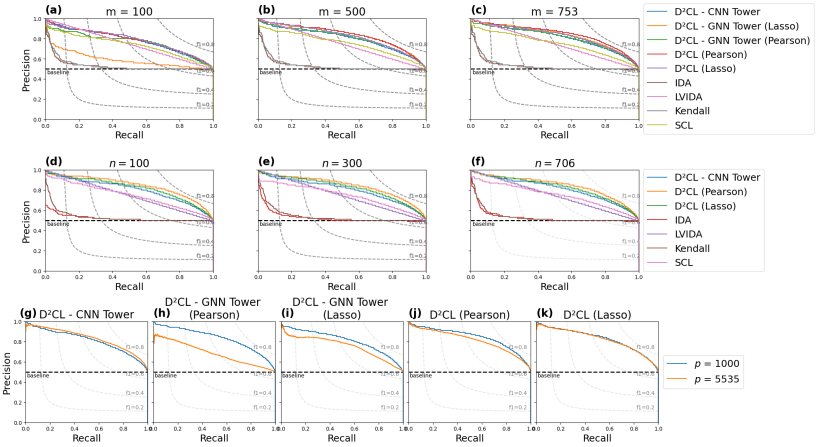


Figure A.1: Results - yeast gene deletion experiments. Causal learning methods, including D^2CL, were applied to gene expression measurements from yeast cells. Performance was quantified using causal Precision-Recall curves (and the area under the PR-curves, or AUPRC see. Table A.6) computed with respect to a causal ground truth obtained from entirely unseen interventional experiments (see Text for details). Panels (a)–(c): the number of interventions whose effects are available to the learner is varied as shown (with problem dimension fixed to p=1000 and sample size to n=706). Panels (d)–(f): the sample size n of the data matrix X is varied as shown (with problem dimension fixed to p=1000 and number of available interventions fixed to to m=753). Panels (g)–(k): analogous results for a higher-dimensional setting covering all available genes (roughly the full yeast genome) with p=5535 (with n=706 and m=753). Here, only D^2CL variants are shown, as the other methods could not be run due to the computational burden in this higher dimensional case. Comparison with the corresponding p=1000 case demonstrates the scalability of D^2CL, with performance broadly maintained in the higher dimensional setting. [D^2CL variants shown include a CNN tower alone, GNN tower alone and the entire D^2CL architecture; methods compared against include IDA, LVIDA, Kendall correlations (as a non-causal baseline) and SCL (see text and SI for details and references). For D^2CL and its variants two different initial graph estimates were used based respectively on Pearson correlation coefficients ("Pearson") and on a lightweight regression ("Lasso"; see Text for details).]

Table A.6: Yeast Gene Deletion - AUPRC scores: Causal learning methods, including D^2CL, were applied to gene expression measurements from yeast cells. Performance was quantified using causal area under Precision-Recall curves. Subtables extend results of Figure A.1, Subtable A.6a extends panels (a)-(c), Subtable A.6b provides AUPRC scores for panels (d)-(f), and Subtable A.6c presents scores for panels (g)-(k).

(a) Varying amount of causal input m

	m=100	m=500	m=753
D²CL - CNN Tower	0.784	0.804	0.815
D²CL - GNN Tower (Lasso)	0.597	0.837	0.835
D²CL - GNN Tower (Pearson)	0.766	0.811	0.814
D²CL (Pearson)	0.787	0.839	0.845
D²CL (Lasso)	0.792	0.822	0.828
IDA	0.533	0.533	0.533
LVIDA	0.499	0.499	0.499
Kendall	0.538	0.538	0.538
SCL	0.737	0.752	0.749

(b) Varying sample size n

	n=100	n=300	n=706
D²CL - CNN Tower	0.806	0.807	0.815
D²CL (Pearson)	0.846	0.845	0.845
D²CL (Lasso)	0.824	0.827	0.828
IDA	0.524	0.533	0.533
LVIDA	0.499	0.499	0.499
Kendall	0.536	0.543	0.538
SCL	0.749	0.749	0.749

(c) Scaling to full yeast genome

	p=1000	p=5535
D²CL - CNN Tower	0.815	0.829
D²CL - GNN Tower (Pearson)	0.835	0.683
D²CL - GNN Tower (Lasso)	0.814	0.753
D²CL (Pearson)	0.845	0.811
D²CL (Lasso)	0.828	0.821

B Supplementary Information: Learning Latent Dynamics via Invariance Decomposition (LaDID)

B.1 From invariances to a simple learning framework

In this Section, we consider an abstract version of the problem of interest, aimed at clarifying the specific invariances that will be needed and gaining understanding of how learning can be conveniently performed in this setting. The notation in this Section is self-contained but (in the interests of expositional clarity) differs from Section 3.3.2 and B.2 providing architectural and implementation details.

General formulation. Consider an entirely general system f which may be deterministic or stochastic (with all random components absorbed for convenience into f). We are interested in settings in which some aspects of the model are realization-specific (RS) while others remain realization-invariant (RI). Let

$$x_t^r = f(t; \Theta_r),\ x_t^r \in \mathbb{R}^p$$

denote the fully general model. Here, Θ_r is the complete parameter needed to specify the time-evolution, including both RS and RI parts. To make the separation clear, we write the two parts separately as $x_t^r = f(t; \theta_r, \theta)$, where θ_r, θ are respectively the RS and RI parameters (together comprising Θ_r).

We call this a *generalized initial condition formulation*, as it generalizes the idea of an initial condition in ODEs. In the case of an ODE, the initial conditions and relevant constants are the information needed, in addition to the model equations themselves, to fully specify the time evolution of any specific instance/realization of the model. In our terms, if a problem/dataset has one model but many such "initial conditions" (more precisely this can be any RS aspect, including constants), then the model itself is RI, while the initial conditions/constants are the RS part. Note that although we do not assume any specific knowledge about the system (other than the motivating invariances), we do assume that we can at the outset block datasets into instances arising from a shared system (whose details are entirely unknown); in this work we do not consider the task of learning the system classification itself from data.

Fully observed case. In the model f above, the true parameter $\Theta_r = (\theta_r, \theta)$ comprises RS and RI parts. We now provide conditions under which learning of the system state at any continuous time t is possible without explicit knowledge of either the model f or the true parameter Θ_r. We start with the simplest case of fully observed data (i.e. no latent dynamics) and then consider the latent case. The idea is to work from a candidate encoding $\hat{\theta}_r$ of the RS information. In practice this would be the output of a neural network (NN) based on initial data from a realization r and itself learned end-to-end jointly with other model components; see subsequent Sections for architectural and implementation details. The encoding $\hat{\theta}_r$ is intended to be a representation of RS information that, as we will see below, under certain assumptions can be combined with a universal model to allow effective prediction.

Specifically, we assume that the encoding $\hat{\theta}_r$, while possibly incorrect (i.e. such that $\hat{\theta}_r \neq \theta_r$) satisfies the property

$$\exists m, \exists \theta_m : \theta_r = m(\hat{\theta}_r; \theta, \theta_m),$$

where m is a function that "corrects" $\hat{\theta}_r$ to give the correct RS parameter. Note that the correction function m can potentially use the RI parameter of the system and possibly additional parameters θ_m. This essentially demands that while the encoding $\hat{\theta}_r$ might be very different from the true value (and may even diverge from it in a complicated way that depends on unknown system parameters), there exists an RI transformation that recovers the true RS parameter from it, and in this sense the encoding contains all RS information. We call this the *sufficient encoding assumption* (SEA). Note that the function m has an oracle-like property in that it may depend on the true RI parameter θ and we will not have access to m in practice.

We would like to learn a mapping that takes as input the RS encoding $\hat{\theta}_r$ and query time point t and yields the correct system output (for any realization and any time). To this end, consider the candidate prediction

$$\hat{x}_t^r = f(t; m(\hat{\theta}_r; \theta, \theta_m), \theta),$$

and observe that we can always write the RHS as $h(t, \hat{\theta}_r; \Xi)$ where $\Xi = (\theta, \theta_m)$ is a RI parameter and h is a function (obtained by combining f and m as above). This latter formulation emphasizes the fact that the RHS is in fact a function (here, h) of only the inputs $(t, \hat{\theta}_r)$, and therefore potentially learnable from training pairs. Note that the parameters of h are entirely RI and hence the only RS information is carried by the encoding $\hat{\theta}_r$. It is easy to see that under SEA this construction provides the correct output since we can write the RHS as $f(t; m(\hat{\theta}_r; \theta, \theta_m), \theta) = f(t; \theta_r, \theta) = x_t^r$. Thus, combining encoding $\hat{\theta}_r$ and function h allows prediction of the time evolution of any realization. In other words, even if the RS encoding is distant from the true RS parameter, under SEA there exists a RI function that can correct it, and we can therefore seek to train a NN aimed at learning a function h which combines these RI elements to provide the desired mapping.

Latent dynamics. In line with the manifold hypothesis, consider now dynamics at the level of latent variables $z \in \mathbb{R}^q$ and again consider a model with RS and RI parts but at the level of the latents, i.e. $z_t^r = f(t, \hat{\theta}_r; \theta_r, \theta)$. We assume that the observables are given by (an unknown) function of the hidden state z. We first consider the case in which the latent-to-observed mapping is RI, and then the more general case of a RS mapping.

Case I: RI mapping. Assume the observable is given as $x_t^r = g(z_t^r; \theta_g)$, where $g : \mathbb{R}^q \to \mathbb{R}^p$ is the (true) observation process and θ_g is an RI parameter. Further, assume that we have an estimate $\hat{\theta}_r$ of the RS encoding that satisfies the sufficient encoding assumption (SEA). In a similar fashion, assume we have an estimate $\hat{\theta}_g$, which may be incorrect (in the sense of $\hat{\theta}_g \neq \theta_g$) but that satisfies: $\exists m_g, \exists \theta_{m_g} : \theta_g = m_g(\hat{\theta}_g; \theta, \theta_{m_g})$. That is, $\hat{\theta}_g$ admits an RI correction. As above, the correction is oracle-like and may potentially depend on true RS parameters. Note also that subject to the existence of a correction the estimate (and implied mapping) may be potentially arbitrarily incorrect. In analogy to SEA, we call this the *sufficient mapping assumption* (SMA).

Now, consider training of a NN, with training (input, output) pairs of the form $\{(t, \hat{\theta}_r), x_t^r\}_{(t,r)\in Train}$. We want to understand whether supervised learning of a universal model to predict output for arbitrary queries (t, r) is possible. This is not obvious, since we now have training data only at the level of the observables, but the actual dynamics operate at the level of latents. Consider the following function h_{SMA}:

$$h_{SMA}(t, \hat{\theta}_r; \Phi) = g(f(t; m(\hat{\theta}_r; \theta, \theta_m); m_g(\hat{\theta}_g; \theta, \theta_{m_g}))$$

where $\Phi = (\theta, \theta_m, \theta_{m_g})$ is an RI parameter.

Under SEA and SMA it is easy to see that h_{SMA} provides the correct output, since:

$$\begin{aligned} h_{SMA}(t, \hat{\theta}_r; \Phi) &= g(f(t; m(\hat{\theta}_r; \theta, \theta_m); m_g(\hat{\theta}_g; \theta, \theta_{m_g})) \\ &= g(f(t; \theta_r, \theta); \theta_g) \\ &= g(z_t^r; \theta_g) \\ &= x_t^r \end{aligned}$$

That is, under SEA and SMA there exists a function of t and $\hat{\theta}_r$ that provides the correct output and that is universal in the sense that (i) the same function applies to any query (t, r) and (ii) its parameter is itself RI and hence the same for all realizations.

Case II: RS mapping. Suppose now the mapping is RS, with the model specification as above but with the observation step $x_t^r = g(z_t^r; \theta_g^r)$, where θ_g^r is an RS parameter. This means that the latent-observable relationship is itself non-constant and instead

varies between realizations.

Assume we have a candidate estimate $\hat{\theta}_g^r$ which may be incorrect in the sense of $\hat{\theta}_g^r \neq \theta_g^r$ but that satisfies: $\exists m_g, \exists \theta_{m_g} : \theta_g = m_g(\hat{\theta}_g^r; \theta, \theta_{m_g})$. In analogy to SMA, we call this the *realization-specific sufficient mapping assumption* or RS-SMA. Now to create training sets, we extend the formulation to require input triples, as: $\{(t, \hat{\theta}_r, \hat{\theta}_g^r), x_t^r\}_{(t,r)\in Train}$. In a similar spirit to the RS encoding above, $\hat{\theta}_g^r$ in the input triples may be incorrect, but only needs to satisfy RS-SMA. As in Case I, we have training data only at the level of observables (not latents) but want to understand whether supervised learning of a model to predict output for arbitrary queries (t, r) is possible. Consider the function h_{RS-SMA}:

$$h_{RS-SMA}(t, \hat{\theta}_r, \hat{\theta}_g^r; \Phi) = g(f(t; m(\hat{\theta}_r; \theta, \theta_m), \theta); m_g(\hat{\theta}_g^r; \theta, \theta_{m_g}))$$

where $\Phi = (\theta, \theta_m, \theta_{m_g})$ is an RI parameter. In a similar manner to Case I, it is easy to see that h_{RS-SMA} provides the correct output under SEA and RS-SMA, since:

$$\begin{aligned} h_{RS-SMA}(t, \hat{\theta}_r, \hat{\theta}_g^r; \Phi) &= g(f(t; m(\hat{\theta}_r; \theta, \theta_m), \theta); m_g(\hat{\theta}_g^r; \theta, \theta_{m_g})) \\ &= g(f(t; \theta_r, \theta); \hat{\theta}_g^r) \\ &= g(z_t^r; \hat{\theta}_g^r) \\ &= x_t^r \end{aligned}$$

The foregoing arguments are based on an abstract view of the task at hand and show that under the assumptions above, there exist universal mappings from the available inputs to the desired outputs whose parameters are themselves RI. As a result, subject to the assumptions above, it may be possible to learn suitable mapping functions from data (without requiring prior access to the various components). We now put forward a specific architecture aimed at learning such a mapping in practice.

B.2 Model Architecture

Encoder. The encoder is a collection of three NNs. First, features from the input observations $x_{t-k:t-1}^r$ are extracted using a convolutional neural network (CNN) parameterized by θ_{enc} and shared across all representations and patches. Specifically, the CNN encoder has the following architecture: three convolution layers (5x5 kernel, stride 2, padding 2) with batch norm and ReLU activations, one convolution layer (2x2 kernel, stride 2) with batch norm and ReLU activation. The channels of the respective CNN layers are doubled throughout. Finally, the downsampled image features are flattened and linearly projected to the output dimension. Hence, our encoder transforms the sequence of input observations to a sequence of feature vectors, $z_{t-k:T}^{(enc),r} = f_{\theta_{enc}}(x_{t-k:T}^r)$.
Then, we compute the trajectory representation and the latent embedding as follows. Each input patch is split into two disjoint sets by time. The first $k \in K$

data points $\mathcal{M}_R{=}\{z^{(enc),r}_{t-k:t}\}$ are used to compute a trajectory specific representation distribution $\psi^r{\sim}q_{\Theta_R}(x^r_{t-k:t-1}) = \mathcal{N}(\mu_r, \sigma_r)$ and $\mu_r, \sigma_r{=}f_{\theta_R}(z^{(enc),r}_{t-k:t})$. In cases of irregularly sampled trajectories, we use a time threshold τ to define the representation set, $\mathcal{M}_R{=}\{z^{(enc),r}_{t_i}\}$, $t_i{\in}\{t{<}\tau\}$. We model f_{θ_R} as a transformer network with temporal attention. In other words, we consider the sequence feature vectors $z^{(enc),r}_{t-k:t}$ as an time-ordered sequence of tokens and transform each token according to the temporal distance to the other tokens. We compared two approaches of temporal attention [117] which performed roughly similar. First, temporal reweighting is performed as introduced [117]: $C_{ij} = c_{ij}/\sum_{k=1}^{K} c_{ik}$ with $c_{ij} = \exp(\left\langle W_Q h_i, W_K hj \right\rangle + \ln(\epsilon)(|t_j - t_i|/\delta)^p)$, where $\langle\cdot,\cdot\rangle$ denotes the dot product, W_K, W_Q, and W_V represent the weight matrices for the query, key, value as in regular attention. ϵ and δ are constants. Hence, the larger the distance $|t_j - t_i|$ grows, the stronger the time-aware attention is reduced [117]. The parameter δ determines the distance threshold beyond which the scaling of regular attention occurs by at least ϵ. Moreover, parameter p governs the shape of the scaling curve. This methods works best for most of the dynamical systems. Second, we tested a temporal attention approach as defined in [129]. This time aware attention is given by $C^{TA}(t) = \sum_{t'=0}^{T-1} softmax(\frac{\langle W_Q \rho_t, W_K \rho_{t'} \rangle}{\sqrt{d}}) W_V \rho_{t'}$.
Finally, the trajectory representation ψ^r is obtained applying a *mean*-aggregation of the temporally transformed representation tokens.

Dynamics model. With initial density given by the encoder networks $q_{\Theta_L}(z_t|x^r_{t-k:T}, \psi^r)$, the density for all queried latent points (on a continuous time grid) can be predicted by $z^r_{t_q}{\sim}\mathcal{N}(\mu^r_{t_q}, \sigma^r_{t_q})$ with $\mu^r_{t_q}, \sigma^r_{t_q}{=}f_{\theta_{dyn}}(t_q, z_t, \psi^r)$. Note that this approach allows for latent state predictions at any time since the learned dynamics module $f_{\theta_{dyn}}$ is continuous in time and our variational model utilizes encoders only for obtaining the initial latent distribution. We also make use of the reparameterization trick to tackle uncertainties in both, the latent states and in the trajectory representations [263]. In our implementation $f_{\theta_{dyn}}$ consists of three linear layers, with the first two followed by a ReLU non-linearity.

Decoder. The decoder maps the latent trajectory points back to the observational space. Hence, our implementation is fairly simple and comprises a set of transposed convolutional layers. In particular, it first projects latent trajectory points linearly followed by four transposed convolution layers (2x2 kernel, stride 2) with batch norm and ReLU non-linearities. Finally, a convolutional layer (5x5 kernel, padding 2) with sigmoid function computes our output distribution. The channel dimension of the four transposed convolution layers is halved subsequently from layer to layer.

B.3 Training details and hyperparameters

Our implementation uses the PyTorch framework [45]. All modules are initialized from scratch using random weights. During training, an AdamW-Optimizer [264] is applied starting at an initial learning rate $\varepsilon_0 = 0.0003$. An exponential learning rate scheduler is applied showing the best results in the current study. Every network is trained for 30 000 epochs. At initialization, we start training at a subpatch length of 1 which is doubled every 3000 epochs. After the CNN encoder, 8 attention layers are stacked each using 4 attention heads. A relative Sin/Cos embedding is used as position encoding followed by a linear layer. The input resolution of the observational image data is 128×128 px. All computations are run on a single GPU node equipped with one NVidia A100 (40 GB) and a global batch size of 16 is used. A full training run on the single pendulum, the double pendulum, the wave equation and the Navier-Stokes equation dataset requires approx. 14 h. A full training run on the reaction-diffusion system and the von Kármán vortex street requires approx. 8 h.

Table B.1: Training hyperparameters

Hyperparameter	Value
LR schedule	Exp. decay
Initial LR	3e-4
Weight Decay	0.01
Global batch size	16
Parallel GPUs	1
Input resolution	128×128 px
Number of input time steps	10
Initial subpatch length	1
Number of epochs per subpatch length	3000
Latent dimension	32
attention mechanism	spatio-temporal
Number of attention blocks	8
Number of attention heads	4
Position Encoding	relative Sin/Cos encoding

C Supplementary Information: Forecasting Responses in Unpaired Interventional Data using Sparse Causal Modeling

C.1 Identifiability of sparse causal relations for unpaired data

Consider a data generating process of the form 4.1. The true parameter vector β^* satisfies the moment condition

$$\mathrm{Cov}\left(\tilde{X}, Y - Y^T\beta^*\right) = 0. \tag{C.1}$$

This scenario can be considered analogous to an instrumental variable setting. That is, the identifiability results in an IV model with a sparse causal effect, as presented by [176], are highly relevant and transferable to our considered setting. Hence, initial notation is directly borrowed from [176] with further case-relevant modifications introduced. To establish this connection, we define the solution space of Eq. C.1 by

$$\mathcal{B} = \left\{\beta \in \mathbb{R}^d | Cov(\tilde{X}, X)\beta = Cov(\tilde{X}, Y)\right\}. \tag{C.2}$$

Following [176], we will see below that - under mild conditions on the interventions I - the causal coefficient β^* is a unique solution to the optimization problem

$$\min_{\beta \in \mathcal{B}} \|\beta\|_0. \tag{C.3}$$

Consider the matrix C, defined as

$$C \coloneqq A^\top (\mathrm{Id} - B)^{-\top}, \tag{C.4}$$

with dimensions $(m \times d)$. Each entry $C_{i,j}$ represents the total effect of intervention $I = e_i$ on X_j, as defined in the structural causal model described in equation 4.1. The entry $C_{i,j}$ expresses the weighted sum of all paths from the instrument variable i to covariate X_j. The matrix C will play a crucial role in the analysis of identifiability. Now, we introduce the following assumptions:

(A1) It holds that $\mathrm{Rank}(C_{Pa(Y)}) = |Pa(Y)|$.

(A2) For all $S \subseteq \{1, \dots, d\}$, it holds that

$$\left.\begin{array}{ll} \operatorname{Rank}(C_S) \leqq \operatorname{Rank}(C_{Pa(Y)}) & \text{and} \\ \operatorname{Im}(C_S) \neq \operatorname{Im}(C_{Pa(Y)}) & \end{array}\right\} \text{ implies}$$
$$\left\{\forall w \in \mathbb{R}^{|S|}: \quad C_S w \neq C_{Pa(Y)} \beta^*_{Pa(Y)}\right.$$

(A3) For all $S \subseteq \{1, \dots, d\}$ with $|S| = |Pa(Y)|$ and $S \neq Pa(Y)$, we have $\operatorname{Im}(C_S) \neq \operatorname{Im}(C_{Pa(Y)})$.

In this context, we adopt a convention where a matrix $D \in \mathbb{R}^{m \times d}$ and a subset $S \subseteq \{1, \dots, d\}$ imply that the subindexed matrix D_S represents the $m \times |S|$ submatrix of D, containing only the columns indexed by S. The assumption (A1) is essential for identifying the true coefficients β^* under the IV framework since it ensures β^* is correctly identifies if the IV regression is based on the correct parent set $Pa(Y)$. Assumption (A2) pertains to the underlying causal model and prevents certain types of cancellations. It can be considered a mild assumption because if we regard the true causal parameter β^* as randomly drawn from a distribution that is absolutely continuous w.r.t. the Lebesgue measure, it would almost surely result in a system that satisfies (A2), see [176].

The subsequent theorem demonstrates that the assumptions (A1) and (A2) are sufficient to guarantee that β^* satisfies the optimization problem $\min_{\beta \in \mathcal{B}} \|\beta\|_0$. Furthermore, by including assumption (A3), we ensure the uniqueness of the solution. This assumption can be interpreted as necessitating an additional level of heterogeneity in a manner in which interventions influence the system, as described in Theorem 5 in [176].

Theorem C.1.1 *(Identifiability of sparse causal parameters). Consider a data generating process of the form 4.1. If (A1) and (A2) hold, then β^* is a solution to $\min_{\beta \in \mathcal{B}} \|\beta\|_0$. Moreover, if, in addition, (A3) holds, then β^* is the unique solution.*

Proof. We use the notation $\xi^X \coloneqq h(H, \epsilon^X)$, $\tilde{\xi}^X \coloneqq h(\tilde{H}, \tilde{\epsilon}^X)$, and $\xi^Y \coloneqq g(H, \epsilon^Y)$. Then, Eq. 4.1 and the assumption $K \perp\!\!\!\perp \xi^X \perp\!\!\!\perp \tilde{\xi}^X \perp\!\!\!\perp \xi^Y$ imply that

$$\begin{aligned} \operatorname{Cov}[\tilde{X}, X] &= \operatorname{Cov}\left[(\mathrm{Id} - B)^{-1}(AI_K + \tilde{\xi}^X), (\mathrm{Id} - B)^{-1}(AI_K + \xi^X)\right] \\ &= (\mathrm{Id} - B)^{-1} A \operatorname{Cov}(I_K) A^\top (\text{Id-B})^{-\top}. \end{aligned} \tag{C.5}$$

Similarly, we find

$$\begin{aligned} \operatorname{Cov}[\tilde{X}, Y] &= \operatorname{Cov}\left[(\mathrm{Id} - B)^{-1}(AI_K + \tilde{\xi}^X), \left((AI_K + \xi^X)^\top (\mathrm{Id} - B)^{-\top} \beta^* + \xi^Y\right)\right] \\ &= (\mathrm{Id} - B)^{-1} A \operatorname{Cov}(I_K) A^\top (\mathrm{Id} - B)^{-\top} \beta^*. \end{aligned} \tag{C.6}$$

Hence, for any $\tilde{\beta} \in \mathcal{B}$, using the definition of $\mathcal{B}$ and Eq. C.5 and C.6, leads to

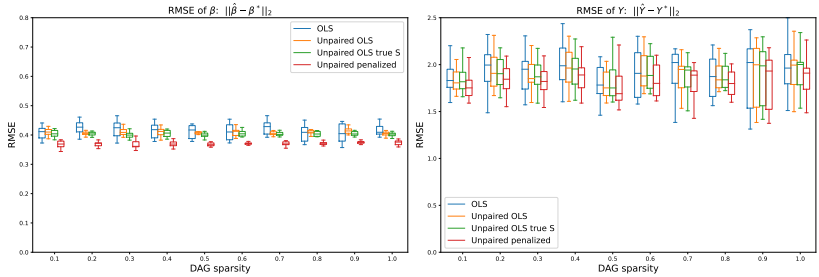
$$A\text{Cov}(I_K)C\tilde{\beta} = A\text{Cov}(I_K)C\beta^*. \tag{C.7}$$

Here, we used the definition of C in Eq. C.4. Finally, since $\text{Rank}(A) = m$ we know that A has a left-inverse, i.e. $A^{-1} = (A^\top A)^{-1}A^\top$ and the fact that $\text{Cov}(I_K)$ is invertible, hence, we can conclude

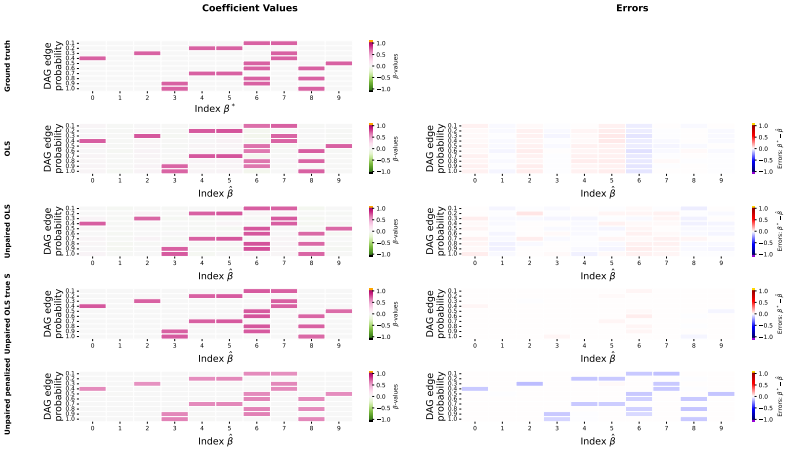
$$C\tilde{\beta} = C\beta^*. \tag{C.8}$$

The remaining of the proof follows exactly the proof of Theorem 3 in [176] and the interested reader is referred to this work and references therein.

C.2 Additional Results

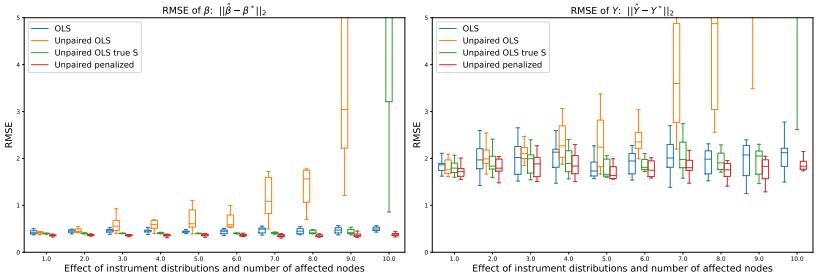


(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$

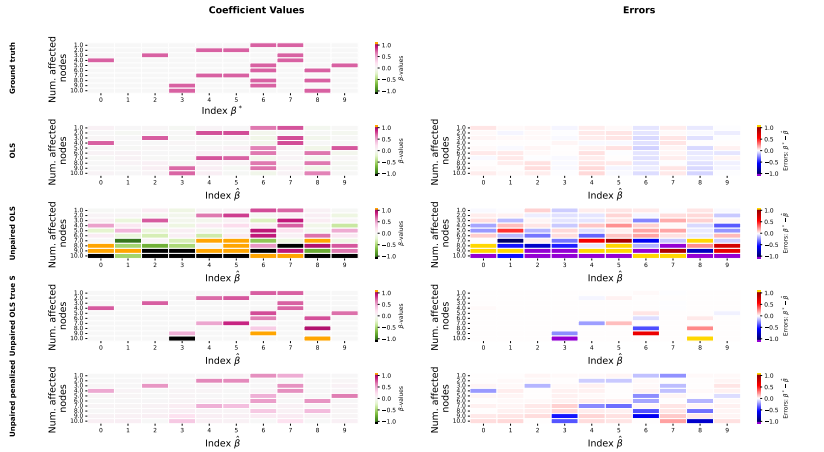


(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure C.1: Experiment (E2): DAG sparsity. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.

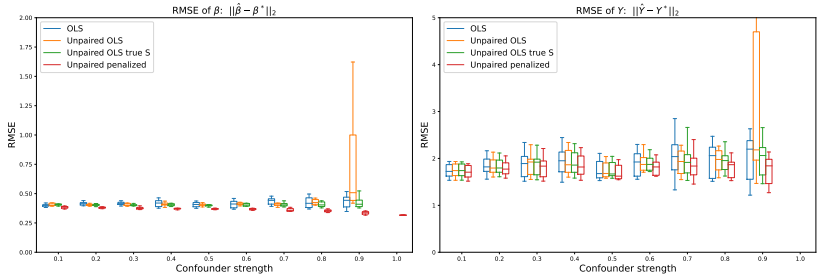


(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$

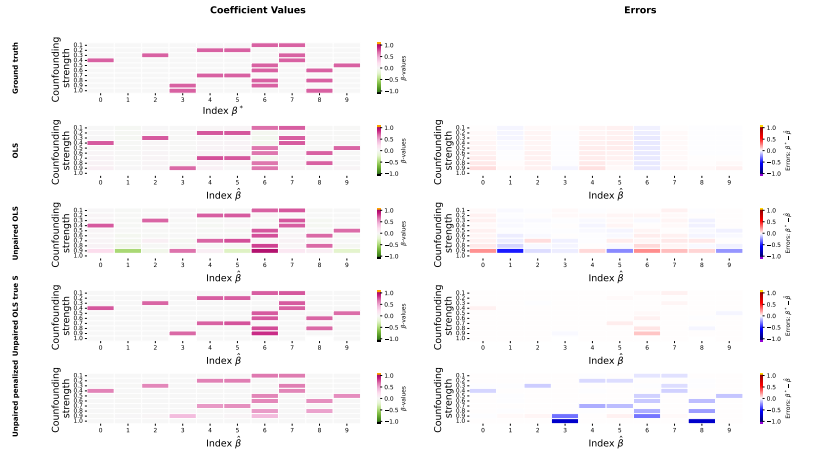


(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure C.2: Experiment (E6): Number of affected nodes and mix of distribution shapes. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.

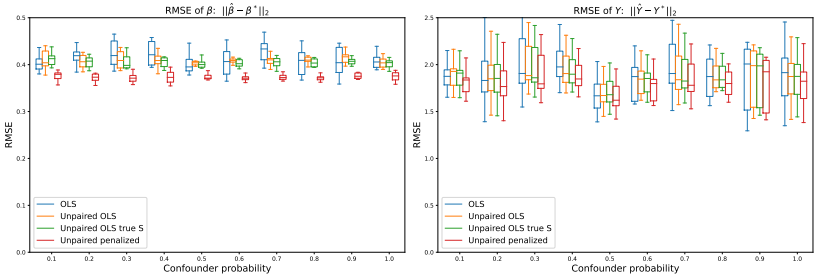


(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$

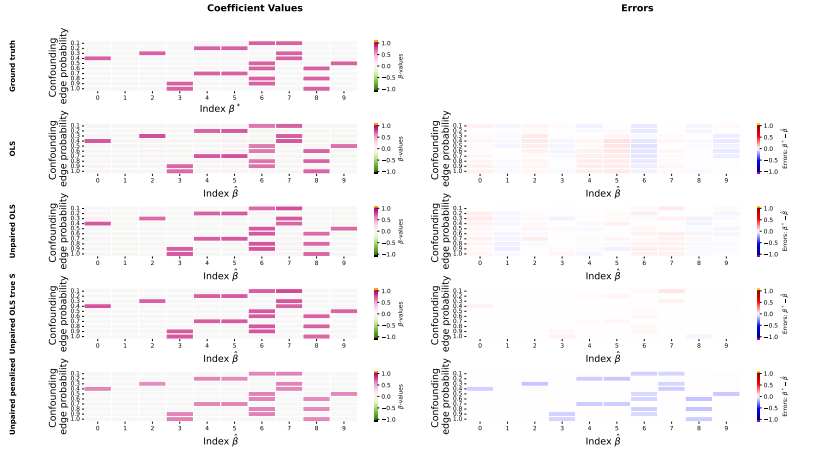


(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure C.3: Experiment (E7): Confounding strength. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.

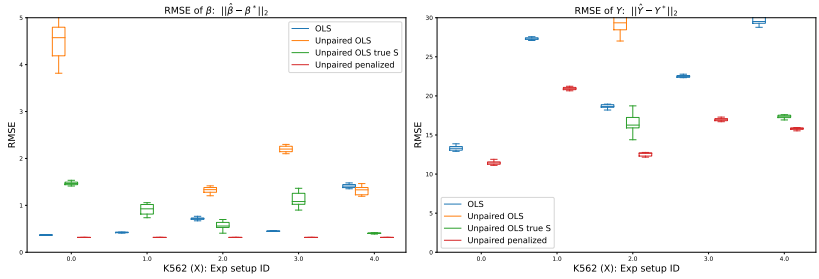


(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$



(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

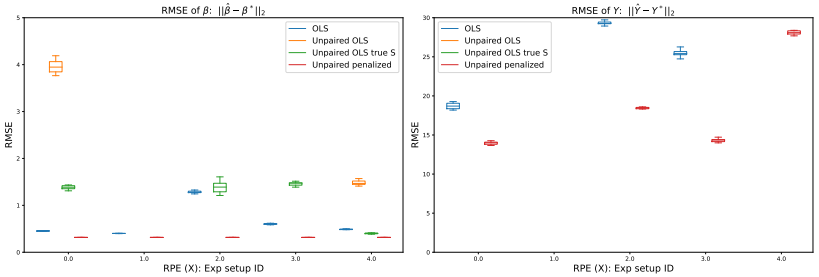
Figure C.4: Experiment (E8): Confounding probability. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.



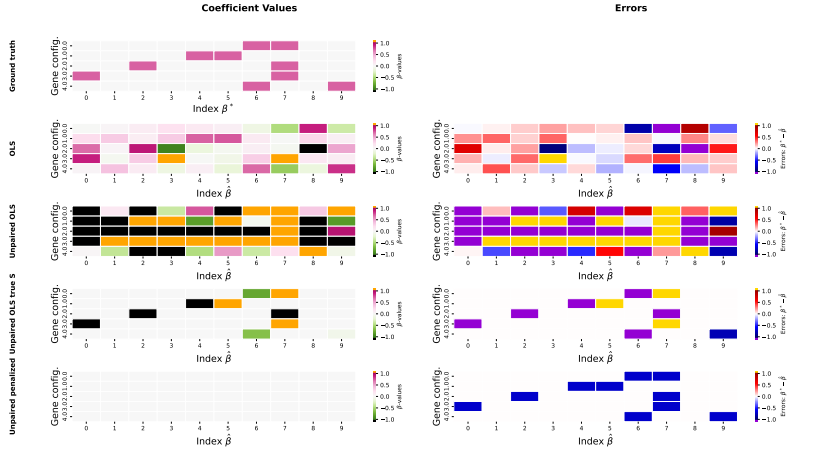
(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$

(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure C.5: K562 gene expression levels as covariates. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.



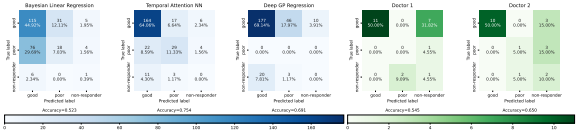
(a) RMSE: left $\|\hat{\beta} - \beta^*\|_2$, right $\|\hat{Y} - Y^*\|_2$



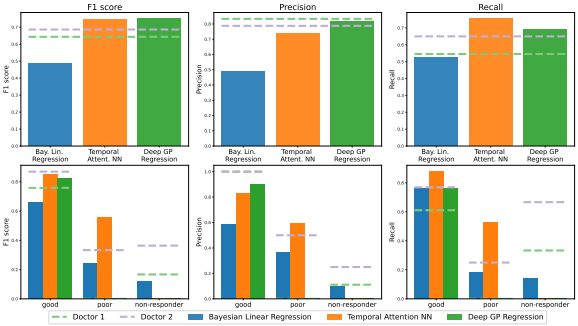
(b) Predictions of coefficients: left: Heatmaps of predicted coefficients $\hat{\beta}$, right: error between ground truth coefficients and predictions

Figure C.6: RPE gene expression levels as covariates. (a): Boxplot of RMSE of predicted coefficients $\hat{\beta}$ (left) and predicted outcomes Y (right) for OLS (blue), Unpaired OLS (orange), Unpaired OLS using solely non-coefficients (green) and our proposed penalized covariance adjusted estimator, Unpaired penalized, (red). (b): Heatmap of ground truth and predicted β-coefficients (left) , error between predictions and ground truth values (right), lighter colors indicate smaller errors. Predicted β-values falling outside the range of $[-1, 1]$ are represented in black and orange, while errors exceeding $[-1, 1]$ are emphasized using purple and gold.

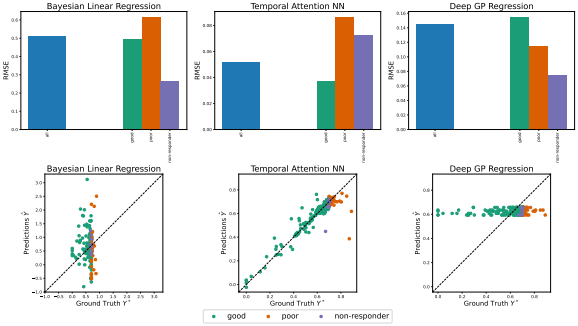
D Supplementary Information: Estimating Treatment Effects using Deep Neural Networks



(a) Confusion Matrix: Initial data: $n_{visits} = 2$

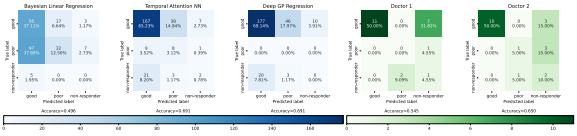


(b) Classification Performance: Initial data: $n_{visits} = 2$

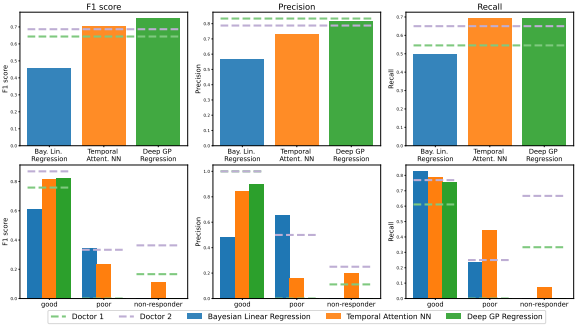


(c) Regression Performance: Initial data: $n_{visits} = 2$

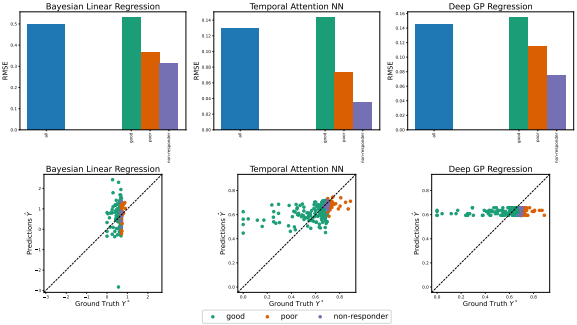
Figure D.1: Entire Population Validation Data: Results of the initial dataset with $n_{visits} = 2$: (a) Confusion matrix: left algorithms (blue colorbar) and right human ophthalmologists (green colorbar) (b) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (c) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.



(a) Confusion Matrix: LMM preprocessed data: $n_{visits} = 2$

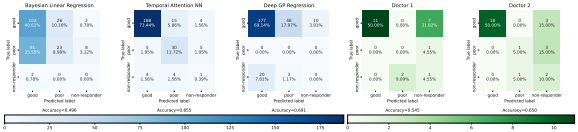


(b) Classification Performance: LMM preprocessed data: $n_{visits} = 2$

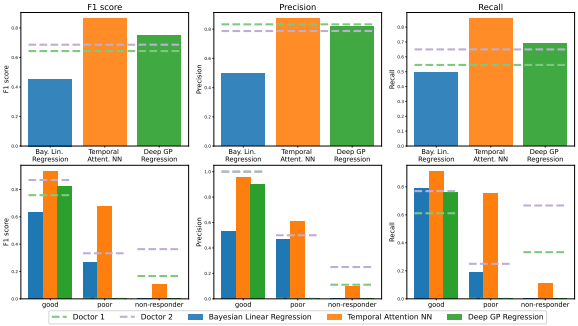


(c) Regression Performance: LMM preprocessed data: $n_{visits} = 2$

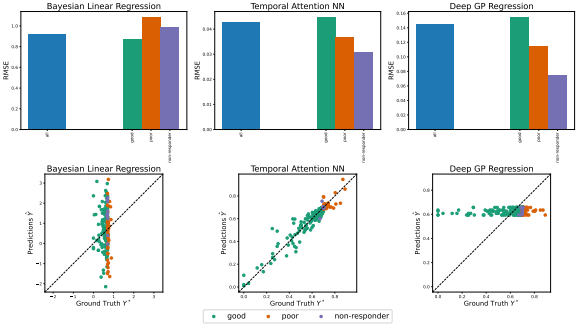
Figure D.2: Entire Population Validation Data: Results of the LMM preprocessed dataset with $n_{visits} = 2$: (a) Confusion matrix: left algorithms (blue colorbar) and right human ophthalmologists (green colorbar) (b) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (c) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.



(a) Confusion Matrix: GP preprocessed data: $n_{visits} = 2$

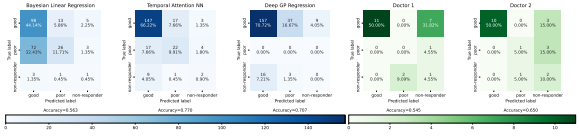


(b) Classification Performance: GP preprocessed data: $n_{visits} = 2$

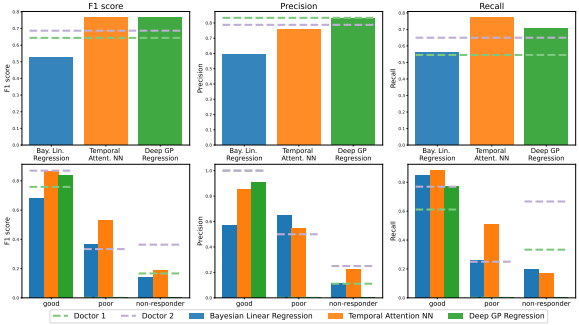


(c) Regression Performance: GP preprocessed data: $n_{visits} = 2$

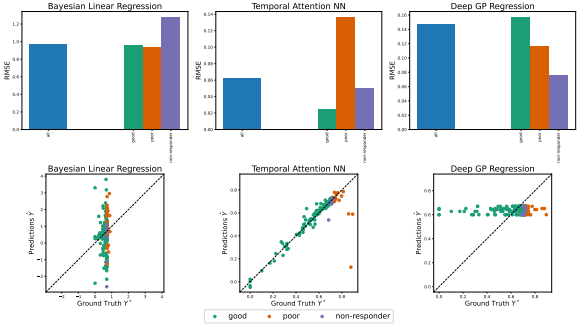
Figure D.3: Entire Population Validation Data: Results of the GP preprocessed dataset with $n_{visits} = 2$: (a) Confusion matrix: left algorithms (blue colorbar) and right human ophthalmologists (green colorbar) (b) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (c) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.



(a) Confusion Matrix: Initial data: $n_{visits} = 3$

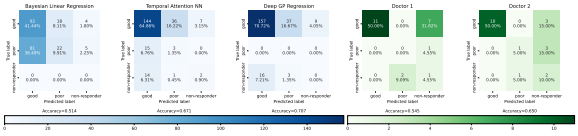


(b) Classification Performance: Initial data: $n_{visits} = 3$

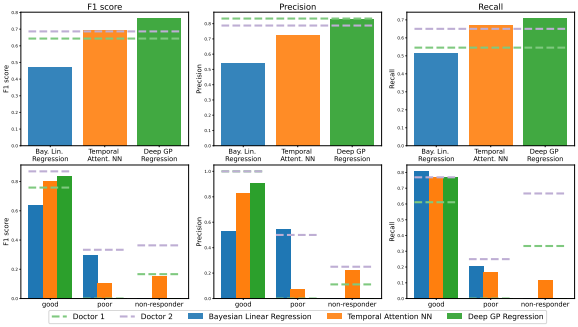


(c) Regression Performance: Initial data: $n_{visits} = 3$

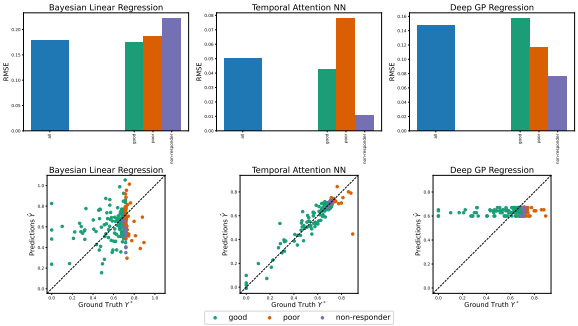
Figure D.4: Entire Population Validation Data: Results of the initial dataset with $n_{visits} = 3$: (a) Confusion matrix: left algorithms (blue colorbar) and right human ophthalmologists (green colorbar) (b) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (c) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.



(a) Confusion Matrix: LMM preprocessed data: $n_{visits} = 3$

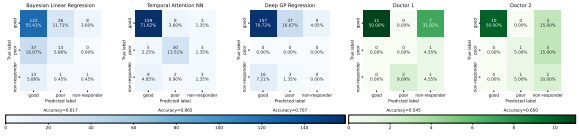


(b) Classification Performance: LMM preprocessed data: $n_{visits} = 3$

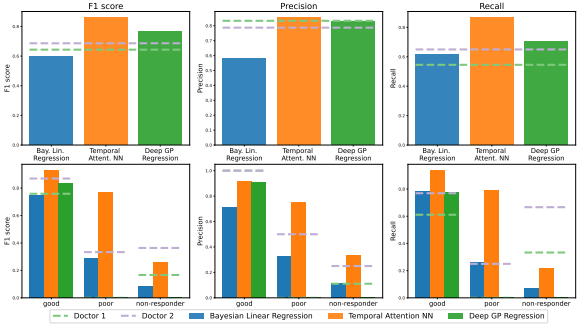


(c) Regression Performance: LMM preprocessed data: $n_{visits} = 3$

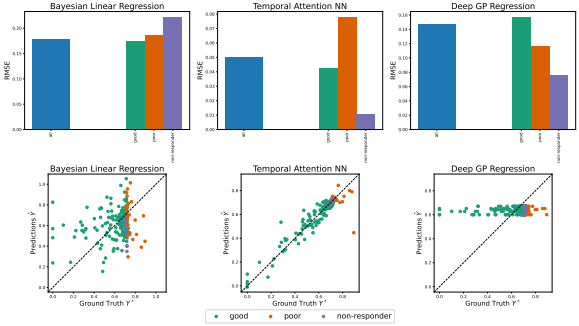
Figure D.5: Entire Population Validation Data: Results of the LMM preprocessed dataset with $n_{visits} = 3$: (a) Confusion matrix: left algorithms (blue colorbar) and right human ophthalmologists (green colorbar) (b) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (c) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.



(a) Confusion Matrix: GP preprocessed data: $n_{visits} = 3$



(b) Classification Performance: GP preprocessed data: $n_{visits} = 3$



(c) Regression Performance: GP preprocessed data: $n_{visits} = 3$

Figure D.6: Entire Population Validation Data: Results of the GP preprocessed dataset with $n_{visits} = 3$: (a) Confusion matrix: left algorithms (blue colorbar) and right human ophthalmologists (green colorbar) (b) Classification performance of the Bayesian linear regression (blue), the temporal attention neural network (orange) and the Deep GP (green) are shown. The top row depicts F1 scores, precision and recall values for all patients while the bottom row presents the same metrics per subpopulation. (c) Regression performance for the same models are shown. Top row: The RMSE value of the entire test data is given on the left while class-wise RMSE values are shown on the right of each subplot. Bottom row: Scatter plot presenting predictions against ground truth.