The following publication X. Zhang, T. Wang, X. -L. Wang, F. -L. Fan, Y. -M. Cheung and I. Bose, "Causality-Informed Neural Networks for Regularized Learning in Regression Problems," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 56, no. 3, pp. 1895-1910, March 2026 is available at https://doi.org/10.1109/TSMC.2025.3646993.

Causality-Informed Neural Networks for Regularized Learning in Regression Problems

Xiaoge Zhang, *Senior Member, IEEE,* Tao Wang, Xiao-Lin Wang, Feng-Lei Fan, *Senior Member, IEEE,* Yiu-Ming Cheung, *Fellow, IEEE,* and Indranil Bose

***Abstract*—Neural networks that overlook the underlying causal relationships among observed variables pose significant risks in high-stake decision-making contexts due to the concerns about the robustness and stability of model performance. To tackle this issue, we present a general approach for embedding hierarchical causal structure among observed variables into neural network to inform its learning. The proposed methodology, termed causality-informed neural network (CINN), exploits hierarchical causal structure learned from observational data as a structurally informed prior to guide the layer-to-layer architectural design of the neural network while maintaining the orientation of causal relationships in the discovered causal graph. The proposed method involves three steps. First, CINN mines causal relationships from observational data via directed acyclic graph (DAG) learning, where causal discovery is recast as a continuous optimization problem to circumvent the combinatorial nature of DAG learning. Second, we encode the discovered hierarchical causal graph among observed variables into neural network via a dedicated architecture and loss function. By classifying observed variables in the DAG as root, intermediate, and leaf nodes, we translate the hierarchical causal DAG into CINN by creating a one-to-one correspondence between DAG nodes and certain CINN neurons. For the loss function, both intermediate and leaf nodes in the DAG are treated as target outputs during CINN training, facilitating the co-learning of causal relationships among the observed variables. Finally, as multiple loss components emerge in CINN, we leverage the projection of conflicting gradients to mitigate gradient interference among the multiple learning tasks. Computational studies indicate that CINN outperforms several state-of-the-art methods across a broad range of datasets. In addition, an ablation study that incrementally incorporates structural and quantitative causal knowledge into the neural network is conducted to highlight the pivotal role of causal knowledge in enhancing neural network's prediction performance.**

***Index Terms*—Deep learning, Causal inference, Causality-informed neural network, Informed learning**

I. Introduction

Deep learning is recognized as a pivotal technology propelling the fourth industrial revolution due to its powerful ability in learning useful representations from massive data [1, 2]. Unlike traditional machine learning methods such as support vector machines and decision trees, neural networks can automatically extract representations from raw data (e.g., images, voices, videos, texts) for detection or classification tasks, thereby supporting various decision-making activities. The end-to-end representation learning paradigm enabled by deep learning has resulted in numerous revolutionary innovations surpassing human-level performance by a large margin across a broad range of tasks, such as medical image analysis [3], object classification [4], fault diagnosis [5, 6], among others.

Despite its impressive performance, deep learning is often criticized for several significant flaws, including learning spurious correlations, vulnerability to small perturbations, and poor generalization [7–9]. These deficiencies ultimately manifest as reliability, safety, and accountability-related issues on the surface when deep learning is deployed in practice. Hence, the absence of a rigorous framework for safety assurance and risk management in deep learning models renders a low rate of translating these models into practical solutions in high-stake decision-making contexts. Take healthcare as an example. Only a few AI-based solutions have been approved by pertinent regulatory agencies for use without human oversight, and these approved applications are primarily limited to low-risk settings [10–12].

Though tremendous progress has been made to address some of these issues, existing attempts to improve the robustness and generalizability of deep learning models remain largely ad-hoc and unprincipled. For example, post-hoc explanation methods, such as salience map, have been developed to uncover the reasoning behind model predictions through local interpretations and sensitivity analysis [13–15]. Some methods emphasize safeguarding AI algorithms against a certain type of adversarial attacks (e.g., data poisoning), resulting in an endless race between "defense mechanisms" and new attacks that appear shortly after [16]. However, most theoretical advancements fail to address a fundamental problem in deep learning: how to induce neural network to learn meaningful causal relationships, rather than merely extracting correlation- or association-based patterns from raw data, as is commonly observed in the current deep learning algorithms. Since most deep learning models overlook the cause-and-effect relationships inherent in observational data, it is challenging to achieve a robust and stable prediction performance. Importantly, deep neural networks tend to learn spurious associations from raw data, and these learned spurious correlations negatively impact the model's ability to generalize [17–19]. For example, Ribeiro et al. [20] demonstrated that a deep neural network classifier trained on a meticulously designed dataset extracted spurious correlations between the presence of snow and the classification of wolves versus Eskimo dogs (huskies) rather than identifying meaningful features that distinguished wolf and husky based on their inherent differences. Poor generalization poses a significant challenge when deploying deep learning in high-stake decision-making settings, where input data are often collected under dynamic environments and might not adhere to the independent and identically distributed (i.i.d.) assumption that underpins deep learning.

Xiaoge Zhang and Tao Wang are with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China.

Xiao-Lin Wang is with the Business School, Sichuan University, Chengdu 610065, China.

Feng-Lei Fan is with the Frontier of Artificial Networks (FAN) Lab, Department of Data Science, City University of Hong Kong, Hong Kong, China.

Yiu-Ming Cheung is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, China.

Indranil Bose is with the NEOMA Business School, 59 rue Pierre Taittinger, Reims 51100, France.

Correspondence to: Room EF622, Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, 11 Yuk Choi Rd, Hung Hom, Hong Kong. Email: xiaoge.zhang@polyu.edu.hk.

To tackle these issues, causal inference offers a nuanced and formal means to describe how humans comprehend the world by enabling a rigorous and sound mechanism to link causes and effects in practice [21–24]. Take computer vision as an example. Unlike deep learning, humans can easily recognize an object in a scene while being less interfered by the variations in other aspects, such as background and viewing angles. If neural network can be induced to concentrate on learning with the causal predictors in distinguishing objects, then the variations in the background and viewing angle will have a trivial effect on the object recognition. The paradigm shift from correlation-based learning to causation-based learning will enable neural networks to achieve a stable and reliable prediction performance. In addition, reasoning with causal drivers could extend neural network's capability beyond mere prediction by enabling "what-if" analysis and interventional queries [25, 26]; such a capability is desirable across a broad range of industrial sectors. For example, in the telecommunication industry, decision-makers are interested not only in identifying users who might terminate the service, but also in determining what actions to take (e.g., offering price discounts or improving network quality) to prevent customer churn.

In fact, the ability to perform causal reasoning highlights the fundamental difference between human and machine intelligence, and it has been recognized as a hallmark of human intelligence [27]. Such capability aids in identifying factors that are closely relevant to the quantity of interest. Given the significant potential of causal knowledge to address the inherent shortcomings of conventional neural networks, existing studies have explored incorporating causal knowledge in several ways to regularize the learning of neural networks. To this end, Kancheti et al. [28] proposed regularizing neural network training by aligning the learned causal effects—including both direct and total effects—with established domain priors. Kyono et al. [29] employed causal graph learning as an auxiliary task for encouraging models to concentrate on learning and making predictions based on stable causal parents. Regularization of predictive models to satisfy causal relationships among input features and target variable guides the model to have better generalization performance. Teshima and Sugiyama [30] augmented training data with synthetic data samples encoding conditional independence relations among observed variables to guide the learning of neural networks. For a detailed review on the incorporation of causal reasoning for neural network learning, refer to the Appendix "Causal Knowledge Integration".

Despite the progress, existing studies still fall short in several aspects. First, as causal knowledge often manifests in various forms, such as *structural forms* (e.g., directed acyclic graphs or DAGs) and *relational forms* (e.g., qualitative or quantitative relationships), most existing studies focus on incorporating a particular form of causal knowledge. The literature still lacks a unified interface for integrating these diverse forms of causal knowledge into neural networks to effectively inform their learning. Second, existing studies tend to inject causal knowledge into neural networks in an opaque manner typically by penalizing the violation of causal relationships in the loss function or by using synthetic samples encoding conditional independence. These black-box approaches for incorporating causal knowledge make it challenging to preserve the causal direction (e.g., effect from cause) among observed variables, even though prediction in the causal direction is known to yield lower testing error than the anti-causal direction [31–34]. Third, most studies leverage causal knowledge merely as an inductive bias to optimize model parameters for a single prediction task while the hierarchical causal relationships among the observed variables are not fully considered in the optimization problem.

Motivated by these observations, we develop a generic Causality-Informed Neural Network (CINN) to offer a flexible interface for integrating both causal structure and causal relationships in qualitative and quantitative forms into neural network simultaneously. These forms of causal knowledge can be either discovered from observational data or elicited from domain experts. Generally, causal knowledge can be incorporated into neural network through two means: (1) embedding the causal structure among observed variables into neural networks; (2) imposing relational (quantitative, qualitative) causal relationships between certain variables as constraints within neural networks. In this paper, we exploit both strategies for devising CINN. Unlike existing studies, the proposed CINN strictly adheres to the orientation of discovered causal relationships among observed variables and provides a more flexible interface for incorporating causal knowledge in both quantitative and qualitative forms. The proposed methodology consists of three coherent steps. First, we formulate causal discovery as a continuous optimization problem in the form of a Directed Acyclic Graph (DAG) using the approach developed by Zheng et al. [35, 36]. This allows us to derive a set of cause-and-effect relationships from observational data. Domain experts then review the discovered causal mechanisms to refine the causal DAG by removing invalid causal links and adding substantiated causal edges. In the second step, we develop a generic methodology to fully integrate the hierarchical causal structure among observed variables into the design of the neural network architecture and loss function. The proposed CINN not only incorporates the causal DAG in a hierarchical form, but also accommodates (partial) domain knowledge represented by quantitative and qualitative relationships among observed variables. In the third step, to address the presence of multiple loss terms in the developed CINN, we adopt the projection of conflicting gradients (PCGrad) to mitigate gradient interference during back-propagation in the optimization of neural network parameters. Finally, we conduct an ablation study by incrementally incorporating structural and relational causal knowledge into neural network to demonstrate the role of causal knowledge in enhancing neural network's prediction performance.

Rather than limiting the application to a specific domain, this study concentrates on developing a *generic* methodology for integrating causal knowledge and validating its performance across multiple benchmark datasets for regression problems. In summary, the contributions of this article are summarized as follows.

1) We develop a generic approach to explicitly map the hierarchical causal structure among observed variables onto neural networks by creating a one-to-one correspondence between nodes in the causal DAG and neurons in the neural network. In doing so, the causal DAG serves as a structurally informative prior to the architectural design of neural network.
2) We design a specialized loss function aimed at minimizing the total loss across nodes in both the intermediate and output layers, driving co-learning of causal relationships among different groups of nodes. Such a loss function design represents a significant departure from existing learning paradigms. Computational comparisons with various baseline models demonstrate that the architecture and loss function design provides a more effective method for encoding causal knowledge from observational data into neural networks.

3) The developed CINN offers an elegant interface for integrating domain expertise in two important ways: human-guided causal discovery and incorporation of domain priors on stable causal relationships. By integrating human knowledge, CINN addresses the limitations of purely algorithmic causal discovery and enables multiple stakeholders (e.g., data scientists, regulators, executives) to align on the fundamental mechanics of the neural network's decision-making process.
4) Comprehensive experiments are conducted to assess the impact of multiple hyperparameters, such as learning rate, seed value, and weighting factor, on the performance of CINN. Thorough computational comparisons on five benchmark datasets demonstrate that CINN significantly outperforms other models. Furthermore, an ablation study underscores the importance of causal knowledge, including causal DAGs and relationships, in enhancing CINN's performance. The causal graph, refined by domain knowledge, along with the integration of quantitative causal relationships, plays a pivotal role in enhancing CINN's performance.

The rest of this paper is structured as follows. In Section II, we introduce the proposed CINN framework to enforce causal structure and causal relationships into neural network. In Section III, we demonstrate the procedures of the proposed methodology, compare its performance against several state-of-the-art methods, and summarize the computational findings. In Section IV, we conclude this paper and discuss future research directions.

II. Methodology

In this section, we describe the proposed methodology for CINN in detail. As shown in Fig. 1, our approach consists of three coherent steps. In the first step, causal relationships among a predefined set of variables are discovered from observational data via DAG learning. The discovered causal relationships are examined by domain experts to eliminate any invalid and groundless cause-and-effect relationships and add substantiated causal links simultaneously. Next, a generic framework is developed to map the hierarchical causal structure among observed variables into the architecture and loss function design of neural network. By aligning the architecture of neural network with the causal structure among observed variables, the developed CINN is capable of accommodating quantitative or qualitative causal relationships elicited from domain experts while strictly adhering to the orientation of each discovered causal relationship. In the final step, since multiple loss components emerge in CINN, we consider loss-specific gradient features and adopt PCGrad to mitigate gradient interference towards the optimization of neural network parameters.

A. Algorithmic Causal Discovery and Refinement

Consider $\mathsf{X}_1, \mathsf{X}_2, \cdots, \mathsf{X}_d$ as a set of input features and Y as the target variable in a regression problem. In the context of causal structure learning, we observe N pairs of data instances (x_i, y_i), $i \in \{1, 2, \cdots, N\}$, where each $x_i = \left(x_i^1, x_i^2, \cdots, x_i^d\right)$ is drawn from the joint probability distribution $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \cdots \times \mathcal{X}_d$ and y_i is drawn from the probability distribution $\mathcal{Y}$. Specifically, $x_i^j \sim \mathcal{X}_j$ for all i and j, and $y_i \sim \mathcal{Y}$, where x_i^j denotes the j-th feature of the i-th instance. Here, $\mathcal{X}_j$ $(j \in \{1, 2, \cdots, d\})$ and $\mathcal{Y}$ represent the underlying distributions of the j-th input feature and the target variable, respectively.

When modeling causal-and-effect relationships among uncertain variables, we represent the causal structure over the input features

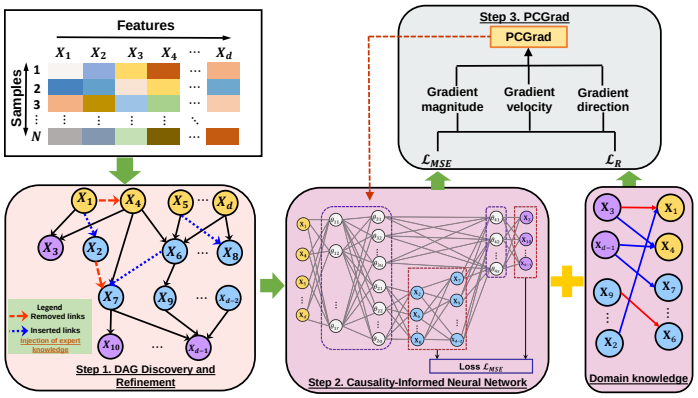


Fig. 1. Flowchart of the developed methodology

$\mathsf{X}_1, \mathsf{X}_2, \cdots, \mathsf{X}_d$ and the target variable Y as a DAG [37]. Given the observational data, the causal structure learning problem is formulated as finding a DAG $\mathsf{G} \in \mathbb{D}$ made of vertices $\mathsf{V} = \{\mathsf{Y}, \mathsf{X}_1, \mathsf{X}_2, \cdots, \mathsf{X}_d\}$ and edges $\mathsf{E} \subseteq \mathsf{V} \times \mathsf{V}$ so as to achieve a reasonable description on the joint distribution $P(\mathcal{X}, \mathcal{Y})$. Suppose the causal relationships among observed variables are linear, the specific relationships among the set of vertices V can be described using structural equation models, which are defined by a weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{(d+1)\times(d+1)}$ with zeros on the diagonal.

Let $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_d]$ denote the $N \times d$ input data matrix, $\mathbf{Y}$ represent the N-dimensional target feature, and $\overline{\mathbf{X}} = [\mathbf{Y}, \mathbf{X}]$ be the $N \times (d+1)$ matrix containing data of all variables in the DAG. In causal discovery, statistical properties of the least-squares loss in scoring the DAG have been extensively investigated in the literature [38, 39]: the minimizer of the least-squares loss $\frac{1}{N}\|\mathbf{Y} - \overline{\mathbf{X}}\mathbf{W}_{:,1}\|^2$ provably recovers a true DAG with high probability on finite samples in high dimensions. Thus, we seek to find a sparse weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{(d+1)\times(d+1)}$ by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{W}\in\mathbb{R}^{(d+1)\times(d+1)}} & = \tfrac{1}{N}\left\|\mathbf{Y} - \overline{\mathbf{X}}\mathbf{W}_{:,1}\right\|^2 + \lambda\|\mathbf{W}\|_1, \\ \text{s.t.} \quad & \mathsf{H}(\mathbf{W}) \in \mathbb{D}, \end{aligned} \tag{1}$$

where $\mathbf{W}_{:,1}$ denotes the first column of $\mathbf{W}$, $\|\mathbf{W}\|_1$ denotes the L_1 norm of $\mathbf{W}$. Essentially, the square matrix $\mathbf{W}$ captures the causal relationships among the $d+1$ variables by learning from the data represented in the expanded matrix $\overline{\mathbf{X}}$. Elements of $\mathbf{W}$ can be negative or positive reflecting that one variable may exert a positive or negative causal influence on another. $\mathsf{H}(\mathbf{W})$ defines the adjacency matrix of graph G (i.e., $\mathsf{H}(\mathbf{W}) \Leftrightarrow \mathbf{W}_{ij} \neq 0$ and zero otherwise) and λ is a factor to weight the L_1 norm of $\mathbf{W}$.

Unfortunately, the DAG constraint $\mathsf{H}(\mathbf{W}) \in \mathbb{D}$ is inherently discrete, which poses a significant challenge for optimization. To make this constraint more tractable, the combinatorial acyclicity constraint is replaced with a continuous and smooth equality constraint $\mathsf{R}(\mathbf{W}) = 0$, where $\mathsf{R}(\mathbf{W})$ measures the DAG-ness of the adjacency matrix $\mathbf{W}$. In particular, $\mathsf{R}(\mathbf{W}) = 0$ if and only if $\mathbf{W}$ is acyclic (i.e., $\mathsf{H}(\mathbf{W}) \in \mathbb{D}$). By doing so, the discrete DAG learning problem is recast as a continuous optimization problem formulated below [35]:

$$\min_{\mathbf{W}\in\mathbb{R}^{(d+1)\times(d+1)}} = \frac{1}{N}\left\|\mathbf{Y} - \overline{\mathbf{X}}\mathbf{W}_{:,1}\right\|^2 + \mathsf{R}(\mathbf{W}) + \lambda\|\mathbf{W}\|_1, \tag{2}$$

where $\mathsf{R}(\mathbf{W}) = (\mathrm{Tr}(e^{\mathbf{W}\odot\mathbf{W}} - d - 1))^2$, $\odot$ is the Hadamard product,

and e^A is the matrix exponential of A.

By setting up a threshold τ, we could identify the causal relationship between any two nodes as follows. If $|\mathbf{W}_{ij}| \geq \tau$, then node V_i causally affects node V_j; otherwise, no causal relationship exists between them. In doing so, $\mathbf{W}$ defines a weighted adjacency matrix with a capacity to represent a broad spectrum of causal DAGs. By formulating causal DAG learning as a continuous optimization problem over the real matrix $\mathbf{W}$, standard numerical algorithms, such as Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm [40], can be leveraged to solve the non-convex optimization problem in Eq. (2). Given a specified threshold τ, we denote the discovered causal DAG as $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, where $\forall (i,j) \in \mathsf{E}, |\mathbf{W}_{ij}| \geq \tau$.

Note that this step primarily serves to probe potential causal relationships among observed variables and derive an approximate structural representation that is most likely to generate the observed data. The discovered causal relationship between some variables might be invalid, as some of them might violate our common sense. To remedy such shortcoming, domain expert knowledge can be exploited to further refine the discovered graph by removing non-causal or implausible links and adding substantiated edges between relevant variables. The resulting refined causal graph is expected to more accurately reflect practical scenarios.

B. Causality-Informed Neural Network

1) DAG Node Classification: Established upon the continuous formulation of DAG learning, recent years have witnessed an increasing interest in leveraging the discovered causal DAG as an informative regularizer to enhance the generalizability of neural networks. However, a noticeable flaw along the existing development of causality-aware neural networks is that they do not strictly respect the orientation of cause-and-effect relationships among observed variables, which can be either given up-front by experts or discovered from observational data. It is well known that causal relationship among variables oftentimes comes with a fixed orientation, and the direction of any causal relationship, if reversed, becomes meaningless. Unfortunately, the current paradigm of regularizing neural networks through causal graph discovery ignores the essential orientation issue. For example, Kyono et al. [29] proposed to regularize a neural network by masking one input feature and reconstructing it from the remaining features, where the effects as a result of the masked feature might be mistakenly used as the inputs of neural networks towards inferring the masked feature. Take the toy causal graph shown in Fig. 2 as an example. In the case that X_2 is masked out, the remaining features that include the effects of variable X_2 (e.g., X_5, X_6, Y) are used to reconstruct the masked feature X_2. Obviously, inferring X_2 from X_5, X_6, Y violates the orientation of the causal relationship between X_2 and X_5, X_6, Y.

To tackle this issue, we develop a generic CINN framework to fully respect the direction of the learned causal relationships. Unlike existing paradigms, CINN offers a rigorous and principled means to incorporate the hierarchical causal structure discovered from observational data into neural networks. To elaborate on the proposed methodology, we first categorize the nodes in the causal DAG G, discovered in the previous step, into four groups:

1) **Isolated nodes** V_S: nodes without any inbound and outbound edges, such as X_{11} and X_{12} in Fig. 2.
2) **Root nodes** V_C: nodes with only outgoing edges, such as X_1, X_2, and X_3 in Fig. 2.
3) **Intermediate nodes** V_B: nodes with both incoming and outgoing edges, such as X_4, X_5, X_6, X_8, and Y in Fig. 2.

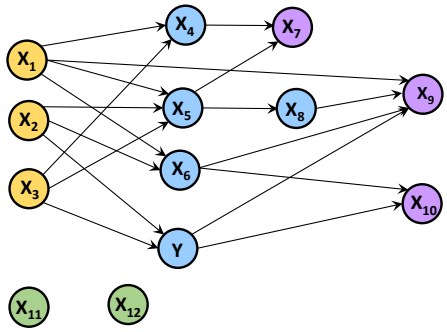


Fig. 2. Demonstration of the orientation of causal relationships and node categorization. Nodes in identical color belong to the same group. Specifically, nodes in green are categorized as isolated nodes, nodes in yellow are root nodes, nodes in blue are intermediate nodes, and nodes in purple indicate leaf nodes.

4) **Leaf nodes** V_O: nodes with only incoming edges but no successors, such as X_7, X_9, and X_{10} in Fig. 2.

If there are multiple layers of intermediate nodes with causal relationships among themselves, their hierarchical structure can be determined by iteratively removing the current root nodes from the graph and re-identifying the sets of root, intermediate, and leaf nodes in the updated graph at each step. This operation continues until no intermediate node remain in the graph. Algorithm 1 outlines the basic steps for identifying the type and hierarchical position of each node in the discovered causal graph G.

Algorithm 1: Type and position of nodes in the causal DAG

Input: Discovered causal DAG $\mathsf{G} = (\mathsf{V}, \mathsf{E})$
Output: Type and hierarchical position of each node in the causal graph G: $\{\mathsf{K}, \mathsf{V}_C, \mathsf{V}_B, \mathsf{V}_O, \mathsf{V}_S\}$

$i \leftarrow 1$;
Group nodes in the graph G into four categories: isolated nodes V_S, root nodes V_C, intermediate nodes V_B, and leaf nodes V_O;
$\mathsf{I} \leftarrow \mathsf{V}_S$, $\mathsf{T} \leftarrow \mathsf{V}_C$
$\mathsf{Z} \leftarrow \emptyset$
while $|\mathsf{Z}| < |\mathsf{V}_B|$ **do**
  $\mathsf{V} \leftarrow \mathsf{V} \setminus (\mathsf{I} \cup \mathsf{T})$; ▷ Remove root and isolated nodes from the graph
  $\mathsf{E} \leftarrow \{(u,v) \in \mathsf{E} \mid u \in \mathsf{V},\ v \in \mathsf{V}\}$; ▷ Remove edges associated with root nodes from the graph
  $\mathsf{G} \leftarrow (\mathsf{V}, \mathsf{E})$; ▷ Update the graph
  $\mathsf{T} \leftarrow$ root nodes of G ; ▷ Identify root nodes T in the updated graph G
  $\mathsf{I} \leftarrow$ isolated nodes of G ; ▷ Identify isolated nodes I in the updated graph G
  $\mathsf{K}[i] \leftarrow \mathsf{T}$; ▷ Save the set of nodes in the i-th intermediate layer
  $\mathsf{Z} \leftarrow \mathsf{Z} \cup \mathsf{T}$; ▷ Track the set of nodes with intermediate layer number determined so far
  $i \leftarrow i + 1$;
return $\{\mathsf{K}, \mathsf{V}_C, \mathsf{V}_B, \mathsf{V}_O, \mathsf{V}_S\}$

For instance, the set of intermediate nodes for the graph shown in Fig. 2 is $\mathsf{V}_B = \{\mathsf{X}_4, \mathsf{X}_5, \mathsf{X}_6, \mathsf{Y}, \mathsf{X}_8\}$. To partition V_B into multiple

layers of intermediate nodes, we perform the following operations iteratively:

1) **Iteration 1**: The set of root nodes $\{\mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3\}$ and their associated links (e.g., $\mathsf{X}_1 \rightarrow \mathsf{X}_4, \mathsf{X}_2 \rightarrow \mathsf{X}_6, \mathsf{X}_3 \rightarrow \mathsf{X}_5$, etc.) are removed from the graph. After that, the set of nodes $\{\mathsf{X}_4, \mathsf{X}_5, \mathsf{X}_6, \mathsf{Y}\}$ become the current root nodes. Thus, we have the first layer of intermediate node as the intersection of current root nodes and V_B; that is, $\mathsf{V}_B^1 = \{\mathsf{X}_4, \mathsf{X}_5, \mathsf{X}_6, \mathsf{Y}\}$.
2) **Iteration 2**: Next, we remove the current root nodes $\{\mathsf{X}_4, \mathsf{X}_5, \mathsf{X}_6, \mathsf{Y}\}$ and their associated links so that $\{\mathsf{X}_7, \mathsf{X}_8, \mathsf{X}_9, \mathsf{X}_{10}\}$ become the current root nodes. Thus, the second layer of intermediate nodes is the intersection of $\{\mathsf{X}_7, \mathsf{X}_8, \mathsf{X}_9, \mathsf{X}_{10}\}$ and V_B; that is, $\mathsf{V}_B^2 = \{\mathsf{X}_8\}$.

The procedure continues until the number of intermediate nodes summed over all the intermediate layers V_B^1 and V_B^2 equals to the number of intermediate nodes V_B in the causal graph. In doing so, the program produces a hierarchical structure to define the type and position of each node in the DAG G.

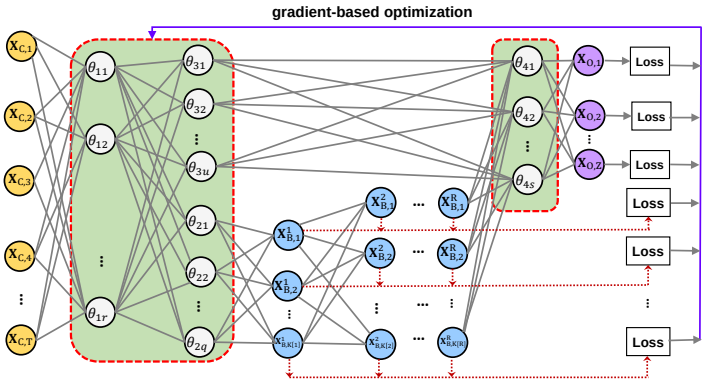


Fig. 3. Proposed CINN architecture. Nodes in yellow, blue, and purple represent the sets of root nodes V_C, intermediate nodes V_B, and output nodes V_O, respectively. Note that V_B might have multiple layers of nodes stacked together with each layer having its own set of features.

2) CINN Development: Suppose the sets of isolated nodes, root nodes, intermediate nodes, and leaf nodes in G are denoted by $\mathsf{V}_S = \{\mathsf{X}_{S,1}, \mathsf{X}_{S,2}, \cdots \mathsf{X}_{S,M}\}$, $\mathsf{V}_C = \{\mathsf{X}_{C,1}, \mathsf{X}_{C,2}, \cdots \mathsf{X}_{C,T}\}$, $\mathsf{V}_B = \{\mathsf{X}_{B,1}^1, \mathsf{X}_{B,2}^1, \cdots \mathsf{X}_{B,K[1]}^1; \mathsf{X}_{B,1}^2, \mathsf{X}_{B,2}^2, \cdots \mathsf{X}_{B,K[2]}^2; \cdots; \mathsf{X}_{B,1}^R, \mathsf{X}_{B,2}^R, \cdots \mathsf{X}_{B,K[R]}^R\}$, and $\mathsf{V}_O = \{\mathsf{X}_{O,1}, \mathsf{X}_{O,2}, \cdots \mathsf{X}_{O,Z}\}$, where subscripts $S,1; S,2; \cdots; S,M$; $C,1; C,2; \cdots; C,T$; $O,1; O,2; \cdots; O,Z$ are the indices of the corresponding feature V_S, V_C, and V_O in the set of vertices V, respectively. Clearly, when building CINN, there is no need to account for the set of isolated nodes because they have no causal relationship with any other variables in the DAG G.[1] Regarding the set of features V_B, $K[i]$ is a function representing the number of features in the i-th layer of intermediate nodes in V_B and $\mathsf{X}_{B,1}^i, \mathsf{X}_{B,2}^i, \cdots, \mathsf{X}_{B,K[i]}^i$ represent the corresponding features in that layer, where the superscript indicates the layer number. Naturally, we have $M + T + \sum_{i=1}^{R} K[i] + Z = d + 1$.

Next, we devise the architecture of CINN in a way such that it adheres to the causal relationships among the three node categories V_C, V_B, and V_O.[2] Fig. 3 illustrates the CINN architecture with two hidden layers, where the causal structure in the form of DAG among the three groups of nodes V_C, V_B, and V_O is encoded into the network architecture design. More specifically, root nodes $\mathsf{V}_C = \{\mathsf{X}_{C,1}, \mathsf{X}_{C,2}, \cdots \mathsf{X}_{C,T}\}$ act as input features of the neural network, while intermediate nodes $\mathsf{V}_B = \{\mathsf{X}_{B,1}^1, \mathsf{X}_{B,2}^1, \cdots \mathsf{X}_{B,K[1]}^1; \cdots; \mathsf{X}_{B,1}^R, \mathsf{X}_{B,2}^R, \cdots \mathsf{X}_{B,K[R]}^R\}$ play a dual role. On the one hand, they act as the outcomes of root nodes V_C or the preceding layer of intermediate nodes; on the other hand, they serve as the causes of the succeeding layer of intermediate nodes or leaf nodes $\mathsf{V}_O = \{\mathsf{X}_{O,1}, \mathsf{X}_{O,2}, \cdots \mathsf{X}_{O,Z}\}$. Finally, leaf nodes V_O represent the outputs of the neural network, and both V_C and V_B serve as its inputs. It should be noted that nodes V_B and V_O share a hidden layer, denoted by $\theta_{11}, \theta_{12}, \cdots, \theta_{1r}$, in addition to their own task-specific layers of neurons as represented by $\theta_{21}, \theta_{22}, \cdots, \theta_{2q}$ and $\theta_{31}, \theta_{32}, \cdots, \theta_{3u}$, respectively. The purpose is to maintain the similarities and differences that coexist between the two types of nodes as observed in the original causal DAG. For example, as illustrated in Fig. 2, X_1 could influence X_9 through multiple pathways: It can affect X_9 directly via the edge $\mathsf{X}_1 \rightarrow \mathsf{X}_9$, or indirectly through the paths $\mathsf{X}_1 \rightarrow \mathsf{X}_5 \rightarrow \mathsf{X}_8 \rightarrow \mathsf{X}_9$ and $\mathsf{X}_1 \rightarrow \mathsf{X}_6 \rightarrow \mathsf{X}_9$. This allows the effect that root nodes have on leaf nodes to be imposed via different pathways so that the effect does not necessarily be enforced through the intermediate layers.

In addition to the causality-aligned network architecture, CINN also gets embodied in the design of loss function. In a conventional neural network, there is only one target variable typically located at the output layer, and the network is trained to minimize the loss between the predicted and observed values with respect to this single target variable. When training CINN, however, we treat both V_B and V_O as target outputs. Our goal is to minimize the overall loss over the observations associated with V_B and V_O using the N observational data so as to synergize co-learning of causal relationships among observed variables. Mathematically, the loss function of CINN is formulated as follows:

$$\min_{\boldsymbol{\theta}} \quad \mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left[\underbrace{\sum_{j=1}^{R}\sum_{k=1}^{K[j]}\left(\overline{\mathbf{X}}_{B,k}^{i,j} - \widehat{\mathbf{X}}_{Bk}^{i,j}\right)^2}_{\mathcal{L}_{MSE}^{B}} + \underbrace{\sum_{z=1}^{Z}\left(\overline{\mathbf{X}}_{O,z}^{i} - \widehat{\mathbf{X}}_{O,z}^{i}\right)^2}_{\mathcal{L}_{MSE}^{O}}\right], \tag{3}$$

where $\widehat{\mathbf{X}}_{B,k}^{i,j}$ denotes the output with respect to neuron $\mathsf{X}_{B,k}^{j}$ in the j-th layer of intermediate nodes, and $\widehat{\mathbf{X}}_{O,z}^{i}$ refers to the output with respect to $\mathsf{X}_{O,z}$ $(z = 1, 2, \cdots, Z)$ when the set of features V_C associated with the i-th observational data is fed into the neural network; $\overline{\mathbf{X}}_{B,k}^{i,j}$ (resp. $\overline{\mathbf{X}}_{O,z}^{i}$) represents the i-th observation associated with the feature $\mathsf{X}_{B,k}^{j}$ (resp. $\mathsf{X}_{O,z}$).

The loss function formulated in Eq. (3) drives co-learning of causal relationships between V_C and V_B, V_C and V_O, as well as V_B and V_O. Such an objective function would induce the trained neural network to respect the causal relationships and the orientation of causal relationships among the observed variables. By adhering to the causal structure, the trained neural network is anticipated to have a stable and augmented performance when making predictions.

3) Integration of Domain Knowledge: The developed CINN architecture illustrated in Fig. 3 not only allows the incorporation of causal structure, but also enables encoding domain knowledge that characterizes any quantitative or qualitative relationship between any pair of cause-and-effect variables. For example, if

[1]For the case that the quantity of interest is one of the isolated nodes, there is no way to build the deep learning model because the isolated node is not causally related to any other observed variables in graph G.

[2]The isolated nodes are discarded when creating the architecture of neural network. Thus, we refer to the number of categories as three instead of four thereafter. If one prefers to incorporate the isolated nodes in CINN, a possible way is to include them as some variables in the set of root nodes.

a domain prior regarding the causal relationship between $\mathsf{X}_{C,t}$ $(t = 1, 2, \cdots, T)$ and $\mathsf{X}^j_{B,k}$ $(k = 1, 2, \cdots, K[j])$ is available, then it can be exploited to guide neural network training. Even though the exact mathematical form of such domain knowledge might be unknown, rough causal relationships in a variety of forms, such as monotonic effect, zero effect, U-shaped effect, and loose differential equation, can be treated as inequality or equality constraints when training the neural network. For instance, if we would like to impose a fairness constraint to ensure that race ($\mathsf{X}_{C,t}$) has no effect on the loan application ($\mathsf{X}^j_{B,k}$) in the learned model [41], a constraint like $\mathrm{d}\widehat{\mathbf{X}}^j_{B,k}/\mathrm{d}\mathbf{X}_{C,t} = 0$ can be encoded into the neural network. In another case, if there is a positive causal relationship between $\mathsf{X}_{C,t}$ and $\mathsf{X}^j_{B,k}$, without an exact form, a soft constraint like $\mathrm{d}\widehat{\mathbf{X}}^j_{B,k}/\mathrm{d}\mathbf{X}_{C,t} > 0$ can be formulated. In general, the domain causal knowledge in most forms can be transformed into an equivalent differential formulation in an unified manner. Therefore, we develop a differential-based formulation to enforce the neural network to keep in compliance with the known domain prior on causal relationships: where M_{OC}, M^k_{BC}, and M_{OB} are binary matrices indicating for which variable prior causal knowledge is available. In particular, M^k_{BC} is a binary indicator characterizing the causal relationship between the k-th layer of intermediate nodes and the root nodes V_C. $\nabla_i \widehat{\mathbf{X}}_O$ is a $Z \times T$ Jacobian of the Z-dimensional neural network prediction $\widehat{\mathbf{X}}_O$ with respect to the i-th T-dimensional input; similarly, $\nabla_i \widehat{\mathbf{X}}^k_B$ is a $K[k] \times T$ Jacobian of the $K[k]$-dimensional prediction $\widehat{\mathbf{X}}^k_B$ with respect to the i-th T-dimensional input; and $\nabla_i \widehat{\mathbf{X}}_O / \nabla_i \widehat{\mathbf{X}}^k_B$ is a $Z \times K[k]$ Jacobian of the Z-dimensional prediction $\widehat{\mathbf{X}}_O$ with respect to the i-th $K[k]$-dimensional output $\widehat{\mathbf{X}}^k_B$; $\odot$ denotes the element-wise product. If there is prior knowledge on the causal relationship between a cause variable i in X_C and an effect variable j in $\widehat{\mathsf{X}}_O$, then $\mathsf{M}^{ij}_{OC} = 1$; otherwise, zero. The construction of M^k_{OB} and M^k_{BC} follows a similar approach. $\delta \mathsf{g}^i_{OC}$, $\delta \mathsf{g}^{i,k}_{BC}$, and $\delta \mathsf{g}^{i,k}_{OB}$ denote matrix derivatives of available domain priors with a size of $Z \times T$, $K[k] \times T$, and $Z \times K[k]$ with respect to X_C, X_C, and $\widehat{\mathsf{X}}_B$, respectively. ε is a hyperparameter to allow a margin of error when regularizing the neural network with prior causal knowledge.

By incorporating domain knowledge into the neural network, we formulate a composite loss function to guide the training of CINN as informed by the rectified causal structure and the prior causal knowledge:

$$\mathcal{L} = \mathcal{L}_{MSE} + \gamma \mathcal{L}_R, \tag{4}$$

where γ denotes the weight associated with the domain prior term $\mathcal{L}_R$. Increasing (resp. lowering) the value of γ strengthens (resp. diminishes) the regularization effect of prior domain knowledge. Devising the architecture and loss function in such a manner brings substantial benefits to neural network learning.

i) Incorporating the hierarchical causal structure can diminish the distraction of spurious relationships derived from noisy data to a neural network [29, 42–44]. Let us revisit the causal graph shown in Fig. 2. Suppose there is a strong spurious correlation between X_9 and X_{10}, and X_9 is the quantity of our interest. If X_{10} is used as an input feature, then deep learning model tends to rely more on X_{10} to predict X_9 rather than other upstream causal factors (e.g., X_1, X_5, X_8, Y), because X_{10} helps to reduce the loss the most. Encoding the causal structure into a neural network prevents the learning of spurious relationships as the network architecture is aligned with the structural causal relationships as defined in the discovered causal graph. The informative network architecture serves as a good starting point towards stable and meaningful representation learning.

ii) Different from existing studies, the developed neural network architecture strictly respects the orientation of causal relationships that are believed to govern the observational data. The orientated causal relationships encoded into neural network can prune the search space over possible combinations of network architecture and parameters, thus providing valuable information to effectively guide neural network training. Such an informative architecture design makes neural network more parsimonious by adhering to the established causal structure among the observed variables. Moreover, the proposed CINN provides straightforward interfaces for integrating domain knowledge into neural networks. On the one hand, domain knowledge can be leveraged to refine the hierarchical causal structure discovered from observational data by pruning the discovered causal graph G, such as removing unreasonable edges, adding causal relationships between certain variables, among others. On the other hand, CINN allows the incorporation of causal relationships among many pairs of observed variables as specified by domain experts. Specifically, it not only allows the incorporation of causal relationships between input features $\mathsf{X}_1, \mathsf{X}_2, \cdots, \mathsf{X}_d$ and target output Y, as some existing studies have done, but also enables encoding causal relationship between two variables in input features $\mathsf{X}_1, \mathsf{X}_2, \cdots, \mathsf{X}_d$, if any.

iii) The causal structure discovered from observational data is amendable to available domain priors on causal relationships, thereby facilitating the combination of well established expert knowledge and observational data. In fact, combining observational data and domain knowledge has been widely adopted in the existing literature to gain a better structural representation of relationships among random variables, such as Bayesian network learning [45, 46].

C. Projection of Conflicting Gradients for CINN

As shown in Eq. (4), the loss function for CINN has two terms, namely, $\mathcal{L}_{MSE}$ and $\mathcal{L}_R$. The former consists of two individual components $\mathcal{L}^B_{MSE}$ and $\mathcal{L}^O_{MSE}$ to quantify the losses specific to the predictions associated with V_B and V_O, respectively, while the latter the extent to which the neural network violates the imposed domain knowledge regarding causal relationships. It is important to note that every single component $\mathcal{L}^B_{MSE}$ and $\mathcal{L}^O_{MSE}$ might consist of a series of loss items; see Fig. 3 for more information. For example, $\mathcal{L}^B_{MSE}$ might be composed of 10 intermediate outputs.

Training CINN to simultaneously master multiple tasks is inherently challenging, as it involves optimizing several learning objectives [47]. During backpropagation with gradient descent-based optimizers, the three loss terms $\mathcal{L}^B_{MSE}$, $\mathcal{L}^O_{MSE}$, and $\mathcal{L}_R$ are typically combined using a weighted sum. In the case that the three loss terms have significant differences in magnitude, the average operation can make some loss component dominate the others during the learning process. Optimizing towards the dominated task leads to performance degradation and sacrifice of other learning tasks. In addition to imbalance in gradient magnitude, the gradients of different learning tasks might be conflicting along the direction of descent with one another, which is detrimental to the optimization of neural network parameters. As a result, the optimizer may oscillate or struggle to make consistent progress when updating the network weights.

$$\mathcal{L}_R = \sum_{i=1}^{N} \max\left\{0, \left\|\nabla_i \widehat{\mathbf{X}}_O \odot \mathsf{M}_{OC} - \delta \mathsf{g}_{OC}^{i}\right\|_1 - \varepsilon\right\} + \sum_{i=1}^{N}\sum_{k=1}^{R} \max\left\{0, \left\|\nabla_i \widehat{\mathbf{X}}_B^k \odot \mathsf{M}_{BC}^k - \delta \mathsf{g}_{BC}^{i,k}\right\|_1 - \varepsilon\right\} + \sum_{i=1}^{N}\sum_{k=1}^{R} \max\left\{0, \left\|\frac{\nabla_i \widehat{\mathbf{X}}_O}{\nabla_i \widehat{\mathbf{X}}_B^k} \odot \mathsf{M}_{OB}^k - \delta \mathsf{g}_{OB}^{i,k}\right\|_1 - \varepsilon\right\},$$

In this study, we tackle this problem from a multi-task learning perspective, where each loss component is treated as an individual learning task. To mitigate gradient interference among different learning tasks, we adopt the PCGrad approach to eliminate conflicting elements from the set of gradients [48]. The key idea is to project the gradient of a task onto the norm plane of any other task if there is a conflict between their gradients. To demonstrate the idea of PCGrad, we take the gradients associated with $\mathcal{L}_{MSE}^{B}$ and $\mathcal{L}_R$ as an example. PCGrad first checks whether there is any conflict between $\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}$ and $\boldsymbol{\Delta}_{\mathcal{L}_R}$ using the cosine similarity metric defined below:

$$\omega\left(\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}, \boldsymbol{\Delta}_{\mathcal{L}_R}\right) = \frac{\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}} \bullet \boldsymbol{\Delta}_{\mathcal{L}_R}}{\left\|\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}\right\|_2 \|\boldsymbol{\Delta}_{\mathcal{L}_R}\|_2}, \tag{5}$$

where $\|\cdot\|_2$ denotes the L_2 norm of a vector. The metric produces a value within the range $[-1, 1]$, where -1 denotes exactly the opposite direction, 1 means exactly the same direction, and 0 indicates orthogonality or decorrelation.

If $\omega(\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}, \boldsymbol{\Delta}_{\mathcal{L}_R}) < 0$, PCGrad projects $\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}$ onto the norm plane of $\boldsymbol{\Delta}_{\mathcal{L}_R}$ or vice versa; otherwise, both $\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}$ and $\boldsymbol{\Delta}_{\mathcal{L}_R}$ remain unchanged (see Fig. 4 for illustration). For demonstration, suppose we project $\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}$ onto the norm plane of $\boldsymbol{\Delta}_{\mathcal{L}_\mathcal{R}}$. After this projection, the gradient $\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}$ is updated as follows:

$$\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}^{\text{PC}} = \boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}} - \frac{\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}} \bullet \boldsymbol{\Delta}_{\mathcal{L}_\mathcal{R}}}{\|\boldsymbol{\Delta}_{\mathcal{L}_\mathcal{R}}\|^2} \boldsymbol{\Delta}_{\mathcal{L}_\mathcal{R}}, \tag{6}$$

where $\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}^{\text{PC}}$ denotes the gradient of $\boldsymbol{\Delta}_{\mathcal{L}_{MSE}^{B}}$ after projection.

Algorithm 2: Projecting conflicting gradients in CINN

Data: Neural network parameters $\boldsymbol{\theta}$
Result: Updated gradient $\boldsymbol{\Delta\theta}$

$\boldsymbol{\Delta}_i \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}),\ \forall i = 1, 2, 3$ ▷ $\mathcal{L}_i$ denotes the i-th loss term. Note $\mathcal{L}_1 = \mathcal{L}_{MSE}^{B}; \mathcal{L}_2 = \mathcal{L}_{MSE}^{O}; \mathcal{L}_3 = \mathcal{L}_R$.
$\boldsymbol{\Delta}_i^{\text{PC}} \leftarrow \boldsymbol{\Delta}_i,\ \forall i$
for ($i = 1;\ i <= 3;\ i = i + 1$) {
  for ($j \overset{uniformly}{\sim} [1, 2, 3]$, where $j \neq i$) {
    if $\omega\left(\boldsymbol{\Delta}_i^{PC}, \boldsymbol{\Delta}_j\right) < 0$ **then**
      Set $\boldsymbol{\Delta}_i^{\text{PC}} = \boldsymbol{\Delta}_i^{\text{PC}} - \frac{\boldsymbol{\Delta}_i^{\text{PC}} \bullet \boldsymbol{\Delta}_j}{\|\boldsymbol{\Delta}_j\|^2} \boldsymbol{\Delta}_j$ ▷ Subtract the projection of $\boldsymbol{\Delta}_i^{\text{PC}}$ onto $\boldsymbol{\Delta}_j$
    end
  }
}
return $\boldsymbol{\Delta\theta} = \sum_{i=1}^{3} \boldsymbol{\Delta}_i^{\text{PC}}$

Algorithm 2 outlines the implementation of PCGrad in CINN. PCGrad repeats the projection operation for all the learning tasks in a random order. In essence, the gradient projection operation in Fig. 4 amounts to removing the conflicting element from the gradients, thus mitigating destructive gradient interferences among different learning tasks. The introduction of PCGrad in CINN frees us from the tuning of weights associated with each loss term. In addition, since the gradient projection operation accounts for gradient information (e.g., magnitude, direction) associated with all the loss components in a holistic manner, it significantly mitigates the conflicts among different learning tasks and produces a set of gradients with a minimal gradient interference, thus leading to a stable convergence in the optimization of neural network parameters.

III. Computational Experiments

In this section, we comprehensively examine the prediction performance of the developed CINN methodology on a broad spectrum of publicly available datasets. The performance of CINN is compared against a wide range of state-of-the-art models in the literature. The robustness of CINN with respect to several hyperparameters (e.g., learning rate, seed value for neural network initialization, weighting factor γ) is examined using a series of computational experiments. In addition, we also conduct an ablation study to demonstrate the value of leveraging causal knowledge in refining the prediction performance of CINN.

A. Datasets

We consider five datasets that are publicly available in the UCI Machine Learning Repository [49], including Boston Housing (BH), Wine Quality (WQ), Facebook Metrics (FB), Bioconcentration (BC), and Community Crime (CM), to demonstrate how CINN generates better predictions compared to other prevailing models. These datasets represent a broad range of application domains with tasks varying from house price prediction (BH), crime rate prediction (CM) to the forecast on the number of interactions to a post in Facebook (FB) as well as wine quality prediction (WQ). These datasets also vary in terms of the number of variables, the number of edges among variables, causal relationships among observed variables, available data for model development, the distribution of target variable values, and represent a diverse spectrum of sectors. Importantly, causal relationships are ubiquitously present in the observed data across the five considered datasets. For example, in the case of BH, per capita crime rate by town (CRIM) and the average number of rooms per dwelling (RM) are causally associated with the median value of owner-occupied homes (MEDV). In fact, these factors make the five datasets a common choice for causal discovery and development of causally-aware ML models [see, e.g., 29, 42, 50].

By demonstrating the CINN's performance on these different prediction tasks, our goal is to showcase the generalization of CINN's prediction performance in different business contexts. Table I briefly describes the basic information of each dataset used for performance validation and comparison, such as the features and the target variable. Note that after an ML model is trained, we

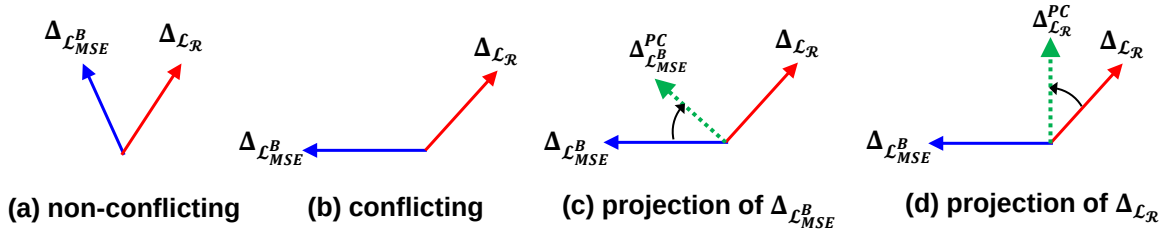


Fig. 4. Demonstration of PCGrad. (a) There is no conflict between $\Delta_{\mathcal{L}^B_{MSE}}$ and $\Delta_{\mathcal{L}_R}$. (b) There is a high conflict between $\Delta_{\mathcal{L}^B_{MSE}}$ and $\Delta_{\mathcal{L}_R}$. (c) PCGrad projects the gradient $\Delta_{\mathcal{L}^B_{MSE}}$ onto the norm vector of the gradient $\Delta_{\mathcal{L}_\mathcal{R}}$. (d) PCGrad projects the gradient $\Delta_{\mathcal{L}_\mathcal{R}}$ onto the norm vector of the gradient $\Delta_{\mathcal{L}^B_{MSE}}$.

verify its prediction performance only regarding the target variable (referred to as the *primary prediction task* throughout the rest of the paper).

B. Baselines

As we are developing a new way to encode cause-and-effect relationships into neural networks, we restrict our comparison to state-of-the-art methods that exploit causality as a mechanism to regularize neural network for improving its prediction performance. In addition, we also compare CINN with other prevailing regularizers commonly adopted to enhance the prediction performance of neural networks. To this end, we benchmark the proposed CINN against the following methods:

- **Early stopping** [51]: Early stopping is a prevailing regularization method used in deep neural networks to halt model training when parameter updates no longer yield an improvement on the validation set. In essence, when this is the case (after a number of iterations), we terminate model training and use the best-performing parameters obtained so far as the parameters of a neural network. Early stopping is used as a baseline in the rest of the paper.
- L_1 **norm**: L_1 regularization (also referred to as Lasso regularization) employs the sum of the absolute values of the weight parameters in neural network as a constitutional part of the loss function to reduce model complexity. In doing so, L_1 norm promotes sparsity by forcing some weights to zero, thus preventing the model from overfitting.
- L_2 **norm**: L_2 regularization adds the sum of squared magnitude of model parameters as a penalty term to the loss function. As model complexity increases, the penalty term penalizes larger weight values to regularize neural network so that the trained model generalizes well to unseen data.
- **Dropout (DO)** [52]: DO randomly drops out hidden and visible units in a neural network to prevent neurons from co-adapting excessively. Through random dropout, DO breaks up co-adaptions by making the presence of any particular hidden unit unreliable. In doing so, DO prevents neural network from overfitting and thus increases its generalization to unseen data.
- **Batch normalization (BN)** [53]: BN is a mechanism to accelerate the training of deep neural network through a normalization step that fixes the means and variances of each layer's inputs over every mini-batch. The normalization operation stabilizes the training of neural network and permits us to use much higher learning rates when training neural network.
- **Input noise** [54]: Adding a small amount of random noise to inputs helps prevent neural network from memorizing all the training examples, particularly in the case of small dataset. The randomly injected noise reduces generalization error and increases robustness.
- **Mixup** [55]: Mixup is a data-agnostic augmentation technique to construct virtual training instances from a pair of samples randomly selected from the training dataset. The mechanism behind mixup is to regularize a neural network by favoring simple linear behavior in-between training examples.
- **CASTLE**: As developed by [29], CASTLE incorporates causal DAG discovery as an auxiliary task to regularize neural network weights such that the weights of those non-causal predictors are shrunk. Fig. 5 gives a schematic description on CASTLE. As can be observed, in the auxiliary task of causal DAG learning, CASTLE reconstructs the masked feature (e.g., $\mathbf{X}_k,\ k = 1, 2, \cdots, d$) from the remaining features $(\mathbf{Y}, \mathbf{X}_1, \cdots, \mathbf{X}_{k-1}, \mathbf{X}_{k+1}, \cdots, \mathbf{X}_d, k = 1, 2, \cdots, d)$ subject to the DAG constraint $\mathsf{R}(\mathbf{W}) = 0$. In doing so, CASTLE imposes the d subnetworks of learning causal DAG as a joint task when training the neural network. Hence, CASTLE exploits causal DAG discovery to implicitly incorporate a causal graph into a neural network via a tailored loss function consisting of DAG loss $\mathsf{R}(\mathbf{W})$, predictor reconstruction loss $f_k(\overline{\mathbf{X}})$, and prediction loss $\|\mathbf{Y} - \overline{\mathbf{X}}\mathbf{W}\|$.

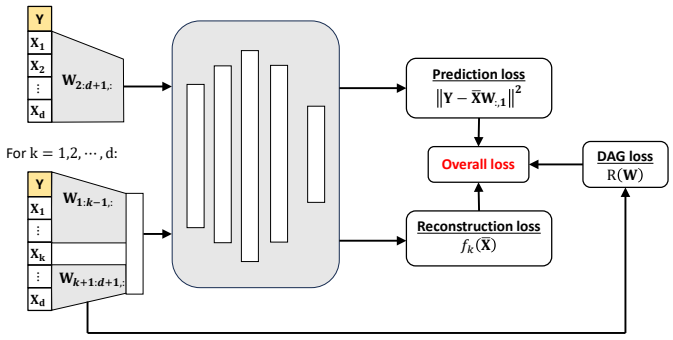


Fig. 5. Graphical illustration of CASTLE developed by Kyono et al. [29]

- **CASTLE+**: While CASTLE reconstructs each masked predictor from the remaining predictors and $\mathbf{Y}$, CASTLE+ [42] attempts to improve the performance of CASTLE by enforcing a causal DAG $\mathsf{G}^c(\mathsf{E}^c, \mathsf{V}^c)$ from domain experts so that the learned model is guaranteed to abide by the causal relationships defined in G^c. Considering a given edge from X_i to X_j, if the edge does not exist in the causal graph G^c approved by domain experts, then CASTLE+ masks the weight of the input layer as zero in the d subnetworks for predictor reconstruction; otherwise, masking is not applied in the d subnetworks.

In this work, we benchmark CINN against the above-described methods in no particular order. Note that BN and DO are applied

TABLE I
DESCRIPTIONS OF DATASETS USED IN THE COMPUTATIONAL EXPERIMENTS

dataset	$d+1$	N	Example features	Target variable
BH	14	506	Crime rate by town, average number of rooms	Median value of owner-occupied homes
WQ	12	4,898	Fixed acidity, volatile acidity, residual sugar	Wine quality
FB	19	500	Category, page total likes, post month, post weekday, post hour, type	Total interactions (comments + likes + shares)
BC	14	779	Number of heavy atoms, average valence connectivity	Bioconcentration factor in log units
CM	128	1994	Population for community, median household income	Number of violent crimes per 100K population

Note: N is the number of observations, and $d+1$ is the number of features including the target variable Y.

after every dense layer, and they are active only during training; L_1 regularization is applied at every dense layer. In the case of CASTLE+, the causal graph used by CINN also serves as the causal DAG G^c specified by domain experts to be injected into CASTLE+. Injecting the same causal DAG enables a fair comparison of the two different means of incorporating causal DAG into neural network. Last but not the least, each benchmark method is initialized and seeded identically with the same random weights.

In all the computational experiments, the entire dataset is randomly split into two parts: 80% is used as the training set, while the remaining 20% is reserved as the test set. Note that in the case of CINN, the training set is used to discover causal relationships and then train the neural network built upon the causal DAG after refinement. Then, ten-fold cross validation is used to examine the stability and robustness associated with the prediction performance of all the investigated models. At each fold, as each model converges at a different learning rate, a validation set (10% of training data) is extracted from the training data to tune the model parameters (e.g., learning rate, weight decay regularizer, the threshold value τ), and those parameters resulting in the lowest mean squared errors (MSE) are treated as the ultimate set of parameters of the trained model. In doing so, the performance of each model is tuned to its best state. Moreover, at each fold, we fix the seed of the program when splitting the entire data into training and test sets so that all the considered models have identical sets of data for training and test. Each model is trained using the Adam optimizer with the default learning rate of 0.001 for 800 epochs. Furthermore, an early stopping regime halts model training with a patience of 30 epochs.

In addition to comparing CINN with multiple state-of-the-art models, we also examine the performance of CINN with and without PCGrad to investigate the benefits of leveraging PCGrad in the optimization of neural network parameters. For a fair comparison, we also fix the seed value of the program when initializing the weights and biases of neural network so that both cases have the same set of initialized parameters as the starting point.

C. Demonstration of CINN Setup

1) Causal Discovery: As introduced earlier, CINN starts from discovering causal relationships from observational data. At the stage of causal discovery across different datasets, the values of λ and τ in Eq. (2) need to be finetuned. For categorical features, one-hot-encoded representation is used for DAG discovery and CINN training, while the non-categorical features are standardized with mean 0 and variance 1. The FB dataset has 60 features in total after categorical features are encoded, because the month, weekday, and hour when a post is published are all categorical variables. In contrast, the BH dataset has only 14 features. Considering the significant variation associated with the number of features in each dataset, we set parameters λ and τ to different values.

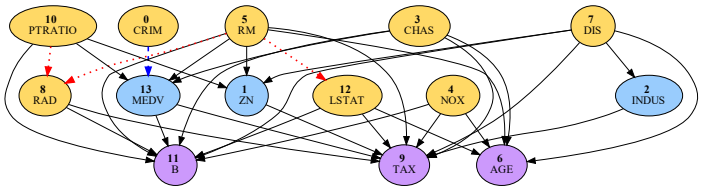


Fig. 6. Discovery and refinement of causal relationships among observed variables for the BH dataset. The bold-faced number in each circle indicates the feature ID, while the associated text represents the feature name. Our primary task is to predict the MEDV value (node 13) using other relevant features. The circles in the same color belong to the same category of nodes. Specifically, circles in yellow refer to the set of root nodes $\mathsf{V}_C = \{\mathsf{X}_0, \mathsf{X}_3, \mathsf{X}_4, \mathsf{X}_5, \mathsf{X}_7, \mathsf{X}_8, \mathsf{X}_{10}, \mathsf{X}_{12}\}$, circles in blue indicate the set of intermediate nodes $\mathsf{V}_B = \{\mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_{13}\}$, while circles in purple are the set of leaf nodes $\mathsf{V}_O = \{\mathsf{X}_6, \mathsf{X}_9, \mathsf{X}_{11}\}$. The red dotted lines denote the edges eliminated from the discovered causal graph by exploiting expert knowledge, while the edge in blue denotes the link that is additionally added to the discovered causal graph.

2) Causal DAG Refinement: After causal discovery, domain knowledge is then exploited to refine the discovered causal relationships for establishing CINN in two different ways: (i) refining the causal structure among observed variables by eliminating invalid causal links and adding substantiated causal edges; (ii) specifying and injecting quantitative relationships between certain variables into CINN following the method described in Section II-B. To demonstrate the specific steps of injecting expert knowledge into CINN construction, take the BH dataset as an example. Fig. 6 illustrates the refined causal graph after expert knowledge is used to eliminate three links from the original causal graph discovered by the algorithm developed by [36]. Towards this goal, it is essential to verify if there is a causal relationship between one variable a and another variable b in a fast manner. A straightforward way to perform such verification is to imagine that we only perturb the value of variable a while keeping the value of all the other variables fixed, and see whether the value of variable b changes as a result of this perturbation [56]. Following this way, it is easy to observe that even if we change the value of attribute RM (average number of rooms, X_5), it does not lead to a change on the values of attributes LSTAT (lower status of the population, X_{12}), RAD (index of accessibility to radial highways, X_8), among others. In contrast, if the value associated with the attribute CRIM (per capita crime rate by town, X_0) is increased or decreased, it has a direct effect on the value of variable MEDV (median home value, X_{13}). The causal relationship between CRIM (X_0) and MEDV (X_{13}) agrees

with our common sense and, in fact, has been confirmed by several other causal discovery algorithms, such as conditional independence test [50], maximum likelihood estimator for causal discovery [57], to name a few. As a result, we add the causal relationship between CRIM (X_0) and MEDV (X_{13}) in the discovered causal graph. It should be noted that the refined graph after elimination and/or addition operations must remain a DAG. If this is not the case, then necessary adjustments need be carried out to turn the refined graph into a DAG.

In the refined causal graph shown in Fig. 6, the features in yellow circles are root nodes acting as input features in CINN, while the features in blue and purple circles are nodes in the intermediate and output layers, respectively. Given the refined causal DAG, we have $\mathsf{V}_C = \{\mathsf{X}_0, \mathsf{X}_3, \mathsf{X}_4, \mathsf{X}_5, \mathsf{X}_7, \mathsf{X}_8, \mathsf{X}_{10}, \mathsf{X}_{12}\}$, $\mathsf{V}_B = \{\mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_{13}\}$, $\mathsf{V}_O = \{\mathsf{X}_6, \mathsf{X}_9, \mathsf{X}_{11}\}$. When training CINN, the parameters are optimized so that the total loss over the intermediate nodes V_B and leaf nodes V_O is minimized. In contrast, when making prediction, our primary task is to infer X_{13} (MEDV), which is a function of the input features V_C.

3) Incorporation of Causal Relationships: In addition to eliminating and inserting causal links in the discovered causal graph, CINN is also equipped with the capability of incorporating quantitative relationship between two variables into the neural network. Again, take the BH dataset as an example. It is easy to understand that LSTAT (X_{12}) has a negative impact on MEDV (X_{13}), provided that all the other contributing factors (e.g., X_5, X_3, X_7) remain fixed. In other words, the higher the LSTAT value, the lower the MEDV value. As a result, we can inject an inequality constraint into CINN, imposing that X_{13} and X_{12} are negatively causally related (i.e., $\mathrm{d}\mathsf{X}_{13}/\mathrm{d}\mathsf{X}_{12} \leq 0$). To allow a margin of error, we further modify the constraint to $\mathrm{d}\mathsf{X}_{13}/\mathrm{d}\mathsf{X}_{12} - \varepsilon \leq 0$, where ε is set to 0.01. We can incorporate the quantitative relationship between MEDV (X_{13}) and CRIM (X_0) in a similar manner. It is worth highlighting that a significant difference between CINN and classical neural networks is that they treat all variables (except target variable X_{13}) as inputs while the relationships among the input variables are ignored. Such an architecture design limits the possible ways of injecting causal knowledge, because it only permits to enforce relationships between target variable X_{13} and input variables. Table .2 of the supplementary material summarizes the values of λ and τ, as well as causal DAG refinement and incorporation of causal relationships along the development of CINN for each dataset, where CINN demonstrates a flexible capability for encoding causal knowledge into the neural network.

We adopt the same architecture as CASTLE [58] for all the alternative models that CINN is compared against (e.g., baseline, L_1, DO, BN). CINN has two hidden layers (Fig. 3) and ReLU is employed as the activation function. The first hidden layer has 32 units ($r = 32$), and both the hidden layers $\theta_{21}, \theta_{22}, \cdots, \theta_{2q}$ ($q = 16$) and $\theta_{31}, \theta_{32}, \cdots, \theta_{3u}$ ($u = 16$) have 16 hidden units. The outputs of the layer $\theta_{2,1}, \theta_{2,2}, \cdots, \theta_{2,q}$ are then passed to the layers of intermediate nodes for further propagation. Finally, the outputs of the last intermediate layer and the outputs of the layer $\theta_{3,1}, \theta_{3,2}, \cdots, \theta_{3,u}$ are concatenated and mapped to the layer of output nodes via a hidden layer denoted by $\theta_{4,1}, \theta_{4,2}, \cdots, \theta_{4,s}$, with $s = 8$.

D. Evaluation Metrics

Because all the datasets considered are related to regression-type problems, we use MSE on the test set to quantify the performance of all the considered models with respect to the primary prediction task (the target variables are indicated in Table I):

$$\mathrm{MSE} = \frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} (y_i - \widehat{y}_i)^2, \tag{7}$$

where N_{test} denotes the number of samples in the test set, y_i indicates the value of the target variable for the i-th sample, while $\widehat{y}_i$ is the prediction for the i-th sample made by the trained model.

E. Performance Comparison

After demonstrating the development process of CINN, we now compare its prediction performance against the other models under consideration. Note that the CINN model is trained for 800 epochs, and Adam optimizer with a learning rate of 0.001 is used to optimize the parameters. After CINN is trained, it can be used to make prediction for any variable in the intermediate and output layers. Since each dataset has a clearly defined target variable (see Table I), we benchmark CINN against the other models with respect to predicting the target variables.

Table II summarizes the performance of these models across all the datasets under consideration. In the first place, we concentrate on comparing the general performance of CINN with other prevailing models in the literature. At a quick glance, CINN (with and without PCGrad) significantly outperforms other state-of-the-art models in terms of test MSE across most datasets; an exception is BC, for which CINN leads to a slight improvement in prediction performance compared to other alternative models. It is worth highlighting that CASTLE only leads to a marginal improvement in prediction performance compared with other prevailing regularizers, while CINN outperforms CASTLE significantly, as reflected by the substantial reduction in the test MSE. For example, with respect to the BH dataset, CASTLE slightly beats the best-performed regularizer (i.e., DO); however, compared with CASTLE, CINN reduces the test MSE by 17.8% and 20%, respectively. Moreover, CINN diminishes the test MSE substantially in a consistent and stable manner as observed in other datasets, such as WQ, FB, and BC. In particular, in the case of CM, CINN reduces the test MSE from 0.383 to 0.319 (without PCGrad) and 0.318 (with PCGrad), which is a nearly 18% reduction against CASTLE. An interesting observation is that CASTLE+ achieves superior performance to CASTLE due to the incorporation of causal DAG, which underscores the importance of encoding causal DAG into neural networks.

Another appealing feature of CINN is its capability of attaining a desirable stability in the prediction performance. As shown in Table II, CINN reduces the standard deviation of test MSE to a large extent when benchmarking against other alternative models. The high variability associated with these alternative models is attributed to the inclusion of spurious correlations and relationships when building the deep learning models. In contrast, structuring the architecture of CINN as informed and guided by the hierarchical causal relationships extracted from observational data and domain experts substantially enhances the stability of prediction performance. In fact, the reduction in the standard deviation of test MSE is much more salient than the corresponding reduction in the associated mean value. The elevated stability in prediction performance is essential to the adoption of neural networks in safety-critical applications.

Next, we focus on performance comparison within the CINN family. As mentioned earlier in Section II-C, CINN involves the

TABLE II
PERFORMANCE COMPARISON IN TERMS OF TEST MSE ± STANDARD DEVIATION USING 10-FOLD CROSS-VALIDATION. BOLD DENOTES THE TEST MSE OF CINN AND UNDERLINE INDICATES THE TEST MSE OF CASTLE AND CASTLE+.

Model \ dataset	BH	WQ	FB	BC	CM
Early stopping (Baseline)	0.117 ± 0.036	0.746 ± 0.020	0.162 ± 0.030	0.293 ± 0.013	0.407 ± 0.027
L_1 norm	0.103 ± 0.033	0.736 ± 0.020	0.120 ± 0.045	0.284 ± 0.016	0.406 ± 0.015
L_2 norm	0.111 ± 0.038	0.741 ± 0.022	0.115 ± 0.042	0.289 ± 0.013	0.396 ± 0.029
Dropout 0.2	0.102 ± 0.034	0.732 ± 0.011	0.142 ± 0.054	0.286 ± 0.006	0.383 ± 0.014
Batch Norm	0.130 ± 0.035	0.740 ± 0.020	0.282 ± 0.042	0.300 ± 0.012	0.435 ± 0.010
Input Noise	0.109 ± 0.040	0.732 ± 0.010	0.120 ± 0.036	0.299 ± 0.017	0.395 ± 0.019
MixUp	0.113 ± 0.030	0.754 ± 0.023	0.159 ± 0.042	0.299 ± 0.021	0.425 ± 0.034
CASTLE	0.101 ± 0.019	0.730 ± 0.043	0.114 ± 0.098	0.283 ± 0.086	0.380 ± 0.032
CASTLE+	0.093 ± 0.015	0.727 ± 0.053	0.112 ± 0.064	0.282 ± 0.050	0.368 ± 0.022
CINN (without PCGrad)	$\mathbf{0.083 \pm 0.006}$	$\mathbf{0.710 \pm 0.020}$	$\mathbf{0.085 \pm 0.024}$	$\mathbf{0.281 \pm 0.006}$	$\mathbf{0.319 \pm 0.007}$
CINN (with PCGrad)	$\mathbf{0.081 \pm 0.005}$	$\mathbf{0.703 \pm 0.011}$	$\mathbf{0.079 \pm 0.021}$	$\mathbf{0.280 \pm 0.004}$	$\mathbf{0.318 \pm 0.006}$

learning of multiple tasks that are conflicting with respect to gradients in nature. Since the gradients associated with these learning tasks differ significantly in magnitude and direction, PCGrad is leveraged to mitigate gradient interferences in multi-task learning by eliminating conflicting components from the set of gradients. Computational results reported in Table II suggest that integrating PCGrad into CINN not only leads to a lower mean MSE, but also results in a slightly contracted standard deviation when compared with the case of no PCGrad. These findings reveal the efficacy of utilizing PCGrad to deconflict the gradients when training the CINN model. In practice, PCGrad is recommended to be used in the context of multi-task learning to enhance model performance in prediction and stability.

F. Robustness Checks

The aim of this subsection is to carry out a series of robustness checks for the developed CINN. Specifically, we investigate the effect of two factors on the CINN's performance. The first is the seed value for initialization of neural network parameters (e.g., bias, weights), and the second is regarding two hyperparameters (i.e., γ in Eq. (4) and learning rate β).

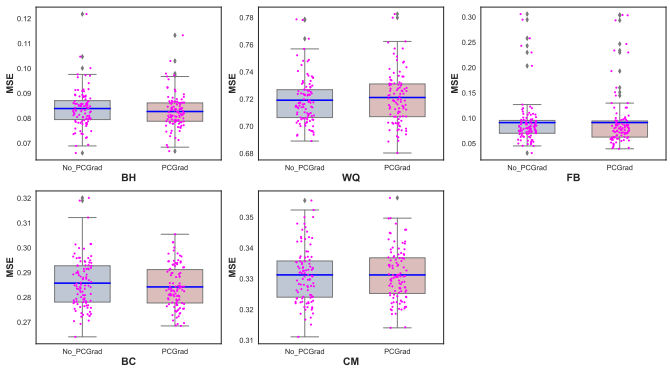


Fig. 7. Effect of seed value on the robustness of CINN performance. The solid blue line indicates the mean value of MSE. Note that the 100 MSE values (10 seed values × ten-fold cross validation) are aggregated in the box plot. The low deviation of MSE value reflects the performance robustness of CINN with respect to the seed value.

For the seed value, we use ten different seed values to initialize the CINN parameters. In each fold of cross validation, we run 10 experiments with an identical seed value for initialization. The other CINN configurations (e.g., learning rate, number of epochs, network architecture) remain the same as before. Fig. 7 shows the relationship between the seed value and the CINN performance. Clearly, as reflected by the variation of MSE value, the effect of different seed values on the CINN's prediction performance is trivial, and CINN exhibits a relatively robust performance that is consistent with the computational results reported in Table II. As shown in Fig. 7, CINN achieves a much lower mean MSE value than the other baseline models across all datasets. Furthermore, when compared with the case without PCGrad, CINN with PCGrad achieves a lower variance for datasets BH, WQ, and CM, while maintaining a comparable performance for FB and BC.

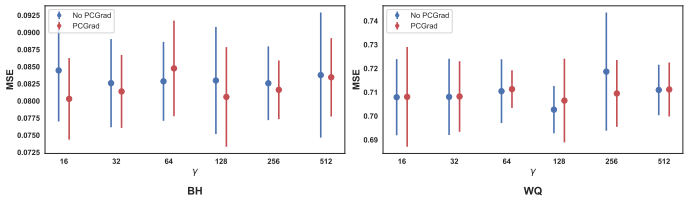


Fig. 8. Impact of hyperparameter γ on the robustness of CINN performance.

We then investigate the impact of hyperparameter γ on the performance of CINN. Recall that γ represents the weight associated with the domain prior on the causal relationship in Eq. (4). As γ is relevant only when domain prior is involved in the construction of CINN, we confine the robustness check to datasets BH and WQ, as no domain prior is imposed with respect to FB, BC, and CM (see the last column in Table .2 in the supplementary material). Fig. 8 illustrates the impact of γ on the robustness of the CINN's performance. At a quick glance, the effect of γ on the performance of CINN is insignificant, as reflected by the negligible variation in the MSE values for the three datasets. Notably, regardless of the value of γ, the mean MSE values stay consistent with those reported in Table II. The insignificant effect of γ on the CINN's performance is primarily attributed to the small magnitude of loss $\mathcal{L}_R$ compared to the overall loss pertaining to the observed variables in the intermediate and output layers.[3]

We further study the robustness of CINN with respect to the learning rate β. For this purpose, we gradually increase the value of β from 0.0001 to 0.02 and observe the change in the CINN's performance. As shown in Fig. 9, the learning rate has a strong effect on the performance of CINN. In particular, when β takes a large value (say, 0.02) or an extremely low value (say, 0.0001), the

[3]Note that $\mathcal{L}_R$ is not always active and it becomes active only when the causal relationship described in the last column of Table .2 in the supplementary material is violated.

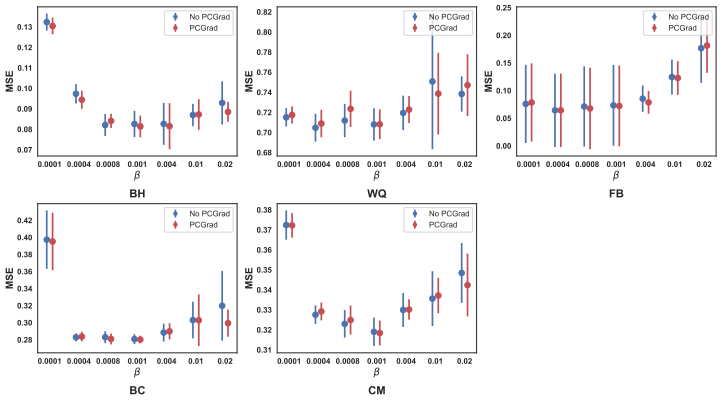


Fig. 9. Impact of learning rate β on the robustness of CINN performance

performance of CINN deteriorates substantially across all the five datasets. For example, with respect to the CM dataset, the MSE value increases to 0.37 when β drops down to 0.0001. While for other values of β, the MSE value fluctuates within a reasonable range. Another interesting observation is that CINN with PCGrad outperforms the case without PCGrad in either the MSE value or the standard deviation—reflected by the spread of the bar.

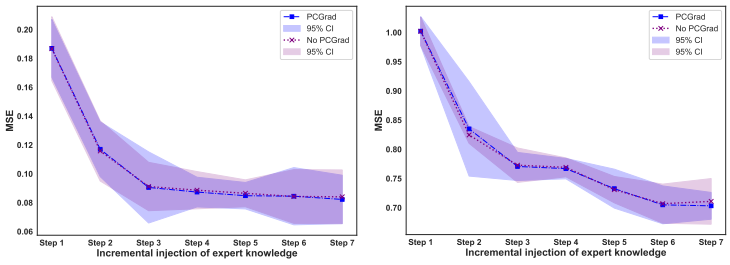
G. Ablation Study of CINN

Intuitively, aligning a neural network with an accurate and informative causal structure should yield an improved prediction performance. The aim of this subsection is to examine the effect of (i) leveraging expert knowledge in rectifying the discovered causal graph and (ii) enforcing quantitative causal relationships in the development of CINN, in an incremental manner. We are particularly interested in whether and how much the prediction performance of CINN can be improved if the neural network is enforced to abide by "guidance" provided by domain experts. Let us consider datasets BH and WQ for demonstration purposes. The ablation study of CINN with respect to the other three datasets can be found in the Appendix. We perform a series of experiments by incrementally injecting expert knowledge in the form of causal links and quantitative causal relationships into CINN so as to examine the effect of incorporating these information on the performance of the primary prediction task. In this ablation study, the specific expert knowledge incorporated into CINN at each step for BH and WQ is summarized in Table III. Specifically, we consider 7 different sets of expert knowledge for both datasets. The steps shaded in gray are expert knowledge in the form of quantitative causal relationships, while those in blue are expert knowledge in the form of trimming the discovered causal graph in structure.

In the ablation study, ten-fold cross validation is used to examine the performance of CINN with respect to the primary prediction task (i.e., inferring the median value of owner-occupied homes (i.e., MEDV) for BH and wine quality for WQ, respectively). Fig. 10 shows the effect of incrementally incorporating expert knowledge on the prediction performance of CINN. Consistent with our expectation, the test MSE drops down in a steady manner when more expert knowledge is utilized to build CINN. In particular, if no expert knowledge is exploited (Step 1), then CINN has an average test MSE of 0.187 for dataset BH, which is much larger than that of CASTLE. In case of BH, after incorporating these quantitative causal relationships, the mean MSE of PCGrad drops

TABLE III
EXPERT KNOWLEDGE INJECTED INTO THE CONSTRUCTION OF CINN AT EACH STEP FOR BH AND WQ

Step	BH	WQ
1	Inject no expert knowledge	Reverse the direction of link [5, 11] to build CINN with the discovered causal graph because node 11 is the target variable in WQ
2	Eliminate edges [5, 8], [10, 8], [5, 12] from the discovered causal graph	Reverse the direction of link [6, 11]
3	Insert edge [0, 13] into the discovered causal graph	Remove [1, 5], [10, 5], [0, 5], [3, 5]
4	(MEDV w.r.t. LSTAT) – Impose the causal relationship between X_{13} and X_{12} as $\frac{dX_{13}}{dX_{12}} - 0.01 \leq 0$ in CINN	Remove [3, 6]
5	(MEDV w.r.t. RM) – Inject the causal relationship between X_{13} and X_5 as $0.01 - \frac{dX_{13}}{dX_5} \leq 0$ in CINN	Remove [8, 3], [10, 3] and add [4, 11], [7, 11]
6	(MEDV w.r.t. CRIM) – Impose the causal relationship between X_{13} and X_0 as $\frac{dX_{13}}{dX_0} - 0.01 \leq 0$ in CINN	Add [3, 11]
7	(B w.r.t. CRIM and NOX) – Inject the causal relationships between X_{11} and X_0, X_4 as $\frac{dX_{11}}{dX_0} - 0.01 \leq 0$ and $\frac{dX_{11}}{dX_4} - 0.01 \leq 0$ in CINN	(total sulfur dioxide w.r.t. alcohol) – Enforce the quantitative causal relationship between X_6 and X_{10} as $\frac{dX_6}{dX_{10}} + 0.01 \leq 0$



(a) BH (b) WQ

Fig. 10. Effect of incrementally injecting expert knowledge on the test MSE for (a) BH and (b) WQ

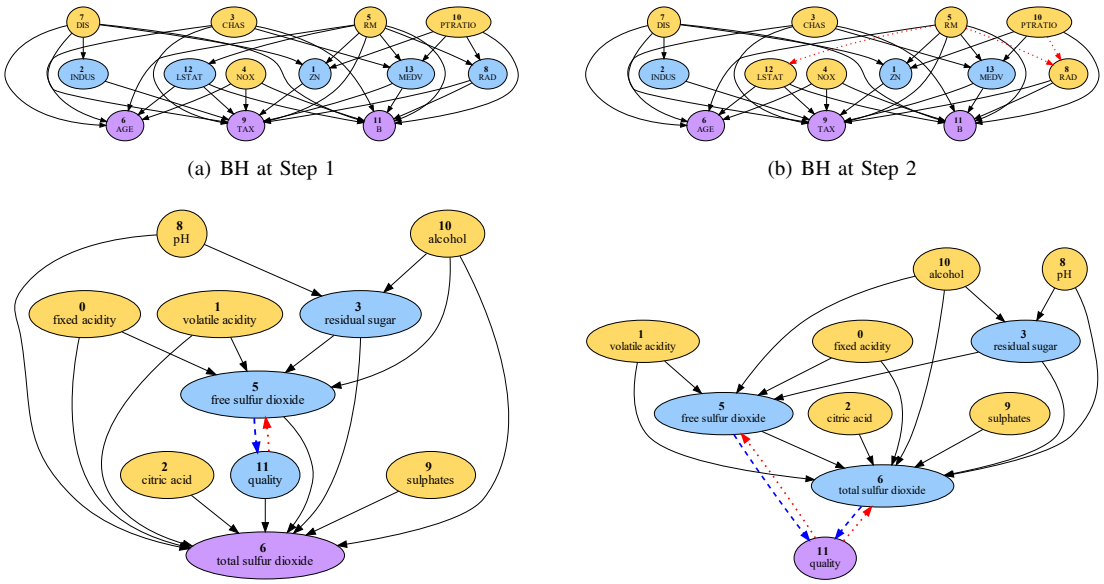
from 0.090553 (Step 3) to 0.087347 (Step 4), 0.084830 (Step 5), 0.084479 (Step 6), and 0.082246 (Step 7).

We have similar finding for WQAfter quantitative causal relationship is injected at Step 7, the mean MSE of PCGrad drops from 0.7049 (Step 6) to 0.7032 (Step 7). In addition, as reflected by the slope of the curve in Fig. 10, the reduction in test MSE is most significant from Step 1 to Step 2 in both cases, as the change in the causal graph structure is most informative. Fig. 11 illustrates the evolution of causal graph from Step 1 to Step 2. In the case of BH, after removing three invalid causal links, the nodes in the only intermediate layer is reduced from 5 to 3 (LSTAT and RAD become input nodes rather than intermediate nodes after refinement). While in the case of WQ, since total sulfur dioxide is also a significant factor affecting wine quality, after reversing the direction of link $[6, 11]$ the test MSE decreases significantly.

The ablation study highlights that the difference in the test MSE values between Step 1 and Step 7 is attributed to the injection of informative causal knowledge in the form of structural and functional causal relationships. In essence, effectively integrating structural and functional causal knowledge drives the learning of relationships among observed variables by the neural network.

IV. CONCLUSION

In this paper, we develop a generic interface for encoding structural and relational causal knowledge among observed variables into neural network to guide its learning. In concert with the causal DAG-informed neural network architecture, CINN is devised to minimize the total loss over the variables in the intermediate and output layers to drive co-learning of causal relationships among observed variables. By exploiting the causal DAG as an inductive



(a) BH at Step 1

(b) BH at Step 2

(c) WQ at Step 1

(d) WQ at Step 2

Fig. 11. Evolution of causal graph. (a) BH Step 1: $\mathsf{V}_C = \{3, 4, 5, 7, 10\}$, $\mathsf{V}_O = \{6, 9, 11\}$, $\mathsf{V}_B = \{1, 2, 8, 12, 13\}$; (b) BH Step 2: $\mathsf{V}_C = \{3, 4, 5, 7, 8, 10, 12\}$, $\mathsf{V}_O = \{6, 9, 11\}$, $\mathsf{V}_B = \{1, 2, 13\}$; (c) WQ Step 1: $\mathsf{V}_C = \{0, 1, 2, 8, 9, 10\}$, $\mathsf{V}_O = \{6\}$, $\mathsf{V}_B = \{[3], [5], [11]\}$; (d) WQ Step 2: $\mathsf{V}_C = \{0, 1, 2, 8, 9, 10\}$, $\mathsf{V}_O = \{11\}$, $\mathsf{V}_B = \{[3], [5], [6]\}$;

bias to the learning of neural network, CINN exhibits a superior performance than other state-of-the-art alternatives. Through robustness checks and ablation study, we demonstrate that causal knowledge contributes to a significantly enhanced performance of neural network. The proposed research deepens our understanding on the nuanced role of causal knowledge in enhancing the neural network performance and highlights the importance of capturing underlying variable relationships when developing deep learning models. Importantly, the devised generic interface moves neural networks beyond pure data-driven learning paradigm by permitting to incorporate domain expertise in various forms in a flexible manner. Integrating domain knowledge into CINN complements the shortcoming of pure data-driven learning in the lack of contexts, thus leading to an augmentation in the prediction performance.

ACKNOWLEDGMENTS

The work described in this paper was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 25206422), the National Natural Science Foundation of China (Grant No. 62406269), the Research Committee of The Hong Kong Polytechnic University (Project code: RKB0, G-UARJ), the NSFC/Research Grants Council (RGC) Joint Research Scheme (Project No: N_HKBU214/21), Seed Funding for Collaborative Research Grants of HKBU (Grant No. RC-SFCRG/23-24/R2/SCI/10), and Guangdong and Hong Kong Universities "1+1+1" Cross-Campus Research Collaboration Scheme (Grant No. 2025A0505000004).

REFERENCES

[1] H. Pei, X.-S. Si, C. Hu, T. Li, C. He, and Z. Pang, "Bayesian deep-learning-based prognostic model for equipment without label data related to lifetime," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 504–517, 2022.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] Y. Yang, H. Fu, A. I. Aviles-Rivero, Z. Xing, and L. Zhu, "DiffMIC-v2: Medical image classification via improved diffusion network," *IEEE Transactions on Medical Imaging*, vol. 44, no. 5, pp. 2244–2255, 2025.

[4] F. Santoso and A. Finn, "A data-driven cyber–physical system using deep-learning convolutional neural networks: Study on false-data injection attacks in an unmanned ground vehicle under fault-tolerant conditions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 346–356, 2022.

[5] Y.-H. Lin and G.-H. Li, "Uncertainty-aware fault diagnosis under calibration," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 10, pp. 6469–6481, 2024.

[6] Y. Jiang, T. Xia, D. Wang, Y. Xu, R. Li, E. Pan, and L. Xi, "A spatiotemporal dynamic wavelet network for infrared thermography-based machine prognostics," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 3, pp. 1658–1665, 2023.

[7] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[8] Q. Li, X. Li, H. Jiang, and X. Qian, "A spatial-transformation-based causality-enhanced model for glioblastoma progression diagnosis," *IEEE Transactions on Artificial Intelligence*, vol. 6, no. 6, pp. 1529–1539, 2025.

[9] X. Zhang, T. Wang, L. Ma, and S. Mahadevan, "Reliability engineering, risk management, and trustworthiness assurance for AI systems," *Journal of Reliability Science and Engineering*, vol. 1, no. 2, p. 022001, 2025.

[10] J. Feng, A. Sondhi, J. Perry, and N. Simon, "Selective prediction-set models with coverage rate guarantees," *Biometrics*, vol. 79, pp. 811–825, 2023.

[11] X. Zhang, F. T. Chan, C. Yan, and I. Bose, "Towards risk-aware artificial intelligence and machine learning systems: An overview," *Decision Support Systems*, vol. 159, p. 113800, 2022.

[12] X. Zhang and I. Bose, "Reliability estimation for individual predictions in machine learning systems: A model reliability-based approach," *Decision Support Systems*, p. 114305, 2024.

[13] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in *Proceedings of the International Conference on Learning Representations*, 2017.

[14] Q. Peng, Y. Liu, Y. Jin, X.-G. Yang, R. Wang, and K. Liu, "Coating feature analysis and capacity prediction for digitalization of battery manufacturing: An interpretable AI solution," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 1, pp. 284–294, 2025.

[15] J. Li, K. Yue, Z. Chen, J. Xia, W. Li, and X. Zhang, "An uncertainty-aware continual learning framework for fault diagnosis of rotating machinery with homogeneous-heterogeneous faults," *IEEE Transactions on Automation Science and Engineering*, 2024.

[16] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[17] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang, "An investigation of why overparameterization exacerbates spurious correlations," in *International Conference on Machine Learning*, 2020, pp. 8346–8356.

[18] N. Wang, L. Qi, J. Guo, Y. Shi, and Y. Gao, "Learning generalizable models via disentangling spurious and enhancing potential correlations," *IEEE Transactions on Image Processing*, vol. 33, pp. 1627–1642, 2024.

[19] W. Xie, Q. Yu, W. Fang, X. Zhang, J. Geng, J. Tang, W. Jing, M. Liu, Z. Ma, J. Yang *et al.*, "Data-driven approaches linking wastewater and source estimation hazardous waste for environmental management," *Nature Communications*, vol. 15, no. 1, p. 5432, 2024.

[20] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[21] K. Zhang, I. Shpitser, S. Magliacane, D. Bacciu, F. Wu, C. Zhang, and P. Spirtes, "IEEE Transactions on Neural Networks and Learning Systems special issue on causal discovery and causality-inspired machine learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 4899–4901, 2024.

[22] Z.-X. Yang, G.-G. Wu, L. Song, and L.-P. Zhang, "Control change cause analysis-based Bayesian network modeling for system risk assessment," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 2958–2968, 2018.

[23] J. Tang, Z. Qi, E. Fang, and C. Shi, "Offline feature-based pricing under censored demand: A causal inference approach," *Manufacturing & Service Operations Management*, vol. 27, no. 2, pp. 535–553, 2025.

[24] X. Wang, T. Ban, L. Chen, D. Lyu, Q. Zhu, and H. Chen, "Large-scale hierarchical causal discovery via weak prior knowledge," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 5, pp. 2695–2711, 2025.

[25] M. Kuzmanovic, D. Frauen, T. Hatt, and S. Feuerriegel, "Causal machine learning for cost-effective allocation of development aid," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5283–5294.

[26] S. Feuerriegel, D. Frauen, V. Melnychuk, J. Schweisthal, K. Hess, A. Curth, S. Bauer, N. Kilbertus, I. S. Kohane, and M. van der Schaar, "Causal machine learning for predicting treatment outcomes," *Nature Medicine*, vol. 30, no. 4, pp. 958–968, 2024.

[27] J. Pearl, "The seven tools of causal inference, with reflections on machine learning," *Communications of the ACM*, vol. 62, no. 3, pp. 54–60, 2019.

[28] S. S. Kancheti, A. G. Reddy, V. N. Balasubramanian, and A. Sharma, "Matching learned causal effects of neural networks with domain priors," in *International Conference on Machine Learning*, 2022, pp. 10 676–10 696.

[29] T. Kyono, Y. Zhang, and M. van der Schaar, "CASTLE: regularization via auxiliary causal graph discovery," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1501–1512, 2020.

[30] T. Teshima and M. Sugiyama, "Incorporating causal graphical prior knowledge into predictive modeling via simple data augmentation," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 86–96.

[31] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters, "Invariant models for causal transfer learning," *Journal of Machine Learning Research*, vol. 19, no. 36, pp. 1–34, 2018.

[32] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij, "On causal and anticausal learning," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012, pp. 459–466.

[33] D. Lopez-Paz, R. Nishihara, S. Chintala, B. Scholkopf, and L. Bottou, "Discovering causal signals in images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6979–6987.

[34] P. Dong, X.-L. Wang, I. Bose, K. K. Ng, X. Zhang, and X. Zhang, "Causally aware spatiotemporal multigraph convolutional network for accurate and reliable traffic prediction," *INFORMS Journal on Computing*, 2025.

[35] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, "DAGs with NO TEARS: Continuous optimization for structure learning," *Advances in Neural Information Processing Systems*, vol. 32, pp. 9492–9503, 2018.

[36] X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. Xing, "Learning sparse nonparametric DAGs," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 3414–3425.

[37] J. Pearl *et al.*, "Causality: Models, reasoning and inference," *Cambridge, UK: Cambridge University Press*, vol. 19, no. 2, 2000.

[38] S. van de Geer and P. Bühlmann, "c0-penalized maximum likelihood for sparse directed acyclic graphs," *The Annals of Statistics*, vol. 41, no. 2, pp. 536–567, 2013.

[39] P.-L. Loh and P. Bühlmann, "High-dimensional learning of linear causal networks via inverse covariance estimation," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3065–3105, 2014.

[40] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[41] N. Kilbertus, M. Rojas Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schölkopf, "Avoiding discrimination through causal reasoning," *Advances in Neural Information Processing Systems*, vol. 30, pp. 656–666, 2017.

[42] F. Russo and F. Toni, "Causal discovery and knowledge injection for contestable neural networks," in *26th European Conference on Artificial Intelligence,*, vol. 372, 2023, pp. 2025–2032.

[43] S. Fan, X. Wang, Y. Mo, C. Shi, and J. Tang, "Debiasing graph neural networks via learning disentangled causal substructure," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 934–24 946, 2022.

[44] Z. Hu, Z. Zhao, X. Yi, T. Yao, L. Hong, Y. Sun, and E. Chi, "Improving multi-task generalization via regularizing spurious correlation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 11 450–11 466, 2022.

[45] D. Heckerman, "Bayesian networks for data mining," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 79–119, 1997.

[46] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick *et al.*, "Informed machine learning-A taxonomy and survey of integrating prior knowledge into learning systems," *IEEE Transactions on Knowledge & Data Engineering*, vol. 35, no. 1, pp. 614–633, 2021.

[47] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3614–3633, 2021.

[48] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020.

[49] A. Asuncion and D. Newman, "UCI machine learning repository," 2007, URL: https://archive.ics.uci.edu/.

[50] K. Zhang, J. Peters, D. Janzing, and B. Schölkopf, "Kernel-based conditional independence test and application in causal discovery," in *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, 2011, pp. 804–813.

[51] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[53] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*. PMLR, 2015, pp. 448–456.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[55] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[56] J. Pearl, "Causal inference in statistics: An overview," *Statistics Surveys*, vol. 3, pp. 96–146, 2009.
[57] W. Wei and L. Feng, "Nonlinear causal structure learning for mixed data," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 709–718.
[58] T. M. Kyono, *Towards Causally-Aware Machine Learning*. University of California, Los Angeles, 2021.

Xiaoge Zhang is an Assistant Professor in the Department of Industrial and Systems Engineering (ISE) at The Hong Kong Polytechnic University. He received his PhD in Systems Engineering and Operations Research from Vanderbilt University, Nashville, Tennessee, United States in 2019. He has won multiple awards, including Peter G. Hoadley Best Paper Award, Chinese Government Award for Outstanding Self-Financed Students Studying Abroad, Bravo Zulu Award, Pao Chung Chen Fellowship, among others. He has published more than 90 research papers in leading academic journals, such as Nature Communications, INFORMS Journal on Computing, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Intelligent Transportation Systems, Reliability Engineering & Systems Safety, Risk Analysis, IEEE Transactions on Industrial Informatics, IEEE Transactions on Artificial Intelligence, IEEE Transactions on Reliability, IEEE Transactions on Cybernetics, and Decision Support Systems, among others. His research has gathered widespread attention from the academic community (4700+ citation, h-index 35 according to Google Scholar). His research interests center on reliable AI, machine learning, reliability and trustworthiness of autonomous systems, and uncertainty quantification. He is a senior member of IEEE and a member of INFORMS and IISE.

Tao Wang received his Bachelor's degree in Automation and his Master's degree in Control Engineering from the School of Automation Science and Electrical Engineering at Beihang University, Beijing, China, in 2020 and 2023, respectively. He is currently pursuing the PhD degree in the Department of Industrial and Systems Engineering at The Hong Kong Polytechnic University, Hong Kong, China. His research interests include conformal prediction, uncertainty quantification, and industrial AI applications.

Xiao-Lin Wang is an associate professor in the Business School at Sichuan University, Chengdu, China. Prior to that, he was a research assistant professor in the Department of Logistics and Maritime Studies at The Hong Kong Polytechnic University, Hong Kong SAR. He received his PhD in industrial engineering from City University of Hong Kong in 2020, and his BS and MS degrees in industrial engineering from Southeast University, Nanjing, China, in 2013 and 2016, respectively. His research interest lies in applying data analytics, stochastic modeling, and optimization techniques to solve maintenance optimization, warranty analytics, and operations management problems. His research outcomes have appeared in IEEE Transactions on Reliability, IISE Transactions, INFORMS Journal on Computing, European Journal of Operational Research, Manufacturing & Service Operations Management, among others.

Feng-Lei Fan received the B.S. degree from Harbin Institute of Technology (HIT), Harbin, China, in 2017, and the Ph.D. degree from the Rensselaer Polytechnic Institute (RPI), Troy, NY, USA, in 2021. He is currently an Assistant Professor with the Department of Data Science, City University of Hong Kong (CityU), and also the Director with the Frontier of Artificial Network (FAN) Group. He was a Research Assistant Professor with the Department of Mathematics, The Chinese University of Hong Kong (CUHK). Before that, he spent one year as a Postdoctoral Researcher with Weill Cornell Medicine. He has authored 26 papers in top-tier venues, such as JMLR and IEEE Transactions on Pattern Analysis and Machine Intelligence. His primary research interests lie in deep learning theory and methodology, neuroscience, and medical image processing. He was a recipient of the OlympusMons Pioneer Award, the CVPR Best Paper Award Candidates, the IEEE TRPMS Best Paper Award from the IEEE Nuclear and Plasma Society, and the International Neural Network Society Doctoral Dissertation Award. He served as a PC Member for many conferences, such as International Joint Conference on Artificial Intelligence and Association for the Advancement of Artificial Intelligence.

Yiu-Ming Cheung received the PhD degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong in Hong Kong. He is a fellow of AAAS, IET, BCS, and AAIA. He is currently a chair professor (Artificial Intelligence) with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China. His research interests include machine learning, visual computing, data science, pattern recognition, multi-objective optimization, and information security. He is currently the editor-in-chief of IEEE Transactions on Emerging Topics in Computational Intelligence. Also, he serves as an associate editor for IEEE Transactions on Cybernetics, IEEE Transactions on Cognitive and Developmental Systems, IEEE Transactions on Neural Networks and Learning Systems (2014-2020), Pattern Recognition, Pattern Recognition Letters, and Neurocomputing, to name a few. For details, please refer to: https://www.comp.hkbu.edu.hk/ymc.

Indranil Bose is Distinguished Professor of Information Systems at NEOMA Business School. He holds a BTech from the Indian Institute of Technology, MS from the University of Iowa, and MS and PhD from Purdue University. His research interests are in business analytics, digital transformation, information security, and management of emerging technologies. His publications have appeared or are forthcoming in MIS Quarterly, INFORMS Journal on Computing, Journal of the MIS, ACM Computing Surveys, Communications of the ACM, Communications of the AIS, Computers and Operations Research, Decision Support Systems, European Journal of Operational Research, IEEE Transactions on Engineering Management, Information & Management, Information Systems Frontiers, International Journal of Production Economics, Journal of the American Society for Information Science and Technology, Technological Forecasting and Social Change, and Tourism Management. He has served as Senior Editor of Decision Support Systems, Journal of Organizational Computing and Electronic Commerce, and Pacific Asia Journal of the AIS; Co-ordinating Editor of Information Systems Frontiers, Associate Editor of Communications of the AIS, Information & Management, and Journal of the AIS; Editorial Board Member of Information Systems Research; and Guest Associate Editor of MIS Quarterly.