mathematics

Article

Hidden Variable Discovery Based on Regression and Entropy

Xingyu Liao and Xiaoping Liu *

Key Laboratory of Systems Health Science of Zhejiang Province, School of Life Science, Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou 310024, China; liaoxingyu22@mails.ucas.ac.cn
* Correspondence: xpliu@ucas.ac.cn

Abstract: Inferring causality from observed data is crucial in many scientific fields, but this process is often hindered by incomplete data. The incomplete data can lead to mistakes in understanding how variables affect each other, especially when some influencing factors are not directly observed. To tackle this problem, we've developed a new algorithm called Regression Loss-increased with Causal Intensity (RLCI). This approach uses regression and entropy analysis to uncover hidden variables. Through tests on various real-world datasets, RLCI has been proven to be effective. It can help spot hidden factors that may affect the relationship between variables and determine the direction of causal relationships.

Keywords: causal discovery; hidden variable; structure learning

MSC: 62D20

1. Introduction

The abundance of data in the natural sciences provides vast opportunities for scientific discovery. Understanding and extracting the underlying mechanisms from these data is a crucial task across many disciplines. While identifying correlations can show how variables move together, understanding cause-and-effect relationships can reveal how a change in one variable can directly influence another. In biology, for instance, discovering how genes interact and influence each other is key to understanding biological processes and can lead to breakthroughs in how we view different biological states or conditions. By mapping out these causal relationships among genes, we can create gene regulatory networks, shedding light on the basic regulatory principles of life. Causal discovery is equally vital in the social sciences, where it helps us understand the complex dynamics of human behavior, social interactions, and social phenomena.

Many works have been developed to uncover causal relationships between variables rather than mere correlations. While randomized controlled experiments are considered the most reliable method for determining causality, they come with significant challenges. These challenges include high costs, technical limitations, and ethical considerations. Beyond experimental approaches, there are several established algorithms for causal discovery. These methods fall into two main categories: analyzing time-series observed data and focusing on non-time-series observed data. Causal discovery methods based on time-series observed data, such as Granger causality [1], utilize time-lagged or time-series data to infer causality, making them suitable for datasets where the order of events is crucial. For non-time-series data, where the order of events is either unknown or irrelevant, causal discovery methods do not account for the timing of data. These methods are further divided into three main types: constraint-based algorithms, score-search-based algorithms, and functional-causal-model-based algorithms. The basic algorithms based on constraint methods are the Inductive Causation (IC) algorithm [2], SGS algorithm [3], and Peter–Clark (PC) algorithm [4]. These algorithms comprise two main steps: In the first step, edge

check for updates

Citation: Liao, X.; Liu, X. Hidden Variable Discovery Based on Regression and Entropy. *Mathematics* **2024**, *12*, 1375. https://doi.org/10.3390/math12091375

Academic Editor: Chin-Shang Li

Received: 25 March 2024
Revised: 22 April 2024
Accepted: 29 April 2024
Published: 30 April 2024

removal is performed on the fully connected network based on statistical methods such as independence hypothesis testing, and an undirected network is obtained. The second step involves identifying unique V-structures and applying directional rules to certain edges. Algorithms based on a scoring search include a DAG space approximation search represented by K2 algorithm [5] and a greedy search such as the hill climbing method [6], and an equivalence class space approximation search represented by Greedy Equivalence Search (GES) [7]. The core of the scoring search class of algorithms consists of two parts, search strategy and scoring function: the search strategy is used to decide how to choose the search path in the search space of the graph structure, and the scoring function is used to evaluate whether to keep the edge or delete the edge. The methods based on functional causal modeling are based on the Structural Equation Model (SEM) [8]. Representative algorithms based on functional causal modeling include the Linear Non-Gaussian Acyclic Model (LiNGAM) [9], Post-NonLinear (PNL) method [10,11] and Additive Noise Model (ANM) [12,13].

However, most of the existing algorithms only consider the causality between observed variables and do not consider the presence of non-observed or hidden variables. The hidden variable is defined as a variable that is not directly detected in the current dataset. By identifying the hidden variables and corresponding causality from the observed dataset, we can obtain deeper insights into the understanding of causal relationships between observed variables. In biology, identifying the potential hidden variables can help to eliminate unobserved genetic influences that may confound observed gene expression data. Accurately identifying and modeling the hidden variables enables researchers to determine the correct regulatory relationships between observed genes, shedding light on the intricate mechanisms driving gene expression and biological processes. In healthcare, hidden variables include unmeasured patient characteristics, genetic factors, or environmental exposures that influence disease progression or treatment response. Identifying these hidden variables can enhance personalized medicine approaches, optimize treatment strategies, and improve patient outcomes. Therefore, hidden variable discovery algorithms can make invaluable contributions in biology and other areas.

In this paper, we proposed a novel method called Regression Loss-increased with Causal Intensity (RLCI), which combines the constraint-based method with the functional-causal-model-based approach to detect potential hidden variables from the observed data. RLCI establishes the basic framework of the network through independence tests and linear regression, determines the presence of hidden variables through causal intensity, and, ultimately, reconstructs the causal structure of observed variables affected by potential confounders. The efficacy of RLCI has been validated through numerous causal simulation experiments and comprehensive comparisons with a wide range of existing classical methods, demonstrating significant improvements in a variety of real systems. For instance, within the BEELINE dataset [14], RLCI successfully identified the concealed gene Nkx22. Similarly, in examining the food chain data [15], RLCI revealed that the populations of rotifers, calanoids, and picophytoplankton are influenced by an undisclosed variable, and a further literature review confirmed the cyclopoids' predation on these three organisms. RLCI stands out as a common method for discovering hidden variables within any observational dataset, capable of elucidating causal relationships between variables and accurately reconstructing the true causal network structure.

2. Methods

The algorithm process of RLCI is depicted in Figure 1 and consists of four stages. It begins with a fully connected network. The initial stage employs independence testing to filter this network into a correlation network, dropping the uncorrelated edges. In the next stage, we refine this correlation network into a pseudo-causal network by eliminating edges that represent indirect causality (e.g., edge 'a' in Figure 1) based on changes in regression loss. This refined network preserves the direct relationships between nodes without considering the effects of hidden variables, but does not explicitly state the direction of these causal relationships, henceforth termed "pseudo-causal network". The third stage introduces a measure of causal intensity to detect spurious edges generated by the influence of hidden variables (e.g., edge 'b' in Figure 1), resulting in a mixed network. In this network, hidden variables are assigned directional edges towards the observed variables, but the observed variables themselves remain directionless among each other. The final stage aims to pinpoint the direction of causality between each pair of directly connected observed variables. This stage completes the process, allowing us to accurately reconstruct the entire causal network. Overall, the RLCI algorithm employs a step-by-step approach to differentiate the types of edges and uncover hidden variables within the causal net-work.

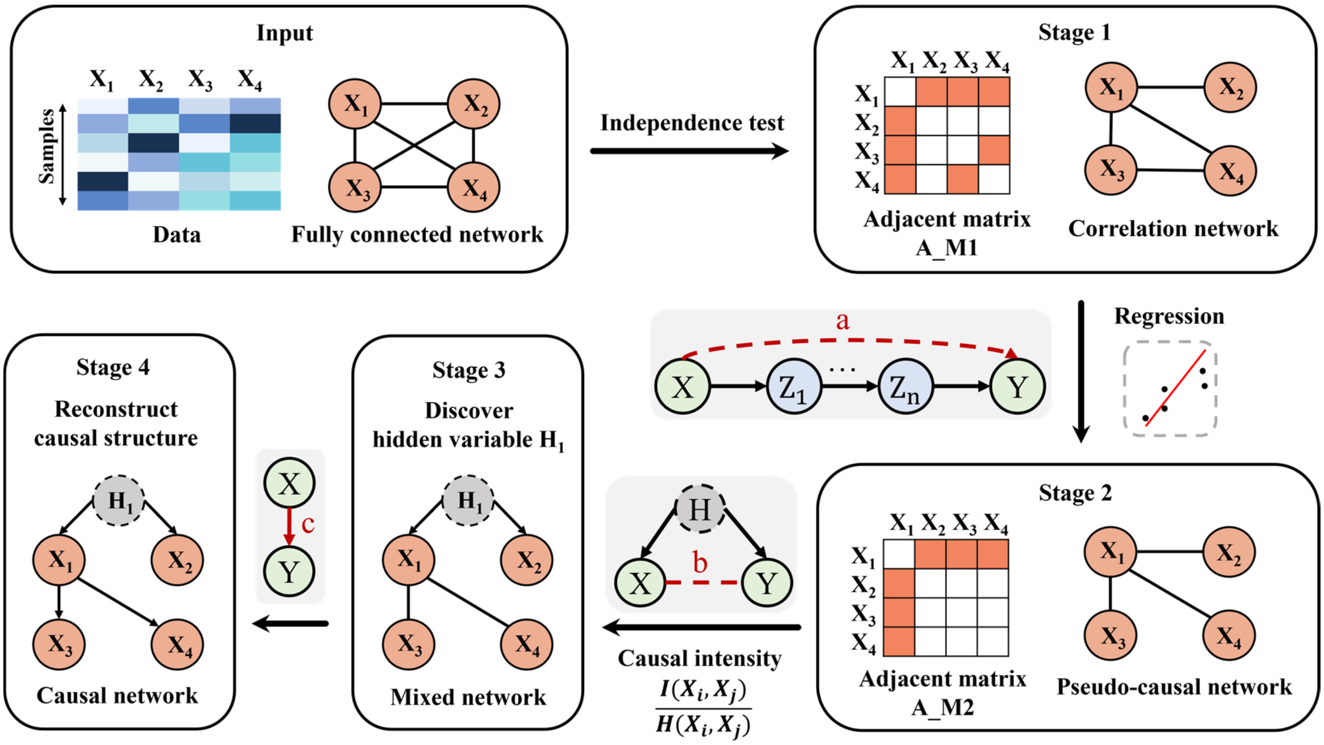

Figure 1. Algorithm procedure. Input: Starting from a fully connected network, each node represents each observed variable (Table 1). Stage 1: The independence test removes edges between independent or weakly correlated nodes to obtain the correlation network. Stage 2: By removing the edges of indirect causal relationships (such as edge 'a': X and Y are indirect causality) through the loss difference of regression fitting, a pseudo-causal network is obtained. Stage 3: Defines the causal intensity indicator to discover the edges of spurious causal relationships caused by hidden variables (such as edge 'b': X is not causally related to Y but X and Y are affected by the hidden variable H (Table 1)), forming a mixed network. Stage 4: Determine the causal direction between two connected variables and reconstruct the final causal network (such as edge 'c': X is directly causal to Y).

Table 1. Main notations used in this article.

Notation	Description
X	Observed variable
H	Hidden variable or reconstructed variable
$\hat{\mu}$	Sample mean
S	Sample standard deviation
$\hat{e}$	Mean squared error
A_M1, A_M2	Adjacency matrix
Adj(X_i)	Variables adjacent to X_i
Adj(X_i)\{X_j}	Variables adjacent to X_i except for X_j
H(X_i)	Entropy of X_i

Table 1. *Cont.*

Notation	Description
$H(X_i, X_j)$	Joint entropy of X_i and X_j
$I(X_i, X_j)$	Mutual information of X_i and X_j
$P(X)$	Observed distribution of X
V_X	Reference distribution of X
$D(P(X)\|V_X)$	KL divergence between the observed and reference distributions of X
$X \to Y$	X is a cause of Y

2.1. Stage 1: Independence Test

In the first stage of the RLCI algorithm, edges caused by indirect causality are identified and removed by performing conditional independence tests between nodes. We used the Fisher Z-test [16] to conduct the conditional independence test (Supplementary Material S1). We start with a fully connected network and apply the conditional independence test (Fisher Z-test) for edge deletion [17]. We first determine the neighboring nodes of each variable. One variable is selected as the conditional variable, and then each pair of connected nodes is traversed to perform the independence test, and, if the nodes are independent from each other, the edges between them are deleted. Subsequently, the independence test is performed again by increasing the number of condition variables and updating the neighboring node information. This process continues, adding one condition variable per round, until the number of condition variables is greater than or equal to the number of neighboring nodes of any node in the network. It is worth noting that, in order to avoid the order of deleting edges affecting the final results, we uniformly delete edges after each round of independence test. Refer to Algorithm 1 for the order in removing the edges. The algorithm flow in Stage 1 is as follows:

Algorithm 1: Independence test

```
Assume that there are K observed variables $X = \{X_1, X_2 ... X_K\}$, N samples.
Input: The dataset matrix of size $N \times K$.
Output: Correlation network C, adjacency matrix A_M1 of size $K \times K$.
1:  Generate an all-one matrix A_M1 of size $K \times K$
2:  From the fully connected undirected network C on the vertex set X
3:  L = 0
4:  Repeat
5:    L = L + 1
6:    For all vertices $X_i$ in C do
7:      Let $A(X_i) = Adj(C, X_i)$
8:    End for
9:    For each neighboring node pair $(X_i, X_j)$ do
10:     For $A(X_i)\backslash\{X_j\}$ all subsets S in which the number of nodes is L do
11:       If $X_i$ and $X_j$ are conditionally independent given S then
12:         Delete edge '$X_i - X_j$' from C, Let $A\_M1_{ij} = 0$, $A\_M1_{ji} = 0$
13:         Break
14:       End if
15:     End for
16:   End for
17: until all pairs of adjacent vertices $(X_i, X_j)$ in C satisfy $|A(X_i)\backslash\{X_j\}| \leq L$
18: return C, A_M1
```

Note: $Adj(C, X_i)$ represents the set of neighbor nodes of X_i in network C. $A(X_i)\backslash\{X_j\}$ represents the set after removing X_j from the set $A(X_i)$.

In Stage 1, we can obtain an adjacency matrix of the correlation network, denoted as A_M1 (Figure 1). $A\_M1_{ij} = 0$ means that there is no edge between variables X_i and X_j, and $A\_M1_{ij} = 1$ means that there is an edge connecting the two variables X_i and X_j.

2.2. Stage 2: Linear Regression

The test of independence can only delete the independent edges from the fully connected network, but it cannot remove the edges with indirect causality, whereas the loss-increased linear regression [18] can be further used to remove the indirect edges, e.g., the dashed edge ‘a’ as in Figure 1.

The specific steps of the second stage in the algorithm are as follows:

Step 1: The data are first pre-processed and standardized for each observed variable X:

$$\mathrm{X}_{\mathrm{std}} = \frac{X - \hat{\mu}}{\mathrm{S}} \tag{1}$$

where X is the original data, $\mathrm{X}_{\mathrm{std}}$ is the normalized value, $\hat{\mu}$ is the mean of the original data set, and S is the standard deviation of the original data set (Table 1).

Step 2: From the adjacency matrix A_M1 generated in the first stage, we can obtain the set of neighboring nodes for each variable. Assuming that $\mathrm{X_j}$ is one of the neighbouring nodes of $\mathrm{X_i}$, we analyze the effect of $\mathrm{X_j}$ on $\mathrm{X_i}$ as follows:

$$\mathrm{X_i} = f_1(\mathrm{Adj}(\mathrm{X_i})) + \hat{e_1} \tag{2}$$

$$\mathrm{X_i} = f_2\big(\mathrm{Adj}(\mathrm{X_i})\backslash\{\mathrm{X_j}\}\big) + \hat{e_2} \tag{3}$$

$$\Delta\mathrm{Loss}_{\mathrm{X_j}\to\mathrm{X_i}} = \hat{e_1} - \hat{e_2} \tag{4}$$

where f_1 and f_2 represent linear functions fitted using the least squares method, while $\hat{e_1}$ and $\hat{e_2}$ are their respective mean squared errors (Table 1). $\mathrm{Adj}(\mathrm{X_i})$ denotes the set of neighbor nodes of $\mathrm{X_i}$, whereas $\mathrm{Adj}(\mathrm{X_i})\backslash\{\mathrm{X_j}\}$ denotes the set obtained after removing $\mathrm{X_j}$ from the set $\mathrm{Adj}(\mathrm{X_i})$ (Table 1). $\Delta\mathrm{Loss}_{\mathrm{X_j}\to\mathrm{X_i}}$ can be used as a measure of the impact of $\mathrm{X_j}$ on the $\mathrm{X_i}$.

Train the linear regression f_1 by the $\mathrm{Adj}(\mathrm{X_i})$, obtain the mean squared error $\hat{e_1}$, and then remove $\mathrm{X_j}$ from the set of neighboring nodes of $\mathrm{X_i}$. Train the linear regression f_2 by the $\mathrm{Adj}(\mathrm{X_i})\backslash\{\mathrm{X_j}\}$, obtain the mean squared error $\hat{e_2}$, and $\hat{e_2} - \hat{e_1}$ is the loss-increased: ΔLoss of $\mathrm{X_j}$ concerning $\mathrm{X_i}$, denoted as $\Delta\mathrm{Loss}_{\mathrm{X_j}\to\mathrm{X_i}}$. The significance of $\Delta\mathrm{Loss}_{\mathrm{X_j}\to\mathrm{X_i}}$ can be interpreted as the effect on the regression of $\mathrm{X_i}$ before and after the removal of $\mathrm{X_j}$. Alternatively, it can be thought of as a measure of the extent of the effect of $\mathrm{X_j}$ on $\mathrm{X_i}$. Similarly, when we take $\mathrm{X_j}$ as the dependent variable, we can obtain $\Delta\mathrm{Loss}_{\mathrm{X_i}\to\mathrm{X_j}}$ in the same way. We obtain the Δloss values accordingly by iterating all the neighboring nodes by the above method based on the adjacency matrix A_M1 generated in the first stage. We consider that, when the regression model is sufficiently convergent, ΔLoss is constant $\geq$0. When $\Delta\mathrm{Loss}_{\mathrm{X_j}\to\mathrm{X_i}}$ and $\Delta\mathrm{Loss}_{\mathrm{X_i}\to\mathrm{X_j}}$ all tend to 0, it is assumed that $\mathrm{X_i}$ and $\mathrm{X_j}$ are indirectly causally related (Supplementary Material S2).

In practice, if there is a Δloss tending to 0, indirect causality may not have been completely removed in Stage 1. Therefore, we normalize each ΔLoss to map the range of values to the interval [0, 1]:

$$\Delta\mathrm{Loss}_{\mathrm{norm}} = \frac{\Delta\mathrm{Loss} - \Delta\mathrm{Loss}_{\mathrm{min}}}{\Delta\mathrm{Loss}_{\mathrm{max}} - \Delta\mathrm{Loss}_{\mathrm{min}}} \tag{5}$$

where $\Delta\mathrm{Loss}_{\mathrm{max}}$ is the maximum of all ΔLoss and $\Delta\mathrm{Loss}_{\mathrm{min}}$ is the minimum of all ΔLoss.

It is easy to find that the Δloss of indirect causality is significantly smaller than the Δloss of direct causality and the Δloss of pseudo-causality under the influence of hidden variables, so a threshold t can be set after normalization (Supplementary Material S3):

When $\Delta\mathrm{Loss}_{\mathrm{X_j}\to\mathrm{X_i}} < \mathrm{t}$ and $\Delta\mathrm{Loss}_{\mathrm{X_i}\to\mathrm{X_j}} < \mathrm{t}$, it is considered as an indirect causality that is not detected in the first stage. The edge ‘$\mathrm{X_i} - \mathrm{X_j}$’ is deleted from the adjacency matrix A_M1;

While $\Delta\mathrm{Loss}_{\mathrm{X_j}\to\mathrm{X_i}} \geq \mathrm{t}$ or $\Delta\mathrm{Loss}_{\mathrm{X_i}\to\mathrm{X_j}} \geq \mathrm{t}$, it is considered to be a direct causality or a false causal relationship affected by hidden variables, and the edge ‘$\mathrm{X_i} - \mathrm{X_j}$’ is still retained.

In Stage 2, a new adjacency matrix A_M2 is obtained. Node $\mathrm{X_i}$ and node $\mathrm{X_j}$ are considered to be connected by an edge if $\mathrm{A\_M2_{ij}} = 1$ or $\mathrm{A\_M2_{ji}} = 1$. Finally, after further correction, we successfully eliminated edges of indirect causality, like edge 'a' shown in Figure 1 (Supplementary Materials S2 and S3).

Refer to Algorithm 2 for the detailed steps involved in calculating the regression loss during Stage 2.

Algorithm 2: Linear regression

Assume that there are K observed variables $X = \{X_1, X_2 ... X_K\}$, N samples.
Input: The dataset matrix of size N × K, The K × K adjacency matrix A_M1.
Output: The adjacency matrix A_M2 of size K × K, the loss-increased matrix L_M of size K × K.

```
generate K×K matrix L_M and A_M2 with all zeros
For each $X_i$ in node list $X = \{X_1, X_2 ... X_K\}$ do
    Adj($X_i$) is obtained according to the matrix A_M1
    Linear fit: $X_i = f_1(\mathrm{Adj}(X_i)) + \hat{e_1}$ Obtain the mean squared error $\hat{e_1}$
    For each $X_j$ in Adj($X_i$) do
        Linear fit: $X_i = f_2(\mathrm{Adj}(X_i) \backslash \{X_j\}) + \hat{e_2}$ Obtain the mean squared error $\hat{e_2}$
        $\Delta \mathrm{Loss}_{X_j \to X_i} = \hat{e_1} - \hat{e_2}$
        $L\_M_{ij} = \Delta \mathrm{Loss}_{X_j \to X_i}$
    End for
End for
Set a threshold t
For each $L\_M_{ij}$ that is not equal to 0 do
    If $L\_M_{ij} \geq t$ then
        $A\_M2_{ij} = 1$
    End if
End for
return L_M, A_M2
```

2.3. Stage 3: Causal Intensity

After Stage 2, we eliminate the indirect causal impacts, and then we only need to distinguish between the edges of direct causality and the spurious edges generated by the influence of hidden variables. We defined a causal intensity indicator and found that the causal intensity of two points with direct causality is significantly greater than the causal intensity of two points that are jointly affected by the hidden variables and do not have direct causality. The causal intensity [19] is denoted as:

$$\mathrm{CI}(X_i, X_j) = \frac{I(X_i, X_j)}{H(X_i, X_j)} = \frac{H(X_i) + H(X_j) - H(X_i, X_j)}{H(X_i, X_j)} \tag{6}$$

where $I(X_i, X_j)$ is the mutual information [20] and $H(X_i, X_j)$ is the joint entropy (Supplementary Material S4).

Mutual Information is a statistical measure of correlation and dependence between two random variables. It measures the information gained about one random variable when we know the value of the other. When the mutual information is zero, it indicates that the variables X and Y are independent of each other; i.e., knowing the value of one variable does not help in predicting the value of the other variable. When the value of mutual information is large, it means that there is a strong correlation between the variables X and Y. Knowing the value of one variable provides more information about the other variable. A larger value of mutual information indicates a stronger correlation between the two variables. Joint Entropy (JE) is a concept in information theory used to measure the uncertainty or amount of information between multiple random variables. It is the entropy of the joint probability distribution of multiple random variables when these variables are known. Joint entropy can be used to characterize the overall uncertainty or amount of information between X and Y. If there is some dependence between X and Y, then their joint entropy will be less than the sum of the entropies when each is independent. It indicates that observing one variable can result in some speculation about the value of the other. If X and Y are independent of each other, then their joint entropy will be equal to the sum of the entropies when each is independent, indicating that there is no correlation between the two variables.

We define "causal intensity" as the ratio of mutual information to joint entropy, offering a metric to gauge the strength of correlations and dependencies between variables. For instance, in Figure 1, the causal intensity for edge 'c' stems from a direct causal link between the variables. Conversely, edge 'b' represents a spurious causality, arising because a hidden variable, labeled H, influences both variables X and Y (Figure 1). To ensure comparability, we normalized the data for each variable. The causal intensity indicator shows, both in theory and through empirical evidence, that the causal intensity for a direct causality like edge 'c' in Figure 1 is significantly higher than that for a spurious edge like edge 'b'. This difference is crucial for detecting the influence of hidden variables (Supplementary Material S4).

2.4. Stage 4: Reconstruction Causal Network

After the above stages, identifying the causal direction can be specific to each pair of variables that have a direct causality. We utilized the Information Geometric Causal Inference (IGCI) model [21], an entropy-based causal inference algorithm, to determine the causal direction between pairs of observed variables. This model infers causality by comparing the Kullback–Leibler (KL) divergence between a reference measure and the distribution $\mathrm{P(X)}$ against the KL divergence between the same reference measure and $\mathrm{P(Y)}$. Specifically, the reference distributions $\mathrm{V_X}$ and $\mathrm{V_Y}$ for variables X and Y are assumed to be Gaussian, reflecting the characteristics of the data (Table 1).

The KL divergence from the observation distribution to the reference distribution for X, denoted as $\mathrm{D(P(X)||V_X)}$, is calculated as follows:

$$\mathrm{D(P(X)||V_X)} = \int \log\frac{\mathrm{P(X)}}{\mathrm{V_X}}\mathrm{P(X)dx} \tag{7}$$

where $\mathrm{P(X)}$ is the true distribution of the variable X, $\mathrm{V_X}$ is the reference distribution of the variable X, and $\mathrm{D(P(X)||V_X)}$ represents the KL divergence between the observed distribution $\mathrm{P(X)}$ and the reference distribution $\mathrm{V_X}$. Similarly, the KL divergence for Y, $\mathrm{D(P(Y)||V_Y)}$, is computed using the same approach (Table 1).

The decision criterion for causality is defined by the equation:

$$\mathrm{V_{X\rightarrow Y}} = \mathrm{D(P(X)||V_X)} - \mathrm{D(P(Y)||V_Y)} \tag{8}$$

If $\mathrm{V_{X\rightarrow Y}} < 0$, it indicates a causal direction of $X \rightarrow Y$ (Table 1). Conversely, if $\mathrm{V_{X\rightarrow Y}} > 0$, the causal direction is determined as $Y \rightarrow X$.

3. Results

3.1. Data

In this work, we compared the analysis of our method with eight classical structure learning algorithms: PC, LiNGAM, DirectLiNGAM [22], FCI [23], GFCI [24,25], RFCI [26], RCD [27], and CAM-UV [28]. Among these algorithms, PC, LiNGAM, DirectLiNGAM, and RCD are linear methods, while the rest of the methods are nonlinear. FCI, GFCI, RFCI, RCD, and CAM-UV consider hidden variables compared with the rest of the algorithms. For the GFCI and RFCI algorithms, we utilized the code available at https://github.com/cmu-phil/tetrad (accessed on 30 April 2024) with default parameters. The remaining algorithms utilized code from https://causal-learn.readthedocs.io/en/latest/getting_started.html (accessed on 30 April 2024).

In total, we tested three simulated datasets (Sim1, Sim6, and DREAM4) and three real datasets (BEELINE, food chain dataset, and TCGA dataset). In addition, since the Sim1 and Sim6 datasets are large and impractical for some algorithms, we randomly selected 1000 samples in the Sim1 and Sim6 datasets, respectively, and tested them again.

To evaluate the performance of these algorithms, we employed precision (Equation (9)), recall (Equation (10)), and the comprehensive evaluation metric F1 score (Equation (11)) as evaluation metrics.

$$\text{precision} = \frac{\{\text{discovered causalities}\} \cap \{\text{actual causalities}\}}{\{\text{discovered causalities}\}} \tag{9}$$

$$\text{recall} = \frac{\{\text{discovered causalities}\} \cap \{\text{actual causalities}\}}{\{\text{actual causalities}\}} \tag{10}$$

$$\text{F1 score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{11}$$

3.2. The Performance of RLCI on Simulation Data

We evaluated our proposed method on Sim1 and Sim6 datasets from FMRI [29] (https://www.fmrib.ox.ac.uk/datasets/netsim/, accessed on 30 April 2024).

3.2.1. Sim1

The Sim1 dataset comprises 10,000 samples for five variables and the true causal relations are shown in Figure 2. We assume that X_1 is the hidden variable, and the remaining four variables are observed variables (Supplementary Material S5). Then, we can obtain a new dataset with only four observed variables, and the network was reconstructed by RLCI and other existing methods for the new dataset. From the reconstruction results, we can see that only RLCI can predict the hidden variable from the observed data and other methods cannot test the hidden variable from the new dataset (Figure 2a). Meanwhile, there are five directed edges in the real network, and RLCI accurately predicts four of them, and the other one is correctly predicted in the position but opposite in the direction, so the precision and recall of RLCI are both 80% in the real network, while all other methods are below 60% (Supplementary Table S1). There is no exact edge to be predicted from the new dataset by RCD and CAM-UV (Figure 2a). Then, we randomly choose 1000 samples from the new dataset and reconstruct the network based on RLCI and other methods (Figure 2b). The RLCI can also identify the hidden variable (Figure 2b), and infer the causal network with a precision of 80% (Supplementary Table S2). The best result among the other algorithms is CAM-UV with a precision of 75% (Supplementary Table S2).

3.2.2. Sim6

The Sim6 dataset contains 60,000 samples of 10 variables. We hid variables X_1 and X_6, and the remaining variables were used as observation variables to form a new dataset with only eight variables (Supplementary Material S5). By reconstructing the causal network by every method on the new dataset, the RLCI can identify all the two hidden variables, while the FCI and GFCI algorithms found only one hidden variable in the remaining methods (Figure 3a). In addition, RLCI recognized nine edges in the real network, and the other two edges were incorrectly predicted in terms of direction but correctly judged in terms of position in the real work. Therefore, RLCI outperforms the other algorithms in all evaluation metrics, with an accuracy and recall of 81.8%, much higher than the rest of the algorithms (Supplementary Table S3). Among other methods, GFCI is the highest at 45.5% in precision, and LiNGAM and DirectLiNGAM have the highest recall at 63.6% (Supplementary Table S3). Although LiNGAM and DirectLiNGAM have identified more correct edges, they have also generated many erroneous edges, resulting in significant false positives (Figure 3a). The comprehensive F1 score of other algorithms is below 47.7%, far lower than RLCI (Supplementary Table S3). Similarly, we randomly selected 1000 samples from the new dataset with variables X_1 and X_6 removed and conducted further testing (Figure 3b). The F1 score of RLCI is still the highest at 72.7% (Supplementary Table S4), and RLCI can still accurately identify two hidden variables, while only RFCI can identify one hidden variable among other algorithms (Figure 3b).

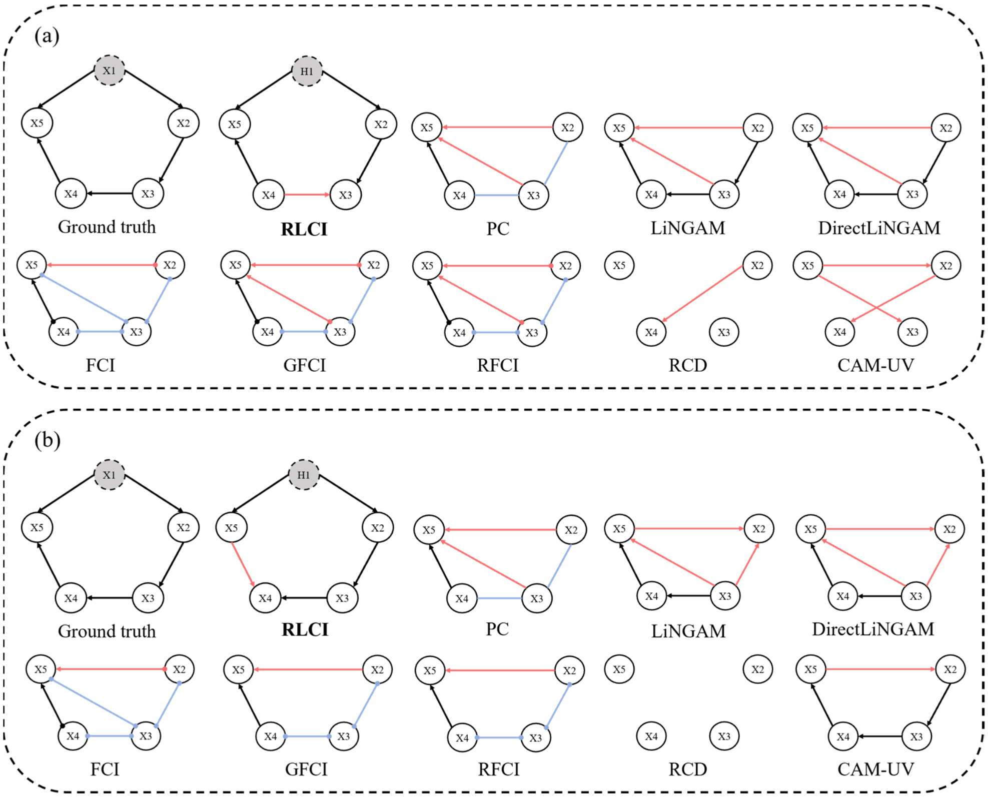

Figure 2. Results of reconstructing causal networks on Sim1. (**a**) Reconstructing causal networks on Sim1. Node X_1 is hidden and treated as a hidden variable, H_1 is the reconstructed variable, and the remaining nodes are observed variables. Black lines indicate correctly predicted edges, while red lines indicate incorrectly predicted edges. The blue line indicates a predicted relationship between two nodes with an unknown causal direction, which is not considered in our analysis. (**b**) Reconstructing causal networks on Sim1 of 1000 samples. The interpretations of different nodes and edges in (**b**) are consistent with those in (**a**).

3.2.3. DREAM4

The DREAM challenges [30,31] are widely regarded as a benchmark dataset for causal inference. In our experiment, we selected a dataset with 10 genes from DREAM4, and the regulatory relationships between genes are shown in Figure 4. We assumed that gene G_1 is a hidden variable and the remaining genes are observed variables; then, we removed G_1 to obtain a new dataset (Supplementary Material S5). The complex regulatory relationships between genes in DREAM4 (represented by bidirectional arrows) pose a challenge for reconstructing the causal network. Despite these challenges, our method stands out by uniquely identifying the hidden variable (Figure 4b). In addition, RLCI scored the highest in precision, recall, and F1 score (Supplementary Table S5). RLCI achieves a 50% accuracy and the rest of the methods are all under 45.5% (Supplementary Table S5). In terms of recall, RLCI and DirectLiNGAM are the highest at 33.3% (Supplementary Table S5). In terms of the composite metrics F1 score, RLCI reaches 40%, while FCI is the highest among the rest of the methods at 38.5% (Supplementary Table S5).

3.3. Performance on BEELINE_VSC

We used the BEELINE-VSC dataset, which consists of 2000 samples and eight genes, with the gene Nkx22 deliberately obscured as a hidden variable (Figure 5a). We assumed that each observed variable is influenced by no more than one hidden variable. Based on this premise, we infer a hidden variable labeled H_1 that simultaneously affects three specific genes: Pax6, Olig2, and Irx3 (Figure 5b and Supplementary Material S5). Except for the RCD and CAM-UV algorithms, which did not predict the results, the rest of the algorithms including RLCI incorrectly assumed that there is a regulatory relationship between the genes Dbx1, Dbx2, and genes Nkx62, Dbx2, respectively (Figure 5). However, RLCI correctly judged that there is no regulatory relationship between Dbx2 and Irx3 (Figure 5b), and the rest of the algorithms misjudged (Figure 5c–j). Due to the complexity of the regulatory relationships between the genes in BEELINE_VSC, many gene-regulated relationships were not identified, and the overall accuracy of the causal networks reconstructed by RLCI and other methods was relatively low (Supplementary Table S5). However, compared with the benchmark network, the precision of RLCI reconstruction can still reach 66.7%, while the precision of other algorithms is below 33.3% (Supplementary Table S5).

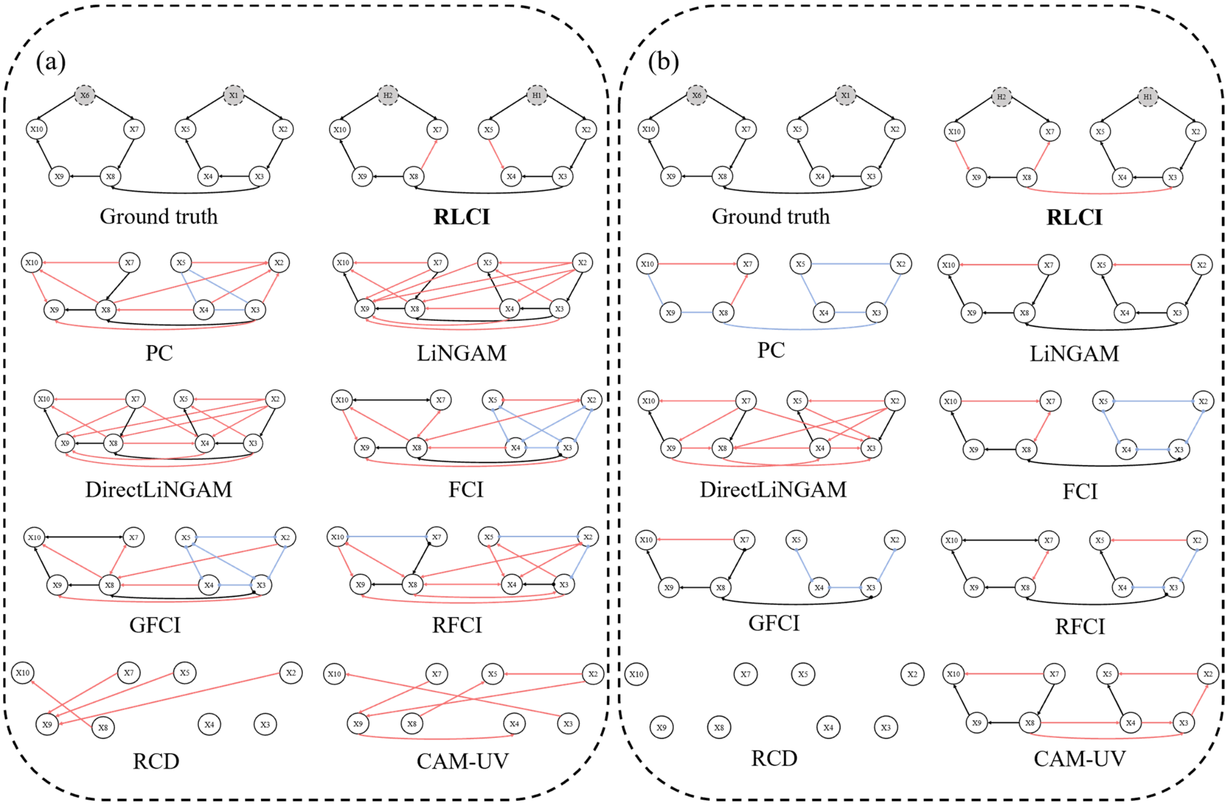

Figure 3. Results of reconstructing causal networks on Sim6. (**a**) Reconstructing causal networks on Sim6. Nodes X_1 and X_6 are hidden and treated as the hidden variables, H_1 and H_2 are reconstructed variables, and the remaining nodes remain as observed variables. Black lines indicate correctly predicted edges and red lines indicate incorrectly predicted edges. The blue line indicates a predicted relationship between two nodes with an unknown causal direction, which is not considered in our analysis. (**b**) Reconstructing causal networks on Sim6 of 1000 samples. The interpretations of different nodes and edges in (**b**) are consistent with those in (**a**).

3.4. *Performance in the Food Chain*

The food chain dataset contains time-series data on the abundance of plankton species isolated from the Baltic Sea. The food web was sampled and observed twice a week for over 2300 days. The network constructed from this dataset involves four planktonic species: rotifers, calanoids, picophytoplankton, and nanophytoplankton (Figure 6a).

During our analysis, we observed a low causal intensity between rotifers and picophytoplankton, as well as between picophytoplankton and calanoids (Supplementary Material S5). Therefore, we hypothesized the existence of a hidden variable H_1, that simultaneously affects rotifers, picophytoplankton, and calanoids (Figure 6b). Since cyclopoids can prey on rotifers [32], we hypothesized that cyclopoids might serve as the hidden variable H_1 (Figure 6b).

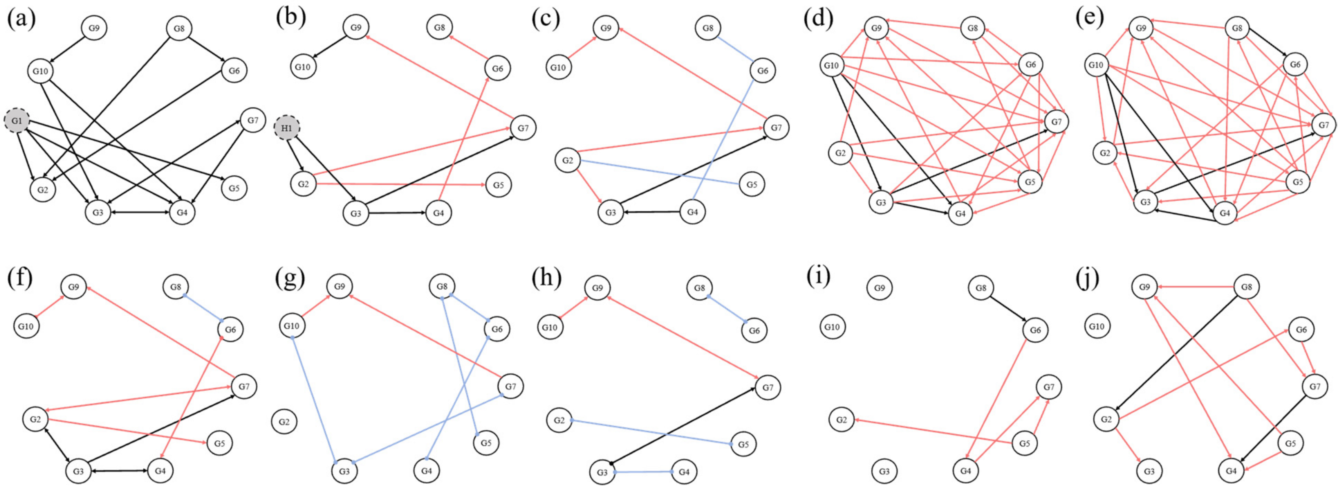

Figure 4. Results of reconstructing causal networks on DREAM4. (**a**) Ground truth. Node G_1 is hidden and treated as a hidden variable and the remaining nodes are observed variables. It is worth noting that DREAM4 is the simulated data of gene regulation, so there are bidirectional edges in the real network representing two genes regulating each other. (**b**) Result of RLCI. H_1 is the reconstructed variable. Black lines represent correctly predicted edges and red lines represent incorrectly predicted edges. The blue line indicates a predicted relationship between two nodes with an unknown causal direction, which is not considered in our analysis. (**c**) Result of PC. (**d**) Result of LiNGAM. (**e**) Result of DirectLiNGAM. (**f**) Result of FCI. (**g**) Result of GFCI. (**h**) Result of RFCI. (**i**) Result of RCD. (**j**) Result of CAM-UV.

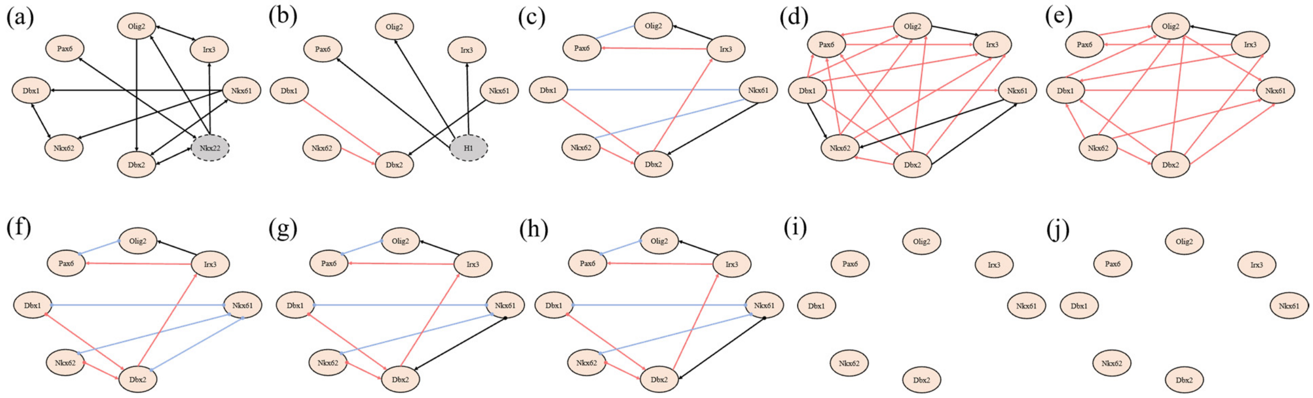

Figure 5. Results of reconstructing causal networks on BEELINE_VSC. (**a**) Ground truth. Gene Nkx22 is used as a hidden variable and the remaining genes are used as observed variables. It is worth noting that BEELINE_VSC is a gene regulation dataset, so there may be mutual regulation between genes; that is, they are represented in the real network as bidirectional edges. (**b**) Result of RLCI. H_1 is the reconstructed variable. Black lines represent correctly predicted edges and red lines represent incorrectly predicted edges. The blue line indicates a predicted relationship between two nodes with an unknown causal direction, which is not considered in our analysis. (**c**) Result of PC. (**d**) Result of LiNGAM. (**e**) Result of DirectLiNGAM. (**f**) Result of FCI. (**g**) Result of GFCI. (**h**) Result of RFCI. (**i**) Result of RCD. (**j**) Result of CAM-UV.

A research study has shown that cyclopoids can take cyanobacteria as food, and the picophytoplankton is a kind of cyanobacteria with the smallest cell size [33,34]. Therefore, the causality from cyclopoids to picophytoplankton inferred by RLCI may be true for the food chain (Figure 6).

Furthermore, nautilus and copepods have been documented as common aquatic herbivores and a common prey for copepods [35], which confirms that cyclopoids can prey on calanoids (Figure 6b). Although there is no confirmed research paper indicating that nanophytoplankton feeds on picophytoplankton, recent findings [36] suggest that phytoplankton can exhibit predatory behavior in addition to photosynthesis. The existence of phytoplankton species that exhibit a combination of two different modes of nutrition, photosynthesis and predation, has been revealed. Therefore, the ability of the nanophytoplankton to prey on the picophytoplankton may also exist, but further investigations are needed to confirm this (Figure 6b).

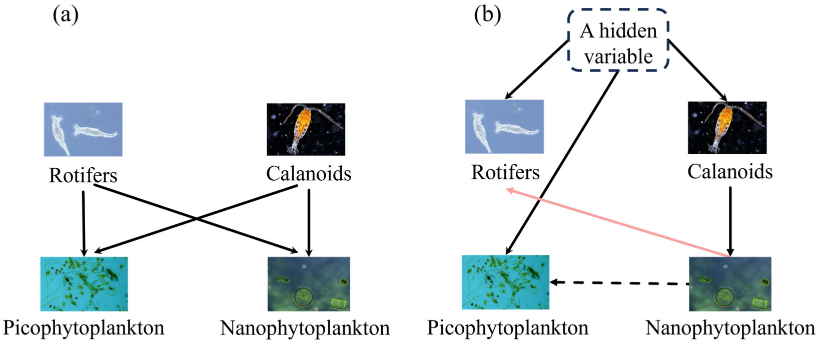

Figure 6. Results of reconstructing causal networks in the food chain. (**a**) Ground truth. The direction of the arrow means the direction of preying. (**b**) Result of RLCI. The black lines represent correctly predicted edges and the red lines represent incorrectly predicted edges. The dotted line represents that the accuracy of the predicted outcome is undetermined.

3.5. Performance on the COAD Dataset in TCGA

The development and progression of cancer involve intricate molecular regulations. Analyzing the gene regulatory network (GRN) of oncogenes can provide valuable insights for cancer prevention and treatment. This study focuses on analyzing the GRN of oncogenes in colon adenocarcinoma (COAD) as an example. The RNA-Seq data for COAD was obtained from the TCGA database, which can be accessed at http://xena.ucsc.edu/, accessed on 30 April 2024. We filtered out the tumor samples and excluded samples labeled as 'cancer stage I' or 'not reported'.

To establish a gold standard regulatory network for COAD data, we used the Colorectal cancer pathway (hsa05210) from the KEGG (Kyoto Encyclopedia of Genes and Genomes) database at https://www.kegg.jp/kegg/pathway.html, accessed on 30 April 2024. We selected a subset of the pathway consisting of 10 nodes, with a total of 27 genes (Figure 7). This subset served as the reference regulatory network for COAD data.

In the experiment, we deliberately concealed the KRAS gene. The results of our method indicate that the edges "PIK3CD-RALGDS" and "BRAF-PIK3R1" are spurious causal relationships, which are caused by the influence of hidden variables. Upon considering the actual network, it was confirmed that KRAS simultaneously affects these four genes. This finding further validates the accuracy of our algorithm.

3.6. Analysis of Evaluation Metrics

The RLCI algorithm was compared with other algorithms on four datasets: Sim1, Sim6, DREAM4, and BEELINE_VSC. The reconstructed causal network is evaluated for precision, recall, and F1 score (Figure 8). Among other algorithms, LiNGAM and DirectLiN-GAM perform better in recall, but their precision is poor due to the high false positivity in the results (Figure 8). The PC algorithm assumes that there are no potential confounding factors or omitted variables in the data generation process; that is, all factors that affect the relationships between variables have been observed. FCI is an improved version of the PC that takes into account the influence of hidden variables. GFCI, RFCI, and FCI are similar methods. GFCI is a variant of FCI, which combines the FCI algorithm and greedy search strategy. RFCI is a further simplification and optimization of FCI, aimed at improving the computational efficiency of the algorithm while maintaining its ability to handle potential confounding factors and select bias. These four algorithms can identify the causal direction between part of the variables, and can also generate some undirected edges, which can only determine whether there is a relationship between the variables, but cannot determine the causal direction between these variables (Figures 2–5). The CAM-UV algorithm is greatly affected by the size of the data and performs well on small datasets, but performs poorly on large datasets. The RCD algorithm is based on the linear and non-Gaussian nature of the data, and identifies causal relationships between variables through an iterative process, resulting in an overall performance bias on nonlinear data. In summary, RLCI outperforms other algorithms in three evaluation metrics and is not affected by the dataset size. RLCI still performs well on a large sample dataset, demonstrating excellent stability (Figure 8).

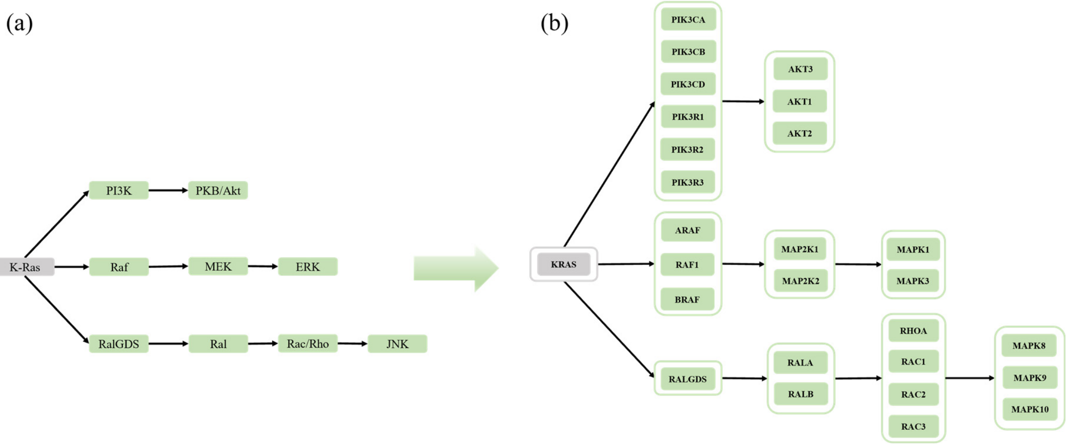

Figure 7. The KEGG pathway in human colorectal cancer. (**a**) Part of the KEGG pathway in human colorectal cancer. Each node may contain multiple genes. (**b**) Genes contained in each node in (**a**).

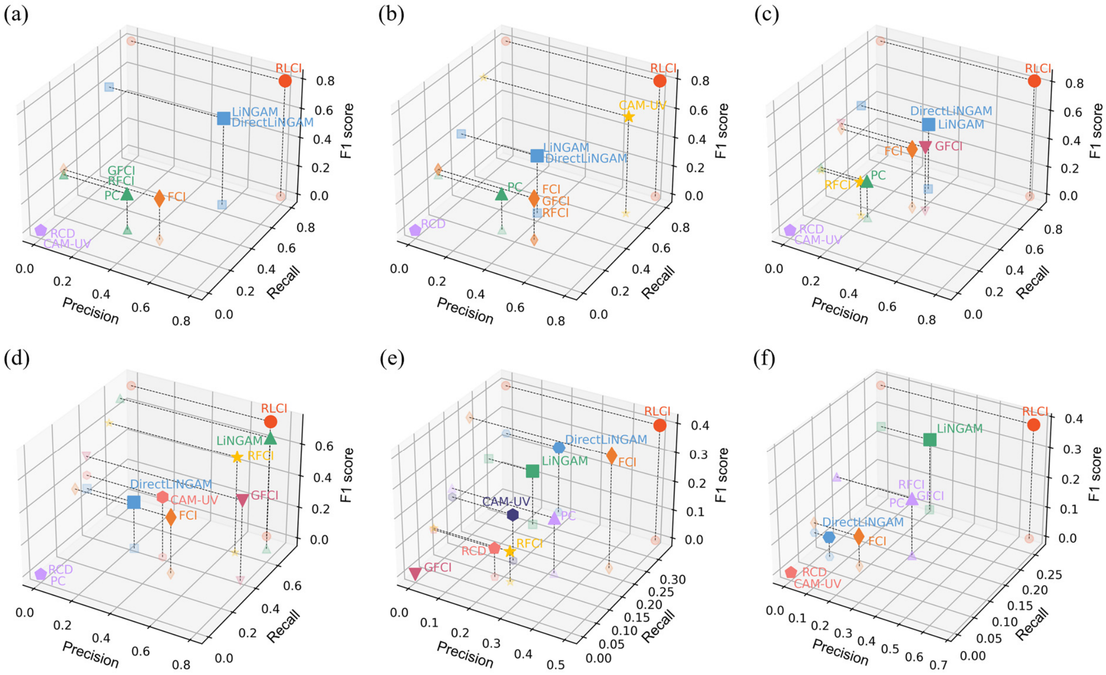

Figure 8. Comparison of algorithm results. The x, y, and z axis represent precision, recall, and F1 score, respectively, to evaluate the results of the reconstructed networks. The same marker and color indicate that the algorithms achieved identical results across three evaluation metrics. (**a**) Performance of the evaluation metrics on Sim1. (**b**) Performance of the evaluation metrics on Sim1 of 1000 samples. (**c**) Performance of the evaluation metrics on Sim6. (**d**) Performance of the evaluation metrics on Sim6 of 1000 samples. (**e**) Performance of the evaluation metrics on DREAM4. (**f**) Performance of the evaluation metrics on BEELINE_VSC.

4. Conclusions

The identification of causal relationships is crucial for advancing scientific knowledge and driving innovation across diverse disciplines. In this study, we have developed the Regression Loss-increased with Causal Intensity (RLCI) algorithm, which effectively uncovers hidden variables. RLCI has demonstrated its efficacy in accurately identifying unobserved variables and determining causal directions between observed variables. The RLCI algorithm is particularly effective when applied to datasets with large samples, outperforming traditional algorithms. These results demonstrate the potential of RLCI to reveal hidden variables and improve causal inference in various research domains.

However, there is still room for improvement in the RLCI algorithm. Currently, the algorithm cannot recognize bidirectional edges and struggles to identify situations where two directly connected observed variables are simultaneously influenced by a hidden variable. Future efforts should focus on developing a nonlinear version of the algorithm to address these limitations or relaxing certain assumptions. For instance, broadening the algorithm's applicability and enhancing its accuracy could be achieved by considering cases where observed variables are affected by multiple hidden variables.

By addressing these challenges and further refining the RLCI algorithm, its capabilities can be expanded, and its utilization can be facilitated beyond the scope of this study. This would enable a deeper understanding of causal relationships and contribute to more precise and comprehensive analyses in various research fields.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/math12091375/s1. Figure S1: Indirect causality and false causality caused by hidden variables. Figure S2: Increased losses in linear regression and causal intensity. Figures S3–S7: Steps in reconstructing causal networks of different datasets. Figure S8: Different types of edges generated by FCI, GFCI, and RFCI algorithms in reconstructing causal networks. Tables S1–S6: Evaluation results on different datasets.

Author Contributions: Conceptualization, X.L. (Xiaoping Liu); data curation, X.L. (Xingyu Liao); formal analysis, X.L. (Xingyu Liao); investigation, X.L. (Xingyu Liao); methodology, X.L. (Xingyu Liao) and X.L. (Xiaoping Liu); project administration, X.L. (Xiaoping Liu); resources, X.L. (Xiaoping Liu); supervision, X.L. (Xiaoping Liu); writing—original draft, X.L. (Xingyu Liao); writing—review and editing, X.L. (Xiaoping Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (NSFC) Grant No. T2341024, Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ22C060001, the research funds of Hangzhou Institute for advanced study, UCAS (No. 2022ZZ01013), the National Key Research and Development Program of China (2022YFA1004800).

Data Availability Statement: For detailed information regarding the datasets analyzed during this study, please refer to Section 3. Results, where the links to the publicly datasets are provided.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Granger, C.W. Investigating causal relations by econometric models and cross-spectral methods. *Econom. J. Econom. Soc.* **1969**, *37*, 424–438. [CrossRef]
2. Verma, T.S.; Pearl, J. Equivalence and synthesis of causal models. In *Probabilistic and Causal Inference: The Works of Judea Pearl*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 221–236.
3. Spirtes, P.; Glymour, C.; Scheines, R. From probability to causality. *Philos. Stud. Int. J. Philos. Anal. Tradit.* **1991**, *64*, 1–36. [CrossRef]
4. Spirtes, P.; Glymour, C.; Scheines, R. *Causation, Prediction, and Search*; MIT Press: Cambridge, MA, USA, 2001.
5. Bouckaert, R.R. Properties of Bayesian belief network learning algorithms. In *Uncertainty in Artificial Intelligence*; Elsevier: Amsterdam, The Netherlands, 1994.
6. Sun, B.; Zhou, Y. Bayesian network structure learning with improved genetic algorithm. *Int. J. Intell. Syst.* **2022**, *37*, 6023–6047. [CrossRef]

7. Teyssier, M.; Koller, D. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. *arXiv* **2012**, arXiv:1207.1429.
8. Pearl, J. *Models, Reasoning and Inference*; Cambridge University Press: Cambridge, UK, 2000; Volume 19, p. 3.
9. Shimizu, S.; Hoyer, P.O.; Hyvärinen, A.; Kerminen, A.; Jordan, M. A linear non-Gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.* **2006**, *7*, 2003–2030.
10. Zhang, K.; Hyvarinen, A. On the identifiability of the post-nonlinear causal model. *arXiv* **2012**, arXiv:1205.2599.
11. Zhang, K.; Chan, L.-W. Extensions of ICA for causality discovery in the hong kong stock market. In Proceedings of the International Conference on Neural Information Processing, Hong Kong, China, 3–6 October 2006; Springer: Berlin/Heidelberg, Germany, 2006.
12. Hoyer, P.; Janzing, D.; Mooij, J.M.; Peters, J.; Schölkopf, B. Nonlinear causal discovery with additive noise models. *Adv. Neural Inf. Process. Syst.* **2008**, *21*, 689–696.
13. Peters, J.; Janzing, D.; Scholkopf, B. Causal inference on discrete data using additive noise models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2436–2450. [CrossRef] [PubMed]
14. Pratapa, A.; Jalihal, A.P.; Law, J.N.; Bharadwaj, A.; Murali, T. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nat. Methods* **2020**, *17*, 147–154. [CrossRef]
15. Benincà, E.; Jöhnk, K.D.; Heerkloss, R.; Huisman, J. Coupled predator–prey oscillations in a chaotic food web. *Ecol. Lett.* **2009**, *12*, 1367–1378. [CrossRef]
16. Fisher, R.A. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika* **1915**, *10*, 507–521. [CrossRef]
17. Colombo, D.; Maathuis, M.H. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.* **2014**, *15*, 3741–3782.
18. Li, Z.; Shahrajabian, H.; Bagherzadeh, S.A.; Jadidi, H.; Karimipour, A.; Tlili, I. Effects of nano-clay content, foaming temperature and foaming time on density and cell size of PVC matrix foam by presented Least Absolute Shrinkage and Selection Operator statistical regression via suitable experiments as a function of MMT content. *Phys. A Stat. Mech. Its Appl.* **2020**, *537*, 122637. [CrossRef]
19. He, C.; Yue, K.; Wu, H.; Liu, W. Structure learning of bayesian network with latent variables by weight-induced refinement. In Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning, Shanghai, China, 3 November 2014.
20. Ross, B.C. Mutual information between discrete and continuous data sets. *PLoS ONE* **2014**, *9*, e87357. [CrossRef] [PubMed]
21. Janzing, D.; Mooij, J.; Zhang, K.; Lemeire, J.; Zscheischler, J.; Daniušis, P.; Steudel, B.; Schölkopf, B. Information-geometric approach to inferring causal directions. *Artif. Intell.* **2012**, *182*, 1–31. [CrossRef]
22. Shimizu, S.; Inazumi, T.; Sogawa, Y.; Hyvarinen, A.; Kawahara, Y.; Washio, T.; Hoyer, P.O.; Bollen, K.; Hoyer, P. DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. *J. Mach. Learn. Res. JMLR* **2011**, *12*, 1225–1248.
23. Spirtes, P.L.; Meek, C.; Richardson, T.S. Causal inference in the presence of latent variables and selection bias. *arXiv* **2013**, arXiv:1302.4983.
24. Glymour, C.; Zhang, K.; Spirtes, P. Review of causal discovery methods based on graphical models. *Front. Genet.* **2019**, *10*, 524. [CrossRef] [PubMed]
25. Jabbari, F.; Cooper, G.F. An instance-specific algorithm for learning the structure of causal Bayesian networks containing latent variables. In Proceedings of the 2020 SIAM International Conference on Data Mining, Cincinnati, OH, USA, 7–9 May 2020.
26. Ogarrio, J.M.; Spirtes, P.; Ramsey, J. A hybrid causal search algorithm for latent variable models. In Proceedings of the Conference on Probabilistic Graphical Models, PMLR, Lugano, Switzerland, 6–9 September 2016.
27. Maeda, T.N.; Shimizu, S. RCD: Repetitive causal discovery of linear non-Gaussian acyclic models with latent confounders. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Online, 26–28 August 2020.
28. Maeda, T.N.; Shimizu, S. Causal additive models with unobserved variables. In Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, PMLR, Online, 27–30 July 2021.
29. Smith, S.M.; Miller, K.L.; Salimi-Khorshidi, G.; Webster, M.; Beckmann, C.F.; Nichols, T.E.; Ramsey, J.D.; Woolrich, M.W. Network modelling methods for FMRI. *Neuroimage* **2011**, *54*, 875–891. [CrossRef]
30. Marbach, D.; Prill, R.J.; Schaffter, T.; Mattiussi, C.; Floreano, D.; Stolovitzky, G. Revealing strengths and weaknesses of methods for gene network inference. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 6286–6291. [CrossRef]
31. Schaffter, T.; Marbach, D.; Floreano, D. GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods. *Bioinformatics* **2011**, *27*, 2263–2270. [CrossRef] [PubMed]
32. Benincà, E.; Huisman, J.; Heerkloss, R.; Jöhnk, K.D.; Branco, P.; Van Nes, E.H.; Scheffer, M.; Ellner, S.P. Chaos in a long-term experiment with a plankton community. *Nature* **2008**, *451*, 822–825. [CrossRef] [PubMed]
33. Tõnno, I.; Agasild, H.; Kõiv, T.; Freiberg, R.; Nõges, P.; Nõges, T. Algal diet of small-bodied crustacean zooplankton in a cyanobacteria-dominated eutrophic lake. *PLoS ONE* **2016**, *11*, e0154526. [CrossRef] [PubMed]
34. Sommer, U.; Sommer, F. Cladocerans versus copepods: The cause of contrasting top–down controls on freshwater and marine phytoplankton. *Oecologia* **2006**, *147*, 183–194. [CrossRef] [PubMed]

35. Soto, D.; Hurlbert, S.H. Long-term experiments on calanoid-cyclopoid interactions. *Ecol. Monogr.* **1991**, *61*, 245–266. [CrossRef]
36. Li, Q.; Edwards, K.F.; Schvarcz, C.R.; Steward, G.F. Broad phylogenetic and functional diversity among mixotrophic consumers of Prochlorococcus. *ISME J.* **2022**, *16*, 1557–1569. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.