entropy

MDPI

Article

Neural Information Squeezer for Causal Emergence

Jiang Zhang [1,2,*] **and Kaiwei Liu** [1]

[1] School of Systems Sciences, Beijing Normal University, Beijing 100875, China
[2] Swarma Research, Beijing 100085, China
* Correspondence: zhangjiang@bnu.edu.cn

Abstract: Conventional studies of causal emergence have revealed that stronger causality can be obtained on the macro-level than the micro-level of the same Markovian dynamical systems if an appropriate coarse-graining strategy has been conducted on the micro-states. However, identifying this emergent causality from data is still a difficult problem that has not been solved because the appropriate coarse-graining strategy can not be found easily. This paper proposes a general machine learning framework called Neural Information Squeezer to automatically extract the effective coarse-graining strategy and the macro-level dynamics, as well as identify causal emergence directly from time series data. By using invertible neural network, we can decompose any coarse-graining strategy into two separate procedures: information conversion and information discarding. In this way, we can not only exactly control the width of the information channel, but also can derive some important properties analytically. We also show how our framework can extract the coarse-graining functions and the dynamics on different levels, as well as identify causal emergence from the data on several exampled systems.

Keywords: causal emergence; coarse-graining; invertible neural network

1. Introduction

Emergence, as one of the most important concepts in complex systems, describes the phenomenon that some overall properties of a system cannot be reduced to the parts [1,2]. Causality, as another significant concept, characterises the connection between cause and effect events through time [3,4] for a dynamical system. As pointed out by Hoel et al. [5,6], causality could be emergent, which means that the events of a system on the macro level may have stronger causal connections than the micro level, where the strength of causality could be measured by effective information (EI) [5,7]. This theoretical framework of causal emergence provides us a new way to understand emergence and other important conceptions in a quantitative way [8–11]. Previous works have shown that causal emergence can be applied in wide areas including studies on ant colony [12], protein interactomes [13], brain [14], and biological networks [15].

Although many concrete examples of causal emergence across different temporal and spatial scales have been shown in [5], a method to identify causal emergence merely from data is needed. This problem is hard because a method to search all possible coarse-graining strategies (functions, mappings) in a systematic and automatic way, in which the causal emergence can be shown [10], is needed.

The problem is difficult because the search space is all possible mapping functions between micro and macro, which is huge. To solve this problem, Klein et al. focuses on the complex systems with network structures [15,16], and converted the problem of coarse-graining into node clustering. That is, to find a way to group nodes into clusters such that the connections on the cluster level has larger EI than the original network. This method has been widely applied in various areas [12–14], nevertheless, it assumes that the underlying node dynamic is diffusion (random walks). Meanwhile, real complex systems have much richer node dynamics. For a general dynamic, even if the node grouping is

check for updates

Citation: Zhang, J.; Liu, K. Neural Information Squeezer for Causal Emergence. *Entropy* **2023**, *25*, 26. https://doi.org/10.3390/e25010026

Academic Editor: José F. F. Mendes

Received: 25 October 2022
Revised: 19 December 2022
Accepted: 19 December 2022
Published: 23 December 2022

given, the coarse-grained strategy still needs to consider how to map the micro-states of all nodes in a cluster to the macro-state of the cluster [5]. The tedious searching on a huge functional space of coarse-graining strategies is also needed.

When we consider all possible mappings, another difficulty is trivial coarse-graining strategies avoiding. An exampled trivial method is to map all the micro-states into an identical value as the macro-state. In this way, the macroscopic dynamics is only an identical mapping that will have large effective information (EI) measure. However, this can not be called causal emergence because all the information is eliminated by the coarse-graining method itself. Thus, we must find a way to exclude such trivial strategies.

An alternative way to identify causal emergence from data is based on partial information decomposition given by [10,17]. Although the methods based on information decomposition can avoid the discussion on coarse-graining strategies, a time consuming search on subsets of the system state space is also needed if we want to obtain the exact value. In addition, the reported numeric approximate method can only provide sufficient condition. Further, the method can not give the explicit coarse-graining strategy and the corresponding macro-dynamics which are useful in practice. Another common shortage shared by the two mentioned methods is that an explicit Markov transition matrix for both macro- and micro-dynamics are needed, and the transitional probabilities should be estimated from data. As a result, large bias on rare events can hardly be avoided, particularly for continuous data.

On the other hand, machine learning methods empowered by neural networks have been developed in recent years, and many cross-disciplinary applications have been made [18–21]. Equipped with this method, automated discovery of causal relationships and even dynamics of complex systems in a data driven way becomes possible [22–29]. Machine learning and neural networks can also help us to find good coarse-graining strategies [30–34]. If we treat a coarse-graining mapping as a function from micro-states to macro-states, then we can certainly approximate this function by a parameterized neural network. For example, Refs. [31,33] used normalized flow model equipped with invertible neural network to learn how to renormalize a multi-dimensional field (quantum field, images or joint probability distributions), and how to generate the field from Gaussian noise. Therefore, both the coarse-graining strategy and the generative model can be learned from data automatically.

These techniques can also help us to reveal causality on macro-level from data. Causal representation learning aims to use unsupervised representation learning to extract causal latent variables behind the observational data [35,36]. The encoding process from the original data to the latent causal variables can be understood as a kind of coarse-graining. This shows the similarity between causal emergence identification and causal representation learning; however, their basic objectives are different. Causal representation learning aims to extract the causality hidden in data, whereas causal emergence identification aims to find a good strategy of coarse-graining to reduce the given micro-level dynamics. Furthermore, introducing multi-scale modeling and coarse-graining operations into causal models brings some new theoretical problems [37–39]. For example, Refs. [38,39] discuss the basic requirements of the model abstraction (coarse-graining). However, these studies only care about static random variables and structural causal models but not Markovian dynamics.

In this paper, we formulate the problem of causal emergence identification as a maximization problem of the effective information (EI) for the macro-dynamics under the constraint of precise prediction of micro-dynamics. We then propose a general machine learning framework called Neural Information Squeezer (NIS) to solve the problem. By using invertible neural network to model the coarse-graining strategy, we can decompose any mapping from $\mathcal{R}^p$ to $\mathcal{R}^q$ ($q \leq p$) into a series of information conversions invertible processes and information discarding processes. In this way, the framework can not only allow us to control information conversion and discarding in a precise way but also enable us to mathematically analyze the whole framework in theory. We prove a series of mathematical

theorems to reveal the properties of NIS. At last, we show how NIS can learn effective coarse-graining strategies and macro-state dynamics numerically on a set of examples.

2. Basic Notions and Problems Formulation

First, we will formulate our problems under a general setting, and layout our framework to solve the problems.

2.1. Background

Suppose the dynamics of the complex system that we consider can be described by a set of differential equations.

$$\frac{d\mathbf{x}}{dt} = g(\mathbf{x}(t), \xi), \tag{1}$$

where $\mathbf{x}(t) \in \mathcal{R}^p$ is the state of the system and $p \in \mathcal{Z}^+$ is a positive integer, ξ is a gaussian random noise. Normally, micro-dynamic g is always Markovian which means it could be also modeled as a conditional probability $Pr(\mathbf{x}(t+dt)|\mathbf{x}(t))$ equivalently.

However, we cannot directly obtain the evolution of the system but the discrete samples of the states, and we define these states as micro-states.

Definition 1. *(Micro-states): Each sample of the state of the dynamical system (Equation (1)) $\mathbf{x}_t$ is called a micro-state at time step t. In addition, the multi-variate time series $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T$ which are sampled with equal intervals and a finite time step T, forms a micro-state time series.*

We always want to reconstruct g according to the observable micro-states. However, an informative dynamical mechanism g with strong causal connections is always hard to be reconstructed from the micro-states when noise ξ is strong. Meanwhile, we can ignore some information in the micro-state data and convert it into macro-state time series. In this way, we may reconstruct a macro-dynamic with stronger causality to describe the evolution of the system. This is the basic idea behind causal emergence [5,6]. We formalize the information ignoring process as a coarse-graining strategy (or mapping, method).

Definition 2. *(q dimensional coarse-graining strategy): Suppose the dimension of the macro-states is $0 < q < p \in \mathcal{Z}^+$, a q dimensional coarse-graining strategy is a continuous and differential function to map the micro-state $\mathbf{x}_t \in \mathcal{R}^p$ to a macro-state $\mathbf{y}_t \in \mathcal{R}^q$. The coarse-graining is denoted as ϕ_q.*

After coarse-graining, we obtain a new time series data of macro-states denoted by $\mathbf{y}_1 = \phi_q(\mathbf{x}_1), \mathbf{y}_2 = \phi_q(\mathbf{x}_2), \cdots, \mathbf{y}_T = \phi_q(\mathbf{x}_T)$. We then try to find another dynamical model (or a Markov chain) $\hat{f}_{\phi_q}$ to describe the evolution of $\mathbf{y}_t$:

Definition 3. *(macro-state dynamics): For the given time series of macro-states $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T$, the macro-state dynamics are a set of differential equations*

$$\frac{d\mathbf{y}}{dt} = \hat{f}_{\phi_q}(\mathbf{y}, \xi'), \tag{2}$$

where $\mathbf{y} \in \mathcal{R}^q$, $\xi' \in \mathcal{R}^q$ is the gaussian noise in the macro-state dynamics, and $\hat{f}_{\phi_q}$ is a continuous and differential function such that the solution of Equation (2), $\mathbf{y}(t)$, can minimize:

$$\langle ||\mathbf{y}_t - \mathbf{y}(t)|| \rangle_{\xi'} \tag{3}$$

for any given time step $t \in [1, T]$ and given vector form $||\cdot||$.

However, this formulation cannot reject some trivial strategies. For example, suppose a $q = 1$ dimensional ϕ_q is defined as $\phi_q(\mathbf{x}_t) = 1$ for $\forall \mathbf{y}_t \in \mathcal{R}^p$. Thus, the corresponding

macro-dynamic is simply $d\mathbf{y}/dt = 0$ and $\mathbf{y}(0) = 1$. However, this is meaningless because the macro-state dynamic is trivial and coarse-graining mapping is too arbitrary.

Therefore, we must set limitations on coarse-graining strategies and macro-dynamics so that such trivial strategies and dynamics could be avoided.

2.2. Effective Coarse-Graining Strategy and Macro-Dynamics

We define an effective coarse-graining strategy to be a compressed map such that the macro-states may preserve the information of micro-states as much as it can. Formally,

Definition 4. *(ϵ-effective q coarse-graining strategy and macro-dynamcis): A q coarse-graining strategy $\phi_q : \mathcal{R}^p \to \mathcal{R}^q$ is ϵ-effective (or abbreviate as effective) if there exists a function $\phi_q^\dagger : \mathcal{R}^q \to \mathcal{R}^p$, such that the following inequality holds for a given small real number ϵ and given vector norm $||\cdot||$:*

$$||\phi_q^\dagger(\mathbf{y}(t)) - \mathbf{x}_t|| < \epsilon, \tag{4}$$

and the derived macro-dynamic $\hat{f}_{\phi_q}$ is also ϵ-effective, where $\mathbf{y}(t)$ is the solution of Equation (2), that is:

$$\mathbf{y}(t) = \phi_q(\mathbf{x}_{t-1}) + \int_{t-1}^{t} \hat{f}_{\phi_q}(\mathbf{y}(\tau), \xi')d\tau \tag{5}$$

for all $t = 1, 2, ..., T$. Thus, we can reconstruct the micro-state time series by $\phi_q^\dagger$ such that the macro-state variables contain the information of micro-states as much as they can.

Notice that this definition is in accordance with the approximate causal model abstraction [40].

2.3. Problem Formulation

Our final objective is to find a most informative macro-dynamic. Therefore, we need to optimize the coarse-graining strategy and the macro-dynamic among all possible effective strategies and dynamics. Therefore, our problem can be formulated as:

$$\max_{\phi_q, \hat{f}_{\phi_q}, \phi_q^\dagger, q} \mathcal{I}(\hat{f}_{\phi_q}), \tag{6}$$

under the constraint Equations (4) and (5), where $\mathcal{I}$ is a measure of effective information, it could be EI, Eff, or dimension averaged EI which is mainly used in this paper and is denoted as dEI (will mention in Section 3.3.3). ϕ_q is an effective coarse-graining strategy, and $\hat{f}_{\phi_q}$ is an effective macro-dynamic.

3. Methods

The problem (Equations (6) and (4)) is hard to solve because the objects that we will optimize are functions: $\phi_q, \hat{f}_{\phi_q}, \phi_q^\dagger$ but not numbers. Thus, we use neural networks to parameterize the functions and convert the function optimization problem into a parameter optimization problem.

3.1. Neural Information Squeezer Model

We propose a new machine learning framework called neural information squeezer (NIS) which is based on invertible neural network to solve the problem (Equation (6)). NIS is composed of three components: encoder, dynamics learner, and decoder. They are represented by neural networks ψ_α, f_β, and ψ_α^{-1} with the parameters α, β, and α respectively. The entire framework is shown in Figure 1. Next, we will describe each module separately.

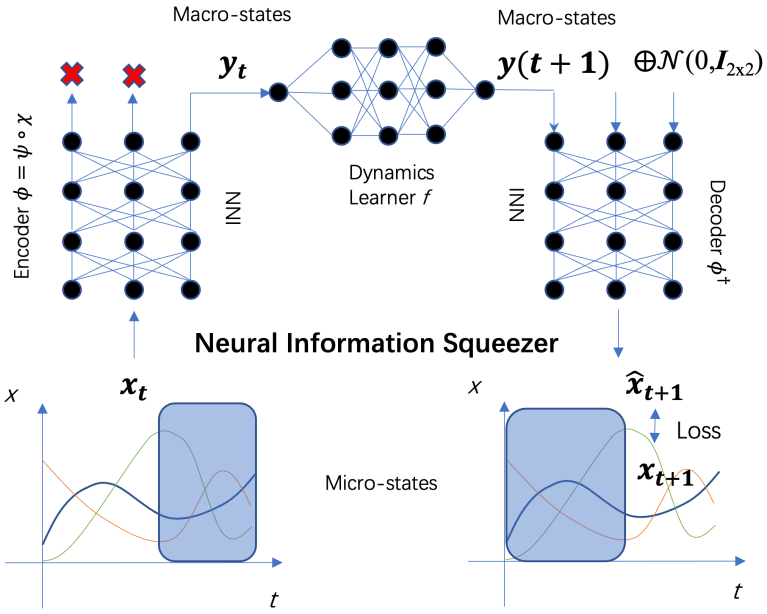


Figure 1. The workflow and the framework of the neural information squeezer. x_t is the data at time t. Encoder ψ_α is an invertible neural network (INN), from which the coarse-grained data y_t can be generated. The dynamics learner f_β is a common feed-forward neural network with parameters β. Through it the evolution from y_t to $y(t+1)$ can be conducted. The decoder converts the predicted macro-state of the next time step $\mathbf{y}(t+1)$ into the prediction of the micro-state at the next time step $\hat{\mathbf{x}}_{t+1}$.

Encoder

To be noticed, ψ_α is an invertible neural network (INN), therefore ψ and ψ^{-1} share the parameters α. However, invertible function has no information loss, we must introduce a new operator, projection.

Definition 5. *(Projection operator): A projection operator $\chi_{p,q}$ is a function from $\mathcal{R}^p$ to $\mathcal{R}^q$, such that:*

$$\chi_{p,q}(\mathbf{x}_q \bigoplus \mathbf{x}_{p-q}) = \mathbf{x}_q, \tag{7}$$

where $\bigoplus$ is the operation of vector concatenation, and $\mathbf{x}_q \in \mathcal{R}^q, \mathbf{x}_{p-q} \in \mathcal{R}^{p-q}$. Sometimes, we abbreviate $\chi_{p,q}$ as χ_q if there is no ambiguity.

Thus, the encoder (ϕ) maps the micro-state $\mathbf{x}_t$ to the macro-state $\mathbf{y}_t$, and this mapping can be separated into two steps. That is,

$$\phi_q = \chi_q \circ \psi_\alpha, \tag{8}$$

where $\circ$ represents the operation of function composition.

The first step is a bijective (invertible) mapping $\psi_\alpha : \mathcal{R}^p \to \mathcal{R}^p$ from $\mathbf{x}_t \in \mathcal{R}^p$ to $\mathbf{x}'_t \in \mathcal{R}^p$ without information lose and is realized by an invertible neural network, the second step is to project the resulting vector to q dimension by mapping $\mathbf{x}'_t \in \mathcal{R}^p$ into $\mathbf{y}_t \in \mathcal{R}^q$ by discarding the information on $p-q$ dimension.

There are several ways to realize an invertible neural network [41,42]. Meanwhile, we select RealNVP module [43] as shown in Figure 2 to concretely implement the invertible computation.

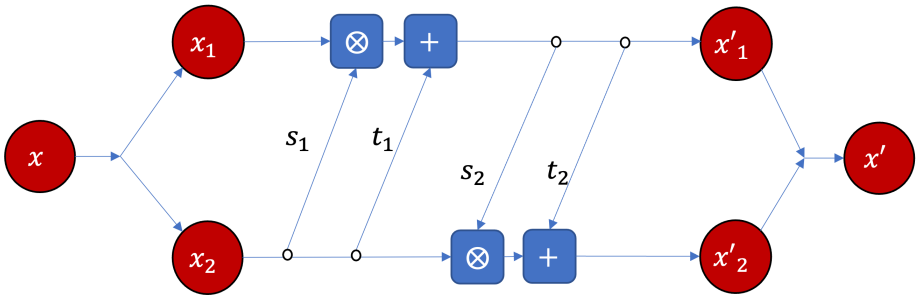


Figure 2. The RealNVP neural network implementation of the basic module of the bijector ψ, where s_1, s_2 and t_1, t_2 are all feed-forward neural networks with three layers, 64 hidden neurons, and ReLU active function. s_is and t_is share parameters, respectively. $\otimes$ and $+$ represent element-wised product and addition, respectively. $\mathbf{x} = \mathbf{x}_1 \bigoplus \mathbf{x}_2$ and $\mathbf{x}' = \mathbf{x}'_1 \bigoplus \mathbf{x}'_2$.

In the module, the input vector $\mathbf{x}$ can be separated into two parts, both vectors will be scaled, translated and merged again. The magnitude of the scaling and translation operations will be adjusted by the corresponding feed-forward neural networks. s_1, s_2 are the same neural networks shared parameters for scaling, $\otimes$ represents element-wised product. In addition, t_1, t_2 are the neural networks shared parameters for translation. In this way, an invertible computation from $\mathbf{x}$ to $\mathbf{y}$ can be realized. The same module can be repeated for multiple times (three times in this paper) to realize complex invertible computation, the details can be referred to Appendix A.

The reasons why we use invertible neural network are: (1) INN can reduce the complexity of the model by multiplexing the structure and the parameters in the encoder to the decoder because we can simply reverse the running direction of the encoder to implement decoding; (2) the encoder equipped with INN can separate out the information conversion process and information discarding process; (3) this enables us to do mathematical analysis on the whole framework, and several theorems reflecting the basic properties can be proved.

3.2. Decoder

The decoder converts the predicted macro-state of the next time step $\mathbf{y}(t+1)$ into the prediction of the micro-state at the next time step $\hat{\mathbf{x}}_{t+1}$. In our framework, because the coarse-graining strategy ϕ_q can be decomposed as a bijector ψ_α and a projector χ_q, we can simply reverse ψ_α to become ψ_α^{-1} as the decoder. However, because the dimension of the macro-state is q and the input dimension of ψ_α is $p > q$, we need to fill the remaining $p - q$ dimensions by a $p - q$ dimensional Gaussian random vector. That is, for any ϕ_q, the decoding mapping can be defined as:

$$\phi_q^\dagger = \psi_\alpha^{-1} \circ \chi_q^\dagger, \tag{9}$$

where ψ_α^{-1} is the inverse function of ψ_α, and $\chi_q^\dagger : \mathcal{R}^q \to \mathcal{R}^p$ is a function defined as follow: for any $\mathbf{x}_q \in \mathcal{R}^p$

$$\chi_q^\dagger(\mathbf{x}_q) = \mathbf{x}_q \bigoplus \mathbf{z}_{p-q}, \tag{10}$$

where $\mathbf{z}_{p-q} \sim \mathcal{N}(0, \mathcal{I}_{p-q})$ is a random Gaussian noise with $p - q$ dimension, and $\mathcal{I}_{p-q}$ is an identity matrix with the same dimension. That is, we can generate a micro-state by composing $\mathbf{x}_q$ and a random sample $\mathbf{z}_{p-q}$ from a $p - q$ dimensional standard normal distribution.

According to the point view of [31,33], the decoder can be regarded as a generative model of the conditional probability $Pr(\hat{\mathbf{x}}_{t+1}|\mathbf{y}(t+1))$, and the encoder just performs a renormalization process.

Dynamics Learner

The dynamics learner f_β is a common feed-forward neural network with parameters β, it will learn the effective Markov dynamic on the macro-level. Concretely, we at first use f_β to replace $\hat{f}_{\phi_q}$ in Equation (2), and second we use Euler method with $dt = 1$ to solve the Equation (2), and suppose the noise is a additive Gaussian (or Laplacian) [44], therefore we can reduce Equation (5) as:

$$\mathbf{y}(t+1) = \mathbf{y}_t + \int_t^{t+1} f_\beta(\mathbf{y}(\tau), \xi')d\tau \approx \mathbf{y}_t + f_\beta(\mathbf{y}_t) + \xi' \tag{11}$$

where $\xi' \sim \mathcal{N}(0, \Sigma)$ or $Laplacian(0, \Sigma)$, $\Sigma = diag(\sigma_1^2, \sigma_2^2, \cdots, \sigma_q^2)$ is the covariance matrix, and σ_i is the standard deviation in the ith dimension which could be learned or fixed. Thus, the transitional probability of this dynamics can be written as

$$P(\mathbf{y}(t+1)|\mathbf{y}_t) = \mathcal{D}(\mu(\mathbf{y}_t), \Sigma), \tag{12}$$

where $\mathcal{D}$ represents the PDF of Gaussian distribution or Laplace distribution, $\mu(\mathbf{y}_t) \equiv \mathbf{y}_t + f_\beta(\mathbf{y}_t)$ is the mean vector of the distribution.

By training the dynamics learner in an end-to-end manner, we can avoid estimating the Markov transitional probabilities from the data to reduce biases because neural networks always have much better ability to fit the data and generalize to unseen cases.

3.3. Two Stage Optimization

Although the functions that will be optimized have been parameterized by neural networks, Equation (6) is still hard to be optimized directly because the objective function and the constraint condition must be combined together to be considered and q as a hyper-parameter can affect the structure of neural networks. Thus, in this paper, we propose a two-stage optimization method. In the first stage, we fix the hyper-parameter q and optimize the difference between the predicted micro-state and the observed data $|\phi_q^\dagger(\mathbf{y}(t)) - \mathbf{x}_t|$, that is Equation (4), to let the coarse-graining strategy ϕ_q and macro-dynamics $\hat{f}_q$ to be effective. Then, we search for all possible q values to find the optimal one such that $\mathcal{I}$ can be maximized.

3.3.1. Stage 1: Training a Predictor

In the first stage, we can use likelihood maximization and stochastic gradient descend techniques to obtain the effective q coarse-graining strategy and the effective predictor of the macro-state dynamics. The objective function is defined on the likelihood of micro-state prediction.

We can understand a feed-forward neural network as a machine to model a conditional probability with Gaussian or Laplacian distribution [44]. Thus, the entire NIS framework can be understood as a model of $P(\hat{\mathbf{x}}_{t+dt}|\mathbf{x}_t)$ with the output $\hat{\mathbf{x}}_{t+1}$ is just the mean value. In addition, the objective function Equation (14) is just the log-likelihood or cross-entropy of the observed data under the given form of the distribution.

$$\mathcal{L} = \sum_t \ln P(\hat{\mathbf{x}}_{t+1} = \mathbf{x}_{t+1}|\mathbf{x}_t), \tag{13}$$

where $P(\hat{\mathbf{x}}_{t+1} = \mathbf{x}_{t+1}|\mathbf{x}_t) \equiv \mathcal{N}(\hat{\mathbf{x}}_{t+1}, \Sigma)$ when $l = 2$ or $Laplace(\hat{\mathbf{x}}_{t+1}, \Sigma)$ when $l = 1$, where Σ is the covariance matrix which is always be a diagonal matrix and the magnitude can be calculated as the mean square error for $l = 2$ or mean absolute value for $l = 1$.

If we take the concrete form of Gaussian or Laplacian distribution into the conditional probability, we will see to maximize the log-likelihood is equivalent to minimize the l-norm objective function:

$$\mathcal{L} = \sum_t ||\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}||_l \tag{14}$$

where $l = 1$ or 2.

Then, we can use stochastic gradient descend technique to optimize Equation (14).

3.3.2. Stage 2: Search for the Optimal Scale

In the previous step, we can obtain the effective q coarse graining strategy and the macro-state dynamics after a large number of training epochs, but the results are dependent on q.

To select the optimized q, we can compare the measure of effective information $\mathcal{I}$ for different q coarse-graining macro-dynamics. Because the parameter q only has one dimension, and its value range is also limited ($0 < q < p$), we can simply iterate all q to find out the optimal q^* and the optimal effective strategy.

3.3.3. About Effective Information

In the second stage, to compare coarse-graining strategies and macro-dynamics, we need to compute the important indicator: effective information (EI); however, the conventional computations of EIs are all for discrete Markov dynamics in most of previous works [5,6], and we may confront difficulties when we apply EI on continuous dynamics [9].

First, the conventional methods on mutual information computation for discrete variables cannot be used here, new methods for continuous variables and mappings especially for high dimensional space must be invented. To solve the problem, we treat the mapping of the dynamics learner neural network as an conditional Gaussian distribution, thereafter, we can calculate EI for this Gaussian distribution. Concretely, we have the following theorem:

Theorem 1. *(EI for feed-forward neural networks) In general, if the input of a neural network is $X = (x_1, x_2, \cdots, x_n) \in [-L, L]^n$, which means X is defined on a hyper-cube with size L, where L is a very large integer. The output is $Y = (y_1, y_2, \cdots, y_m)$, and $Y = \mu(X)$. Here μ is the deterministic mapping implemented by the neural network: $\mu : \mathcal{R}^n \to \mathcal{R}^m$, and its Jacobian matrix at X is $\partial_{X'}\mu(X) \equiv \left\{ \frac{\partial \mu_i(X')}{\partial X'_j} |_{X'=X} \right\}_{nm}$. If the neural network can be regarded as a Gaussian distribution conditional on given X:*

$$p(Y|X) = \frac{1}{\sqrt{(2\pi)^m |\Sigma|}} \exp\left(-\frac{1}{2} (Y - \mu(X))^T \Sigma^{-1} (Y - \mu(X)) \right) \tag{15}$$

where $\Sigma = diag(\sigma_1^2, \sigma_2^2, \cdots, \sigma_m^2)$ is the co-variance matrix, and σ_i is the standard deviation of the output y_i which can be estimated by the mean square error of y_i, then the effective information (EI) of the neural network can be calculated in the following way:

(i) If there exists X such that $\det(\partial_{X'}\mu(X)) \neq 0$, then the effective information (EI) can be calculated as:

$$\begin{aligned} EI_L(\mu) = I(do(X \sim U([-L, L]^n); Y) \approx & -\frac{m + m\ln(2\pi) + \sum_{i=1}^m \sigma_i^2}{2} \\ & + n\ln(2L) + \mathbb{E}_{X \sim U([-L,L]^n)} (\ln |\det(\partial_{X'}\mu(X))|). \end{aligned} \tag{16}$$

where $U([-L, L]^n)$ is the uniform distribution on $[-L, L]^n$, and $|\cdot|$ is absolute value, and det *is determinant.*

(ii) If $\det(\partial_{X'}\mu(X)) \equiv 0$ for all X, then $EI \approx 0$.

Although Theorem 1 can solve the problem of EI computation for continuous variables and functions, new problems must be confronted which are: (1) EI will be affected by the output dimension m easily, this may trouble the comparison of EI for different dimensional dynamics, and (2) EI is dependent on L, and will be divergent when L is very large.

To solve the first problem, we define a new indicator which is called dimension averaged effective information or effective information per dimension. Formally,

Definition 6. *(Dimension Averaged Effective Information (dEI)): For a dynamic f with n dimensional state space, then the dimension averaged effective information is defined as:*

$$dEI(f) = \frac{EI(f)}{n} \tag{17}$$

Therefore, if the dynamic f is continuous and can be regarded as a conditional Gaussian distribution, then according to Theorem 1, the dimension averaged EI can be calculated as ($m = n$):

$$dEI_L(f) = -\frac{1 + \ln(2\pi) + \sum_{i=1}^{n} \sigma_i^2 / n}{2} + \ln(2L) + \frac{1}{n}\mathbb{E}_{X\sim U([-L,L]^n}(\ln|\det(\partial_{X'} f(X))|). \tag{18}$$

It is easy to see that all the terms related with dimension n in Equation (18) is eliminated. However, there is still L in the equation which may cause divergent when L is very large.

Therefore, to solve this problem, we can calculate the dimension averaged causal emergence (dCE) to eliminate the influence of L.

Definition 7. *(Dimension averaged causal emergence (dCE)): for macro-dynamics f_M with dimension n_M and micro-dynamics f_m with dimension n_m, we define dimension averaged causal emergence as:*

$$dCE(f_M, f_m) = dEI(f_M) - dEI(f_m) = \frac{EI(f_M)}{n_M} - \frac{EI(f_m)}{n_m}. \tag{19}$$

Thus, if the dynamics f_M and f_m are continuous and can be regarded as conditional Gaussian distributions, then according to Definition 7 and Equation (18), the dimension averaged causal emergence can be calculated as:

$$\begin{aligned} dCE(f_M, f_m) = &\left(\frac{1}{n_M}\mathbb{E}_{X_M} \ln|\det \partial_{X_M} f_M| - \frac{1}{n_m}\mathbb{E}_{X_m} \ln|\det \partial_{X_m} f_m|\right) \\ &- \left(\frac{1}{n_M}\sum_{i=1}^{n_M} \ln \sigma_{i,M}^2 - \frac{1}{n_m}\sum_{i=1}^{n_m} \ln \sigma_{i,m}^2\right) \end{aligned} \tag{20}$$

Therefore, all the effects of dimension n and L have been eliminated in Equation (20), and the result is only influenced by the relative values of the variances and the logarithmic values of the determinant of the jacobian matrices. In the following numeric computations, we will mainly use Equation (20). The reason why we not use Eff is also because it contains L.

4. Results

In this section, we will layout several theoretic properties of NIS at first; then, we will apply it on some numeric examples.

4.1. Theoretical Analysis

To understand why the neural information squeezer framework can find out the most informative macro-dynamics and how the effective strategy and dynamics change with q, we at first layout some major theoretical results through mathematical analysis. Notice that although all of the theorems are about mutual information, these conclusions are also suitable for effective information because all the theoretical results are irrelevant to the distribution of input data.

4.1.1. Squeezed Information Channel

First, we notice that the framework (Figure 1) can be regarded as an information channel as shown in Figure 3, and due to the existence of the projection operation, the

channel is squeezed in the middle. Therefore, we call that a squeezed information channel (see also Appendix B for formal definition for the sqeezed information channel).

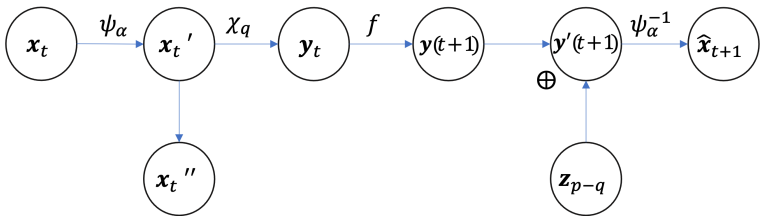


Figure 3. The graphic model of the neural information squeezer as a squeezed information channel.

As proved in Appendix B, we have a theorem for the squeezed information channel:

Theorem 2. *(Information bottleneck of Information Squeezer): For the squeezed information channel as shown in Figure 3 and for any bijector ψ, projector χ_q, macro-dynamics f, and the random noise $\mathbf{z}_{p-q} \sim \mathcal{N}(0, \mathcal{I}_{p-q})$, we have:*

$$I(\mathbf{y}_t; \mathbf{y}(t+1)) = I(\mathbf{x}_t; \hat{\mathbf{x}}_{t+1}), \tag{21}$$

where $\hat{\mathbf{x}}_{t+1}$ is the prediction of NIS, and $\mathbf{y}(t+1)$ follows Equation (2).

That is, for any neural network that implements the general framework as shown in Figure 3, the mutual information of macro-dynamic f_{ϕ_q} is identical to the entire dynamical model, i.e., the mapping from $\mathbf{x}_t$ to $\hat{\mathbf{x}}_{t+1}$ for any time. Theorem 2 is fundamental for NIS. Actually, the macro-dynamics f is the information bottleneck of the entire channel [45].

4.1.2. What Happens during Training

With Theorem 2, we can understand what happens when the neural squeezer framework is trained by data in an intuitive way.

First, we know as the neural networks are trained, the output of the entire framework $\hat{\mathbf{x}}_{t+1}$ is closed to the real data $\mathbf{x}_{t+1}$ under any given $\mathbf{x}_t$, so do the mutual information, that is the following theorem:

Theorem 3. *(Mutual information of the model will be closed to the data for a well trained framework): If the neural networks in NIS framework are well-trained (which means for any $t \in [1, T]$, the Kullback-Leibler divergence between $Pr_\tau(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_t)$ and $Pr(\mathbf{x}_{t+1}|\mathbf{x}_t)$ approaches 0 when the training epoch $\tau \to \infty$), then:*

$$I(\hat{\mathbf{x}}_{t+1}; \mathbf{x}_t) \simeq I(\mathbf{x}_{t+1}; \mathbf{x}_t) \tag{22}$$

for any $t \in [1, T]$, where $\simeq$ means asymptotic equivalence when $\tau \to \infty$.

The proof is in the Appendix C.

Second, we suppose that the mutual information $I(\mathbf{x}_t, \mathbf{x}_{t+1})$ is always large because the time series of micro-states $\mathbf{x}_t$ contains information. Otherwise, we may not be interested in $\{\mathbf{x}_t\}$. Therefore, as the neural network is trained, $I(\mathbf{x}_t; \hat{\mathbf{x}}_{t+1})$ will increase to be closed to $I(\mathbf{x}_t; \mathbf{x}_{t+1})$.

Third, according to Theorem 2, $I(\mathbf{y}_t; \mathbf{y}(t+1)) = I(\mathbf{x}_t, \hat{\mathbf{x}}_{t+1})$ will also be increased such that it can be closed to $I(\mathbf{x}_t; \mathbf{x}_{t+1})$.

Because the macro-dynamics is the information bottleneck of the entire channel, therefore its information must be increased as training. At the same time, the determinant of the Jacobian of ψ_α and the entropy of $\mathbf{y}_t$ will also be increased in a general case. This conclusion is implied in Theorem 4.

Theorem 4. *(Information on bottleneck is the lower bound of the encoder): For the squeezed information channel shown in Figure 3, the determinant of the Jacobian matrix of ψ_α and the Shannon entropy of $\mathbf{y}_t$ are lower bounded by the information of the entire channel:*

$$H(\mathbf{x}_t) + \mathbb{E}(\ln|\det(J_{\psi_\alpha}(\mathbf{x}_t))|) \geq H(\mathbf{y}_t) + \mathbb{E}(\ln|\det(J_{\psi_\alpha,\mathbf{y}_t}(\mathbf{x}_t))|) \geq I(\mathbf{x}_t;\hat{\mathbf{x}}_{t+1}), \tag{23}$$

where H is the Shannon entropy measure, $J_{\psi_\alpha}(\mathbf{x}_t)$ is the Jacobian matrix of the bijector ψ_α at the input $\mathbf{x}_t$, and $J_{\psi_\alpha,\mathbf{y}_t}(\mathbf{x}_t)$ is the sub-matrix of $J_{\psi_\alpha}(\mathbf{x}_t)$ on the projection $\mathbf{y}_t$ of $\mathbf{x}'_t$.

The proof is also given in Appendix D.

Because the distribution of $\mathbf{x}_t$ and its Shannon entropy are given, thus, Theorem 4 states that the expectation of the logrithim of $|\det(J_{\psi_\alpha}(\mathbf{x}_t))|$ and the entropy of $\mathbf{y}_t$ must be larger than the information of the entire information channel.

Therefore, once the initial values of $\mathbb{E}|\det(J_{\psi_\alpha}(\mathbf{x}_t))|$ and $\mathbf{y}_t$ are small, as the model is trained, the mutual information of the entire channel increases, the determinant of the Jacobian must also be increased, and the distribution of the macro-state $\mathbf{y}_t$ must be more disperse. However, these may not happen if the information $I(\mathbf{x}_t;\hat{\mathbf{x}}_{t+1})$ has been closed to $I(\mathbf{x}_t;\mathbf{x}_{t+1})$ or $\mathbb{E}|\det(J_{\psi_\alpha}(\mathbf{x}_t))|$ and $H(\mathbf{y}_t)$ have been already large enough.

4.1.3. The Effective Information Is Mainly Determined by the Bijector

The previous analysis is about the mutual information but not the effective information of the macro-dynamic which is the key ingredients about causal emergence. Actually, with the good properties of the squeezed information channel, we can write down an expression of the EI for the macro-dynamic but without the explicit form of it. Accordingly, we find the major ingredient to determine causal emergence is the bijector ψ_α.

The proof is detailed in Appendix D.

Theorem 5. *(The mathematical expression for effective information of the macro-dynamics): Suppose the probability density of $\mathbf{x}_{t+1}$ under given $\mathbf{x}_t$ can be described by a function $Pr(\mathbf{x}_{t+1}|\mathbf{x}_t) \equiv G(\mathbf{x}_{t+1},\mathbf{x}_t)$, and the Neural Information Squeezer framework is well trained, then the effective information of the macro-dynamics of f_β can be calculated by:*

$$EI_L(f_\beta) = \frac{1}{(2L)^p} \cdot \int_\sigma \int_{\mathcal{R}^p} G(\mathbf{y},\psi_\alpha^{-1}(\mathbf{x})) \ln \frac{(2L)^p G(\mathbf{y},\psi_\alpha^{-1}(\mathbf{x}))}{\int_\sigma G(\mathbf{y},\psi_\alpha^{-1}(\mathbf{x}'))d\mathbf{x}'} d\mathbf{y}d\mathbf{x}, \tag{24}$$

where $\sigma \equiv [-L,L]^p$ is the integration region for $\mathbf{x}$ and $\mathbf{x}'$.

4.1.4. Change with the Scale (Q)

According to Theorems 2 and 3, we have the following Corollary 1:

Corollary 1. *(The mutual information of macro-dynamics will not change if the model is well trained): For the well trained NIS model, the Mutual Information of the macro-dynamics f_β will be irrelevant of all the parameters, including the scale q.*

If the neural networks are well-trained, the mutual information on the macro-dynamics will approach to the information in the data $\{\mathbf{x}_t\}$. Therefore, no matter how small q is (or how large is the scale), the mutual information of the macro-dynamics f_β will keep constant.

It seems that the scale q is an irrelevant parameter on causal emergence. However, according to Theorem 6, smaller q will lead to the encoder carrying more effective information.

Theorem 6. *(Narrower is Harder): If the dimension of $\mathbf{x}_t$ is p, then for $0 < q_1 < q_2 < p$:*

$$I(\mathbf{x}_t;\hat{\mathbf{x}}_{t+1}) \leq I(\mathbf{x}_t;\mathbf{y}_t^{q_1}) \leq I(\mathbf{x}_t;\mathbf{y}_t^{q_2}), \tag{25}$$

where $\mathbf{y}_t^q$ *denotes the q-dimensional vector* $\mathbf{y}_t$.

The mutual information in Theorem 6 is about the encoder, i.e., the micro-state $\mathbf{x}_t$ and the macro-state $\mathbf{y}_t$ in different dimension q. The theorem states that as q decreases, the mutual information of the encoder part must also decrease and more closed to the information limitation $I(\mathbf{x}_t; \hat{\mathbf{x}}_{t+1}) \simeq I(\mathbf{x}_t; \mathbf{x}_{t+1})$. Therefore, the entire information channel becomes narrower, the encoder must carry more useful and effective information to transfer to the macro-dynamics. In addition, the prediction becomes harder.

4.2. Empirical Results

We test our model on several data sets. All the data are generated by the simulated dynamical models. In addition, the models include continuous dynamics and discrete Markovian dynamics.

4.2.1. Spring Oscillator with Measurement Noise

The first experiment to test our model is a simple spring oscillator following the dynamical equations:

$$\begin{cases} dz/dt = v \\ dv/dt = -z \end{cases} \tag{26}$$

where z and v are position and velocity of the oscillator in one dimension, respectively. The states of the system can be represented as $\mathbf{x} = (z, v)$.

However, we can only observe the state from two sensors with measurement errors. Suppose the observational model is

$$\begin{cases} \tilde{\mathbf{x}}_1 = \mathbf{x} + \zeta \\ \tilde{\mathbf{x}}_2 = \mathbf{x} - \zeta \end{cases} \tag{27}$$

where $\zeta \sim \mathcal{N}(0, \sigma)$ is a random number following two dimensional Gaussian distribution, and σ is the vector of the standard deviations for position and velocity. In this example, we can understand the states $\mathbf{x}$ as latent macro-states and the measurements $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2$ are micro-states. What the NIS will do is recover the latent macro-state $\mathbf{x}$ from the measurements.

According to Equation (27), although there is noise to disturb the measurement of the state, it can be easily eliminated by adding the measurements on the two channels together. Therefore, if NIS can discover a macro-state which is the addition of the two measurements, then it can easily obtain the correct dynamics. We sample the data for 10,000 batches (with Euler method and $dt = 1$), and in each batch, we randomly generate 100 random initial states and perform one step dynamic to obtain the state at the next time step. We use these data to train the neural network. To compare, we also use the same data set to train an ordinary feed-forward neural network with the same number of parameters.

The results are shown in Figure 4. To test if NIS can learn the real latent macro state, we directly plot the predicted and the real latent states. As shown in Figure 4a, the predicted and the real curves collapse together which means NIS can recover the macro state in the data although it is unknown. As a comparison, the feed-forward neural network cannot recover the macro state. We can also check if the NIS can learn the dynamic of the macro states by plotting the derivatives of the states $(dz/dt, dv/dt)$ against the macro state variables (v, z). If the learned dynamics follows Equation (26), then two cross-over lines for $dz/dt = v$ and $dv/dt = -z$ can be observed as shown in Figure 4c. However, the same pattern cannot be reproduced on the common feed-forward network as shown in Figure 4d. We also test the well-trained NIS by multiple-step prediction as shown in Figure 4e. Although there are larger and larger deviation from the prediction and the real data, the general trends can be captured by NIS model. We further study how the dimension averaged causal emergence dCE changes with the scale q which is measured by

the number of effective information channels on the well-trained NIS model as shown in Figure 4f. dCE peaks at $q = 2$ which is exactly same as in the ground truth.

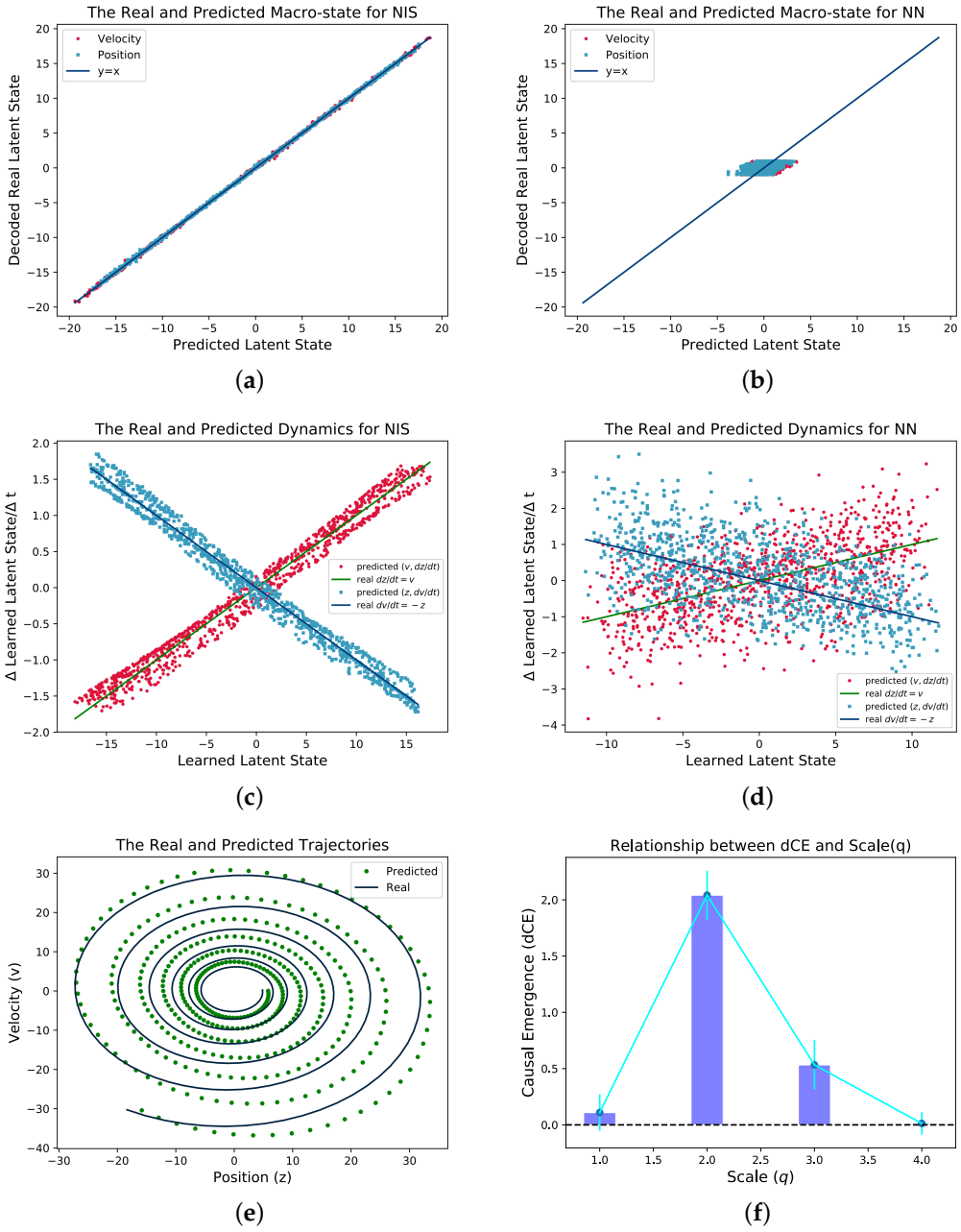


Figure 4. Experimental Results for the Simple Spring Oscillator with Measurement Noise. We sample data from Equations (26) and (27), and we use Euler method to simulate by taking $dt = 0.1$. (**a**,**b**) show the real macro-state versus the predicted ones both for NIS and the ordinary feed-forward neural network, respectively; (**c**,**d**) show the real and predicted dynamics, i.e., the dependence between dz/dt and v, and dv/dt with z for both neural networks for comparison; (**e**) shows the real and predicted trajectories with 400 time steps starting from the same latent state; (**f**) shows the dependence of the dimension averaged of *Causal Emergence* (dCE) on q (the number of effective channels).

Further, we use experimental results to verify the theorems mentioned in the previous section and the theory of information bottleneck [45]. First, we show how the mutual information of $I(\mathbf{x}_t, \mathbf{x}_{t+1})$, $I(\mathbf{y}_t, \mathbf{y}(t+1))$, and $I(\mathbf{x}_t, \hat{\mathbf{x}}_{t+1})$ change with time (epoch) when q takes different values as shown in Figure 5c,d. The results show that all the mutual information converge as predicted by Theorems 2 and 3. We also plot the mutual information

between $\mathbf{x}_t$ and $\mathbf{y}_t$ with different q to test Theorem 6, and the results show that the mutual information increases when q increases as shown in Figure 5a.

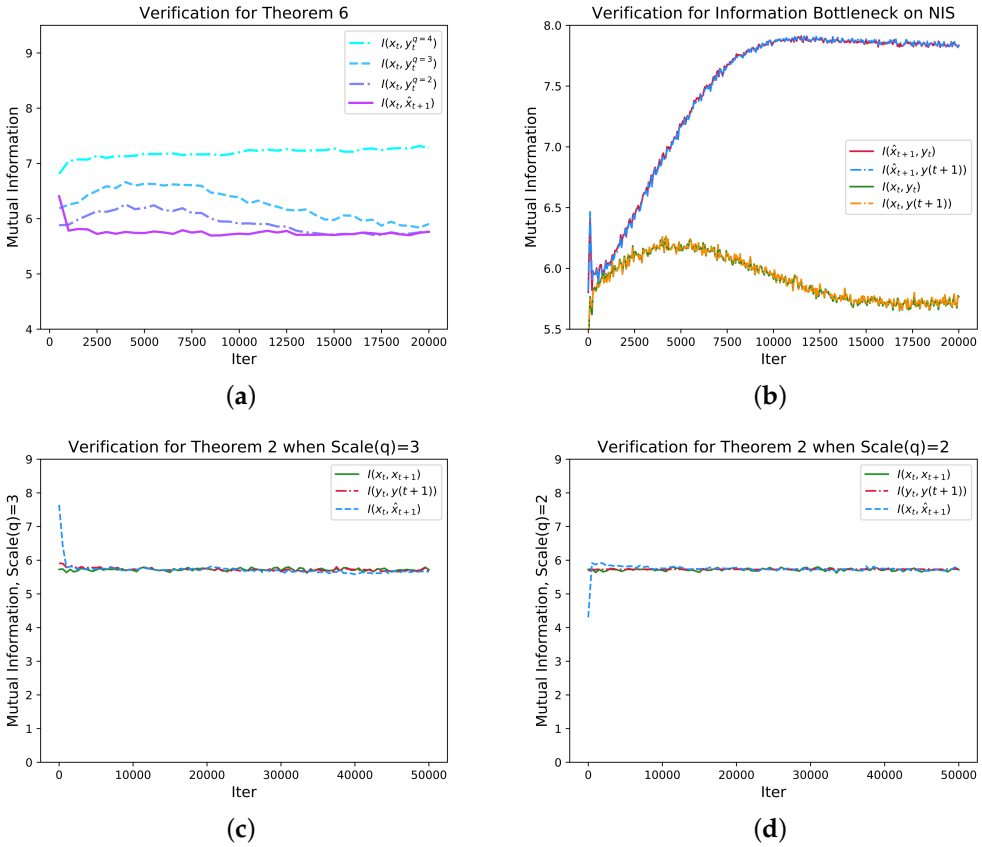


Figure 5. Various mutual information between variables change with training iterations. (**a**) shows the change of mutual information $I(x_t, y_t^{q=4})$, $I(x_t, y_t^{q=3})$, $I(x_t, y_t^{q=2})$ and $I(x_t, \hat{x}_t)$ with the increase of the number of iterations. From the figure, we can see that within the specified number of iterations, $I(x_t, \hat{x}_t) \leq I(x_t, y_t^{q=2}) \leq I(x_t, y_t^{q=3}) \leq I(x_t, y_t^{q=4})$. Among them, q is the dimension of the coarse-grained system. (**b**) verifies the theory of information bottleneck on NIS when Scale (q) = 2. (**c**,**d**) show the change of mutual information $I(x_t, x_{t+1})$, $I(y_t, y_{t+1})$ and $I(x_t, \hat{x}_t)$ with the increase of the number of iterations. It can be seen that under different scales, the three mutual information values are close to each other. The standard deviations of the three variables gradually decrease with iteration. Therefore, $I(x_t, x_{t+1}) \approx I(y_t, y_{t+1}) = I(x_t, \hat{x}_t)$ is reflected.

According to the information bottleneck theory [45], the mutual information between latent variable and output may increase while the information between input and latent variable should increase in the early stage and then decrease as training process proceed. As shown in Figure 5b, this conclusion is confirmed by the NIS model where the macro-states $\mathbf{y_t}$ and the prediction $\mathbf{y(t+1)}$ are all latent variables. Although the same conclusion is obtained, the information bottleneck can be reflected by the architecture in NIS model much clearer than the general neural networks because $\mathbf{y}_t$ and $\mathbf{y}(t+1)$ is the bottleneck and all other irrelevant information is discarded by the variable $\mathbf{x}_t''$ as shown in Figure 3.

4.2.2. Simple Markov Chain

In the second example, we show NIS can work on discrete Markov chain, and the coarse-graining strategy can work on state space. The Markov chain to generate the data is the following probability transition matrix:

$$\begin{pmatrix} 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 0 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 0 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 0 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 0 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 0 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 0 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{28}$$

The system has 8 states, and seven of them can transfer each other. The last state is standalone. We use a one-hot vector to encode the states. Therefore, for example, state 2 will be represented as $(0,1,0,0,0,0,0,0)$. We sample the initial state for 50,000 batches to generate data. We then feed these one-hot vectors into the NIS framework, after training for 50,000 epochs, we can obtain an effective model. The results are shown in Figure 6.

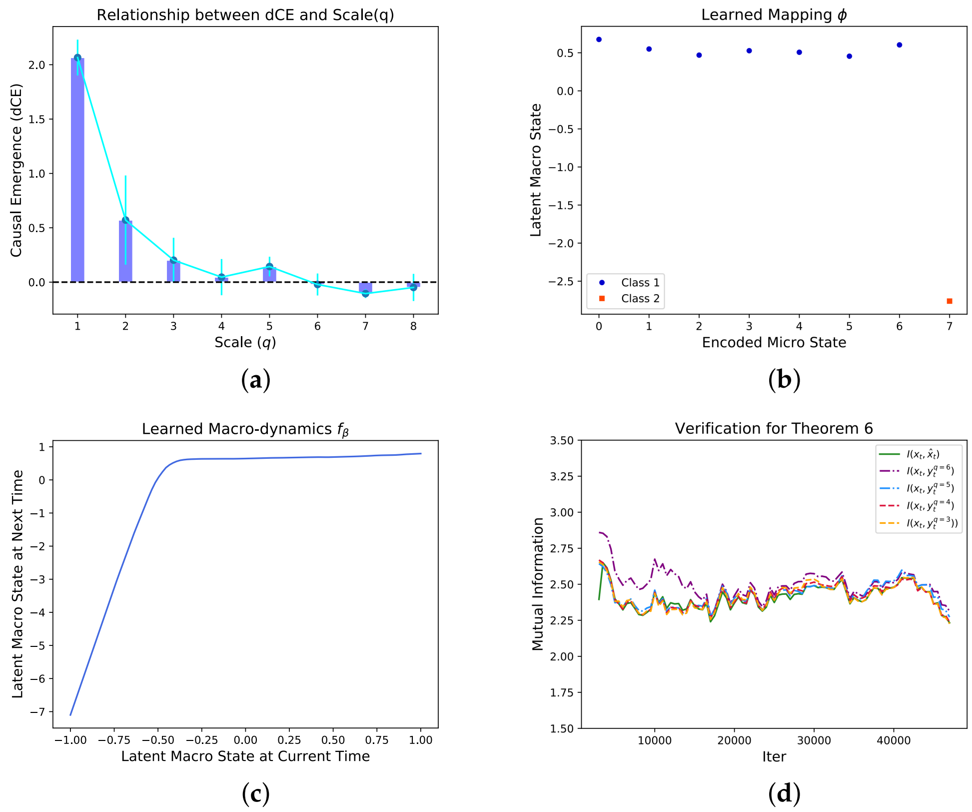


Figure 6. The dependence of the dimension averaged *Causal Emergence* (dCE) on different scales (q) of the Markov dynamics (**a**), the learned mapping between micro states and macro states on the optimal scale (q) (b), and the learned macro-dynamics the mapping from $\mathbf{y}_t$ to $\mathbf{y}(t+1)$ (**c**). There are two clear separated clusters on the y-axis in (**b**) which means the macro states are discrete. We found that the two discrete macro states and the mapping between micro and discrete macro states are identical as the example in Ref. [6] which means the correct coarse-graining strategy can be discovered by our algorithm automatically under the condition without any prior information. In (**d**), $I(x_t, \hat{x}_t) \le I(x_t, y_t^{q=3}) \le I(x_t, y_t^{q=4}) \le I(x_t, y_t^{q=5}) \le I(x_t, y_t^{q=6})$ is reflected. In order to make the data clearer, we have taken a moving average for each group of data. This result can be regarded as the verification of Theorem 6.

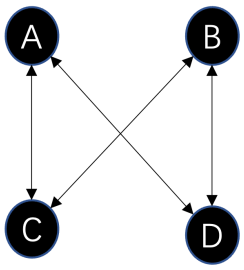
By systematically search for different q, we found that the dimension averaged causal emergence (dCE) peaks at $q = 1$ as shown in Figure 6a. On the optimal scale, we can visualize the coarse-graining strategy by Figure 6b, in which the x-coordinate is the decimal coding for different states, and the y-coordinate represents the coding for the macro-states. We find that the coarse-graining mapping successfully classifies the first seven states into a one macro-state, and leaves the last state stay alone. This learned coarse-graining strategy is identical as the example shown in [6].

We also visualize the learned macro-dynamics as shown in Figure 6c. This is a linear mapping when $y_t < 0$ and almost a constant for $y_t > 0$. Therefore, the dynamics can guarantee that all the first seven micro-states can be separated with the last state. We also verify Theorem 2 in Figure 6d.

4.2.3. Simple Boolean Network

Our framework can not only work on continuous time series and Markov chain, but also can work on a networked system in which each node follows a discrete micro mechanism.

For example, boolean network is a typical discrete dynamical system in which the node contains two possible states (0 or 1), and the state of each node is affected by the state of the neighbors connected to it. We follow the example in [5]. Figure 7 shows an exampled boolean network with four nodes, and each node follows the same micro mechanism as shown in the table of Figure 7. In the table, each entry is the probability of each node's state conditions on the state combination of its neighbors. For example, if the current node is A, then the first entry is $Pr(x_A^{t+1} = 0 | x_C^t = 0, x_D^t = 0) = 0.7$, which means that A will take value 0 with probability 0.7 when the state combination of C and D is 00. By taking all the single node mechanisms together, we can obtain a large Markovian transition matrix with $2^4 = 16$ states which is the complete micro mechanism of the whole network.



Micro mechanisms

t-1\t	0	1
00	0.7	0.3
01	0.7	0.3
10	0.7	0.3
11	0	1

Figure 7. An exampled Boolean network (**left**) and its micro mechanisms on nodes (**right**). Each node's state on the next time step is affected by its neighboring nodes' state combination randomly. The transition probabilities (micro mechanisms) on each case are shown in the table.

We sample the one step state transition of the entire network for 50,000 batches and each batch contains 100 different initial conditions which are randomly sampled from the possible state space evenly, and we then feed these data to the NIS model. By systematically search for different q, we found that the dimension averaged causal emergence peaks at $q = 1$ as shown in Figure 8a. Under this condition, we can visualize the coarse-graining strategy by Figure 8b, in which the x-coordinate is the decimal coding for the binary micro-states (e.g., 5 denotes for the state 0101), and the y-coordinate represents the codes for macro-states. The data points can be clearly classified into four clusters according to their y-coordinate. This means the NIS network found four discrete macro-states although the states are continuous real numbers. Interestingly, we found that the mapping between the 16 micro states and four macro states are identical as the coarse-graining strategy shown in the example in Ref. [5]. However, any prior information neither the method on how

to group the nodes nor the coarse graining strategy, nor the dynamics are known by our algorithm. Finally, Theorems 2 and 6 are verified in this example as shown in Figure 8c,d.

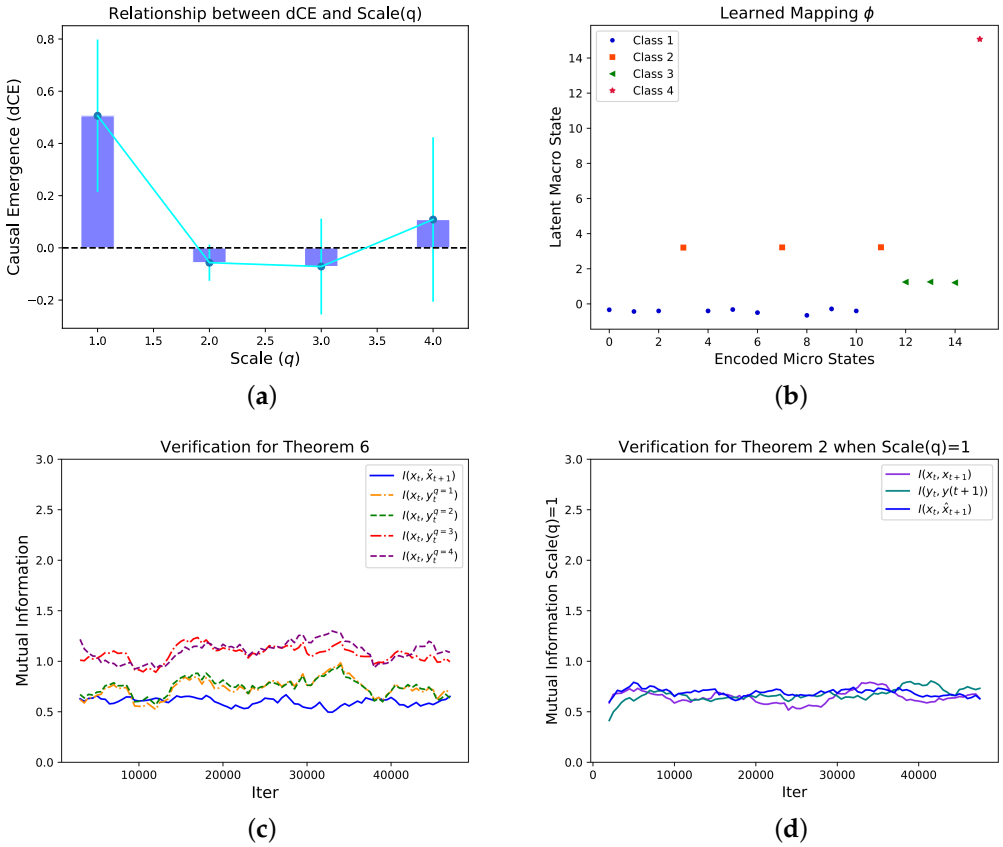


Figure 8. Experimentsl Results for the Boolean Network. The dependence of the dimension averaged *Causal Emergence* (dCE) on different scales (q) (**a**) and the learned mapping between micro states and macro states on the optimal scale (q) (**b**). There are four clear separated clusters on the y-axis in (**b**) which means the macro states are discrete. We found that the four discrete macro states and the mapping between micro and discrete macro states are identical as the example in Ref. [5] which means the correct coarse-graining strategy can be discovered by our algorithm automatically under the condition without any prior information. (**c**) shows the change of mutual information $I(x_t, y_t^{q=4})$, $I(x_t, y_t^{q=3})$, $I(x_t, y_t^{q=2})$,$I(x_t, y_t^{q=1})$ and $I(x_t, \hat{x}_t)$ with the increase of the number of iterations. From the figure, we can see approximately that within the specified number of iterations, $I(x_t, \hat{x}_t) \leq I(x_t, y_t^{q=1}) \leq I(x_t, y_t^{q=2}) \leq I(x_t, y_t^{q=3}) \leq I(x_t, y_t^{q=4})$. (**d**) shows the change of mutual information $I(x_t, x_{t+1})$, $I(y_t, y_{t+1})$ and $I(x_t, \hat{x}_t)$ with the increase of the number of iterations. In order to easily observe the trend of data changes, we added a moving average curve for each group of data. It can be seen that under different scales, the three mutual information values are close to each other. $I(x_t, x_{t+1}) \approx I(y_t, y_{t+1}) = I(x_t, \hat{x}_t)$ is reflected. Considering the experimental error, the overall trend of the data still conforms to the theorem.

5. Concluding Remarks

In this paper, we propose a novel neural network framework, Neural Information Squeezer, for discovering coarse-graining strategy, macro-dynamic and emergent causality in time series data. We first define effective coarse-graining strategy and macro-dynamic by constraining the coarse-graining strategies to predict the future micro-state with a precision threshold. Then, the causal emergence identification problem can be understood as a maximization problem for effective information under the constraint.

We then use an invertible neural network incorporating with the projection operation to realize the coarse-graining strategy. The usage of invertible neural network can not only allow us to reduce the number of parameters by sharing them between the encoder and the

decoder but also can facilitate us to analyze the mathematical properties of the whole NIS architecture.

By treating the framework as a squeezed information channel, we can prove four important theorems. The results show that if the causal connection in the data is strong, then as we train the neural networks, the macro-dynamics will increase its informativeness. In addition, during this process, the determinant of the Jacobian of the bijector will increase at the same time. We also found a mathematical expression for the effective information of the macro-dynamics without the explicit dependence on the macro-dynamics, and it is determined solely by the bijector and the data when the whole framework is well trained. Furthermore, if the framework has been trained in a sufficient time, the mutual information of the macro-dynamics will keep a constant no matter the scale q is. However, as q decreases, the mutual information or the bandwidth on the encoder part also decreases and closed to the information limitation on the entire channel such that it can make correct prediction for the future micro-states. Thus, the task becomes harder for the encoder because more effective information must be encoded and pass to the dynamics learner such that it can make correct prediction with less information. Numerical experiments show that our framework can reconstruct the dynamics in different scales and also can discover emergent causality in data on several classic causal emergence examples.

There are several weak points in our framework. First, it can only work on small data set. The major reason is the invertible neural network is very difficult to train on large data set. Therefore, we will use some special techniques to optimize the architecture in future. Second, the framework is still lack of explainability, the grouping method for variables is implicitly encoded in the invertible neural network although we can illustrate what the coarse-graining mapping is, and decompose it into information conversion and information discarding parts clearly. A more transparent neural network framework with more explanatory power is deserved for future studies. Third, the conditional distribution that the model can predict actually is limited as Gaussian or Laplacian, and it should be extended to more general distributional forms in future studies.

There are several theoretical problems left for future studies. For example, we conjecture that all coarse-graining strategies can be decompose into a bijection and a projection, but this needs strict mathematical proof. Second, although an explicit expression for EI on macro-dynamics has been derived under NIS, we still cannot directly predict the causal emergence in the data. We believe that a more concise analytic results on the EI should be derived by setting some constraints on the data. Furthermore, we think the meaning and the usage of the discarding variable $\mathbf{x}_t''$ should be further explored since that it may relate with the redundant information of a pair of variables toward a target [46]. Therefore, we guess more deep connections between the framework of NIS and the mutual information decomposition may exist and NIS may work as a numeric tool to decompose the mutual information.

Author Contributions: Conceptualization and methodology, J.Z.; coding, J.Z. and K.L.; writing, J.Z. and K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by by the National Natural Science Foundation of China (NSFC) under Grant No. 61673070 at https://www.nsfc.gov.cn/, accessed on24 October 2022.

Data Availability Statement: All the codes and data are available at: https://github.com/jakezj/NIS_for_Causal_Emergence, accessed on 24 October 2022.

Acknowledgments: J.Z. acknowledge the discussion in the reading group on "Causal Emergence" organized by Swarma Club, particularly for Yanbo Zhang, Everrete You and Erik Hoel. We thank the support from Causal Emergence Reading Group supported by the Save 2050 Programme jointly sponsored by Swarma Club and X-Order; We thank the support from Swarma Research.

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

Appendix A. RealNVP Implementation of Invertible Neural Network

In the main text, we mentioned that the invertible neural network can be realized by a RealNVP module. The architecture of a RealNVP module can be visualized by Figure 2. Concretely, if the input of the module is $\mathbf{x}$ with dimension n and the output is $\mathbf{x}'$ with the same dimension, then the RealNVP module can perform the following computation steps:

$$\begin{cases} \mathbf{x_1} = \mathbf{x}_{1:m} \\ \mathbf{x_2} = \mathbf{x}_{m:n} \end{cases} \tag{A1}$$

where m is an integer in between 1 and n.

$$\begin{cases} \mathbf{x}'_\mathbf{1} = \mathbf{x_1} \bigotimes s_1(\mathbf{x_2}) + t_1(\mathbf{x_2}) \\ \mathbf{x}'_\mathbf{2} = \mathbf{x_2} \bigotimes s_2(\mathbf{x}'_\mathbf{1}) + t_2(\mathbf{x}'_\mathbf{1}) \end{cases} \tag{A2}$$

where s_1, s_2, t_1 and t_2 are feed-forward neural networks with arbitrary architectures, while their input-output dimensions must match with the data. In practice, s_1 or s_2 always do an exponential operation on the output of the feed-forward neural network [43] to facilitate the inverse computation.

Finally,

$$\mathbf{x}' = \mathbf{x}'_\mathbf{1} \bigoplus \mathbf{x}'_\mathbf{2} \tag{A3}$$

It is not difficult to verify that all three steps are invertible. Equation (A2) is invertible because the same form but with negative signs can be obtained by solving the expressions of $\mathbf{x_1}$ and $\mathbf{x_2}$ with $\mathbf{x}'_\mathbf{1}$ and $\mathbf{x}'_\mathbf{2}$ from Equation (A2).

To simulate more complex invertible functions, we always duplex the basic RealNVP modules by stacking them together. In the main text, we use duplex the basic RealNVP module by three times.

Appendix B. Approximated Calculation of Effective Information for Neural Networks

In this paper, we propose an approximated method to calculate EI for a neural network. Conventional methods usually coarse-grain the input and output spaces into small regions, and estimate the probability of each region by the frequency. However, this estimation is inaccurate especially for the regions with small probability.

To avoid this problem, we propose a new method to estimate the mutual information of a neural network. The key idea is to treat a Neural Network as a conditional probability with a Gaussian (or Laplacian) distribution in which the mean value is the output vector of the neural network, and the standard deviation takes the Mean Square Error of the prediction. The concrete distributional form (Gaussian or Laplacian) is determined by the types of Loss function. If MSE (Mean Square Error) is taken then Gaussian distribution is considered, otherwise if MAE (Mean Absolute Error) is considered then Laplacian distribution is considered. Without lose genarlity, here, we take the distributional form as Gaussian.

We restate Theorem 1:

Theorem 1 In general, if the input of a neural network is $X = (x_1, x_2, \cdots, x_n) \in [-L, L]^n$, where L is a big integer, the output is $Y = (y_1, y_2, \cdots, y_m)$, and $Y = \mu(X)$. Here μ is the deterministic mapping implemented by the neural network: $\mu : \mathcal{R}^n \to \mathcal{R}^m$, and its Jacobian matrix at X is $\partial_{X'}\mu(X) \equiv \left\{ \frac{\partial \mu_i(X')}{\partial X'_j}|_{X'=X} \right\}_{nm}$, and if the neural network can be regarded as an Gaussian distribution conditional on given X:

$$p(Y|X) = \frac{1}{\sqrt{(2\pi)^m |\Sigma|}} \exp\left(-\frac{1}{2}(Y - \mu(X))^T \Sigma^{-1} (Y - \mu(X)) \right) \tag{A4}$$

where $\Sigma = diag(\sigma_1^2, \sigma_2^2, \cdots, \sigma_m^2)$ is the co-variance matrix, and σ_i is the standard deviation of the output y_i which can be estimated by the mean square error of y_i. Then the effective information (EI) of the neural network can be calculated in the following way:

(i) If there exists X such that $\det(\partial_{X'}\mu(X)) \neq 0$, then the effective mutual information (EI) can be calculated as:

$$\begin{aligned} EI_L(\mu) = I(do(X \sim U([-L,L]^n); Y) \approx & -\frac{m + m\ln(2\pi) + \sum_{i=1}^{m} \ln\det(\sigma_i^2)}{2} \\ & + n\ln(2L) + \mathbb{E}_{X\sim U([-L,L]^n)}(\ln|\det(\partial_{X'}\mu(X))|). \end{aligned} \tag{A5}$$

where $U([-L,L]^n)$ is the uniform distribution on $[-L,L]^n$, and $|\cdot|$ is absolute value, and det is determinant.

(ii) If $\det(\partial_{X'}\mu(X)) \equiv 0$ for all X, then $EI \approx 0$.

Proof. Because the calculation of mutual information can be separated into two parts:

$$\begin{aligned} I(X;Y) &= \int_{XY} p(X,Y)\ln\frac{p(X,Y)}{p(X)p(Y)}dXdY = \int_{XY} p(X)p(Y|X)\ln\frac{p(Y|X)}{p(Y)}dYdX \\ &= \int_X p(X)\left(\int_Y p(Y|X)\ln p(Y|X)dY\right)dX \\ &- \int_X p(X)\left(\int_Y p(Y|X)\ln p(Y)dY\right)dX \end{aligned} \tag{A6}$$

By inserting Equation (A4) into Equation (A6), the first term becomes (the Shannon entropy of the Gaussian Distribution):

$$\int_Y p(Y|X)\ln p(Y|X)dY = -\frac{m + m\ln(2\pi) + \ln\det(\Sigma)}{2} \tag{A7}$$

However, it is hard to derive an explicit expression of the second term in Equation (A6) because it contains integration. Therefore, we can expand $\mu(X')$ into Taylor series on the point X and keep only the first order term:

$$\mu(X') \approx \mu(X) + \partial_{X'}\mu(X)(X' - X), \tag{A8}$$

where $\partial_{X'}\mu(X) \equiv \frac{\partial\mu(X')}{\partial X'}|_{X'=X} = \left(\frac{\partial\mu_i(X)}{\partial X'_j}\right)_{m,n}$.

(i) If there exists X: $|\det(\partial_{X'}\mu(X))| \neq 0$, thus:

$$\begin{aligned} p(Y) &= \int_{X'} p(X',Y)dX' = \int_{X'} p(X')p(Y|X')dX' = \int_{X'} \rho p(Y|X')dX' \\ &\approx \rho\frac{1}{\sqrt{|\Sigma|}}\frac{1}{\sqrt{|\det(\partial_{X'}\mu(X)^T\cdot\Sigma^{-1}\cdot\partial_{X'}\mu(X))|}} = \frac{\rho}{|\det(\partial_{X'}\mu(X))|}, \end{aligned} \tag{A9}$$

where $\rho = (2L)^{-n}$. This is the multivariate Gaussian integral. Therefore:

$$\begin{aligned} \int_X p(X)\left(\int_Y p(Y|X)\ln p(Y)dY\right)dX &\approx \ln\rho - \rho\int_X \ln|\det(\partial_{X'}\mu(X))|dX \\ &\approx \ln\rho - \mathbb{E}_{X\sim U([-L,L]^n)}\ln|\det(\partial_{X'}\mu(X))| \end{aligned} \tag{A10}$$

Thus, EI can be derived by combining the two terms together:

$$EI_L \approx -\frac{m + m\ln(2\pi) + \ln|\det(\Sigma)|}{2} - \ln\rho + \mathbb{E}_{X\sim U([-L,L]^n)}\ln|\det(\partial_{X'}\mu(X))| \tag{A11}$$

To insert $\rho = (2L)^{-n}$ and $\det(\Sigma) = \prod_{i=1}^{m} \sigma_i$ into Equation (A11), we obtain Equation (A5).
(ii) If $\det(\partial_{X'} \mu(X)) = 0$ for all X, which means $\mu(X) \equiv \mu_0$ where μ_0 is a constant, then:

$$p(Y) \approx \frac{\exp\left(-\frac{1}{2}(Y-\mu_0)^T \Sigma^{-1}(Y-\mu_0)\right)}{\sqrt{(2\pi)^m |\det(\Sigma)|}}, \tag{A12}$$

so,

$$\int_Y p(Y|X) \ln p(Y) dY \approx -\frac{m + m\ln(2\pi) + \ln|\det(\Sigma)|}{2}, \tag{A13}$$

Combining with Equation (A7), we have:

$$EI_L \approx 0 \tag{A14}$$

□

With this theorem, we can numerically calculate the EI of a neural network in an approximate way. The mathematical expectation can be approximated by averaging the logarithm of the determinant of the Jacobian on the samples of X drawn on the hyper-cube $[-L, L]^n$ uniformly. This method can avoid partitioning intervals and counting frequencies which are very difficult when the dimension is large.

Lemma A1. *(Projection does not affect mutual information): Suppose $X \in \mathcal{R}^{p+q}$ and $Y \in Dom(Y)$, where $p, q \in \mathcal{Z}^+$. In addition, X can be decomposed as two components $U \in \mathcal{R}^p, V \in \mathcal{R}^q$, that is:*

$$X = U \bigoplus V, \tag{A15}$$

where $\bigoplus$ represents vector concatenation. We call that U and V are X's projections on p or q dimensional sub spaces, respectively. If U and Y form a Markov chain $U \to Y$, and V is independent on Y, then we have:

$$I(X;Y) = I(U;Y) \tag{A16}$$

Proof. Notice that the joint distribution of X and Y can be written as:

$$p(X,Y) \equiv p(U \bigoplus V, Y) \equiv p(U,V,Y) \tag{A17}$$

Further, because $X \to U$ forms a Markov chain, but V is not, thus:

$$p(V,Y|U) = p(V|U)p(Y|U,V) = p(V|U)p(Y|U), \tag{A18}$$

thus, we have:

$$p(X,Y) = p(U,V,Y) = p(U)p(V,Y|U) = p(U)p(V|U)p(Y|U) \tag{A19}$$

Therefore:

$$\begin{aligned} I(X;Y) &= \int_{X,Y} p(X,Y) \ln \frac{p(X,Y)}{p(X)p(Y)} dXdY \\ &= \int_{U,V,Z} p(U)p(V|U)p(Y|U) \ln \frac{p(U)p(V|U)p(Y|U)}{p(U)p(V|U)p(Y)} dUdVdZ \\ &= \int_{U,Y} p(U)p(Y|U) \ln \frac{p(U)p(Y|U)}{p(U)p(Y)} dUdY \\ &= I(U;Y) \end{aligned} \tag{A20}$$

□

Lemma A2. *(Mutual information will not be affected by concatenating independent variables): If $X \in Dom(X)$ and $Y \in Dom(Y)$ form a Markov chain $X \to Y$, and $Z \in Dom(Z)$ is a random variable which is independent on both X and Y, then:*

$$I(X;Y) = I(X;Y \bigoplus Z) \tag{A21}$$

Proof. Because:

$$p(X, Y \bigoplus Z) = p(X, Y, Z) = p(X)p(Y, Z|X), \tag{A22}$$

furthermore, because $X \to Y$ and Z is independent on both X and Y, therefore:

$$p(Y, Z|X) = p(Y|X)p(Z). \tag{A23}$$

Thus:

$$\begin{aligned} I(X;Y \bigoplus Z) = I(X;Y,Z) &= \int_{XYZ} p(X,Y,Z) \log \frac{p(X,Y,Z)}{p(X)p(Y,Z)} dXdYdZ \\ &= \int_{XYZ} p(X)p(Y,Z|X) \ln \frac{p(X)p(Y,Z|X)}{p(X)p(Y)p(Z)} dXdYdZ \\ &= \int_{XYZ} p(X)p(Y|X)p(Z) \ln \frac{p(X)p(Y|X)}{p(X)p(Y)} dXdYdZ \\ &= \int_{XY} p(X,Y) \ln \frac{p(X,Y)}{p(X)p(Y)} dXdY = I(X;Y) \end{aligned} \tag{A24}$$

□

Definition A1. *(Squeezed Information Channel): A squeezed information channel is a graphic model as shown in Figure A1 which also satisfies the following requirements: (1) the mapping ψ from X to X' is a bijection; (2) χ_q is a q dimensional projector, that is U is a q dimensional projection of X'; (3) U and V form a Markov chain $U \to V$, and f is the conditional probability $P(V|U)$; (3) $\mathcal{N}$ is a random noise which is independent on all other variables; (4) $Y = V \bigoplus N$.*

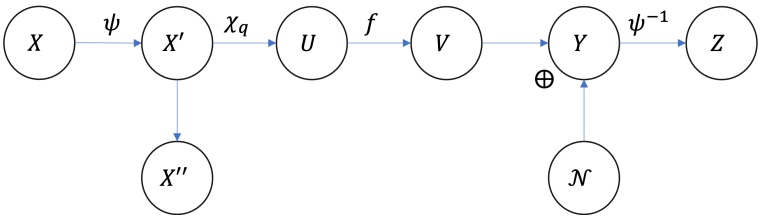


Figure A1. The graphic model of a squeezed information channel.

It is not hard to know that Figure 1 is actually a special case of the squeezed information channel, where $\mathbf{y}_t$, $\mathbf{y}(t+1)$, and $\hat{\mathbf{x}}_{t+1}$ correspond to U, V, and Z respectively.

For this general graphic model, we can prove Theorem 2:

Theorem 2 (Information bottleneck of the Squeezed Information Channel): For the squeezed information channel as shown in Figure A1 and for any ψ, f and $\mathcal{N}$, we have:

$$I(X;Z) = I(U;V) \tag{A25}$$

Proof. Because ψ and ψ^{-1} are all one to one mappings, thus, according to Lemmas A1–A3:

$$I(X;Z) = I(X';Z) = I(X';Y) = I(U;Y) = I(U;V) \tag{A26}$$

□

Therefore, the information of the whole squeezed channel is determined only by the Markov chain f, i.e., the macro-dynamics. Thus f is the bottleneck of the whole squeezed channel.

The Neural Information Squeezer framework can be converted as a Squeezed Information Channel as shown in Figure 3, in which, $\mathbf{x}_t, \mathbf{x}'_t, \mathbf{y}_t, \mathbf{y}(t+1), \mathbf{y}(t+1)', \hat{\mathbf{x}}_{t+1}$, and $\mathbf{z}_{p-q}$ correspond to X, X', U, V, Y, Z, and $\mathcal{N}$, respectively. Therefore,

$$I(\mathbf{y}_t; \mathbf{y}(t+1)) = I(\mathbf{x}_t; \hat{\mathbf{x}}_{t+1}) \tag{A27}$$

We can further extend Theorem 2 to the case of stacked neural information squeezer by the following corollary:

Corollary A1. *Theorem 2 can be extended to stacked neural information squeezer models.*

Proof. Because Equation (A27) holds for any neural information squeezer model whose encoder and decoder are composed by basic units of NIS, therefore, theorem 2 holds for stacked neural information squeezer models. □

Appendix C. Proof of Theorem 3

At first, we restate Theorem 3 as follow:

Theorem 3 (Mutual information of the model will be closed to the data for a well trained framework): If the neural networks in NIS framework are well-trained, which means

$$p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_t) \simeq p(\mathbf{x}_{t+1}|\mathbf{x}_t) \tag{A28}$$

where $p \simeq q$ means $D_{KL}(p, q) \to 0$ when the training epoch $\tau \to \infty$, then:

$$I(\hat{\mathbf{x}}_{t+1}; \mathbf{x}_t) \simeq I(\mathbf{x}_{t+1}; \mathbf{x}_t) \tag{A29}$$

where $\simeq$ means asymptotic equivalence when $\tau \to \infty$.

Proof. According to the objective function, i.e., Equations (13) and (14), we know that if the neural networks in NIS framework are well-trained, Equation (A28) is hold. Therefore:

$$p(\mathbf{x}_t, \hat{\mathbf{x}}_{t+1}) = p(\mathbf{x}_t) \cdot p(\hat{\mathbf{x}_{t+1}}|\mathbf{x}_t) \simeq p(\mathbf{x}_t) \cdot p(\mathbf{x}_{t+1}|\mathbf{x}_t) = p(\mathbf{x}_t, \mathbf{x}_{t+1}), \tag{A30}$$

and also:

$$p(\hat{\mathbf{x}}_{t+1}) = \int_{\mathbf{x}_t} p(\mathbf{x}_t, \hat{\mathbf{x}}_{t+1}) d\mathbf{x}_t \simeq \int_{\mathbf{x}_t} p(\mathbf{x}_t, \mathbf{x}_{t+1}) d\mathbf{x}_t = p(\mathbf{x}_{t+1}), \tag{A31}$$

so,

$$I(\hat{\mathbf{x}}_{t+1}; \mathbf{x}_t)) \simeq I(\mathbf{x}_{t+1}; \mathbf{x}_t) \tag{A32}$$

□

Corollary A2. *(The mutual information of macro-dynamics will not change if the model is well trained): For the well trained NIS model, the Mutual Information of the macro-dynamics f_β will be irrelevant on all the parameters, including the scale q.*

Proof. According to Theorems 2 and 3:

$$I(\mathbf{y}_t; \mathbf{y}(t+1)) = I(\mathbf{x}_t; \hat{\mathbf{x}}_{t+1}) \approx I(\mathbf{x}_t; \mathbf{x}_{t+1}). \tag{A33}$$

This equality is irrelevant with all parameters in neural network and q. □

Appendix D. Proof for Theorem 4

Lemma A3. *For any continuous random variable X and Y, we have:*

$$I(X;Y) = H(Y) + \mathbb{E}_X(\ln|\det(\frac{\partial Y}{\partial X})|), \tag{A34}$$

where $\frac{\partial Y}{\partial X}$ *is the Jacobian matrix.*

Proof. According to the computation of the mutual information by a continuous mapping:

$$I(X;Y) = H(Y) - H(Y|X), \tag{A35}$$

and:

$$H(Y|X) = -\mathbb{E}_X(\ln|\det(\frac{\partial Y}{\partial X})| \tag{A36}$$

according to [47], thus:

$$I(X;Y) = H(Y) + \mathbb{E}_X(\ln|\det(\frac{\partial Y}{\partial X})|). \tag{A37}$$

□

Theorem 4 (Information on bottleneck is the lower bound of the encoder): For squeezed information chain shown in Figure A1, the information of $U \to V$ is bounded by:

$$I(U;V) \leq I(X;U) = H(U) + \mathbb{E}_X(\ln|\det(\frac{\partial U}{\partial X})|) \leq I(X;X') = H(X) + \mathbb{E}_X(\ln|\det(\frac{\partial X'}{\partial X})|)). \tag{A38}$$

Proof. Because both ψ, χ_q, and f are Markovian, so $X \to X' \to U \to V$ forms a Markov chain. Thus the data processing inequality holds:

$$I(U;V) = I(X;Z) \leq I(X;U) \leq I(X;X'), \tag{A39}$$

and according to Lemma A3:

$$I(U;V) \leq H(U) + \mathbb{E}_X(\ln|\det(\frac{\partial U}{\partial X})|) \leq H(X) + \mathbb{E}_X(\ln|\det(\frac{\partial X'}{\partial X})|)). \tag{A40}$$

□

Applying this theorem in the information squeezed channel (Figure A1), we can obtain the form of Equation (23).

Proof for Theorem 5

We re-state Theorem 5:

Theorem 5 (The mathematical expression for effective information of macro-dynamics): Suppose the probability density of $\mathbf{x}_{t+1}$ under given $\mathbf{x}_t$ can be described by a function $p(\mathbf{x}_{t+1}|\mathbf{x}_t) \equiv G(\mathbf{x}_{t+1}, \mathbf{x}_t)$, and the Neural Information Squeezer framework is well trained, then the effective information of the macro-dynamics of f_β can be calculated by:

$$EI_L(f_\beta) = \frac{1}{(2L)^p} \cdot \int_\sigma \int_{\mathcal{R}^p} G(\mathbf{y}, \psi_\alpha^{-1}(\mathbf{x})) \ln \frac{(2L)^p G(\mathbf{y}, \psi_\alpha^{-1}(\mathbf{x}))}{\int_\sigma G(\mathbf{y}, \psi_\alpha^{-1}(\mathbf{x}'))d\mathbf{x}'} d\mathbf{y}d\mathbf{x}, \tag{A41}$$

where $\sigma \equiv [-L, L]^p$ is the integration region for $\mathbf{x}$ and $\mathbf{x}'$.

Proof. According to Definition of effective information (EI),

$$EI_L(f_{\phi_q}) \equiv I(do(\mathbf{y}_t \equiv \mathbf{u} \sim U([-L, L]^q)); \mathbf{y}(t+1)). \tag{A42}$$

The effect of the do operator can be understood by another graphic model which is shown in Figure A2a.

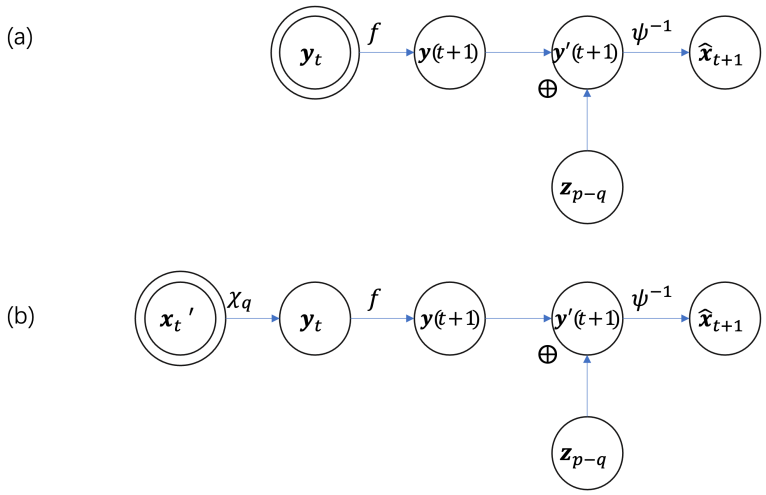


Figure A2. The graphic model of the squeezed information channel Figure 3 after do operation. (**a**) do operator acts on the $\mathbf{y}_t$ node, and (**b**) do operator acts on the $\mathbf{x}'_t$ node. The nodes with double circles are the nodes that the do operator acts on.

For the squeezed information channel as shown in Figure 3, because $\mathbf{y}_t$ is the projection of $\mathbf{x}'_t$ on q dimension, so if $\mathbf{x}'_t \sim U([-L,L]^p)$, then $\mathbf{y}_t \sim U([-L,L]^q)$, but the density increased by a factor $(2L)^{p-q}$. In addition, because $\mathbf{x}_t \to \mathbf{x}'_t \to \mathbf{y}_t \to \mathbf{y}(t+1)$, So, the graphic model of Figure A2a is equivalent to the graph in Figure A2b. And according to Lemmas A1 and A2:

$$I(\mathbf{y}_t; \mathbf{y}(t+1)) = I(\mathbf{x}'_t; \mathbf{y}(t+1)), = I(\mathbf{x}'_t; \hat{\mathbf{x}}_{t+1}) \tag{A43}$$

therefore,

$$\begin{aligned} EI_L(f_{\phi_q}) &\equiv I(do(\mathbf{y}_t \sim U([-L,L]^q)); \mathbf{y}(t+1)) \\ &= I\big(do(\mathbf{x}'_t \sim U([-L,L]^p)); \mathbf{y}(t+1)\big) \\ &= I\big(do(\mathbf{x}'_t \sim U([-L,L]^p)); \hat{\mathbf{x}}_{t+1}\big). \end{aligned} \tag{A44}$$

And,

$$\begin{aligned} &I\big(do(\mathbf{x}'_t \sim U([-L,L]^p)); \hat{\mathbf{x}}_{t+1}\big) \\ &= \int_\sigma \rho \cdot \left(\int_{\hat{\mathbf{x}}_{t+1} \in \mathcal{R}^q} p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}'_t) \ln \frac{p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}'_t)}{\rho \int_\sigma p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}'_t) d\mathbf{x}'_t} \cdot d\hat{\mathbf{x}}_{t+1} \right) d\mathbf{x}'_t \end{aligned} \tag{A45}$$

where $\sigma = [-L,L]^p$ is the integration region, and $\rho = (2L)^{-p}$. Meanwhile, the conditional probability of $\hat{\mathbf{x}}_{t+1}$ under given $\mathbf{x}_t$ is a function $p(\hat{\mathbf{x}}_{t+1} = y|\mathbf{x}_t = x) \equiv F(y,x)$, so:

$$p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}'_t) = p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_t = \psi^{-1}(\mathbf{x}'_t)) = F(\hat{\mathbf{x}}_{t+1}, \psi^{-1}(\mathbf{x}'_t)) \tag{A46}$$

and according to Theorem 3, if the NIS framework is well trained, we have:

$$p(\hat{\mathbf{x}}_{t+1}|\mathbf{x}_t) = F(\hat{\mathbf{x}}_{t+1}, \mathbf{x}_t) \approx G(\mathbf{x}_{t+1}, \mathbf{x}_t) \tag{A47}$$

Therefore:

$$\begin{aligned} & I\left(do(\mathbf{x}_t' \sim U([-L,L]^p)); \hat{\mathbf{x}}_{t+1}\right) \\ &= \int_\sigma \rho \cdot \left(\int_{\hat{\mathbf{x}}_{t+1} \in \mathcal{R}^q} F(\hat{\mathbf{x}}_{t+1}, \psi^{-1}(\mathbf{x}_t')) \ln \frac{F(\hat{\mathbf{x}}_{t+1}, \psi^{-1}(\mathbf{x}_t'))}{\rho \int_\sigma F(\hat{\mathbf{x}}_{t+1}, \psi^{-1}(\mathbf{x}_t'))d\mathbf{x}_t'} \cdot d\hat{\mathbf{x}}_{t+1} \right) d\mathbf{x}_t' \\ &\approx \int_\sigma \rho \cdot \left(\int_{\mathbf{x}_{t+1} \in \mathcal{R}^q} G(\mathbf{x}_{t+1}, \psi^{-1}(\mathbf{x}_t')) \ln \frac{G(\mathbf{x}_{t+1}, \psi^{-1}(\mathbf{x}_t'))}{\rho \int_\sigma G(\mathbf{x}_{t+1}, \psi^{-1}(\mathbf{x}_t'))d\mathbf{x}_t'} \cdot d\mathbf{x}_{t+1} \right) d\mathbf{x}_t' \end{aligned} \tag{A48}$$

We then use $\mathbf{x}$ or $\mathbf{x}'$ to replace $\mathbf{x_t}'$ and use $\mathbf{y}$ to replace $\mathbf{x}_{t+1}$ in the integrations; then, we have:

$$EI_L(f_\beta) \approx \frac{1}{(2L)^p} \int_\sigma \left(\int_{\mathbf{y} \in \mathcal{R}^q} G(\mathbf{y}, \psi^{-1}(\mathbf{x})) \ln \frac{G(\mathbf{y}, \psi^{-1}(\mathbf{x}))}{\rho \int_\sigma G(\mathbf{y}, \psi^{-1}(\mathbf{x}'))d\mathbf{x}'} \cdot d\mathbf{y} \right) d\mathbf{x} \tag{A49}$$

□

Appendix E. Proof for Theorem 6

Theorem 6 (Narrower is Harder): If X is random variable with dimension p, and if the q_1 dimensional random variable U_{q_1} is the projection of a q_2 dimensional variable U_{q_2}, and $0 < q_1 < q_2 < p$, then:

$$I(X; U_{q_1}) \leq I(X; U_{q_2}). \tag{A50}$$

Proof. Because $q_1 < q_2$, therefore, U_{q_2} contains U_{q_1} as the component, thus, there exists a $q_2 - q_1$ dimensional random variable $U'_{q_2-q_1}$ such that:

$$U_{q_2} = U_{q_1} \bigoplus U'_{q_2-q_1} \tag{A51}$$

Therefore:

$$H(U_{q_2}) = H(U_{q_1}, U'_{q_2-q_1}) = H(U_{q_1}) + H(U'_{q_2-q_1}|U_{q_1}) \geq H(U_{q_1}) \tag{A52}$$

because $H(U'_{q_2-q_1}|U_{q_1}) \geq 0$, and:

$$\mathbb{E}_X(\ln|\det(\frac{\partial U_{q_2}}{\partial X})| \geq \mathbb{E}_X(\ln|\det(\frac{\partial U_{q_1}}{\partial X})| \tag{A53}$$

because the matrices of $\frac{\partial U_{q_2}}{\partial X}$ and $\frac{\partial U_{q_2}}{\partial X}$ are all sub-matrices of $\frac{\partial \psi}{\partial X}$ and the former contains the latter. Thus, according to Lemma A3:

$$I(X; U_{q_2}) = H(U_{q_2}) + \mathbb{E}_X(\ln|\det(\frac{\partial U_{q_2}}{\partial X})| \geq H(U_{q_1}) + \mathbb{E}_X(\ln|\det(\frac{\partial U_{q_1}}{\partial X}) = I(X; U_{q_1}) \tag{A54}$$

□

Thus, if the number of dimension is smaller, the mutual information between X and U will also be smaller. That means, narrower channel is harder to transfer information.

Combined with Theorems 2 and 4, we have the following inequalities for the squeezed information channel of Figure 3:

$$I(\mathbf{x}_t; \hat{\mathbf{x}}_{t+1}) \leq I(\mathbf{x}_t; \mathbf{y}_t^{q_1}) \leq I(\mathbf{x}_t; \mathbf{y}_t^{q_2}). \tag{A55}$$

This is the form of Theorem 6 in the main text.

References

1. Holland, J.H. *Emergence: From Chaos To Order*; Illustrated edition; Basic Books: New York, NY, USA, 1999.
2. Bedau, M.A. Weak Emergence. *Philos. Perspect.* **1997**, *11*, 375–399.
3. Pearl, J. *Causality: Models of Reasoning and Inference*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2009.

4. Granger, C.W.J. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica* **1969**, *37*, 424–438.
5. Hoel, E.P.; Albantakis, L.; Tononi, G. Quantifying causal emergence shows that macro can beat micro. *Proc. Natl. Acad. Sci. USA* **2013**, *110*, 19790–19795.
6. Hoel, E.P. When the Map Is Better Than the Territory. *Entropy* **2017**, *19*, 188.
7. Tononi, G.; Sporns, O. Measuring information integration. *BMC Neurosci.* **2003**, *4*, 31.
8. Varley, T.; Hoel, E. Emergence as the conversion of information: A unifying theory. *arXiv* **2021**, arXiv:2104.13368.
9. Chvykov, P.; Hoel, E. Causal Geometry. *Entropy* **2021**, *23*, 24.
10. Rosas, F.E.; Mediano, P.A.M.; Jensen, H.J.; Seth, A.K.; Barrett, A.B.; Carhart-Harris, R.L.; Bor, D. Reconciling emergences: An information-theoretic approach to identify causal emergence in multivariate data. *PLoS Comput. Biol.* **2020**, *16*, e1008289.
11. Varley, T.F. Flickering emergences: The question of locality in information-theoretic approaches to emergence. *arXiv* **2022**, arXiv:2208.14502.
12. Swain, A.; Williams, S.D.; Di Felice, L.J.; Hobson, E.A. Interactions and information: Exploring task allocation in ant colonies using network analysis. *Anim. Behav.* **2022**, *189*, 69–81.
13. Klein, B.; Hoel, E.; Swain, A.; Griebenow, R.; Levin, M. Evolution and emergence: Higher order information structure in protein interactomes across the tree of life. *Integr. Biol.* **2021**, *13*, 283–294.
14. Ravi, D.; Hamilton, J.L.; Winfield, E.C.; Lalta, N.; Chen, R.H.; Cole, M.W. Causal emergence of task information from dynamic network interactions in the human brain. *Rev. Neurosci.* **2022**, *31*, 25–46.
15. Klein, B.; Swain, A.; Byrum, T.; Scarpino, S.V.; Fagan, W.F. Exploring noise, degeneracy and determinism in biological networks with the einet package. *Methods Ecol. Evol.* **2022**, *13*, 799–804.
16. Klein, B.; Hoel, E. The Emergence of Informative Higher Scales in Complex Networks. *Complexity* **2020**, *2020*, 8932526.
17. Varley, T.; Hoel, E. Emergence as the conversion of information: A unifying theory. *Philos. Trans. R. Soc. A* **2022**, *380*, 20210150.
18. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359.
19. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
20. Reichstein, M.; Camps-Valls, G.; Stevens, B.; Jung, M.; Denzler, J.; Carvalhais, N. Deep learning and process understanding for data-driven Earth system science. *Nature* **2019**, *566*, 195–204.
21. Senior, A.W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A.W.R.; Bridgland, A.; et al. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, *577*, 706–710.
22. Tank, A.; Covert, I.; Foti, N.; Shojaie, A.; Fox, E. Neural Granger Causality. *arXiv* **2018**, arXiv:1802.05842.
23. Löwe, S.; Madras, D.; Zemel, R.; Welling, M. Amortized causal discovery: Learning to infer causal graphs from time-series data. *arXiv* **2020**, arXiv:2006.10833.
24. Glymour, C.; Zhang, K.; Spirtes, P. Review of Causal Discovery Methods Based on Graphical Models. *Front. Genet.* **2019**, *10*, 524.
25. Casadiego, J.; Nitzan, M.; Hallerberg, S.; Timme, M. Model-free inference of direct network interactions from nonlinear collective dynamics. *Nat. Commun.* **2017**, *8*, 1–10.
26. Sanchez-Gonzalez, A.; Heess, N.; Springenberg, J.T.; Merel, J.; Riedmiller, M.; Hadsell, R.; Battaglia, P. Graph networks as learnable physics engines for inference and control. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4470–4479.
27. Zhang, Z.; Zhao, Y.; Liu, J.; Wang, S.; Tao, R.; Xin, R.; Zhang, J. A general deep learning framework for network reconstruction and dynamics learning. *Appl. Netw. Sci.* **2019**, *4*, 1–17.
28. Kipf, T.; Fetaya, E.; Wang, K.C.; Welling, M.; Zemel, R. Neural relational inference for interacting systems. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 2688–2697.
29. Chen, B.; Huang, K.; Raghupathi, S.; Chandratreya, I.; Du, Q.; Lipson, H. Discovering State Variables Hidden in Experimental Data. *arXiv* **2021**, arXiv:2112.10755.
30. Koch-Janusz, M.; Ringel, Z. Mutual information, neural networks and the renormalization group. *Nat. Phys.* **2018**, *14*, 578–582.
31. Li, S.H.; Wang, L. Neural Network Renormalization Group. *Phys. Rev. Lett.* **2018**, *121*, 260601.
32. Hu, H.Y.; Li, S.H.; Wang, L.; You, Y.Z. Machine learning holographic mapping by neural network renormalization group. *Phys. Rev. Res.* **2020**, *2*, 023369.
33. Hu, H.; Wu, D.; You, Y.Z.; Olshausen, B.; Chen, Y. RG-Flow: A hierarchical and explainable flow model based on renormalization group and sparse prior. *Mach. Learn. Sci. Technol.* **2022**, *3*, 035009.
34. Gökmen, D.E.; Ringel, Z.; Huber, S.D.; Koch-Janusz, M. Statistical physics through the lens of real-space mutual information. *Phys. Rev. Lett.* **2021**, *127*, 240603.
35. Chalupka, K.; Eberhardt, F.; Perona, P. Causal feature learning: An overview. *Behaviormetrika* **2017**, *44*, 137–164.
36. Schölkopf, B.; Locatello, F.; Bauer, S.; Ke, N.R.; Kalchbrenner, N.; Goyal, A.; Bengio, Y. Toward causal representation learning. *Proc. IEEE* **2021**, *109*, 612–634.
37. Iwasaki, Y.; Simon, H.A. Causality and model abstraction. *Artif. Intell.* **1994**, *67*, 143–194.
38. Rubenstein, P.K.; Weichwald, S.; Bongers, S.; Mooij, J.; Janzing, D.; Grosse-Wentrup, M.; Schölkopf, B. Causal consistency of structural equation models. *arXiv* **2017**, arXiv:1707.00819.

39. Beckers, S.; Eberhardt, F.; Halpern, J.Y. Approximate causal abstractions. In Proceedings of the Uncertainty in Artificial Intelligence, Virtual, 3–6 August 2020; pp. 606–615.
40. Beckers, S.; Eberhardt, F.; Halpern, J.Y. Approximate Causal Abstraction. *arXiv* **2019**, arXiv:1906.11583v2.
41. Teshima, T.; Ishikawa, I.; Tojo, K.; Oono, K.; Ikeda, M.; Sugiyama, M. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 3362–3373.
42. Teshima, T.; Tojo, K.; Ikeda, M.; Ishikawa, I.; Oono, K. Universal approximation property of neural ordinary differential equations. *arXiv* **2017**, arXiv:2012.02414.
43. Dinh, L.; Sohl-Dickstein, J.; Bengio, S. Density estimation using real nvp. *arXiv* **2016**, arXiv:1605.08803.
44. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
45. Shwartz-Ziv, R.; Tishby, N. Opening the black box of deep neural networks via information. *arXiv* **2017**, arXiv:1703.00810.
46. Williams, P.L.; Beer., R.D. Nonnegative decomposition of multivariate information. *arXiv* **2017**, arXiv:1004.2515.
47. Geiger, B.C.; Kubin, G. On the information loss in memoryless systems: The multivariate case. *arXiv* **2011**, arXiv:1109.4856.