Supplementary Information for

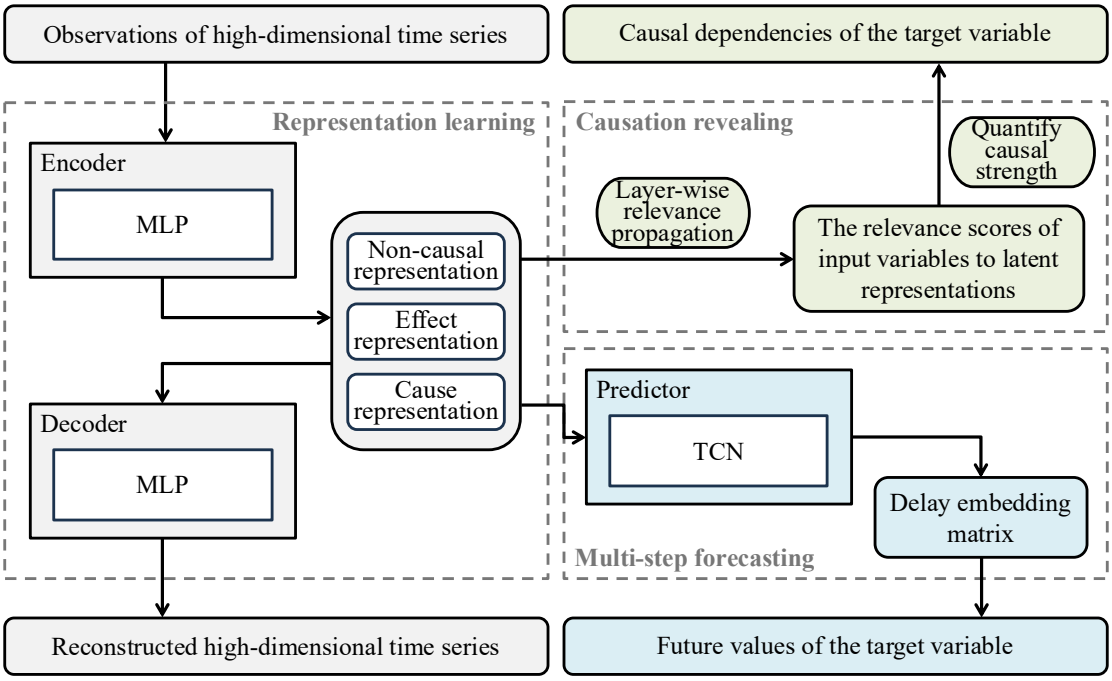
Causal-oriented representation learning for time-series forecasting based on the spatiotemporal information transformation

Sihua Cai[1,†] , Hao Peng[1,†], Rui Liu[1,*] , Pei Chen[1,*]

Content

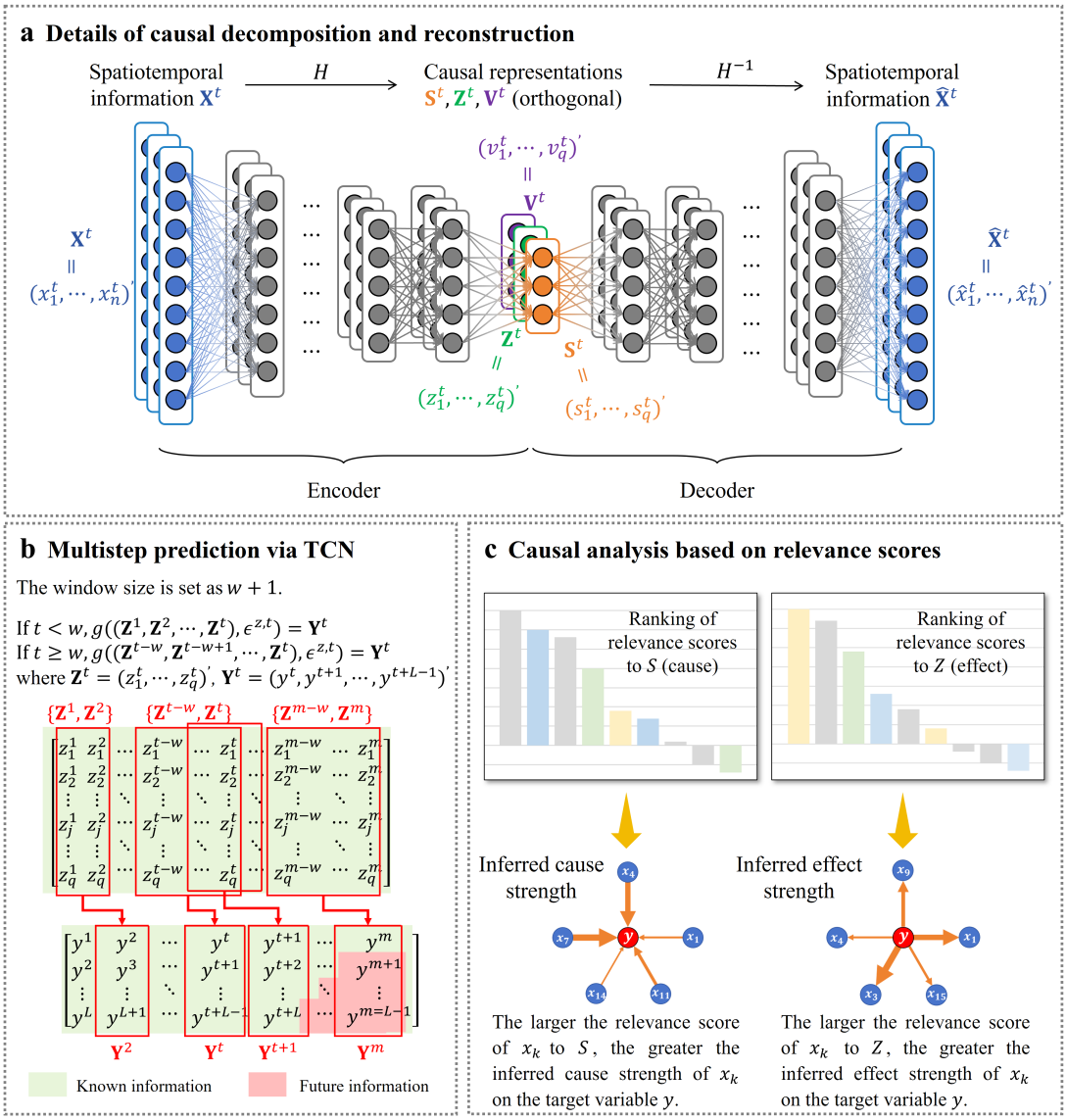
Supplementary Figures

Supplementary Figure 1. The flowchart of CReP method.



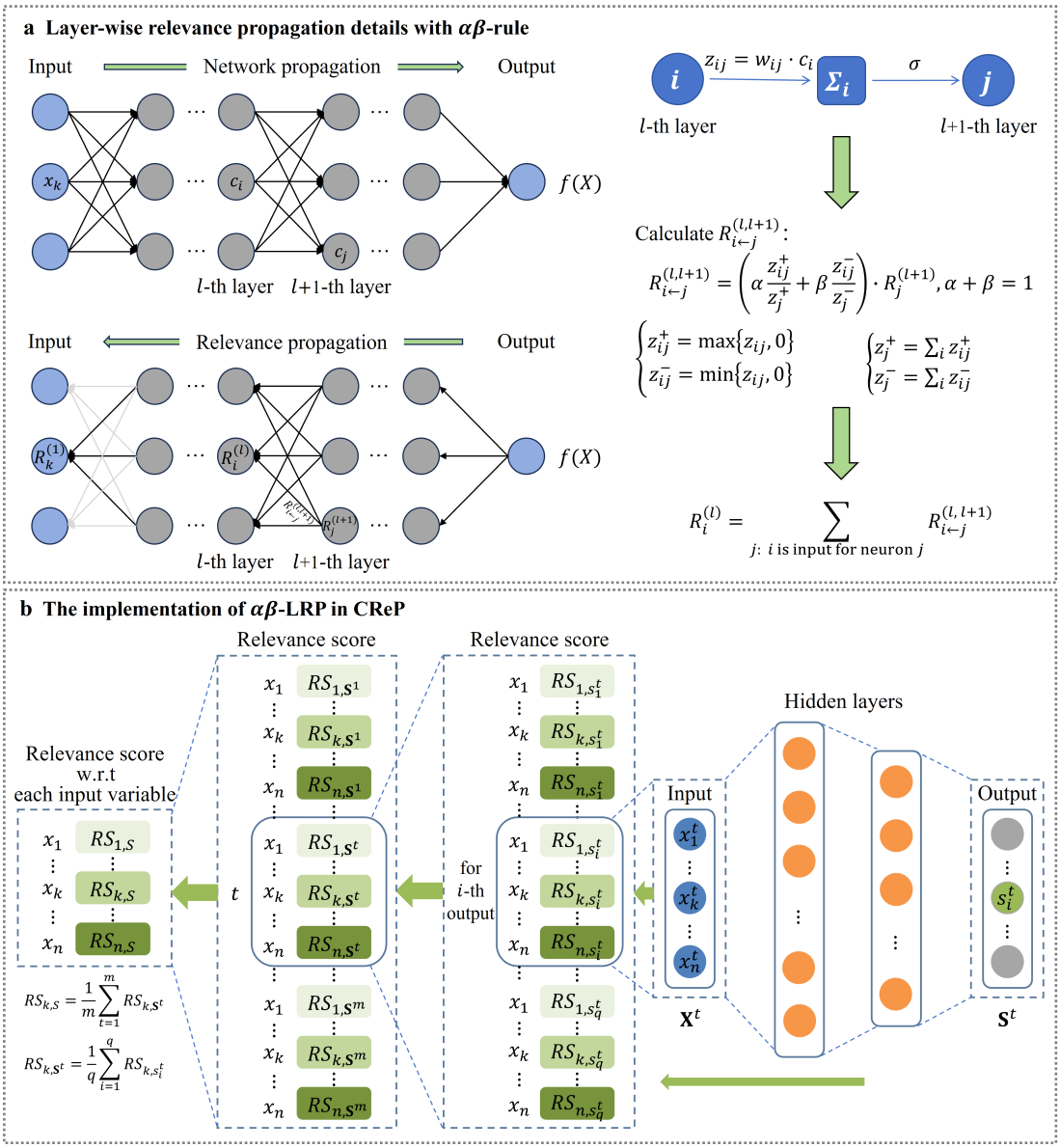
Supplementary Figure 1. The flowchart of CReP method. In general, CReP consists of three modules: representation learning, multi-step forecasting, and causation revealing. Given observations from high-dimensional time series, three implicit causal representations can be learned via an encoder-decoder structure, with each representation capturing different causal relationships with the target variable. Based on the dynamic causation, the temporal information of the target variable is predicted by the effect representation through embedding mapping. Additionally, to reveal explicit causal relationships between the observed variables and the target variable, we employ the LRP (Layer-wise Relevance Propagation) method to interpret to what extent the input variables contribute to the causal representations, thereby indirectly quantifying causal strengths of input variables on the target variable.

Supplementary Figure 2. Implementation details of CReP framework.



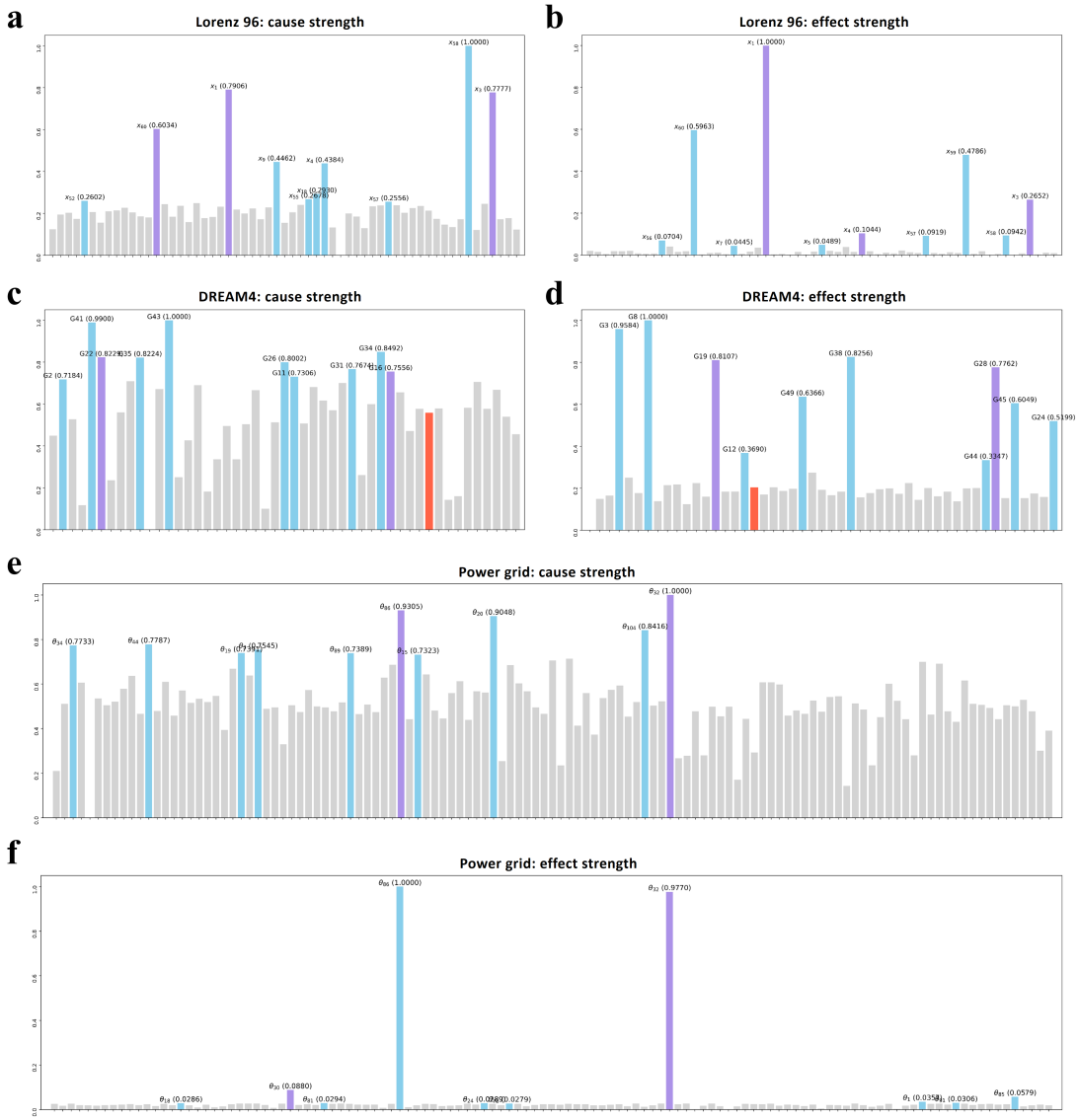
Supplementary Figure 2. Implementation details of CReP framework. (a) Schematic illustration of causal information decomposition and reconstruction based on CReP. The original high-dimensional data is transformed into three causal factors through an autoencoder, each representing distinct causal relationships with the target variable y. (b) The sliding window scheme on the effect representation of CReP to make multistep forecasting. (c) The explicit explanation of latent causal representations through computing relevance scores. The contribution of each input variable to causal representations quantifies their causal strength on the target variable, visualized in the inferred causal strength networks of y.

Supplementary Figure 3. The causal analysis process through interpretation method $\alpha\beta$-LRP.



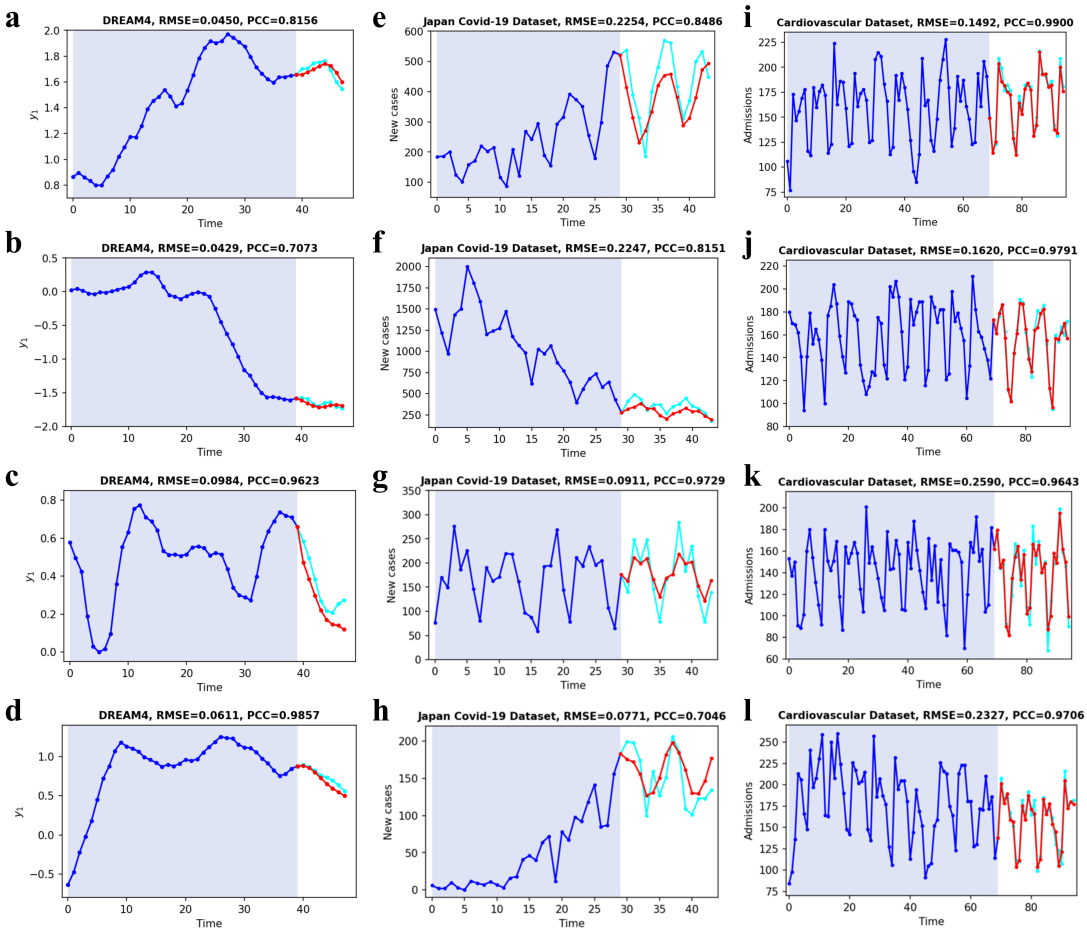
Supplementary Figure 3. The causal analysis process through interpretation method $\alpha\beta$-LRP. (a) The calculation details of LRP with $\alpha\beta$ rule. The relevance scores are calculated starting from the output layer and then moving backward to the input layer, which is in the opposite direction of network propagation. This method utilizes the weights and structure of the network to determine the importance of input variables for the output. (b) The implementation of $\alpha\beta$-LRP in CReP. Since the learnt causal representations (*e.g.*, $\boldsymbol{S}^t$) are multidimensional output, the relevance scores of input variables calculated using the $\alpha\beta$-LRP method are integrated across all output neurons. Additionally, the relevance scores calculated across all time points are averaged to assess the importance of input variables to the causal information, independent of specific time points.

Supplementary Figure 4. The normalized causal results of CReP on simulation datasets.



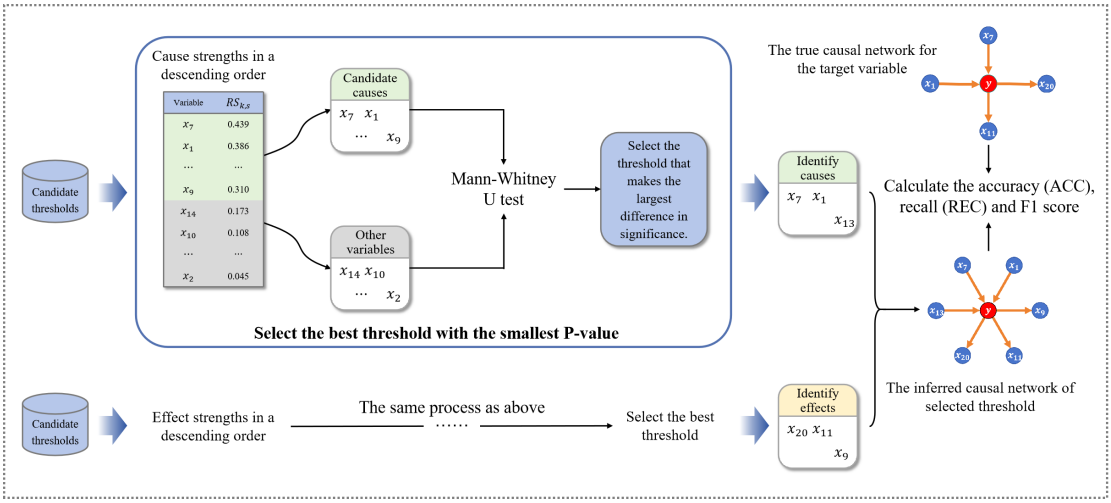
Supplementary Figure 4. The normalized causal results of CReP on simulation datasets. The causal strength of input variables on the target variable y is quantified using the relevance scores. After normalization, we display the causal strength of all variables on the target variable. Blue bars correspond to the top 10 variables in terms of causal strength, while purple bars mark the true causes or effects among these top 10 variables. Red bars represent true causes or effects that are not identified in the top 10. Based on the CReP, the cause strength and effect strength results for the Lorenz 96 simulation data are shown in (a) and (b), respectively. In (c) and (d), the causal results are displayed for the DREAM4 gene regulatory data, while (e) and (f) present the causal results for the power grid dataset modeled by the Kuramoto equations.

Supplementary Figure 5. More forecasting results of CReP.



Supplementary Figure 5. More forecasting results of CReP. By utilizing historical data from the past m time steps, CReP can predict information for the next $L-1$ steps. For the DREAM4 gene regulatory data, the cardiovascular admissions, and the COVID-19 inpatients in Japan, CReP is employed to make multistep predictions in four periods, as shown in (a)-(d), (e)-(h), and (i)-(l), respectively. The average prediction performance across all samples is shown in Table 1 in the main text.

Supplementary Figure 6. The quantified process for evaluating causal results on ablation study.



Supplementary Figure 6. The quantified process for evaluating causal results on ablation study. To assess the importance of each loss term in the objective function, quantitative metrics are required to evaluate the model's causal results. For the inferred causal strength calculated by relevance scores, Mann-Whitney U test is employed to select the optimal threshold for variable classification, thereby identifying the causes and effects for the target variable y. Based on this, a directed causal graph centered on y can be constructed, without differentiation in strength. Using the inferred causal graph and the gold standard, we calculate metrics such as accuracy and recall to evaluate the causal results.

Supplementary Tables

Supplementary Table 1. The numbers of the parameters and the known data for each dataset.

Dataset	Number of known time points (m)	Number of total known data $(m \times D)$	To-be-predicted length of unknown y $(L-1)$
Lorenz 96 system	50	50×60	15
Power grid	30	30×120	9
Gene regulatory network	40	40×50	8
Cardiovascular inpatients	70	70×14	25
Japan Covid-19 transmission	30	30×47	14

Supplementary Table 2. Summary of parameters and variables in CReP framework.

Symbols	Dimensions	Explanation
m	1	Scalar, the length of known time series
$\mathbb{R}^n$	n	n-dimension real number space
d	1	Scalar, box-counting dimension
t	1	Scalar, time point
n	1	Scalar, the dimension of the observed variables in time-series
q	1	Scalar, the dimension of causal representation
H	$q+q+q$	The decomposition function
H^{-1}	n	The reconstruction function
f	q	The function for recovering the cause representation
g	L	The function for making multistep forecasting of y
a^t	1	Scalar, the value of variable a at time point t
b^t	1	Scalar, the value of variable b at time point t
L	1	Scalar, the embedding dimension
$\mathbf{A}^t$	L	Vector, the time delay embedding vector of a at time point t
$\mathbf{B}^t$	$L+1$	Vector, the time delay embedding vector of b at time point t
x_k^t	1	Scalar, the value of observed variable x_k at time point t
y^t	1	Scalar, the value of the target variable y at time point t
$\hat{y}^t$	1	Scalar, the estimation of the target variable y at time point t
X	$n \times m$	Matrix, the high-dimensional time series for the observed variables during m time points
$\hat{X}$	$n \times m$	Matrix, the reconstructed high-dimensional time series by CReP for the observed variables during m time points
$\mathbf{X}^t$	n	Vector, the observed variables at time point t
$\widehat{\mathbf{X}}^t$	n	Vector, the reconstructed observed variables at time point t
Y	$L \times m$	Matrix, the delay embedding matrix of y during m time points
$\hat{Y}$	$L \times m$	Matrix, the reconstructed delay embedding matrix by CReP for the target variable y during m time points
Y_{win}	$L \times (w+1)$	Matrix, the sliding window matrix contains delay embedding vectors for the target variable with $(w+1)$ time points
$\mathbf{Y}^t$	L	Vector, the time delay embedding vector of y at time point t
S	$q \times m$	Matrix, the cause representation of original space for the target variable y
$\hat{S}$	$q \times m$	Matrix, the estimated cause representation using the temporal

		information of y
$\mathbf{S}^t$	q	Vector, the cause representation of original space for the target variable y at time point t
$\hat{\mathbf{S}}^t$	q	Vector, the estimated cause representation using the temporal information of y at time point t
Z	$q \times m$	Matrix, the effect representation of original space for the target variable y
Z_{win}	$q \times (w+1)$	Matrix, the sliding window matrix contains the effect representation for the target variable with $(w+1)$ time points
$\mathbf{Z}^t$	q	Vector, the effect representation of original space for the target variable y at time point t
V	$q \times m$	Matrix, the non-causal representation of original space for the target variable y
$\mathbf{V}^t$	q	Vector, the non-causal representation of original space for the target variable y at time point t
c_i	1	Scalar, the neuron with index i
w_{ij}	1	Scalar, the weight between neuron c_i and neuron c_j
z_{ij}	1	Scalar, the weighted value of neuron c_i into neuron c_j
$R_i^{(l)}$	1	Scalar, the relevance score of neuron c_i in layer l
$R_{i\leftarrow j}^{(l,l+1)}$	1	Scalar, the amount of relevance score that spread to neuron c_i from neuron c_j
$\circ$		Function composition operation
$'$		Transpose of a vector
mean		The average function
max		The maximum function
min		The minimum function
$\mathcal{L}_{\mathrm{DS}}$		The determined-state loss
$\mathcal{L}_{\mathrm{REC}}$		The reconstruction loss
$\mathcal{L}_{\mathrm{REC\_X}}$		The reconstruction loss of observed high-dimensional information
$\mathcal{L}_{\mathrm{REC\_S}}$		The reconstruction loss of latent cause representation information
$\mathcal{L}_{\mathrm{ORTH}}$		The orthogonal loss of latent causal representations
$\mathcal{L}_{\mathrm{FC}}$		The future-consistency loss

Supplementary Table 3. Summary of default hyperparameters for CReP.

Dataset	**Number of Encoder/ Decoder layers**	**Number of TCN filters**	**Dilation factors of TCN filters**	**Weights of loss terms**	**Initial training rate**
Lorenz 96 system	[128, 64, 32]	[128, 64, 32, 16]	[1, 2, 4, 8]	$\{\lambda_1$: 0.6, λ_2: 0.15, λ_3: 0.05, λ_4: 0.1$\}$	0.01
Power grid	[128, 64, 32]	[128, 64, 32, 10]	[1, 2, 4, 8]	$\{\lambda_1$: 0.6, λ_2: 0.15, λ_3: 0.05, λ_4: 0.1$\}$	0.01
Gene regulatory network	[128, 64, 32]	[128, 64, 32, 9]	[1, 2, 4, 8]	$\{\lambda_1$: 0.6, λ_2: 0.15, λ_3: 0.05, λ_4: 0.1$\}$	0.01
Cardiovascular inpatients	[128, 64, 32]	[128, 64, 32, 26]	[1, 2, 4, 8]	$\{\lambda_1$: 0.6, λ_2: 0.15, λ_3: 0.05, λ_4: 0.1$\}$	0.01
Japan Covid-19 transmission	[128, 64, 32]	[128, 64, 32, 15]	[1, 2, 4, 8]	$\{\lambda_1$: 0.6, λ_2: 0.15, λ_3: 0.05, λ_4: 0.1$\}$	0.01

Supplementary Table 4. Comparison to other forecasting methods.

	Lorenz 96		Power grid		Dream4	
	RMSE	PCC	RMSE	PCC	RMSE	PCC
CReP	**0.1086**	**0.9908**	**0.1127**	**0.9868**	**0.0984**	0.8894
ARNN	0.2328	0.8206	0.1501	0.9784	0.2281	0.8167
STICM	0.1822	0.9752	0.1530	0.9854	0.1080	**0.9928**
LSTM	0.4883	0.8303	0.3916	0.9659	0.8063	0.0615
ARIMA	0.4204	0.6156	0.2558	0.9583	0.2650	0.1218
SVR	0.8311	0.1618	0.4856	0.9540	0.6779	0.0566
Informer	0.1543	0.9005	0.1163	0.9837	0.4716	0.0489

Supplementary Table 5. The performance of CReP on different target variables.

Lorenz 96	**y2**	**y17**	**y21**	**y39**	**y29**
RMSE	0.1086	0.1332	0.1284	0.1212	0.1334
PCC	0.9908	0.9624	0.9570	0.9533	0.9522
TPR	1.0	1.0	0.83	0.83	1.0
FPR	0.05	0.05	0.06	0.06	0.05
Power grid	**y33**	**y68**	**y38**	**y103**	**y28**
RMSE	0.1127	0.1638	0.0550	0.0939	0.1152
PCC	0.9868	0.9832	0.9979	0.9946	0.9909
TPR	1.0	0.80	0.80	1.0	0.67
FPR	0.03	0.03	0.03	0.04	0.04
DREAM4	**y18**	**y5**	**y39**	**y11**	**y32**
RMSE	0.0984	0.0429	0.0536	0.1257	0.0691
PCC	0.8894	0.9608	0.9283	0.8506	0.9209
TPR	0.67	1.0	0.67	1.0	0.67
FPR	0.10	0.05	0.11	0.06	0.09

Supplementary Notes

Supplementary Note 1. Dynamical systems and delay embedding theorem

For a general discrete time dissipative system, the dynamics can be defined as

$$\mathbf{X}^{t+1} = \phi(\mathbf{X}^t),$$

where $\phi: \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear map, and its values are defined in the n-dimensional state space $\mathbf{X}^t = (x_1^t, x_2^t, \cdots, x_n^t)'$ at a time point t, where symbol " $'$ " is the transpose of a vector, and any time eval between two consecutive time points is equal. After a sufficiently long time, all of states are converged into a compact manifold $\mathcal{V}$. Denote $\mathcal{A}$ as the system's attractor contained in the manifold $\mathcal{V}$ with the box-counting dimension $d_{\mathcal{A}}$. The delay embedding theorem claims that the original high dimensional system's attractor $\mathcal{A}$ can be topologically reconstructed by only using the long-term observation of a single variable under certain conditions. The Takens' embedding theorem is stated as follows [1–3].

Theorem 1: Let $\mathcal{V}$ be a compact manifold with the box-counting dimension d. For a smooth (we mean at least $\mathbb{C}^2$) diffeomorphism $\phi: \mathcal{V} \to \mathcal{V}$ and a smooth (we mean at least $\mathbb{C}^2$) function $h: \mathcal{V} \to \mathbb{R}^1$, it is a generic property that the mapping $\Phi_{\phi,h}(X): \mathcal{V} \to \mathbb{R}^L$ is an embedding when $L > 2d$, that is,

$$\Phi_{\phi,h}(X) = (h(X), h \circ \phi(X), \cdots, h \circ \phi^{L-1}(X))',$$

where symbol " $\circ$ " is the function composition operation.

Takens proved that there is an open and dense set of pairs (ϕ, h) for which the mapping $\Phi_{\phi,h}$ is an embedding [1]. Moreover, the generalized embedding theorem is also a kind of embedding theorem which is stated as follows [4].

Theorem 2: Let $\mathcal{V}$ be a compact manifold with the box-counting dimension d. For a set of L smooth (we mean at least $\mathbb{C}^2$) observation functions $\{y_1, y_2, \cdots, y_L\}$ where $y_k: \mathcal{V} \to \mathbb{R}^1$, it is a generic property that the mapping $\phi_{y_k}(X): \mathcal{V} \to \mathbb{R}^L$ is an embedding when $L > 2d$, that is,

$$\phi_{y_k}(X) = \big(y_1(X), y_2(X), \cdots, y_L(X)\big).$$

Generally, the dimension d of manifold $\mathcal{V}$ is much larger than the dimension $d_{\mathcal{A}}$, which limits the application. Then the generalized embedding theorem for fractal attractors [2] was proposed to solve this limitation. For the reason that the attractor's dimension $d_{\mathcal{A}}$ is usually smaller than manifold's dimension d, as long as $L > 2d_{\mathcal{A}}$, the maps mentioned in Theorems 1 and 2 are still

one-to-one on $\mathcal{A}$ and constrained on each compact subset of a smooth manifold contained in $\mathcal{A}$. Thus when $L > 2d_{\mathcal{A}}$, the dynamics can be reconstructed by the data observed from single variable $x_k,\ k = 1,2,\cdots,n$.

In particular, letting $X = \mathbf{X}^t$ and $h(\mathbf{X}^t) = y^t$ where $y^t \in \mathbb{R}$, then the mapping above has the following form with $\Phi_{\phi,h} = \Phi$ and

$$\Phi(\mathbf{X}^t) = (y^t, y^{t+1}, \cdots, y^{t+L-1})' = \mathbf{Y}^t$$

which is used in the primary STI equations. Denote $\mathcal{M}$ as the reconstructed attractor in $\mathbb{R}^L$ through the mapping Φ. Moreover, since the embedding is a one-to-one mapping, its conjugate form $\mathbf{X}^t = \Phi^{-1}(\mathbf{Y}^t) = \Psi(\mathbf{Y}^t)$ with $\Psi: \mathcal{M} \to \mathcal{A}$ can be derived. Note that $\mathbf{X}^t$ is n-dimensional with spatial information, and $\mathbf{Y}^t$ is a L-dimensional vector with temporal information. Therefore, the delay embedding theorem establishes a solid theoretical foundation for the spatiotemporal information transformation mechanism which will be discussed in the following section.

To make Takens' embedding theorem applicable to more real-world scenarios, it has been extended to stochastic systems [5,6]. Specifically, consider a stochastic system whose state at time $t \in \mathbb{Z}$ evolves under the influence of a noise term ω_t, randomly drawn from a compact manifold $\mathcal{N}$. Formally, the stochastic version of Takens' embedding theorem can be stated as follows.

Theorem 3: Let $\mathcal{V}$ and $\mathcal{N}$ be compact manifolds of dimension $d \geq 1$ and $\bar{d}$ respectively. Let $\Sigma = \mathcal{N}^{\mathbb{Z}}$ be the space of bi-infinite sequences $\omega = (\cdots, \omega_{-1}, \omega_0, \omega_1, \cdots)$ of elements of $\mathcal{N}$ with the product topology. Suppose that $L > 2d$. Then for $r \geq 1$, there exists a residual set of $(\phi, h) \in \mathcal{D}^r(\mathcal{V} \times \mathcal{N}, \mathcal{V}) \times \mathcal{C}^r(\mathcal{V}, \mathbb{R}^1)$ such that for any (ϕ, h) in this set there is an open dense set of ω in Σ such that $\Phi_{\phi,h,\omega}$ is an embedding, that is,

$$\Phi_{\phi,h,\omega} = (h(X), h \circ \phi_{\omega_0}(X), \cdots, h \circ \phi_{\omega_{L-2}\cdots\omega_0}(X))',$$

where $\phi_{\omega_t\cdots\omega_0} = \phi_{\omega_t} \circ \cdots \circ \phi_{\omega_0}$ and $\phi_{\omega_i}(\cdot) = \phi(\cdot, \omega_i)$.

According to **Theorem 3**, even in the presence of noise, the original dynamical system can be topologically reconstructed through a delay mapping, which is generically an embedding under certain conditions. Building on these theorems, dynamic causation has been theoretically formulated and is applicable to a wide range of real-world scenarios [7,8].

Supplementary Note 2. Spatiotemporal information transformation (STI) mechanism

The steady-state or the attractor is constrained in a low dimensional space for a high-dimensional system, which also holds for many real-world systems. By exploring such a low-dimensional feature, spatiotemporal information (STI) transformation [3,9,10] has theoretically been derived from the delay embedding theorem [2], which can transform spatial information of high-dimensional data to the temporal information of any target variable. Assuming $L > 2d > 0$ where d is the box-counting dimension of the attractor $\mathcal{A}$ or manifold $\mathcal{V}$, L is the number of embeddings, the STI equations can be stated as follows at $t = 1, 2, \dots, m$ (m is the length of X),

$$\begin{cases} \Phi(\mathbf{X}^t) = \mathbf{Y}^t \\ \mathbf{X}^t = \Psi(\mathbf{Y}^t)' \end{cases} \quad (1)$$

where $\Phi: \mathcal{A} \to \mathcal{M}$ and $\Psi: \mathcal{M} \to \mathcal{A}$ are differentiable functions satisfying $\Phi \circ \Psi = id$. Here, note that $\mathbf{X}^t$ is n-dimensional. Clearly, the left-hand side of Supplementary Eq. (1) is the spatial information of n variables while the right-hand side is the temporal information of the target variable. The first equation is the primary form and the second equation is the conjugate form of the STI equations in Supplementary Eq. (1).

Through the STI transformation equation, the state space of high-dimensional complex system is topologically conjugated to the reconstructed state space of a single variable from the system using a delay embedding scheme. Furthermore, the generalized embedding theorem illustrates that the reconstructed delay embedding manifold of a single variable can be replaced by a non-delay embedding manifold involving multiple variables as long as certain dimensionality conditions are met. Based on this mechanism, the concept of dynamic causation can be extended to a more generalized form, which will be elaborated in detail in the following section.

Supplementary Note 3. Dynamic causation with the STI mechanism

According to dynamical systems theory, if two variables (say, a and b) are from the same dynamic system, they are causally related. Specifically, when the variable a acts as an environmental driver of the variable b, the historical information from a is contained within the time series of b. Therefore, the temporal information of a can be recovered from b. To illustrate this more clearly, we consider a discrete model which describes the causality from a to b based on the instant dynamic causality (iDC) [7] as follows:

$$\begin{cases} a^{t+L\Delta t} = g\left(a^{t+(L-1)\Delta t}, \cdots, a^{t}\right) + \epsilon^{a,t}, \\ b^{t+L\Delta t} = f\left(a^{t+(L-1)\Delta t}, \cdots, a^{t}, b^{t+(L-1)\Delta t}, \cdots, b^{t}\right) + \epsilon^{b,t}, \end{cases}$$

where f and g are two nonlinear functions, and $\epsilon^{a,t}$ and $\epsilon^{b,t}$ include the intrinsic noise and the discretization error. The variables $a^{t+i\Delta t}$ ($i = 0,1,\cdots,L-1$) are denoted as $\mathbf{A}^t = (a^t, a^{t+\Delta t}, \cdots, a^{t+(L-1)\Delta t})'$, and the variables $b^{t+i\Delta t}$ ($i = 0,1,\cdots,L$) are denoted as $\mathbf{B}^t = (b^t, b^{t+\Delta t}, \cdots, b^{t+L\Delta t})'$. Based on the assumption of steady dynamics, $((\mathbf{A}^t)', (\mathbf{B}^t)')'$ is on an attractive manifold with box-counting dimension d. As the necessary conditions are satisfied according to the concept of dynamical causality [11], we can apply the implicit function theorem to derive $\mathbf{A}^t$ as

$$\mathbf{A}^t = h(\mathbf{B}^t, \epsilon^{b,t}), \tag{2}$$

where h sufficiently exists when $L > 2d$, and h is generically the embedding map from the delay manifold $M_b = \{\mathbf{B}^t | t \in \mathbb{R}\}$ to $M_a = \{\mathbf{A}^t | t \in \mathbb{R}\}$ according to Takens' embedding theorem and its stochastic version [5–8]. The causal equation illustrates the mapping direction from the effect variable to the cause variable, and serves as a theoretically grounded principle to identify causality. On the other hand, we cannot apply the implicit function theorem to derive $\mathbf{B}^t$ using $\mathbf{A}^t$, given that the necessary conditions are not satisfied in this case. Therefore $\mathbf{A}^t$ can be predicted reliably using $\mathbf{B}^t$, but not vice versa [12–15]. Note that Convergent Cross Mapping (CCM) [16] has been proposed as a method for constructing h through a linear combination of local neighbors.

The dynamic causation is based on delay embedding theorem, and establishes the embedding mapping from delay manifold of the effect variable to that of the cause variable [7,13,15]. In other words, it emphasizes on the temporal information recovery. To fully utilize the rich spatial information of high-dimensional and short-term data, the dynamic causation can be extended to a more general version, not constrained on the temporal information. Specifically, the mapping h in the causal

equation can be extended to map a non-delay manifold to a delay manifold. The generalized embedding theorem (see Theorem 2) [4], which expands Takens' embedding theorem [1], is capable of addressing more generic scenarios. According to the theorem, the delay embedding manifold of the effect variable b can be topologically conjugated to a non-delay embedding manifold when the embedding dimension meets certain conditions, which is summarized as the spatiotemporal information (STI) transformation [17,18]. Let $\mathbf{C}^t = (c_1^t, c_2^t, \cdots, c_q^t)$ represent the multidimensional representation of M_b at time t in terms of topologically conjugacy, where $q > 2\tilde{d}$ and $\tilde{d}$ is the box-counting dimension of M_b. Clearly, the vector $\mathbf{C}^t$ is non-delay embedding. When the variable a causes variable b, $\mathbf{A}^t$ can be reliably predicted using $\mathbf{C}^t$ and we can also obtain the equation:

$$\mathbf{A}^t = \tilde{h}(\mathbf{C}^t, \epsilon^{c,t}), \tag{3}$$

where $\epsilon^{c,t}$ is the approximation error, and $\tilde{h}$ maps from the non-delay manifold $M_c = \{\mathbf{C}^t | t \in \mathbb{R}\}$ to the delay manifold $M_a = \{\mathbf{A}^t | t \in \mathbb{R}\}$. Therefore, we generalize dynamic causation with the STI transformation, which is illustrated with the Lorenz example (see Fig. 1a in main text). Similarly, we can use the time delay embedding $\mathbf{B}^t$ to reliably predict a non-delay embedding vector which is topologically conjugated to $\mathbf{A}^t$ under certain conditions.

The generalized dynamic causation with the STI transformation, provides a promising perspective for causal analysis and multistep forecasting of high-dimensional but short-term time series. Given the observed spatiotemporal information, it is promising to causally decompose it into different latent factors which represent different causal relationships with the target variable. The decomposition process can be realized through causal representation learning based on dynamic causation. Therefore, the rich spatial information of original data is effectively employed to predict future values of the target variable. Correspondingly, the causal representations can be utilized to recover the original high-dimensional data.

Supplementary Note 4. Layer-wise relevance propagation interpretation method.

With the rapid development of deep learning, artificial neural networks are known for their ability to learn complex relationships/functions between inputs and outputs [19]. For example, they can learn the mapping from observed data X to its cause representation S. However, interpreting how a neural network performs such transformation remains challenging. Layer-wise relevance propagation (LRP), introduced by Sebastian Bach in 2015 [20,21], is an effective method for explaining deep neural network models by considering the network's weights and structure. Its core idea is to redistribute the output backwards to the input variables layer by layer based on relevance propagation rules between two adjacent layers [19]. This method has demonstrated its efficacy and accuracy for both classification and regression tasks [19,22,23].

In contrast to the direction of network propagation, LRP computes the relevance scores of input variables for the output by backpropagating from the output layer to the input layer, as shown in Supplementary Figure 2. It fully leverages the network's weights and parameters to estimate the importance of input variables for the output. Specifically, given a trained neural network, such as a Multilayer Perceptron (MLP) [24], the observed n variables are processed through the network to produce a prediction output. LRP then redistributes the output back to the input variables by assigning relevance scores to assess the importance of each input variable to the network's output. For a multilayer neural network, let c_i and c_j represent neurons in adjacent layers l and $l+1$, respectively, where i and j denote the indices of the neurons. A common mapping from one layer to the next one can be expressed in the following form:

$$c_j = \sigma\left(\sum_{i:i \in K} z_{ij}\right), \tag{4}$$

where $z_{ij} = w_{ij}c_i$, σ is the activation function, and w_{ij} is the weight between neurons c_i and c_j. The set K contains all neurons in the l-th layer. LRP calculates the relevance scores of all neurons from the output $f(X)$ backward to the input variables. Let $R_i^{(l)}$ and $R_j^{(l+1)}$ denote the relevance scores of neurons c_i and c_j in layers l and $l+1$, respectively. The amount of relevance score $R_j^{(l+1)}$ that spread to neuron c_i from neuron c_j is denoted as $R_{i\leftarrow j}^{(l,l+1)}$. By posing the following constraint:

$$R_i^{(l)} = \sum_{j:\, c_i \text{ is input for neuron } c_j} R_{i\leftarrow j}^{(l,l+1)}, \tag{5}$$

the conservation property for the relevance scores would hold between layers globally [19]:

$$\sum_{k=1}^{n} R_k^{(1)} = \cdots = \sum_{i=1}^{n_l} R_i^{(l)} = \sum_{j=1}^{n_{l+1}} R_j^{(l+1)} = \cdots = f(X), \tag{6}$$

where n is the number of input variables, n_l and n_{l+1} are the number of neurons in layer l and $l+1$, and the output value $f(X)$ is the relevance score of the output neuron. The key process for LRP is to calculate the message $R_{i\leftarrow j}^{(l,l+1)}$, for which various rules have been proposed [20]. The adopted propagation method in this work is the $\alpha\beta$-rule [20] defined as

$$R_{i\leftarrow j}^{(l,l+1)} = \left(\alpha \frac{z_{ij}^{+}}{z_j^{+}} + \beta \frac{z_{ij}^{-}}{z_j^{-}}\right) R_j^{(l+1)}, \tag{7}$$

where $z_{ij}^{+} = \max\{z_{ij}, 0\}$, $z_{ij}^{-} = \min\{z_{ij}, 0\}$, $z_j^{+} = \sum_i z_{ij}^{+}$, and $z_j^{-} = \sum_i z_{ij}^{-}$. The parameters α and β need to satisfy the constraint $\alpha + \beta = 1$. By propagating relevance layer by layer, the relevance scores $R_k^{(1)}$ $(k = 1,2,\cdots,n)$ are obtained to evaluate the contribution of each input variable to the network output. Higher relevance scores indicate greater contributions of the input variables to the network's output.

In the CReP framework, causal representations are too abstract to interpret directly, so LRP is employed to explicitly explain the learned causal information. Given that the causal representations are multidimensional outputs, interpreting implicit representations with LRP requires computing the relevance scores of input variables for all output neurons and then integrating these scores. Additionally, we aggregate the relevance scores of input variables across all time points to assess the contribution of each input variable to the causal representations, making these scores independent of specific time points, as shown in Supplementary Figure 2. In this study, these relevance scores are used to measure the strength of each input variable as a cause of y and as an effect of y. The specific LRP computation process within the CReP framework is detailed in the algorithm section.

Supplementary Note 5. The CReP algorithm

The CReP framework effectively leverages the observed spatiotemporal information via causal decomposition, thereby improving the accuracy and robustness of multistep forecasting. The determination of H, g, f, H^{-1} relies on a self-supervised learning scheme. In this study, each layer of the nonlinear mappings is followed by the LeakyReLU activation function. The CReP algorithm is carried out to predict the future values $\{y^{m+1}, y^{m+2}, \cdots, y^{m+L-1}\}$ of the target variable y, which is selected from $\{x_1, x_2, \cdots, x_n\}$, and explore the causal relationships for y through the following procedure.

Step1. Construct the STI-based causal equations. To efficiently utilize the known spatiotemporal information X, we decompose it into three mutually orthogonal causal representations implicitly based on their causal relationship with the target variable y through the nonlinear function H, and then recover the original information using the causal representations through the inverse mapping H^{-1} (Eq. (6) in main text). Therefore, assuming $g = (g_1, g_2, \cdots, g_L)'$ and $f = (f_1, f_2, \cdots, f_q)'$, we can establish the following causal equations with regard to the target variable y according their implicit causal information based on the TCN scheme:

$$\begin{pmatrix} g_1\big((\mathbf{Z}^1), \epsilon^{z,t}\big) & g_1\big((\mathbf{Z}^1, \mathbf{Z}^2), \epsilon^{z,t}\big) & \cdots & g_1\big((\mathbf{Z}^{m-w}, \cdots, \mathbf{Z}^m), \epsilon^{z,t}\big) \\ g_2\big((\mathbf{Z}^1), \epsilon^{z,t}\big) & g_2\big((\mathbf{Z}^1, \mathbf{Z}^2), \epsilon^{z,t}\big) & \cdots & g_2\big((\mathbf{Z}^{m-w}, \cdots, \mathbf{Z}^m), \epsilon^{z,t}\big) \\ \vdots & \vdots & \ddots & \vdots \\ g_L\big((\mathbf{Z}^1), \epsilon^{z,t}\big) & g_L\big((\mathbf{Z}^1, \mathbf{Z}^2), \epsilon^{z,t}\big) & \cdots & g_L\big((\mathbf{Z}^{m-w}, \cdots, \mathbf{Z}^m), \epsilon^{z,t}\big) \end{pmatrix} = \begin{pmatrix} y^1 & y^2 & \cdots & y^m \\ y^2 & y^3 & \cdots & y^{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ y^L & y^{L+1} & \cdots & y^{m+L-1} \end{pmatrix}, \quad (8)$$

$$\begin{pmatrix} f_1\big((\mathbf{Y}^1), \epsilon^{y,t}\big) & f_1\big((\mathbf{Y}^1, \mathbf{Y}^2), \epsilon^{y,t}\big) & \cdots & f_1\big((\mathbf{Y}^{m-w}, \cdots, \mathbf{Y}^m), \epsilon^{y,t}\big) \\ f_2\big((\mathbf{Y}^1), \epsilon^{y,t}\big) & f_2\big((\mathbf{Y}^1, \mathbf{Y}^2), \epsilon^{y,t}\big) & \cdots & f_2\big((\mathbf{Y}^{m-w}, \cdots, \mathbf{Y}^m), \epsilon^{y,t}\big) \\ \vdots & \vdots & \ddots & \vdots \\ f_q\big((\mathbf{Y}^1), \epsilon^{y,t}\big) & f_q\big((\mathbf{Y}^1, \mathbf{Y}^2), \epsilon^{y,t}\big) & \cdots & f_q\big((\mathbf{Y}^{m-w}, \cdots, \mathbf{Y}^m), \epsilon^{y,t}\big) \end{pmatrix} = \begin{pmatrix} s_1^1 & s_1^2 & \cdots & s_1^m \\ s_2^1 & s_2^2 & \cdots & s_2^m \\ \vdots & \vdots & \ddots & \vdots \\ s_q^1 & s_q^2 & \cdots & s_q^m \end{pmatrix}. \quad (9)$$

By leveraging dynamic causation with y, the CReP framework provides the multistep forecasting of the target variable, *i.e.*, $\{y^{m+1}, y^{m+2}, \cdots, y^{m+L-1}\}$, simultaneously uncovering the implicit causal interactions for y through analyzing and interpreting the causal representations S, Z, V.

Step2. Train the CReP network. Given that the future values of y are unknown information, we employ a self-supervised learning scheme to train the network. Specifically, the nonlinear functions H, H^{-1}, f, g are optimized using a "consistently self-constraint scheme" [17] which focuses on both the prediction accuracy of observed historical values and the preservation of temporal

consistency for unknown future values. According to the CReP framework, there are some high-level requirements for the network training.

For the delay embedding matrix Y (Eq. (5) in main text), there are m known historical values $\{y^1, y^2, \cdots, y^m\}$ and $L-1$ unknown future values $\{y^{m+1}, y^{m+2}, \cdots, y^{m+L-1}\}$ remaining to be predicted. The estimated values of Y in each iteration, obtained through the functions H and g, are given as follows:

$$\hat{Y} = \begin{pmatrix} (\hat{y}^1)_1 & (\hat{y}^2)_1 & \cdots & (\hat{y}^m)_1 \\ (\hat{y}^2)_2 & (\hat{y}^3)_2 & \cdots & (\hat{y}^{m+1})_2 \\ \vdots & \vdots & \ddots & \vdots \\ (\hat{y}^L)_L & (\hat{y}^{L+1})_L & \cdots & (\hat{y}^{m+L-1})_L \end{pmatrix}, \tag{10}$$

where $(\hat{y}^t)_j$ denotes the estimated value of y at time point t ($t \in \{1,2,\cdots,m+L-1\}$) using the j-th sub-mapping function g_j ($j \in \{1,2,\cdots,L\}$) according to Supplementary Eq. (8). Since the historical information is known at time points $\{1,2,\cdots,m\}$, we establish $m-1$ self-constrained conditions to ensure the accuracy of the estimated values $\{\hat{y}^1, \hat{y}^2, \cdots, \hat{y}^m\}$:

$$g_{j-1}\left((\mathbf{Z}^{t-w}, \mathbf{Z}^{t-w+1}, \cdots, \mathbf{Z}^t), \epsilon^{z,t}\right) = g_j\left((\mathbf{Z}^{t-w-2}, \mathbf{Z}^{t-w-1}, \cdots, \mathbf{Z}^{t-1}), \epsilon^{z,t}\right), \tag{11}$$

where $j \in \{2,3,\cdots,L\}$ and $\mathbf{Z}^t = (z_1^t, z_2^t, \cdots, z_q^t)'$ is an embedding vector representing effect information of y at time point t. For the unknown future values, to ensure the temporal consistency of the estimated values $\{\hat{y}^{m+1}, \hat{y}^{m+2}, \cdots, \hat{y}^{m+L-1}\}$, we establish a similar set of $L-2$ self-constrained conditions. In total, these $m+L-3$ conditions regulate the network training process by enforcing temporal sequence consistency across samples.

Through the auto perception procedure mentioned above, the optimization of the CReP framework is accomplished through minimizing the following loss function, which comprises four weighted components:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\mathrm{DS}} + \lambda_2 \mathcal{L}_{\mathrm{FC}} + \lambda_3 \mathcal{L}_{\mathrm{REC}} + \lambda_4 \mathcal{L}_{\mathrm{ORTH}}. \tag{12}$$

In Eq. (12), the first part $\mathcal{L}_{\mathrm{DS}}$ is the determined-state loss for the target variable y, calculated from the estimated values of the known states $\{y^1, y^2, \cdots, y^m\}$ with the following form:

$$\mathcal{L}_{\mathrm{DS}} = \sqrt{\frac{1}{2mL - L^2 + L} \sum_{j=1}^{L} \sum_{t=j}^{m} \left((\hat{y}^t)_j - y^t\right)^2}, \tag{13}$$

where $(\hat{y}^t)_j$ ($t = 1,2,\cdots,m$) denotes the estimated value of y at time point t by the sub-mapping function g_j, and y^t ($t = 1,2,\cdots,m$) is the true value. The loss $\mathcal{L}_{\mathrm{DS}}$ quantifies the error

between the estimation $(\hat{y}^t)_j$ and the ground truth y^t for all historical time points t ($t = 1,2,\cdots,m$).

In Eq. (12), the second part $\mathcal{L}_{\mathrm{FC}}$ is the future consistency loss for the target variable y, calculated from the estimated unknown values $\{y^{m+1}, y^{m+2}, \cdots, y^{m+L-1}\}$ with the following form:

$$\mathcal{L}_{\mathrm{FC}} = \sqrt{\frac{1}{2mL - L^2 + L} \sum_{j=2}^{L} \sum_{t=m+1}^{m+j-1} \left((\hat{y}^t)_j - \mathrm{mean}(\hat{y}^t) \right)^2}, \tag{14}$$

where $(\hat{y}^t)_j$ $(t = m+1, m+2, \cdots, m+L-1)$ is the estimated future value of y at time point t by the sub-mapping function g_j, and $\mathrm{mean}(\hat{y}^t)$ $(t = 1,2,\cdots,m)$ is the average of all predicted values $(\hat{y}^t)_j$ for the same time point t. Minimizing the loss $\mathcal{L}_{\mathrm{FC}}$ ensures that predictions for different sub-mapping functions but corresponding to the same future time point t remain consistent, thus preserving temporal consistency and enhancing the stability of predictions.

In Eq. (12), the third part $\mathcal{L}_{\mathrm{REC}}$ is the reconstruction loss for latent representation S and the spatiotemporal information X. It contains two parts:

$$\mathcal{L}_{\mathrm{REC}} = \mathcal{L}_{\mathrm{REC\_X}} + \mathcal{L}_{\mathrm{REC\_S}}, \tag{15}$$

where $\mathcal{L}_{\mathrm{REC\_X}}$ is the reconstruction loss for the original information X, and $\mathcal{L}_{\mathrm{REC\_S}}$ is the reconstruction loss for the latent cause representation S, recovered by Y through the function f. Specifically,

$$\mathcal{L}_{\mathrm{REC\_X}} = \left|\left| X - \hat{X} \right|\right|_{\mathrm{F}}, \tag{16}$$

and

$$\mathcal{L}_{\mathrm{REC\_S}} = \left|\left| S - \hat{S} \right|\right|_{\mathrm{F}}, \tag{17}$$

where $X = [\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^m]$, $\hat{X} = [\hat{\mathbf{X}}^1, \hat{\mathbf{X}}^2, \cdots, \hat{\mathbf{X}}^m]$, $S = [\mathbf{S}^1, \mathbf{S}^2, \cdots, \mathbf{S}^m]$, and $\hat{S} = [\hat{\mathbf{S}}^1, \hat{\mathbf{S}}^2, \cdots, \hat{\mathbf{S}}^m]$, with $||\cdot||_{\mathrm{F}}$ representing the Frobenius norm. The loss $\mathcal{L}_{\mathrm{REC}}$ quantifies the information recovery for both the overall spatiotemporal data and the latent cause factor.

The last part $\mathcal{L}_{\mathrm{ORTH}}$ in Eq. (12) is the orthogonal loss for the causal representations S, Z, V defined as:

$$\mathcal{L}_{\mathrm{ORTH}} = \sum_{i=1}^{q} \sum_{j=1}^{q} \langle \mathbf{S}_i, \mathbf{Z}_j \rangle + \sum_{i=1}^{q} \sum_{k=1}^{q} \langle \mathbf{S}_i, \mathbf{V}_k \rangle + \sum_{j=1}^{q} \sum_{k=1}^{q} \langle \mathbf{Z}_j, \mathbf{V}_k \rangle, \tag{18}$$

where $\mathbf{S}_i = [s_i^1, s_i^2, \cdots, s_i^m]$, $\mathbf{Z}_j = [z_j^1, z_j^2, \cdots, z_j^m]$, $\mathbf{V}_k = [v_k^1, v_k^2, \cdots, v_k^m]$, and $\langle \cdot, \cdot \rangle$ is the inner

product operation. The loss $\mathcal{L}_{\mathrm{ORTH}}$ enforces orthogonality among the three causal representations, thus ensuring efficient use of the original information X and enhancing the robustness of the causal analysis and multistep forecasting.

Based on the loss function in Eq. (12) with the four important components, the network is trained in a self-supervised manner. The combination of $\mathcal{L}_{\mathrm{DS}}$ and $\mathcal{L}_{\mathrm{FC}}$ ensures accurate fitting of the nonlinear mapping g and robust prediction based on causal decomposition. The reconstruction loss $\mathcal{L}_{\mathrm{REC}}$ guarantees the consistency of the information recovery in the decomposition and reconstruction process. The orthogonal loss $\mathcal{L}_{\mathrm{ORTH}}$ facilitates the efficient exploitation of observed information X, and the reliable causal analysis for the target variable in the dynamic system. In the end, the $L-1$ future values $\{y^{m+1}, y^{m+2}, \cdots, y^{m+L-1}\}$ can be determined from the estimated delay embedding matrix $\hat{Y}$ as follows:

$$y^{m+i} = \mathrm{mean}\left(\hat{y}^{m+i}\right) = \frac{1}{L-i}\sum_{j=i+1}^{L}\left(\hat{y}^{m+i}\right)_j, \tag{19}$$

where $i = 1,2,\cdots,L-1$.

Step3. Infer the causal network based on $\alpha\beta$-LRP.

The CReP framework is capable of not only making multistep prediction for the target variable, but also discovering the causal interactions in the dynamic system. In CReP, causal representations S, Z, V are derived based on their causal relationships with the target variable y. However, these latent representations may be too abstract to interpret directly in a practical context. To uncover the causal links associated with y—the causes and effects—$\alpha\beta$-LRP is employed to explain the transformation mechanism H from X to S and Z. Specifically, for a trained encoder that maps the original information to the latent causal representations, $\alpha\beta$-LRP is utilized to compute the relevance scores of each input variable with respect to S and Z (see Fig. 1c in main text). This approach assesses the contribution of each variable to different causal representations. The procedure is as follows:

Given a trained neural network, where $\mathbf{X}^t = (x_1^t, x_2^t, \cdots, x_n^t)$ is mapped to $\mathbf{S}^t = (s_1^t, s_2^t, \cdots, s_q^t)$ and $\mathbf{Z}^t = (z_1^t, z_2^t, \cdots, z_q^t)$, the relevance scores of each variable x_k^t $(k = 1, 2, \cdots, n)$ to the output $\mathbf{S}^t$ and $\mathbf{Z}^t$ at time point t are calculated respectively. For a multioutput neural network, the relevance scores of x_k^t to all q output neurons $\{s_1^t, s_2^t, \cdots, s_q^t\}$ and $\{z_1^t, z_2^t, \cdots, z_q^t\}$ are averaged:

$$RS_{k,\mathbf{S}^t} = \frac{1}{q}\sum_{i=1}^{q} RS_{k,s_i^t}, \tag{20}$$

$$RS_{k,\mathbf{Z}^t} = \frac{1}{q}\sum_{i=1}^{q} RS_{k,z_i^t}, \tag{21}$$

where RS_{k,s_i^t} and RS_{k,z_i^t} denote the relevance scores of the k-th input variable to the output neurons s_i^t and z_i^t, $RS_{k,\mathbf{S}^t}$ and $RS_{k,\mathbf{Z}^t}$ denote the relevance scores of the k-th input variable to the vector outputs $\mathbf{S}^t$ and $\mathbf{Z}^t$. To aggregate the relevance scores across all time points, we then average $RS_{k,\mathbf{S}^t}$ and $RS_{k,\mathbf{Z}^t}$ for $t = 1,2,\cdots,m$ to calculate $RS_{k,S}$ and $RS_{k,Z}$, which refer to the contribution of the k-th input variable x_k to the cause representation S and the effect representation Z:

$$RS_{k,S} = \frac{1}{m}\sum_{t=1}^{m} RS_{k,\mathbf{S}^t}, \tag{22}$$

$$RS_{k,Z} = \frac{1}{m}\sum_{t=1}^{m} RS_{k,\mathbf{Z}^t}. \tag{23}$$

A higher value of $RS_{k,S}$ indicates a stronger causal link from x_k to the target variable y, while a higher value of $RS_{k,Z}$ indicates a stronger causal link from y to x_k. Consequently, the causal strength networks of y can be inferred by interpreting the latent causal representations S and Z from a practical perspective.

Supplementary Note 6. Datasets used in this study

Lorenz 96 system

The CReP framework was first applied to simulated models, one of which is a synthetic 60-dimensional coupled Lorenz 96 system, introduced by Edward Lorenz in 1996 [25]. It is a simplified mathematical model used to study atmospheric dynamics and chaotic behavior in weather prediction. The system consists of a set of ordinary differential equations for n variables that describe how a series of variables evolve over time with the following equations:

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F, \tag{24}$$

where $i = 1,2,\cdots,n$ and the parameter F is a constant external forcing term that drives the system. The value of F controls the level of external forcing applied to the system, and higher values of F introduce more turbulence and complexity. In the implementation of Lorenz 96 system, the initial values of all variables were set by a normal distribution and the time interval was set as $\Delta t = 0.03$. The level of external forcing was set as $F = 5$, and the data was collected after transient dynamics. In this work, CReP used the generated time-course datasets from Supplementary Eq. (24) to forecast the time evolution as illustration examples. For this dataset, $m = 50$ was used as the length of known time points and the next $L - 1 = 15$ steps of the target variable were forecasted.

Power grid system

The second simulated dataset used to validate the performance of our model is the power grid system modeled by Kuramoto equations, which is used to study synchronization phenomena in systems of coupled oscillators. It was introduced by Yoshiki Kuramoto in 1975 [26]. The model explores how a large number of oscillators, each with its own natural frequency, can synchronize when coupled together. We applied the CReP framework to a power grid system involving 120 units, which can be represented as follows:

$$\frac{d\theta_i}{dt} = \omega_i + \gamma \sum_{j=1}^{N} A_{ij} \sin(\theta_j - \theta_i), \tag{25}$$

where θ_i and ω_i are the phase and natural frequency of the i-th oscillator respectively, γ is the coupling strength, while pairwise interactions are encoded in the adjacency matrix A [27]. In certain coupling conditions, this type of system displays highly intricate chaotic behavior instead of achieving synchronization.

In the implementation of power grid, the initial values of all variables were set as zeros. The coupling strength was set as $\gamma = 0.4$, the time interval was set as $\Delta t = 0.02$, and the data was collected after transient dynamics. Additionally, to more accurately capture the important dynamic information within the data, we applied trigonometric transformations to the generated time-course datasets from Supplementary Eq. (25) and sampled with a step size of 5, thereby enhancing the accuracy and reliability of the predictions [27]. For this dataset, the length of known time points was set as $m = 30$, and the next $L - 1 = 9$ steps of the target variable were forecasted. The detailed forecasting results and causal analysis performance are illustrated in Supplementary Figure 3 and Figure 4.

Gene regulatory network

The DREAM4 (Dialogue for Reverse Engineering Assessments and Methods) dataset [28–30] is a benchmark for evaluating algorithms in gene regulatory network inference, featuring simulated gene expression time series data generated from known biological networks. This dataset, produced using GeneNetWeaver (GNW) [31], includes various challenges, each characterized by distinct network structures, noise levels, and the number of genes. Numerous researchers utilize DREAM4 to assess the performance of causal inference methods and enhance the understanding of underlying biological mechanisms. By integrating GNW's capabilities, DREAM4 provides a robust framework for testing and refining methodologies in systems biology.

In this work, we utilized the DREAM in silico dataset with 50 nodes $(n = 50)$, one of the DREAM challenges in GNW. The generated time series shows how the gene network responds to a perturbation and how it relaxes upon removal of the perturbation. This enables us to explore the complex relationships among biological gene expressions and make predictions for specific gene expression. The dataset was simulated using Stochastic Differential Equations (SDEs), with the noise coefficient set to 0.01. The time step for the time series in GNW was set as 1 during dataset generation. Due to the instability of the simulated time series data containing stochastic noise term, we performed a simple smoothing procedure on the dataset before applying the CReP method, using a moving average with a sliding window size of 5. The target variable was selected in this work as x_{18} due to its complex interaction profile. For this dataset, the past 40 time points $(m = 40)$ were treated as known information, while the future 8 time points $(L - 1 = 8)$ were designated for prediction.

Hongkong cardiovascular inpatients dataset

In addition to the simulated datasets, CReP has been applied to two real-world datasets, one of which is the Hongkong cardiovascular inpatients. This dataset encompasses multiple time series, including indices of air pollutants and the daily counts of cardiovascular inpatients at major hospitals in Hong Kong [32]. Based on the assumption that the number of cardiovascular inpatients is strongly correlated with air pollutant levels [33], the CReP method was employed to forecast inpatient numbers. For the 14-dimensional system ($n = 14$), the 70 determined-state time points were used ($m = 70$), and the 25 future values remained to be predicted ($L - 1 = 25$). Using daily concentrations of nitrogen dioxide (NO_2), sulphur dioxide (SO_2), ozone (O_3), respirable suspended particulates (Rspar), along with mean daily temperature and relative humidity data from air monitoring stations in Hong Kong (1994-1997), the CReP method aids in predicting the short-term dynamics of daily cardiovascular disease admissions.

COVID-19 spread dataset in Japan

COVID-19, as a highly contagious disease, had an initial reproduction number estimated at 6.47 [34]. Research consistently shows that early interventions—such as mask usage, social distancing, self-isolation, quarantine, and even large-scale lockdowns—are crucial in controlling, or at the very least, slowing down the transmission of the virus [35]. Consequently, predicting the spread of COVID-19 is critical for the implementation of timely public health measures designed to minimize the outbreak's severity and its further dissemination. The COVID-19 spread dataset contains a series of number representing COVID-19 daily new cases of 47 districts ($D = 47$) in Japan [36]. The CReP framework took the past $m = 30$ steps time series as known information and provided a 14-step-ahead prediction ($L - 1 = 14$) of COVID-19 case numbers in Tokyo. Fig. 5 in the main text and Supplementary Figure 4 have demonstrated the excellent accuracy of CReP. Moreover, the inferred causality of COVID-19 transmission with regard to the target variable is shown in Supplementary Figure 3.

Supplementary Note 7. The procedure of ablation study on loss function in CReP.

To further demonstrate the effectiveness of CReP framework, we conduct an ablation study by eliminating different loss terms from our training loss in the main text. The experiment results on three simulation datasets are presented in Table 2 in main text. The "without $\mathcal{L}_{\mathrm{REC}}$", "without $\mathcal{L}_{\mathrm{FC}}$" and "without $\mathcal{L}_{\mathrm{ORTH}}$" columns indicate that the reconstruction loss, the future-consistency loss and the orthogonal loss are eliminated from loss function during training CReP, respectively. Removing $\mathcal{L}_{\mathrm{REC}}$ prevents CReP from guaranteeing the recovery of original high-dimensional information. Without $\mathcal{L}_{\mathrm{FC}}$, the temporally self-constrained conditions are not enforced, and the removal of $\mathcal{L}_{\mathrm{ORTH}}$ means that CReP is prone to extracting overlapping or insufficient causal information from observed data.

In this ablation study, the model performance is evaluated from two perspectives: time series forecasting and causal analysis. Forecasting performance is assessed using the root mean square error (RMSE), while causal performance requires further quantification of inferred causal strengths to enable accuracy as a metric. To evaluate the causal results, we separated the cause variables and effect variables from the set of all input variables based on their relevance scores (contribution weights) to the corresponding representations, *i.e.*, S (cause representation) and Z (effect representation), as shown in Supplementary Figure 5. The separation standard is determined by the statistical significance of the relevance scores. Below, we illustrate this process using the cause variables identification as an example.

Step1. Score ranking.

For all input variables $\{x_1, x_2, \cdots, x_n\}$, let $\{RS_{\pi(1),S}, RS_{\pi(2),S}, \cdots, RS_{\pi(n),S}\}$ denote their relevance scores to the cause representation ordering in descending order, *i.e.*, $RS_{\pi(1),S} \geq RS_{\pi(2),S} \geq \cdots \geq RS_{\pi(n),S}$, where π denotes the permutation of indices corresponding to the sorted scores. A higher score implies a stronger causal influence on the target variable.

Step2. Candidate thresholds and grouping.

We considered several candidate thresholds, specifically the top 5%, 10%, 15% and 20% of the ranked scores, to distinguish between cause variables and non-cause variables. Formally, for a threshold $\tau \in \{0.05, 0.10, 0.15, 0.20\}$, the set of causes is defined as

$$\mathcal{C}(\tau) = \left\{ x_{\pi(i)} \,\middle|\, i \leq \lceil \tau \cdot n \rceil \right\}$$

Step3. Statistical significance testing.

To determine the optimal threshold, we applied the Mann-Whitney U test [37,38] to evaluate the significance of the difference between the medians of the two groups—cause variables and non-cause variables. This non-parametric test is appropriate as the relevance scores may not follow a normal distribution. For each candidate threshold τ, we performed the test with the null hypothesis H_0 stating that the two groups have identical distributions. The U statistic was used to compute the P-value, and we selected the threshold τ^* with the smallest P-value:

$$\tau^* = \underset{\tau}{\operatorname{argmin}}\, p(\tau)$$

The variables classified under the selected threshold $\mathcal{C}(\tau^*)$ were identified as causes of the target variable.

Step4. Identification of effect variables.

The same procedure was followed to determine the effect variables based on their relevance scores to the effect representation Z.

With the causes and effects identified, we then constructed a directed causal network centered around the target variable for evaluating (shown in Supplementary Figure 5). Each directed edge $e_{ij}: x_i \to x_j$ in the inferred network was treated as a sample. The edge was labeled as:

$$\text{label}(e_{ij}) = \begin{cases} 1, & \text{if the edge } x_i \to x_j \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

This enabled us to quantitatively assess the causal analysis results using metrics commonly employed in classification tasks. In the end, we evaluated the causal discovery performance using accuracy (ACC) and recall (REC), which are defined as follows:

$$\text{ACC} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}},$$

$$\text{REC} = \frac{\text{TP}}{\text{TP+FN}},$$

where TP, TN, FP and FN denote the numbers of true positives, true negatives, false positives, and false negatives, respectively. These two metrics provide a comprehensive assessment of the model's ability to distinguish between causal and non-causal relationships. Additionally, we evaluated the forecasting performance using the root mean squared error (RMSE) across all samples:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}^i - y^i)^2}$$

where $\hat{y}^i$ and y^i are the estimation and ground truth, respectively, at time point t. The RMSE

measures the accuracy of the model's predictions over the entire time series.

The results of our evaluations (see Table 2 in main text) demonstrate that the model trained with the full loss function achieves the best overall performance in both forecasting and causal analysis. This indicates that each component of the loss function in the CReP framework plays a vital role in enhancing both the accuracy of time-series forecasting and the robustness of causal analysis.

Supplementary Note 8. Methods for comparison in this study.

In this study, the forecasting performance of CReP was compared with other six reprentative methods as following.

Auto-reservoir neural network (ARNN)

Auto-reservoir neural network (ARNN) [18] is an auto-reservoir computing framework to efficiently and accurately make multi-step-ahead predictions for short-term high-dimensional time series. It is based on Takens' delay embedding theorem and establishes ARNN-based spatiotemporal information (STI) equations. By integrating the RC structure with STI equations, ARNN effectively leverages the rich spatial information inherent in high-dimensional data, thereby enabling robust and efficient multi-step forecasting of future values.

Spatiotemporal information conversion machine (STICM)

Spatiotemporal information conversion machine (STICM) [17] is a neural network computing framework to efficiently and accurately render a forecasting of a time series by employing a spatial-temporal information (STI) transformation. By combining the STI equations and Granger causality, STICM fully explores the nonlinear features and spatiotemporal causal relations of the observed high-dimensional variables. Therefore, STICM not only provides accurate multi-step forecasts but also infers the causal factors of the target variable in the sense of Granger causality.

Long short-term memory network (LSTM)

Long short-term memory (LSTM) [39] is a extensively adopted neural network-based approach, particularly effective for processing sequential and temporal data. Built upon a refined recurrent neural network (RNN) architecture, LSTM networks are specifically designed to capture long-term dependencies in time series data. A typical LSTM unit consists of a memory cell along with three key gates: the input gate, the output gate, and the forget gate. The memory cell is responsible for retaining historical information, while the gates regulate the flow of information into, out of, and within the cell. This gating mechanism enables LSTM networks to selectively preserve or discard information, effectively addressing the vanishing gradient problem common in traditional RNNs.

Autoregressive integrated moving average (ARIMA)

The autoregressive integrated moving average (ARIMA) [40] is a well-established univariate approach for time series forecasting. It effectively addresses various characteristics of temporal data, including trends, seasonality, cyclical patterns, random errors, and non-stationarity. The ARIMA model is characterized by three parameters, denoted as ARIMA(p, d, q), which correspond to the autoregressive (AR), integrated (I), and moving average (MA) components, respectively. The general form of the ARIMA(p, d, q) model is expressed as:

$$\Delta^d(z^t) = \mu + \beta_1\Delta^d(z^{t-1}) + \cdots + \beta_p\Delta^d(z^{t-p}) + \varepsilon^t + \theta_1\varepsilon^{t-1} + \cdots + \theta_q\varepsilon^{t-q},$$

where $\Delta^d(z^t)$ represents the dth-order differenced value of the time series at time t, addressing non-stationarity. The term ε^t denotes the error at time t. The coefficients β_i correspond to the autoregressive component, while θ_i represent the moving average component of the model.

Support vector regression (SVR)

Support vector regression (SVR) employs the principles of support vector machines to perform curve fitting and regression analysis [41,42]. As a multivariate method, it predicts future values using the following function:

$$f(x) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)G(x_n, x) + b,$$

where x_n is a multivariate set of N observations with observed values x and $G(x_n, x) = e^{-|x_n - x|^2}$.

Informer

Informer [43] is a neural network model specifically designed for efficient and scalable time series forecasting. It addresses the limitations of traditional Transformer models, particularly their high computational complexity when handling long sequences. By introducing the ProbSparse self-attention mechanism, Informer significantly reduces computation costs while maintaining forecasting accuracy. Additionally, it incorporates a generative style decoder and distillation techniques to enhance long-term forecasting performance. These features make Informer particularly effective for multi-step prediction tasks in large-scale time series data.

Supplementary References

1. Takens, F. Detecting strange attractors in turbulence. in *Dynamical Systems and Turbulence, Warwick 1980* (eds. Rand, D. & Young, L.-S.) vol. 898 366–381 (Springer Berlin Heidelberg, Berlin, Heidelberg, 1981).
2. Sauer, T., Yorke, J. A. & Casdagli, M. Embedology. *J. Stat. Phys.* **65**, 579–616 (1991).
3. Ma, H., Zhou, T., Aihara, K. & Chen, L. Predicting Time Series from Short-Term High-Dimensional Data. *Int. J. Bifurc. Chaos* **24**, 1430033 (2014).
4. Deyle, E. R. & Sugihara, G. Generalized Theorems for Nonlinear State Space Reconstruction. *PLoS ONE* **6**, e18295 (2011).
5. Stark, J., Broomhead, D. S., Davies, M. E. & Huke, J. Takens embedding theorems for forced and stochastic systems. *Nonlinear Anal. Theory Methods Appl.* **30**, 5303–5314 (1997).
6. Stark, J., Broomhead, D. S., Davies, M. E. & Huke, J. Delay Embeddings for Forced Systems. II. Stochastic Forcing. *J. Nonlinear Sci.* **13**, 519–577 (2003).
7. Shi, J., Chen, L. & Aihara, K. Embedding entropy: a nonlinear measure of dynamical causality. *J. R. Soc. Interface* **19**, 20210766 (2022).
8. Yan, J. *et al.* Dynamical causality under invisible confounders. Preprint at https://doi.org/10.48550/ARXIV.2408.05584 (2024).
9. Ma, H., Leng, S., Aihara, K., Lin, W. & Chen, L. Randomly distributed embedding making short-term high-dimensional data predictable. *Proc. Natl. Acad. Sci.* **115**, (2018).
10. Chen, C. *et al.* Predicting future dynamics from short-term time series using an Anticipated Learning Machine. *Natl. Sci. Rev.* **7**, 1079–1091 (2020).
11. Shi, J., Chen, L. & Aihara, K. Embedding entropy: a nonlinear measure of dynamical causality.

J. R. Soc. Interface **19**, 20210766 (2022).

12. Ma, H., Leng, S. & Chen, L. Data-based prediction and causality inference of nonlinear dynamics. *Sci. China Math.* **61**, 403–420 (2018).
13. Butler, K., Feng, G. & Djurić, P. M. On Causal Discovery With Convergent Cross Mapping. *IEEE Trans. Signal Process.* **71**, 2595–2607 (2023).
14. Gao, B. *et al.* Causal inference from cross-sectional earth system data with geographical convergent cross mapping. *Nat. Commun.* **14**, 5875 (2023).
15. Runge, J. *et al.* Inferring causation from time series in Earth system sciences. *Nat. Commun.* **10**, 2553 (2019).
16. Sugihara, G. *et al.* Detecting Causality in Complex Ecosystems. *Science* **338**, 496–500 (2012).
17. Peng, H., Chen, P., Liu, R. & Chen, L. Spatiotemporal information conversion machine for time-series forecasting. *Fundam. Res.* S2667325822004538 (2022) doi:10.1016/j.fmre.2022.12.009.
18. Chen, P., Liu, R., Aihara, K. & Chen, L. Autoreservoir computing for multistep ahead prediction based on the spatiotemporal information transformation. *Nat. Commun.* **11**, 4568 (2020).
19. Sun, J., Zhou, S. & Veeramani, D. A neural network-based control chart for monitoring and interpreting autocorrelated multivariate processes using layer-wise relevance propagation. *Qual. Eng.* **35**, 33–47 (2023).
20. Bach, S. *et al.* On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE* **10**, e0130140 (2015).
21. Achtibat, R. *et al.* From attribution maps to human-understandable explanations through Concept Relevance Propagation. *Nat. Mach. Intell.* **5**, 1006–1019 (2023).

22. Berrone, S., Della Santa, F., Mastropietro, A., Pieraccini, S. & Vaccarino, F. Layer-wise relevance propagation for backbone identification in discrete fracture networks. *J. Comput. Sci.* **55**, 101458 (2021).

23. Kim, D. *et al.* Untangling the contribution of input parameters to an artificial intelligence PM2.5 forecast model using the layer-wise relevance propagation method. *Atmos. Environ.* **276**, 119034 (2022).

24. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).

25. Lorenz, E. N. Predictability: A Problem Partly Solved. in *Proceedings of the Seminar on Predictability* vol. 1 1–18 (European Centre for Medium-Range Weather Forecasts (ECMWF), Reading, UK, 1996).

26. Kuramoto, Y. Self-entrainment of a population of coupled non-linear oscillators. in *International Symposium on Mathematical Problems in Theoretical Physics* (ed. Araki, H.) vol. 39 420–422 (Springer-Verlag, Berlin/Heidelberg, 1975).

27. Li, X. *et al.* Higher-order Granger reservoir computing: simultaneously achieving scalable complex structures inference and accurate dynamics prediction. *Nat. Commun.* **15**, 2506 (2024).

28. Stolovitzky, G., Monroe, D. & Califano, A. Dialogue on Reverse-Engineering Assessment and Methods: The DREAM of High-Throughput Pathway Inference. *Ann. N. Y. Acad. Sci.* **1115**, 1–22 (2007).

29. Stolovitzky, G., Prill, R. J. & Califano, A. Lessons from the DREAM2 Challenges: A Community Effort to Assess Biological Network Inference. *Ann. N. Y. Acad. Sci.* **1158**, 159–195 (2009).

30. Marbach, D., Schaffter, T., Mattiussi, C. & Floreano, D. Generating Realistic *In Silico* Gene Networks for Performance Assessment of Reverse Engineering Methods. *J. Comput. Biol.* **16**, 229–239 (2009).

31. Schaffter, T., Marbach, D. & Floreano, D. GeneNetWeaver: *in silico* benchmark generation and performance profiling of network inference methods. *Bioinformatics* **27**, 2263–2270 (2011).

32. Wong, T. W. *et al.* Air pollution and hospital admissions for respiratory and cardiovascular diseases in Hong Kong. *Occup. Environ. Med.* **56**, 679–683 (1999).

33. Xia, Y. & Härdle, W. Semi-parametric estimation of partially linear single-index models. *J. Multivar. Anal.* **97**, 1162–1184 (2006).

34. Tang, B. *et al.* Estimation of the Transmission Risk of the 2019-nCoV and Its Implication for Public Health Interventions. *J. Clin. Med.* **9**, 462 (2020).

35. Hellewell, J. *et al.* Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts. *Lancet Glob. Health* **8**, e488–e496 (2020).

36. Wahltinez, O. COVID-19 Open-Data: curating a fine-grained, global-scale data repository for SARS-CoV-2. (2020).

37. Mann, H. B. & Whitney, D. R. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Ann. Math. Stat.* **18**, 50–60 (1947).

38. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biom. Bull.* **1**, 80 (1945).

39. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **9**, 1735–1780 (1997).

40. Box, G. E. P. & Pierce, D. A. Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *J. Am. Stat. Assoc.* **65**, 1509–1526 (1970).

41. *Support Vector Machines: Theory and Applications*. vol. 177 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005).

42. Tong, H. & Ng, M. K. Calibration of $\epsilon -$ insensitive loss in support vector machines regression. *J. Frankl. Inst.* **356**, 2111–2129 (2019).

43. Zhou, H. *et al.* Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proc. AAAI Conf. Artif. Intell.* **35**, 11106–11115 (2021).