C O M M O N S   D E E D

저작자표시-비영리-변경금지 **2.0** 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

**저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.**

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

교육학석사 학위논문

# Binary Data Augmentation via Diffusion Model for Solving Inverse Ising Problem

역 이징 문제 해결을 위한 Diffusion Model을 이용한 이진 데이터 증강

2024년 8월

서울대학교 대학원

과학교육과 물리전공

임 예 찬

# Binary Data Augmentation via Diffusion Model for Solving Inverse Ising Problem

## 역 이징 문제 해결을 위한 Diffusion Model 을 이용한 이진 데이터 증강

지도 교수 조정효

이 논문을 교육학석사 학위논문으로 제출함
2024 년 8 월

서울대학교 대학원
과학교육과(물리전공)
임예찬

임예찬의 교육학석사 학위논문을 인준함
2024 년 8 월

위 원 장 채승철 (인)

부위원장 석효준 (인)

위 원 조정효 (인)

# Abstract

In the pursuit of identifying model parameters from observed configurations, we address the Inverse Ising Problem, a fundamental challenge in statistical physics. Our study introduces a novel approach to augment Ising data using a diffusion model. Diffusion model, unlike the Boltzmann machine, learns the score function of a given distribution, enabling the learning of data distribution without the need for intractable normalization term calculations. This allows for Ising data augmentation through Diffusion model. We employ the erasure machine to efficiently utilize the augmented data for parameter inference. we reveal that samples generated by the diffusion model improve the accuracy of inferring Ising model parameter. We present results across various system sizes, discuss the impact of observed and augmented data quantity on the performance, and demonstrate the effectiveness of the approach in real-world applications, such as inferring missing values in neuronal activity data. This work not only introduces a new paradigm for Ising data augmentation but also provides insights into the broader applicability of diffusion models in data science and physics-inspired machine learning.

**Keywords** : Diffusion Model, Inverse Ising problem, data Augmentation

**Student Number** : 2022-27415

# Table of Contents

# List of figure

# Ⅰ. Introduction

The pursuit of identifying model parameters from given configurations stands as a paramount objective in data science. In statistical physics, deriving observable quantities from microscopic laws governing the constituents of a system holds significant importance. In the context of the Ising model, the goal is to specify interactions among spins, unraveling spin magnetization and correlations.

The Inverse Ising Problem commences with unknown microscopic parameters, where only the observable quantities of the systems are known. Despite the unknown interactions among spins, our aim is to deduce uncharted parameters from the given data, driven by the desire to understand thermodynamic observables such as magnetization and correlation. The Inverse statistical problem finds application in elucidating diverse phenomena, ranging from protein structures[1,2] to the potentials between atoms leading to specific crystal lattices[3], financial markets[4], gene recombinations[5], and human interactions[6]. Its utility extends to the design of complex systems, making it a vital tool in scientific inquiry and technological innovation.

As a concrete illustration, consider a system comprising $\mathrm{N}$ observed configurations of $\mathrm{M}$ binary variables $\sigma_i, i = 1, \ldots, M, \sigma_i = \pm 1$. These binary variables are correlated with Ising spins, and the likelihood of observing a particular configuration $\sigma =$

$(\sigma_1, \sigma_2, \ldots, \sigma_M)$ is postulated to follow the normalized Boltzmann weight:

$$\mathrm{p}(\sigma_i) = \frac{\exp[-H(\sigma_i)]}{Z},\ Z = \ \textstyle\sum_\sigma \exp[-H(\sigma_i)] \tag{1}$$

these spins are coupled by pairwise couplings $J_{ij}$ and are subject to external magnetic fields $h_i$. The Hamiltonian

$$\mathrm{H}(\sigma_i) = -\textstyle\sum_i h_i\, \sigma_i - \sum_{i<j} J_{ij}\, \sigma_i \sigma_j \tag{2}$$

The spin system' s energy is determined by microscopic spin variables, local fields, and pairwise couplings. In the context of the Inverse Ising problem, the task is to deduce the couplings $J_{ij}$ and local field $h_i$ based on a given set of $\mathrm{N}$ observed spin configurations. Employing Maximum Likelihood Estimation (MLE), our goal is to identify $\mathrm{p}(\sigma)$ that maximizes the likelihood. This involves comparing the Boltzmann distribution with the empirical distribution of the data in the sample set $\mathrm{D}$, denoted as $\mathrm{f}(\sigma) = \frac{1}{N} \sum_\mu \delta_{\sigma^\mu, \sigma}$ , $\mu = 1, 2, \ldots, \mathrm{N}$. The disparity between two probability distributions, $\mathrm{f}(\sigma)$ and $\mathrm{p}(\sigma)$, can be quantified using the Kullback–Leibler (KL) divergence.

$$D_{KL} = \ \textstyle\sum_\sigma f(\sigma) \log \frac{f(\sigma)}{p(\sigma)} = -L_D(J, h) + \ \sum_\sigma f(\sigma) \log f(\sigma) \tag{3}$$

The second term is independent of model parameter, so minimizing $D_{KL}$ is achieved by maximizing likelihood. we have

$$\frac{\partial \mathrm{L_D}}{\partial h_i}(\mathrm{J},\mathrm{h}) = \langle \sigma_\mathrm{i} \rangle_\mathrm{f} - \langle \sigma_\mathrm{i} \rangle_\mathrm{p} \tag{4}$$

$$\frac{\partial \mathrm{L_D}}{\partial J_{ij}}(\mathrm{J},\mathrm{h}) = \langle \sigma_\mathrm{i}\sigma_\mathrm{j} \rangle_\mathrm{f} - \langle \sigma_\mathrm{i}\sigma_\mathrm{j} \rangle_\mathrm{p} \tag{5}$$

Then we can update parameter as follow

$$h_i^{n+1} = h_i^n + \alpha(\langle \sigma_\mathrm{i} \rangle_\mathrm{f} - \langle \sigma_\mathrm{i} \rangle_\mathrm{p}) \tag{6}$$

$$J_{ij}^{n+1} = J_{ij}^n + \alpha(\langle \sigma_\mathrm{i}\sigma_\mathrm{j} \rangle_\mathrm{f} - \langle \sigma_\mathrm{i}\sigma_\mathrm{j} \rangle_\mathrm{p}) \tag{7}$$

With a tunable parameter of learning rate $\alpha$.

The dependence on parameters is entirely encapsulated in $\langle \sigma_\mathrm{i} \rangle_\mathrm{p}$ and $\langle \sigma_\mathrm{i}\sigma_\mathrm{j} \rangle_\mathrm{p}$. Utilizing Eq. (4, 5), gradient ascent for parameter determination encounters typical challenges associated with local maxima, especially in case of sparse data. Regardless of the dataset size, N, the computation involves a summation with $2^M$ terms.

Addressing the computational intractability of the partition function has been a significant hurdle for physicists. Various approximate solutions have been developed, including mean-field methods [7], Bethe approximations [8], and machine learning techniques utilizing variational autoregressive networks [9]. Notably, the Erasure machine offers precise inference with both synthetic data and real-world examples [10].

To obtain optimal model parameters, a substantial amount of data is required, yet real-world scenarios often present limited data availability.

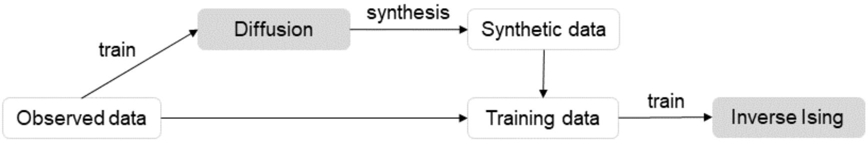


Figure 1. data augmentation process

Our goal is to augment Ising data for solving Inverse Ising problem using diffusion model (Figure 1). Data augmentation is the process of transforming existing training data to generate new training samples, aiming to address the insufficient quantity of data. This helps improve the generalization performance of models and prevent overfitting. Data augmentation is not limited to image data [11] but can also be applied to various types of data, including text data [12], time-series data [13], and others. This technique can be utilized across a range of tasks to enhance the performance of machine learning model.

Diffusion models [14-16] are physics inspired latent variable models. Recently, diffusion-based tools and applications have grown rapidly and a lot of related research is being conducted. The diffusion model is applied to various fields and show good performance. It is a state-of-the-art family of deep generative models that covers not only computer vision but also audio [17], natural language [18], temporal data [19], and drug molecules design [20], As an image data augmentation method, diffusion model shows better performance than other generative model like GAN [21].

$$\mathrm{s}(\sigma) \equiv \nabla_\sigma \log p(\sigma) = -\nabla_\sigma H(\sigma) - \cancel{\nabla_\sigma \log Z} = -\nabla_\sigma H(\sigma) \qquad (8)$$

The diffusion model learns the score of $\mathrm{p}(\sigma)$ without directly estimating $\mathrm{p}(\sigma)$ Eq. (8). By leveraging Langevin Markov Chain Monte Carlo sampling, it generates data following the inferred $\mathrm{p}(\sigma)$ without explicitly computing the partition function. This method allows the model to learn a smooth distribution rather than being solely dependent on the data. Furthermore, as a generative model, the diffusion model can generate an infinite amount of data, this making data augmentation feasible.

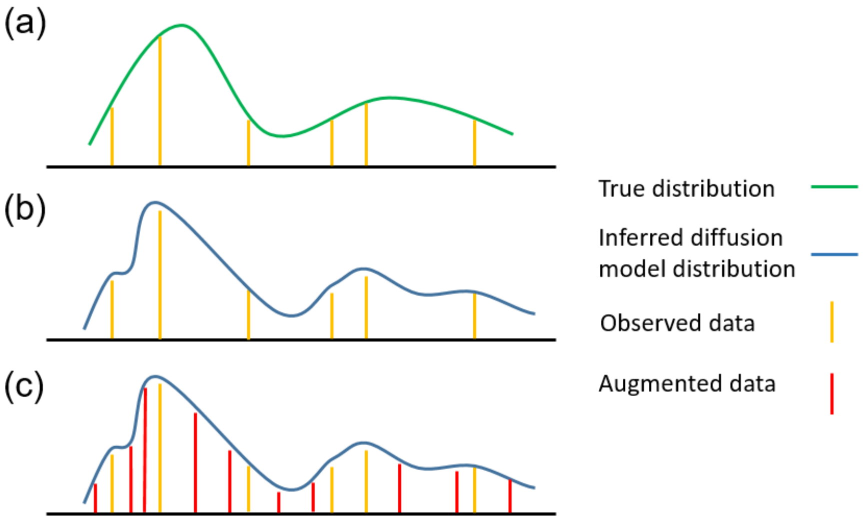


Figure 2. data augmentation as a perspective of filling missing data points of observed data. (a) Observed data sampled from True distribution. (b) Diffusion model’s distribution trained by observed data. (c) Augmented data sampled from diffusion model.

We treat augmented data as missing data points of observed data(training data)[22]. We cannot collect the entire data set representing the true distribution, known as the population (Figure 2(a)). Instead, we only obtain a subset of the population as observed data. we hope that the diffusion model infers the true distribution from the observed data (Figure 2(b)) and generate missing points outside of the observed data (Figure 2(c)).

We have observed that augmented data generated by the diffusion model impute the true parameter $h_i$, $J_{ij}$ more accurately than only using observed data. this means diffusion model can fill missing point of the observed data. This is the first study on Ising data augmentation.

# Ⅱ. Methods

## 2.1 Erasure Machine method

A very effective algorithm called EM(Erasure Machine) was proposed as an inverse Ising problem method [10]. EM solved intractability of the partition function through approximation by reweighting the observed data distribution $\mathrm{f}(\sigma)$ and model distribution $\mathrm{p}(\sigma)$ . $f_{\epsilon}(\sigma) \propto f(\sigma) p^{\epsilon-1}(\sigma)$, $p_{\epsilon}(\sigma) \propto p(\sigma) p^{\epsilon-1}(\sigma) = p^{\epsilon}$ , respectively, with the tunable reweighting parameter $(0 < \epsilon < 1)$. $\epsilon$ plays an important role in EM algorithm. $p^{\epsilon}$ put the system at higher temperature from $\beta$ to $\beta\epsilon$, this means that reweighting process resembles the high-temperature approximation in statistical mechanics. Then Eq. (4, 5) transform as follow

$$\frac{\partial \mathrm{L_D}}{\partial h_i}(\mathrm{J}, \mathrm{h}) = \langle \sigma_{\mathrm{i}} \rangle_{f_{\epsilon}} - \langle \sigma_{\mathrm{i}} \rangle_{p_{\epsilon}} \tag{9}$$

$$\frac{\partial \mathrm{L_D}}{\partial J_{ij}}(\mathrm{J}, \mathrm{h}) = \langle \sigma_{\mathrm{i}} \sigma_{\mathrm{j}} \rangle_{f_{\epsilon}} - \langle \sigma_{\mathrm{i}} \sigma_{\mathrm{j}} \rangle_{p_{\epsilon}} \tag{10}$$

The first expectation still needs to consider only observed configurations and second expectation follow

$$p_{\epsilon}(\sigma) = \frac{\exp[-\epsilon E(\sigma)]}{Z_{\epsilon}} \text{ with } Z_{\epsilon} = \sum_{\sigma} \exp[-\epsilon E(\sigma)] \tag{11}$$

The EM algorithm's advantage is that we can approximately compute the partition function for given $\epsilon$.

$$Z_\epsilon = \sum_\sigma \exp[-\epsilon E(\sigma)] = \sum_\sigma \exp\left(\sum_i \epsilon h_i \sigma_i + \sum_{j<k} \epsilon J_{ik}\sigma_j\sigma_k\right)$$

$$= \sum_\sigma \prod_i \cosh(\epsilon h_i)\,[1 + \sigma_i \tanh(\epsilon h_i)] \prod_{j<k} \cosh(\epsilon J_{jk})\,[1 + \sigma_j\sigma_k \tanh(\epsilon J_{jk})]$$

$$= 2^M \prod_i \cosh(\epsilon h_i) \prod_{j<k} \cosh(\epsilon J_{jk}) \left[1 + \sum_{l<m} \tanh(\epsilon h_l) \tanh(\epsilon h_m) \tanh(\epsilon h_n)\right.$$

$$\left. + \sum_{l<m<n} \tanh(\epsilon J_{lm}) \tanh(\epsilon J_{ln}) \tanh(\epsilon J_{mn}) + \mathcal{O}(\epsilon^4)\right] \tag{12}$$

Then, Logarithm of $Z_\epsilon$ is

$$\ln Z_\epsilon = M \ln 2 + \textstyle\sum_i \ln\cosh(\epsilon h_i) + \sum_{i<j} \ln\cosh\left(\epsilon J_{ij}\right) + \mathcal{O}(\epsilon^3) \tag{13}$$

Which leads to

$$\langle \sigma_\mathrm{i} \rangle_{p_\epsilon} = \frac{1}{\epsilon}\frac{\partial \ln Z_\epsilon}{\partial h_i} \approx \epsilon h_i \tag{14}$$

$$\langle \sigma_\mathrm{i}\sigma_\mathrm{j} \rangle_{p_\epsilon} = \frac{1}{\epsilon}\frac{\partial \ln Z_\epsilon}{\partial J_{ij}} \approx \epsilon J_{ij} \tag{15}$$

Therefore, update algorithm Eq. (6, 7) transform to

$$h_i^{n+1} = h_i^n + \alpha(\langle \sigma_\mathrm{i} \rangle_{f_\epsilon} - \epsilon h_i^n) \tag{15}$$

$$J_{ij}^{n+1} = J_{ij}^n + \alpha(\langle \sigma_\mathrm{i}\sigma_\mathrm{j} \rangle_{f_\epsilon} - \epsilon J_{ij}^n) \tag{16}$$

With a tunable parameter of learning rate $\alpha$. We use $\alpha = 0.1$ in this study.

## 2.2 Diffusion model

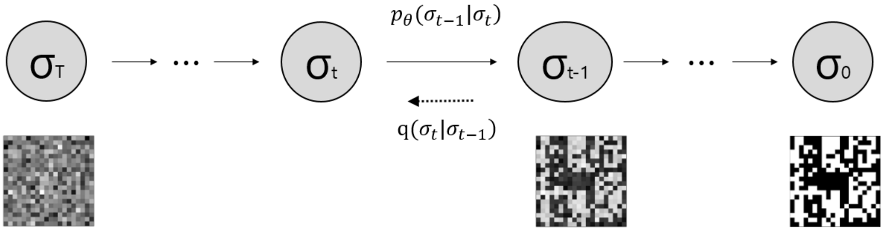


Figure 3. Diffusion process visualization considered in this work

Diffusion models are latent variable generative models characterized by a forward and a reverse Markov process. The forward process corrupts the data $q(\sigma_{1:T}|\sigma_0) = \prod_{t=1}^{T} q(\sigma_t|\sigma_{t-1})$ into a sequence of increasingly noisy latent variables $\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_T$. Construct a forward process in $\sigma_T$ to become a normal Gaussian $p(\sigma_T) = \mathcal{N}(\sigma_T; 0, I)$ , then generate samples through the backward process using a Markov chain with Gaussian transition $p_\theta(\sigma_{0:T}) = p(\sigma_T)\prod_{t=1}^{T} p_\theta(\sigma_{t-1}|\sigma_t)$ . Forward process gradually adds Gaussian noise, backward process learns remove Gaussian noise (Figure 3).

Given observed data $\sigma_0$ and noise $\epsilon \sim \mathcal{N}(0, I)$, one can obtain the Langevin equation that allows direct transition from $\sigma_0$ to $\sigma_t$.

$$\sigma_t = \sqrt{\bar{\alpha}_t}\sigma_0 + \sqrt{1 - \bar{\alpha}_t}\bar{\epsilon}_t \quad (17)$$

Here we use $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i, \alpha_t = 1 - \beta_t$ , $\beta_t$ is a parameter determining the intensity of noise over time, serving as the noise schedule. We defined a new standard normal variable, $\bar{\epsilon}_t$, as a linear combination of $\epsilon_1, \epsilon_2, \ldots, \epsilon_t$ each following a standard normal distribution.

To precisely define spin system' s probability model Eq. (1) one needs to determine the normalization constant Z. However, by utilizing the score function,

$$\mathrm{s}(\sigma) \equiv \nabla_\sigma \log p(\sigma) = -\nabla_\sigma H(\sigma) \tag{18}$$

which is the variation of the log-likelihood with respect to the change in $\sigma_t$, instead of the probability function itself, it is possible to bypass the computation of the normalization constant Z. Instead of observing a probability distribution as a scalar function in space, this becomes a vector function corresponding to a defined flow along the gradients of the probability distribution.

For a Gaussian variable $\sigma \sim \mathcal{N}(\sigma; \mu_\sigma, s^2 \mathrm{I})$, Tweedie' s Formula states that:

$$\mathrm{E}[\mu|\sigma] = \sigma + s^2 \nabla_\sigma \log p(\sigma) \tag{19}$$

It can be applied to a diffusion model following a normal distribution $\sigma_t \sim q(\sigma_t|\sigma_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\sigma_0, (1-\bar{\alpha}_t)\mathrm{I}\right)$.

$$\sqrt{\bar{\alpha}_t}\sigma_0 = \sigma_t + (1-\bar{\alpha}_t)\nabla_{\sigma_t} \log q(\sigma_t|\sigma_0) \tag{20}$$

Upon comparing with equation Eq. (17) ($\sigma_t = \sqrt{\bar{\alpha}_t}\sigma_0 + \sqrt{1-\bar{\alpha}_t}\bar{\epsilon}_t = \sqrt{\bar{\alpha}_t}\sigma_0 - (1-\bar{\alpha}_t)\nabla_{\sigma_t} \log q(\sigma_t|\sigma_0)$), a relationship between $\bar{\epsilon}_t$ and $\mathrm{s}(\sigma_t)$ can be elucidated.

$$\bar{\epsilon}_t = -\sqrt{1-\bar{\alpha}_t}\nabla_{\sigma_t}\log q(\sigma_t|\sigma_0) = -\sqrt{1-\bar{\alpha}_t}s(\sigma_t) \quad (21)$$

If we consider $\bar{\epsilon}_t$ as representing the direction from $\sigma_0 \to \sigma_t$, then the score $s(\sigma_t)$ indicates the direction from $\sigma_t \to \sigma_0$. Therefore, by training the noise term $\bar{\epsilon}_t$ it can be aligned with the corresponding score and used accordingly. Ho et al [15] employed a simple objective function for training (see Appendix A), as follow

$$L_{simple} = \mathbb{E}_{t,\sigma_0,\bar{\epsilon}_t} \parallel \bar{\epsilon}_t - \hat{\epsilon}(\sigma_t) \parallel^2 \quad (22)$$

## 2.3 Analog Bit Diffusion

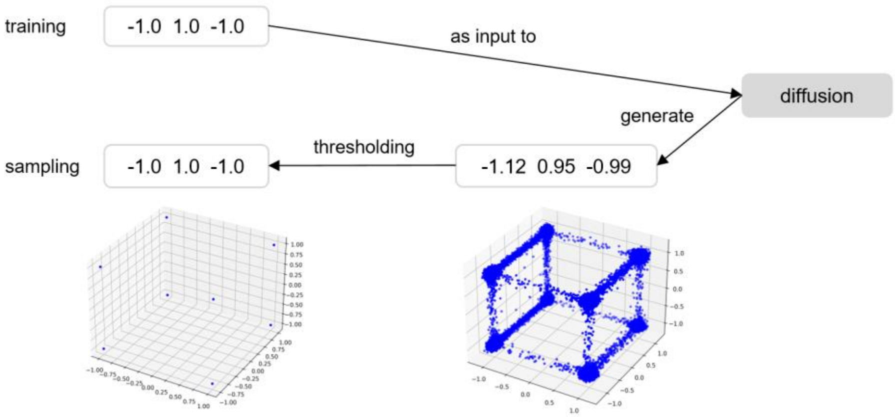


**Figure 4. Analog Bit Diffusion method : generate discrete data using continuous diffusion model.**

Because the diffusion model generates data by removing Gaussian noise from a normal distribution, diffusion can only generate continuous state diffusion model. So we use Analog Bit

Diffusion method (Figure 4) [23]. this approach can be directly modeled by continuous state diffusion model, without any other discrete space or re−formulation of the continuous diffusion process. At sampling process, the generated continuous data are decoded into discrete variables through a thresholding operation.

After learning the diffusion model with the observed data, we can augment the Ising data using this Analog Bit diffusion approach.

# Ⅲ. Results and discussion

## 3.1 Experiment detail

Now we demonstrate that diffusion model is capable of Ising data augmentation. After obtaining $h_i$, $J_{ij}$ from Ising data generated by the diffusion model, which demonstrated greater proximity to the true $h_i$, $J_{ij}$ than when inferred solely from observed data, we applied this approach to experimental recordings of neuronal activities. Subsequently, we conducted a missing data inference task.

We adopt an energy function of Ising model, $\mathrm{H}(\sigma) = -\sum_i h_i \sigma_i - \sum_{i<j} J_{ij}\sigma_i\sigma_j$, which has $\mathrm{L} = \mathrm{M} + \mathrm{M}(\mathrm{M}-1)/2$ parameters $\{h_i, J_{ij}\}$. For simulating data, we randomly set the parameter values from a Gaussian distribution with zero mean and some variance $\boldsymbol{\mathcal{N}}\left(\boldsymbol{0}, \frac{\boldsymbol{g^2}}{\boldsymbol{M}}\right)$, g is hyperparameter, that specifies bias and coupling strength, and define them as $h^{true}$, $J^{true}$. We then generate observed data $\hat{\sigma}$ from $\mathrm{p}(\hat{\sigma}|h^{true}, J^{true})$. The objective of Inverse Ising is to set parameters, creating a model distribution that best represents the distribution of the observed configuration.

In this experiment, we adopted the EM(Erasure Machine) as the algorithm for inferring parameters. Unlike other inverse Ising algorithms, the erasure machine requires only the observed ensemble without considering all unseen configurations. This characteristic makes the computation speed very fast, and the performance is also superior compared to other methods. To compare the accuracy between the inferred parameter $h_i^{obs}$, $J_{ij}^{obs}$

based on the observed configuration and the inferred $h_i^{aug}$, $J_{ij}^{aug}$ based on the configuration augmented by the diffusion model, we assess the concordance of each inferred $h_i^{obs\ or\ aug}$, $J_{ij}^{obs\ or\ aug}$ with the true parameter $h_i^{true}$,$J_{ij}^{true}$.

In conventional diffusion models, the architecture usually employs a U−net [15], but in this work, because ising data does not have locational characteristics, a multi−layer perceptron(MLP) with residual connections architecture was used as image data was not involved. Training and sampling were conducted using the DDPM approach [15](see Appendix A, B).

## 3.2 Early stopping to avoid overfitting

The diffusion model undergoes training with observed data $\hat{\sigma}$ as input, infer the true distribution $p(\hat{h}^{true}, \hat{J}^{true}|\hat{\sigma})$. As iterations progress, the diffusion model better captures the distribution, and it can generate high−quality samples that reflect the true distribution inferred from $\hat{\sigma}$.

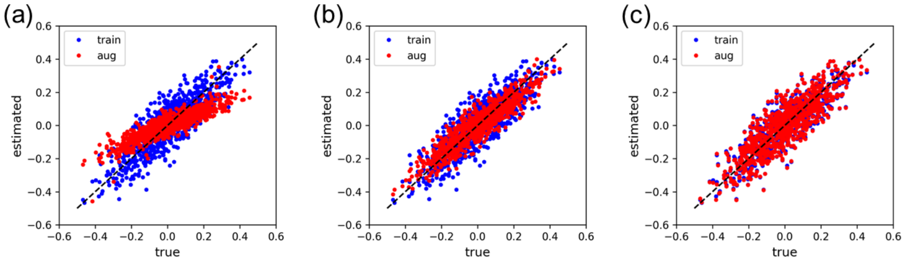


**Figure 5. Inference performance of augmented data. Inferred interactions vs actual interactions graph across iterations. M=40 system, 2,000ea observed samples, 100,000 augmented data. (a) 1,000 iterations. (b) 16,000 iterations. (c) 1,000,000 iterations.**

In the early stage of training, the diffusion model fails to sufficiently capture the distribution, resulting in the generation of random samples. Therefore, when plotting the estimated vs true interaction graph (Figure 5(a)), the dots form a skewed pattern because random samples imply weak interaction. However, at a certain point, the model can generate samples that better represent the true distribution than the observed data $\hat{\sigma}$ (Figure 5(b)).

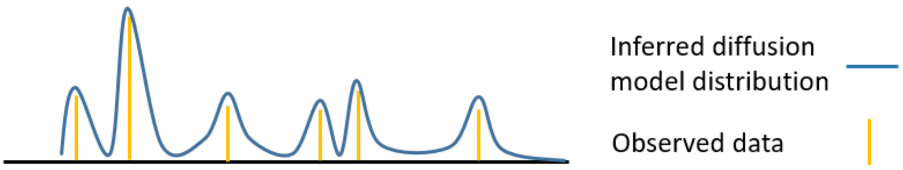


Figure 6. Overfitting to observed data

If too many iterations are given, the diffusion model may overfit to data (Figure 6), generating samples that just mimic the observed samples ($\hat{\sigma}$) rather than being close to the true distribution (Figure 5(c)). Therefore, it is necessary to employ a method to halt the training at an appropriate iteration to prevent overfitting in the diffusion model.

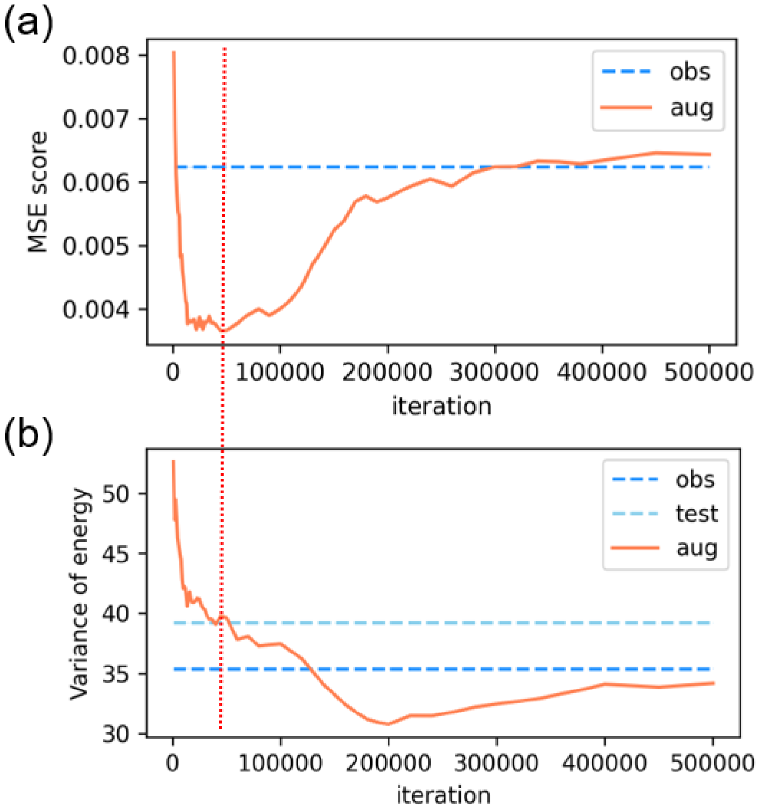


Figure 7. (a) Inferring performance across iterations (M=40, 4,000ea observed data). (b) Variance of energy $\langle E^2\rangle - \langle E\rangle^2$. ($\mathrm{E}(\sigma^{aug}|h^{obs}, J^{obs})$, Orange line), ($\mathrm{E}(\sigma^{obs}|h^{obs}, J^{obs})$, dashed blue line), ($\mathrm{E}(\sigma^{test}|h^{obs}, J^{obs})$, dashed skyblue line).

To quantify inference performance, Mean Squared Error (MSE) was used to compare the inferred parameters obtained from observed data ($h^{obs}$, $J^{obs}$) and augmented data ($h^{aug}$, $J^{aug}$) with the true T parameters ($h^{true}$, $J^{true}$).

$$\mathrm{MSE} = \frac{1}{M}\left(\sum_i\left(h_i^{true} - h_i^{obs\ or\ aug}\right)^2 + \sum_{i<j}\left(J_{ij}^{true} - J_{ij}^{obs\ or\ aug}\right)^2\right) \quad (23)$$

The MSE score decreases as the iteration progresses, but at some point, it starts to rise (Figure 7(a)). This is because the diffusion model initially finds the optimal distribution that can be inferred from the observed data, but as the training continues, it begins to overfit to the observed data.

MSE score is a metric that can be applied only when we know the true parameters. In real-world problems, since we do not know the true parameters, we need other indicators to determine when to stop the training of diffusion model.

The reason for the high MSE in the early stage of training is that the diffusion model has not learned well, resulting in the generation of random samples. Generating random samples implies a high variance in the generated samples, and as training progresses, the variance decreases. We represent this variance using Energy term $\langle E^2\rangle - \langle E\rangle^2$ , $\mathrm{E}(\sigma|h^{obs}, J^{obs}) = -\sum_i h_i^{obs}\sigma_i - \sum_{i<j} J_{ij}^{obs}\sigma_i\sigma_j$ (Figure 7(b)). In statistical physics it represents thermodynamics, specific heat $C_v = \frac{(\langle E^2\rangle - \langle E\rangle^2)}{MT^2}$. In this case $\mathrm{M}, \mathrm{T} = 1$. The variance of energy is initially high during the early iterations and gradually decrease as the iteration progresses. If the variance of $\mathrm{E}(\sigma^{aug}|h^{obs}, J^{obs})$ is below than that of $\mathrm{E}(\sigma^{obs}|h^{obs}, J^{obs})$, it indicates overfitting, and giving too many iterations would lead to convergence to the observed line as in MSE. When create a test set with the same number of observed data and plotting the variance of test data $\mathrm{E}(\sigma^{test}|h^{obs}, J^{obs})$ line on the graph, there exist a point where it intersects with variance of $\mathrm{E}(\sigma^{aug}|h^{obs}, J^{obs})$. This point serves as an appropriate early

stopping point where overfitting is avoided and the variance matches that of test data sampled from the true distribution.

In this study, we experimented by setting the point at which the variance of $\mathrm{E}(\sigma^{aug}|h^{obs}, J^{obs})$ intersects with that of $\mathrm{E}(\sigma^{test}|h^{obs}, J^{obs})$ as stopping point when variance stabilizes and $\mathrm{E}(\sigma^{aug}|h^{obs}, J^{obs})$ does not decrease below $\mathrm{E}(\sigma^{obs}|h^{obs}, J^{obs})$.

## 3.3 Quality of augmented data

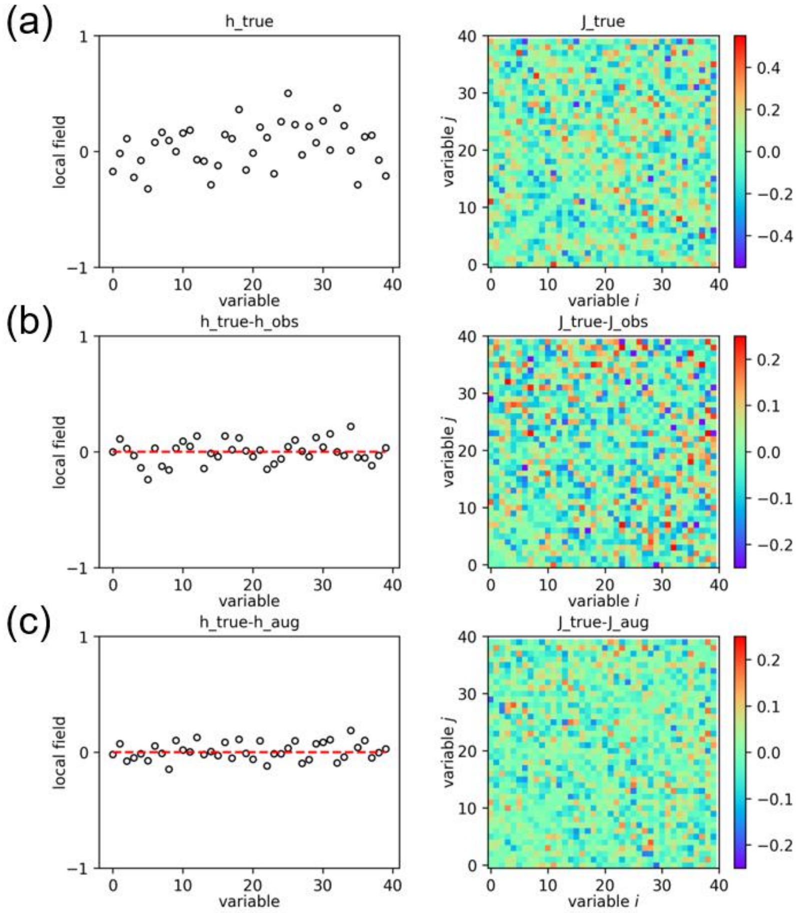


Figure 8. Quality of inference. $\mathbf{M} = \mathbf{40}$ system, 4,000 observed samples. Note that the true parameter $\{\boldsymbol{h}_i^{true}, \boldsymbol{J}_{ij}^{true}\}$ are sampled from a normal distribution $\boldsymbol{\mathcal{N}}\left(\mathbf{0}, \frac{g^2}{M}\right)$. We use g=0.2 for $\mathbf{h}$, g=1 for $\mathbf{J}$. (a) left true local field $\boldsymbol{h}^{true}$, right true pairwise interaction $\boldsymbol{J}^{true}$. (b) inference performance comparison between $\{\boldsymbol{h}_i^{true}, \boldsymbol{J}_{ij}^{true}\}$ and $\{\boldsymbol{h}_i^{obs}, \boldsymbol{J}_{ij}^{obs}\}$. Left $\boldsymbol{h}_i^{true} - \boldsymbol{h}_i^{obs}$, right $\boldsymbol{J}_{ij}^{true} - \boldsymbol{J}_{ij}^{obs}$ matrix. (c) comparison between $\{\boldsymbol{h}_i^{true}, \boldsymbol{J}_{ij}^{true}\}$ and $\{\boldsymbol{h}_i^{aug}, \boldsymbol{J}_{ij}^{aug}\}$. $\{\boldsymbol{h}_i^{aug}, \boldsymbol{J}_{ij}^{aug}\}$ is only use augmented data.

Now, We are evaluating the performance of augmented data. For a M = 40 system, we trained the diffusion model with 4,000ea observed samples, and then generating 100,000ea augmented data. To compare the performance of parameters inferred from observed data and data augmented by the diffusion model, we visually depicted the difference between true parameters and inferred parameters (Figure 8(b)-(c)). Using data purely generated by diffusion model allows for a more accurate inference of parameters.

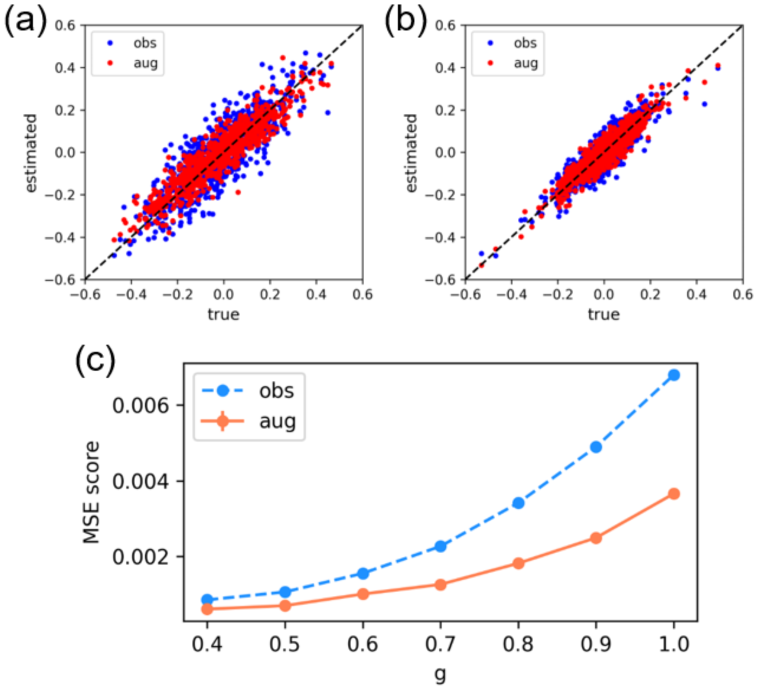


Figure 9. Inferred parameters vs actual parameters for strong and weak interactions. True interaction weights $\boldsymbol{h_i^{true}}, \boldsymbol{J_{ij}^{true}}$ are sampled from a normal distribution $\boldsymbol{\mathcal{N}\left(0, \frac{g^2}{M}\right)}$, $\mathbf{M = 40}$. To generate strong and weak interactions, we use g=1.0 (a), g=0.6 (b) respectively. Inference performance of only use 4,000ea observed data (obs, filled blue circles), only use 100,000ea augmented data (aug, filled red circles). (c) Inferring performance

**based on the parameter g. (obs. Filled blue circles), (aug, filled orange circles).**

Diffusion model even can augment Ising data in both weak and strong interaction weight regime (Figure 9). When inferring using only observed samples for both strong interaction (Figure 9(a)) and weak interaction (Figure 9(b)), Regardless of the hyperparameter **g** value that affects the strength of interactions, augmented data consistently improves performance across all levels of interaction strength(Figure 9(c)). it can be seen that using augmented data leads to better inference compared to relying solely on observed samples in both strong, weak interaction system.

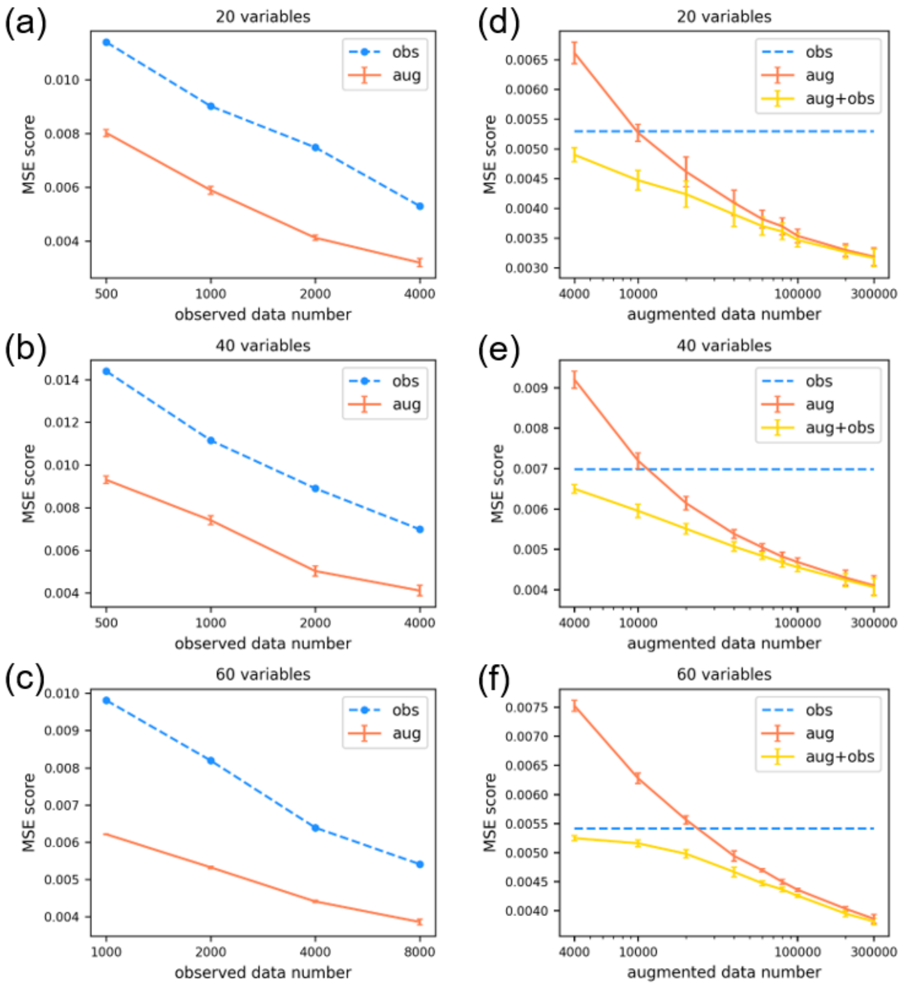


Figure 10. Inferring performance based on the number of observed data and augmented data. (a−c) Inference performance based on the number of observed data. After training the diffusion model with observed data, we augmented data (300,000ea) using DDPM method. All true parameters followed a normal distribution $\boldsymbol{\mathcal{N}\left(0,\frac{g^2}{M}\right)}$, with g=1. (a) $\mathbf{M = 20}$. (b) $\mathbf{M = 40}$. (c) $\mathbf{M = 60}$. (d−f) Inferring performance based on the number of augmented data diffusion model learned by 4,000ea(d−e), 8,000ea(f) observed data. (d) 20 variables. (e) 40 variables. (f) 60 variables.

The performance of data generated by the diffusion model varies with the number of observed and augmented data. To evaluate the inference performance based on the quantity of observed data, we examined the trend using the MSE score between true parameters and inferred parameters (Figure 10). Regardless of the number of observed data, data generated by diffusion models shows better performance (Figure 10 (a)-(c)).

Regarding number of augmented data, the accuracy of inference improves with an increasing number of augmented data (Figure 10 (d)-(f)). Utilizing not only augmented data but also observed data in the inference process enhances overall performance.

### 3.4 neuronal network imputation

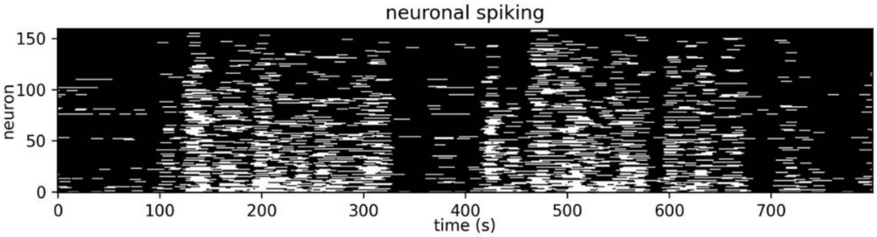


Figure 11. Activities of tiger salamander retina 160 neurons

We applied augmented data to investigate whether it enhances inference in real-world problems. We conducted a task to recover missing values in data using neuronal network derived from temporal neuronal activities in the tiger salamander (Ambystoma tigrinum) retina [24]. The dataset consists of neuronal spike trains from 160 neurons stimulated by a film clip of fish swimming, with a bin size of 20 ms (Figure 11).

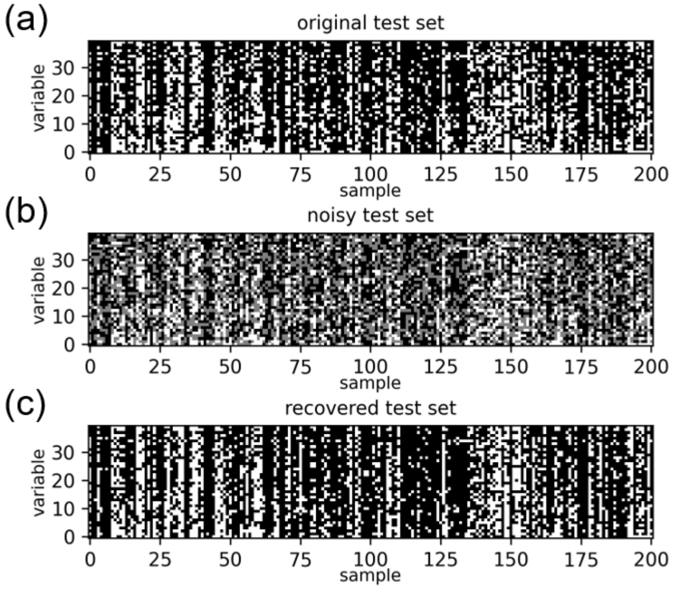


Figure 12. 40 most active neurons activities without considering time sequence. (a) original configuration of some test samples with black dots representing −1 value and white dots representing +1 value. (b) Noisy test samples by randomly missing 14 variables from original test sample (gray dots). (c) Recovered test samples using the inference of the erasure machine.

We considered only the 40 most active neurons. Since the temporal sequence is irrelevant in this model, we randomly arranged the data without considering the order (Figure 12(a)). Initially, we divided the data into 8,000ea training samples and 1,000ea test samples. We applied the EM to the training samples to infer $\{h_i^{obs}, J_{ij}^{obs}\}$. Since we do not know the true parameters $\{h_i^{true}, J_{ij}^{true}\}$, the inferred local fields and interactions cannot be directly compared. Instead, we evaluated the accuracy of the inferred parameters through missing values prediction on test samples. In the test samples, we randomly selected 14 variables

(35% of the total 40 variables) and set them as missing variables ($\sigma_i = 0$) (Figure 12(b)). To recover the missing values, we divided all variables $\sigma = (\sigma_m, \sigma_m^c)$ into missing variables $\sigma_m$ and observed variables $\sigma_m^c$. Next, we reconstructed the missing variables by fining $\sigma_m$ that maximizes $p(\sigma_m|\sigma_m^c) \propto p(\sigma_m, \sigma_m^c) \equiv p(\sigma)$ (Figure 12(c)). We measured the accuracy of matching the missing variables for evaluating reconstruction performance and compared the accuracy obtained from observed data with that from augmented data.

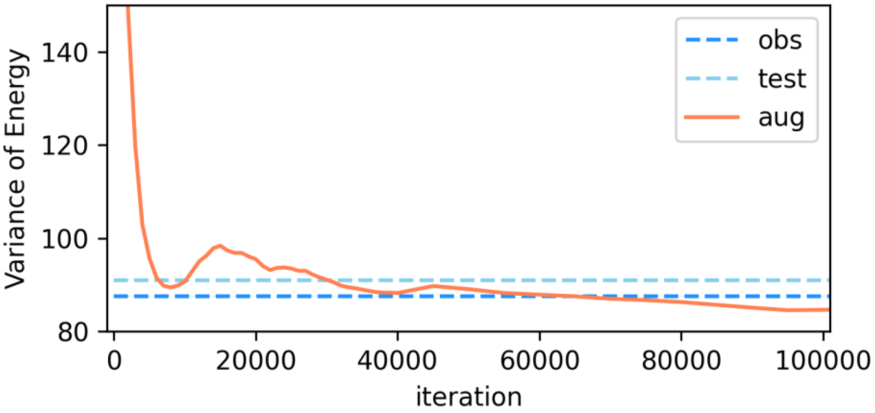


Figure 13. variance of energy of augmented data across iteration

Even when using real-world neuronal data, we confirmed that the variance of energy decreases over iterations (Figure 13). We identified the point where the variance of energy $E(\sigma^{aug}|h^{obs}, J^{obs})$ intersect with $E(\sigma^{test}|h^{obs}, J^{obs})$ as stopping point for training the diffusion model.

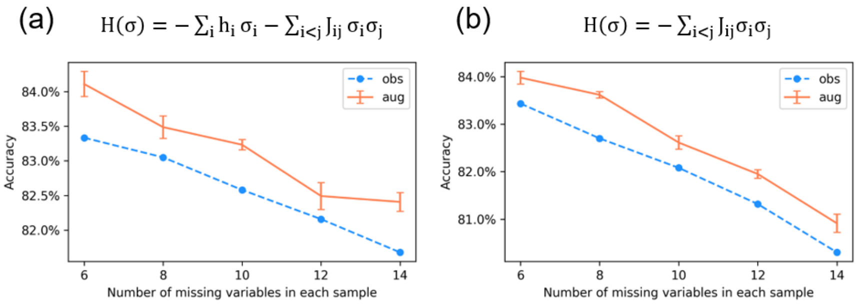


**Figure 14. Inferring missing data with observed and augmented data . (a) Recovery accuracy for varying numbers of missing variables. (b) same task for the Hamiltonian without a local field term $\mathbf{H}(\boldsymbol{\sigma}) = -\sum_{i<j} \boldsymbol{J_{ij}\sigma_i\sigma_j}$.**

Augmented data improved missing values prediction accuracy regardless of the number of missing data (Figure 14(a)). It could be that it performs well because a lot of information is contained in the local term, so we experimented using only the interaction term without local term $\mathrm{H}(\sigma) = -\sum_{i<j} J_{ij}\sigma_i\sigma_j$ (Figure 14(b)). Even when considering only the interaction term, it exhibits augmentation performance similar to when a local field is present. It represents that the interaction term contains the key information.

# Ⅳ. Conclusion

In a low−data setting, increasing the amount of data to improve model performance is a crucial task in the era of big data. In this study, we explored the potential of the first Ising data augmentation. We confirmed the possibility of generating binary Ising data without additional structural changes using continuous state diffusion with an Analog Bit diffusion approach.

Diffusion model learns the interaction weight, and since the weight is embedded inside the model, it is not possible to know directly what the value is, but it can generate an infinite number of samples that fit the weight. As the number of augmented data increase, a more accurate true weight value can be found. In this study, we were able to use a very large number of data generated by diffusion model using an efficient method called Erasure Machine.

In our experiment, fully connected MLP(Multi−Layer Perceptron) was used as the architecture of the diffusion model (Appendix B). This places a lot of load on learning and is not an efficient method. Nevertheless, the experimental results were surprising. Just as the CNN(Convolutional Neural Network) [25] Structure is effective for image data, performance will improve even further if the optimal architecture for generating Ising data is found. Additionally, In recent time, this powerful generative model, diffusion model, has been extended to time series applications[26]. Data augmentation using diffusion model can be applied to kinetic Ising problem. Deriving time series augmented Ising data is an intriguing avenue for future work.

Diffusion model's ability to augment Ising data demonstrates the effectiveness of the score−based approach, allowing them to bypass the computation of intractable normalization terms. This study not only introduce a new paradigm for Ising data augmentation but also provides insights into the broader applicability of diffusion models in data science and physics−inspired machine learning.

# Reference

[1] Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., & Hwa, T. (2009), Identification of direct residue contacts in protein-protein interaction by message passing, Proceedings of the National Academy of Sciences of the United States of America 106(1) 67-72. https://doi.org/10.1073/pnas.0805923106

[2] Cocco, S., Feinauer, C., Figliuzzi, M., Monasson, R., & Weigt, M. (2018), Inverse statistical physics of protein sequences: a key issues review, Reports on progress in physics. Physical Society (Great Britain), 81(3) 032601. https://doi.org/10.1088/1361-6633/aa9965

[3] Zhang, G., Stillinger, F. H., & Torquato, S. (2013), Probing the limitations of isotropic pair potentials to produce ground-state structural extremes via inverse statistical mechanics, Physical review. E, Statistical, nonlinear, and soft matter physics 88(4) 042309. https://doi.org/10.1103/PhysRevE.88.042309

[4] Zhao. Longfeng, Bao. Weiqi, Li. Wei (2018), The stock market learned as Ising model, Journal of Physics: Conference Series 1113(1) 012009, https://dx.doi.org/10.1088/1742-6596/1113/1/012009

[5] Thierry Mora, Aleksandra M. Walczak, William Bialek, Curtis G. Callan (2010), Maximum entropy models for antibody diversity, Proceedings of the National Academy of Sciences

107(12) 5405-5410. https://www.pnas.org/doi/abs/10.1073/pnas.1001705107

[6] Nathan Eagle, Alex (sandy) Pentland, David Lazer (2009), Inferring friendship network structure by using mobile phone data, Proceedings of the National Academy of Sciences 106(36) 15274-15278 https://www.pnas.org/doi/abs/10.1073/pnas.0900272106

[7] Tanaka, Toshiyuki (1998), Mean-field theory of Boltzmann machine learning, Phys. Rev. E. 58(2) 2302-2310, https://link.aps.org/doi/10.1103/PhysRevE.58.2302

[8] Federico Ricci-Tersenghi (2012), The Bethe approximation for solving the inverse Ising problem: a comparison with other inference methods, Journal of Statistical Mechanics: Theory and Experiment 08 P0815 https://dx.doi.org/10.1088/17425458/2012/08/P08015

[9] Wu, Dian and Wang, Lei and Zhang, Pan (2019), Solving Statistical Mechanics Using Variational Autoregressive Networks, Phys. Rev. Lett. 122(8) 080602 https://link.aps.org/doi/10.1103/PhysRevLett.122.080602

[10] Jo, Junghyo and Hoang, Danh-Tai and Periwal, Vipul (2020), Erasure machine: Inverse Ising inference from reweighting of observation frequencies, Phys. Rev. E 101(3) 032107 https://link.aps.org/doi/10.1103/PhysRevE.101.032107

[11] Shorten, C., Khoshgoftaar, T.M. (2019), A survey on Image Data Augmentation for Deep Learning, *J Big Data* 6, 60 https://doi.org/10.1186/ s40537-019-0197-0

[12] Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021), Text Data Augmentation for Deep Learning, *Journal of big data 8*(1) 101 https:// doi.org/10.1186/s40537-021-00492-0

[13] Wen, Qingsong, et al. (2021), Time series data augmentation for deep learning: A survey, International Joint Conferences on Artificial Intelligence Organization 21(8) 4653-4660 https://doi.org/10.24963/ijcai.2021/631

[14] Sohl-Dickstein, et al. (2015), Deep Unsupervised Learning using Nonequilibrium Thermodynamics , Proceedings of the 32nd International Conference on Machine Learning 2256-2265 https://proceedings.mlr.press/v37/sohl-dickstein15.html

[15] Ho, Jonathan and Jain, Ajay and Abbeel, Pieter (2020), Denoising diffusion probabilistic models, Proceedings of the International Conference on Neural Information Processing Systems 574(12) https://api.semanticscholar.org/CorpusID:219955663

[16] Song, Yang and Ermon, Stefano (2020), Improved techniques for training score-based generative models, Proceedings of the 34th International Conference on Neural Information Processing Systems 1043(11) https://api,semanticscholar.org/CorpusID:219708245

[17] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, Bryan Catanzaro (2020), 2021 International Conference on Learning Representations, https://doi.org/10.48550/arXiv.2009.09761

[18] Jacob Austin, Daniel Dun-ning Woo Johnson, Jonathan Ho, Danny Tarlow, Rianne van den Berg (2021), Structured denoising diffusion models in discrete state-spaces, 2021

Advances in Neural Information Processing System https://arxiv.org/abs/ 2107.03006

[19] Tashiro, Yusuke, Song Jiaming, Song Yang, Ermon, Stefano (2021), CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation, Advances in Neural Information Processing System 2021 24804-24816, https://proceedings.neurips.cc/paper_files/paper/2021/file/cfe8504bda37b575c70ee1a8276f3486-Paper.pdf

[20] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, Jianzhu Ma (2023), 3D Equivariant Diffusion for target-Aware Molecule Generation and Affinity Prediction, The International Conference on Learning Representations 2023, https://doi.org/10.48550/arXiv.2303.03543

[21] Ian Goodfellow, Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherji, Courville Aaron, Bengio Yoshua (2014), Generative adversarial networks, Proceedings of the 27th Inthernational Conference on Neural Information processing Systems 2 2672-2680 https://doi.org/10.48550/arXiv.1406.2661

[22] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, Sungwoong Kim (2019), Fast AutoAugment, Proceedings of the 33rd International Conference on Neural Information Processing Systems 598 6665-6675 https://doi.org/10.48550/arXiv.1905.00397

[23] Ting Chen, Ruixiang Zhang, Geoffrey Hinton (2023), Analog Bits: Generating Discrete Data Using Diffusion Models with Self-Conditioning, International Conference on Learning

Representations 2023, https://doi.org/10.48550/arXiv.2208.04202

[24] Marre O., Tkacik G., Amodei D., Schneidman .E., Bialek W., Berry M.(2017), Multi-electrode array recording from salamander retinal ganglion cells. Institute of Science and Technology Austria. https://doi.org/10.15479/AT:ISTA:61

[25] Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E (2012), ImageNet Classification with Deep Convolutional neural Networks, Advances in Neural Information Processing System 012, https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[26] Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, Junbin Gao (2023), Diffusion Models for Time Series Applications: A Survey, arXiv 2305.00624 https://doi.org/10.48550/arXiv.2305.00624

# Appendix

## <A.1> diffusion model (Deep Unsupervised Learning None-quilibrium Thermodynamics, sohl-Dickstein et al., ICML 2015)

The diffusion process where $\sigma_0$ loses existing information and transforms into noise can be expressed using the langevin equation.

$$\sigma_1 = \sqrt{\alpha_1}\sigma_0 + \sqrt{1-\alpha_1}\epsilon_1 \tag{24}$$

$\alpha_1$ is a parameter that determines the speed of diffusion. If $\alpha_1 = 1$, it corresponds to a case where no diffusion occurs. When $\alpha_1 = 0$, $\sigma_0$ becomes noise following a standard normal distribution $\epsilon_1 \sim \mathcal{N}(0, I)$.

When we further advance the diffusion process to generalize, it can be expressed as follows:

$$\sigma_t = \sqrt{\alpha_t}\sigma_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \tag{25}$$

$\alpha_t$ is also be described as $\beta_t = 1 - \alpha_t$, which is parameter determining the intensity of noise over time, referred to as the noise schedule. The results of this Langevin equation can be expressed through the following conditional probability

$$\mathrm{q}(\sigma_t|\sigma_{t-1},\sigma_0) \propto \exp\left[-\frac{(\sigma_t - \sqrt{\alpha_t}\sigma_{t-1})^2}{2(1-\alpha_t)}\right] \tag{26}$$

In this conditional probability, we can observe that the forward diffusion process is a Markov process dependent only on the previous time state, $\sigma_{t-1}$. This diffusion dynamics exhibit an interesting property

$$\begin{aligned}\sigma_2 &= \sqrt{\alpha_2}\sigma_1 + \sqrt{1-\alpha_2}\epsilon_2 = \sqrt{\alpha_1\alpha_2}\sigma_0 + \sqrt{(1-\alpha_1)\alpha_2}\epsilon_1 + \sqrt{1-\alpha_2}\epsilon_2 \\ &= \sqrt{\alpha_1\alpha_2}\sigma_0 + \sqrt{1-\alpha_1\alpha_2}\bar{\epsilon}_2 \qquad (27)\end{aligned}$$

In the final expression $\bar{\epsilon}_2$ is defined as linear combination of $\epsilon_1$ and $\epsilon_2$, which follow a standard normal distribution, resulting in a new standard normal distribution $\bar{\epsilon}_2$. Generalize this expression we can obtain new Langevin equation :

$$\sigma_t = \sqrt{\bar{\alpha}_t}\sigma_0 + \sqrt{1-\bar{\alpha}_t}\bar{\epsilon}_t \qquad (28)$$

Here $\bar{\alpha}_t = \alpha_1\alpha_2 \dots \alpha_t$

Also we can express through the following conditional probability:

$$\mathrm{q}(\sigma_t|\sigma_0) \propto \exp\left[-\frac{(\sigma_t - \sqrt{\bar{\alpha}_t}\sigma_0)^2}{2(1-\bar{\alpha}_t)}\right] \qquad (29)$$

The inverse process of the forward diffusion defined above can be expressed using Bayes' theorem as the following conditional probability:

$$\mathrm{q}(\sigma_{t-1}|\sigma_t, \sigma_0) = \frac{q(\sigma_t|\sigma_{t-1}, \sigma_0)q(\sigma_{t-1}|\sigma_0)}{q(\sigma_t|\sigma_0)} \qquad (30)$$

As the three probability on the right side of the equation are all normal distributions, multiplying them together results in a normal distribution.

$$\mathrm{q}(\sigma_{t-1}|\sigma_t,\sigma_0) = \mathcal{N}(\mu_q(\sigma_t,\sigma_0,\mathrm{t}),\sigma_q^2(\mathrm{t})) \tag{31}$$

$$\mu_q(\sigma_t,\sigma_0,\mathrm{t}) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}\sigma_t + \frac{(1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}}{(1-\bar{\alpha}_t)}\sigma_0 \tag{32}$$

$$\sigma_q^2(\mathrm{t}) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)} \tag{33}$$

To determine the values of variables obtained through the inverse process at any arbitrary $\sigma_T$, not just $\sigma_T$ obtained from starting at $\sigma_0$, we envision a model that describes the inverse process. We assume that the forward diffusion process $\sigma_{t-1} \rightarrow \sigma_t$ occurs over a very short period, and we also assume that its inverse process is nearly in thermodynamic equilibrium. Therefore, it can be considered as a Markov process following a normal distribution. We hope that this inverse process model, denoted as $\mathrm{p}(\sigma_{t-1}|\sigma_t)$, approximately matches the inferred $\mathrm{q}(\sigma_{t-1}|\sigma_t,\sigma_0)$ from the data. To simplify the problem, we assume that the variance of the model, $\sigma_p^2(\mathrm{t})$, depends only on the noise schedule, similar to $\sigma_q^2(\mathrm{t})$.

Now, let's define the likelihood of the samples $\sigma_0$ generated through the inverse process model $\mathrm{p}(\sigma_{t-1}|\sigma_t)$. Finding the parameters that can maximize this objective function is precisely what constitutes "training".

log−likelihood :

$$\sum_{\sigma_0} \log p(\sigma_0) = \int d\sigma_0 q(\sigma_0) \log p(\sigma_0) \tag{34}$$

$$\begin{aligned}
\mathrm{p}(\sigma_0) &= \int d\sigma_1 \cdots d\sigma_T p(\sigma_0, \sigma_1, \cdots, \sigma_T) \\
&= \int d\sigma_1 \cdots d\sigma_T p(\sigma_T) \prod_{t=1}^{T} p(\sigma_{t-1}|\sigma_t) \frac{q(\sigma_1, \cdots, \sigma_T|\sigma_0)}{\prod_{t=1}^{T} q(\sigma_T|\sigma_{t-1})} \\
&= \int d\sigma_1 \cdots d\sigma_T q(\sigma_1, \cdots, \sigma_T|\sigma_0) p(\sigma_T) \prod_{t=1}^{T} \frac{q(\sigma_{t-1}|\sigma_t)}{q(\sigma_t|\sigma_{t-1})} \qquad (35)
\end{aligned}$$

Inserting the expressed $\mathrm{p}(\sigma_0)$ into the log-likelihood and applying Jensen's inequality, let's expand it as follow:

$$\begin{aligned}
&\int d\sigma_0 q(\sigma_0) \log p(\sigma_0) = \\
&\int d\sigma_0 q(\sigma_0) \log \int d\sigma_1 \cdots d\sigma_T q(\ \sigma_1, \cdots, \sigma_T|\sigma_0) p(\sigma_T) \prod_{t=1}^{T} \frac{p(\sigma_{t-1}|\sigma_t)}{q(\sigma_t|\sigma_{t-1})} \\
&\geq \int d\sigma_0 d\sigma_1 \cdots d\sigma_T q(\sigma_0, \sigma_1, \cdots, \sigma_T) \left[\log p(\sigma_T) + \sum_{t=1}^{T} \log \frac{p(\sigma_{t-1}|\sigma_t)}{q(\sigma_t|\sigma_{t-1})}\right] \\
&= \int d\sigma_T p(\sigma_T) \log p(\sigma_T) + \\
&\ \sum_{t=1}^{T} \int d\sigma_0 d\sigma_1 \cdots d\sigma_T q(\sigma_0, \sigma_1, \cdots, \sigma_T) \log \frac{p(\sigma_{t-1}|\sigma_t)}{q(\sigma_{t-1}|\sigma_t)} \frac{q(\sigma_{t-1}|\sigma_0)}{q(\sigma_t|\sigma_0)} \\
&= \int d\sigma_T p(\sigma_T) \log p(\sigma_T) - \sum_{t=1}^{T} L_{t-1} - \\
&\ \int d\sigma_0 d\sigma_T q(\sigma_0, \sigma_T) \log p(\sigma_T|\sigma_0) \qquad (36)
\end{aligned}$$

Since $\sigma_T$ is designed to follow a standard normal distribution, the crucial factor here becomes the second term.

$$L_{t-1} = \int d\sigma_0 d\sigma_T q(\sigma_0, \sigma_t) D_{KL}[q(\sigma_{t-1}|\sigma_t, \sigma_0) \parallel p(\sigma_{t-1}|\sigma_t)] \qquad (37)$$

Calculating the Kullback-Leibler divergence between these two normal distributions yields the following result:

$$D_{KL}[q(\sigma_{t-1}|\sigma_t, \sigma_0) \parallel p(\sigma_{t-1}|\sigma_t)] = \frac{1}{2\sigma_q^2(t)} \parallel \mu_p - \mu_q \parallel^2 \qquad (38)$$

By training in the direction of reducing this objective function and obtaining $p(\sigma_{t-1}|\sigma_t)$, we can now generate samples $\sigma_0$ through $\sigma_T \rightarrow \sigma_{T-1} \rightarrow \cdots \rightarrow \sigma_1 \rightarrow \sigma_0$ process. In practice, sampling is done through the Langevin equation following the conditional probability. Here, $\epsilon \sim \mathcal{N}(0, I)$ plays a role in increasing the diversity of samples generated, as it is a sample generated from noise following a standard normal distribution.

### <A.2> DDPM (Denoising Diffusion Probabilistic Models, Ho et al., NeurIPS 2020)

Eq. (25) that connecting $\sigma_0$ to $\sigma_t$ in the original diffusion model can be expressed as follows :

$$\sigma_0 = \frac{\sigma_t - \sqrt{1-\bar{\alpha}_t}\bar{\epsilon}_t}{\sqrt{\bar{\alpha}_t}} \tag{39}$$

Then, Eq. (32) is also newly expressed as follow:

$$\mu_q(\sigma_t, \sigma_0, \mathrm{t}) = \frac{1}{\sqrt{\alpha_t}}\sigma_t + \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\bar{\epsilon}_t \tag{40}$$

By using this, we can design the $\mu_p(\sigma_t, \mathrm{t})$ of $p(\sigma_{t-1}|\sigma_t)$ as follow :

$$\mu_p(\sigma_t, \mathrm{t}) = \frac{1}{\sqrt{\alpha_t}}\sigma_t + \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha_t}}\sqrt{\alpha_t}}\hat{\epsilon}(\sigma_t) \tag{41}$$

Using this expression, rewriting the Kullback−Leibler divergence, Eq. (38)

$$D_{KL}[q(\sigma_{t-1}|\sigma_t,\sigma_0) \parallel p(\sigma_{t-1}|\sigma_t)] = \frac{1}{2\sigma_q^2(t)}\frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \parallel \bar{\epsilon}_t - \hat{\epsilon}(\sigma_t) \parallel^2 \qquad (42)$$

Therefore, the overall objective function for the entire time is as follows:

$$\begin{aligned} \mathrm{L} = \sum_{t=1}^{T} L_{t-1} &= \sum_{t=1}^{T} \int d\sigma_0 d\sigma_t q(\sigma_0,\sigma_t) \frac{1}{2\sigma_q^2(t)}\frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \parallel \bar{\epsilon}_t - \hat{\epsilon}(\sigma_t) \parallel^2 \\ &= \sum_{t=1}^{T} \gamma_t \int d\sigma_0 d\sigma_t q(\sigma_0,\sigma_t) \parallel \bar{\epsilon}_t - \hat{\epsilon}(\sigma_t) \parallel^2 \end{aligned}$$

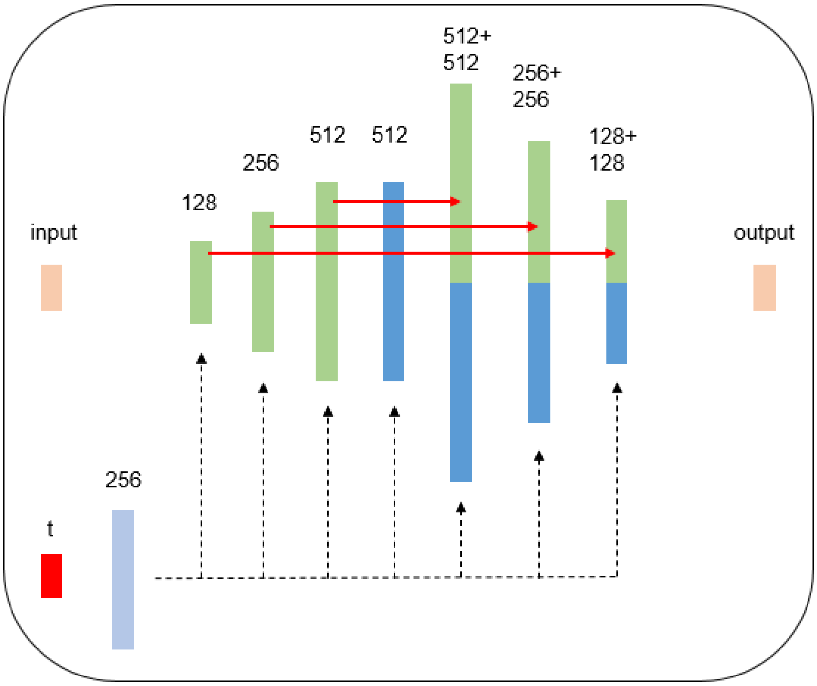
(43)

Ho et al. [15] simplified above objective function as follows:

$$L_{simple} = \mathbb{E}_{t,\sigma_0,\bar{\epsilon}_t} \parallel \bar{\epsilon}_t - \hat{\epsilon}(\sigma_t) \parallel^2 \qquad (44)$$

$L_{simple}$ can be interpreted as simplifying the $\gamma_t$, which represents the weight for each time, to 1. In reality $\gamma_t$ is larger as t approaches 0. However, setting all of them to1 reduces the weight for smaller values of t, effectively downplaying the importance of samples with more noise at larger t. This can be interpreted as giving more weight to removing noise in samples with significant noise at larger t. In DDPM, the diffusion model evolved by predicting the noise $\bar{\epsilon}_t$ and removing it, instead of predicting the original $\sigma_0$.

## <B> Diffusion model architecture code



```
# activation function

nonlinearity = nn.GELU()


class SinusoidalPosEmb(nn.Module):
    def __init__(self, dim):
        super().__init__()
        self.dim = dim

    def forward(self, x):
        device = x.device
        half_dim = self.dim // 2
        emb = math.log(10000) / (half_dim - 1)
        emb = torch.exp(torch.arange(half_dim, device=device) * -emb)
        emb = x[:, None] * emb[None, :]
        emb = torch.cat((emb.sin(), emb.cos()), dim=-1)
        return emb

class ResNetBlock(nn.Module):
    def __init__(self, in_hidden, out_hidden, temb):
        super().__init__()
        self.in_hidden = in_hidden
        self.out_hidden = out_hidden

        # timestep embedding
        self.temb_proj = nn.Linear(temb, out_hidden)

        # mlp layer
        self.mlp1 = nn.Linear(in_hidden, out_hidden)
        self.mlp2 = nn.Linear(out_hidden, out_hidden)


        if self.in_hidden != self.out_hidden:
            self.conv_shortcut = nn.Linear(in_hidden, out_hidden)

    def forward(self, x, temb):
        h = x
        h = nonlinearity(h)
        h = self.mlp1(h)

        h = h + self.temb_proj(nonlinearity(temb))

        h = nonlinearity(h)
        h = self.mlp2(h)

        if self.in_hidden != self.out_hidden:
            x = self.conv_shortcut(x)

        return x + h

class MLP(nn.Module):
    def __init__(self, n_var,n_dim=128, n_hidden1=256,n_hidden2=512):
        super().__init__()

        self.t_emb = nn.Sequential(
            SinusoidalPosEmb(n_steps),
            nn.Linear(n_steps, n_hidden1),
            nn.GELU(),
            nn.Linear(n_hidden1, n_hidden1),
        )
        self.mlp_in = nn.Linear(n_var, n_dim)
        self.downs = nn.ModuleList([
            ResNetBlock(in_hidden=n_dim, out_hidden=n_hidden1, temb=n_hidden1),
            ResNetBlock(in_hidden=n_hidden1, out_hidden=n_hidden2, temb=n_hidden1),
            ResNetBlock(in_hidden=n_hidden2, out_hidden=n_hidden2, temb=n_hidden1),
        ])

        self.mid = ResNetBlock(in_hidden=n_hidden2, out_hidden=n_hidden2,
temb=n_hidden1)

        self.ups = nn.ModuleList([
            ResNetBlock(in_hidden=2*n_hidden2, out_hidden=n_hidden2, temb=n_hidden1),
            ResNetBlock(in_hidden=2*n_hidden2, out_hidden=n_hidden1, temb=n_hidden1),
            ResNetBlock(in_hidden=2*n_hidden1, out_hidden=n_dim, temb=n_hidden1),
        ])


        self.mlp_out = nn.Linear(2*n_dim,n_var)

    def forward(self, x, time):
        # timestep embedding
        t = self.t_emb(time)
        x = self.mlp_in(x)
        r = x.clone()

        h = []
        for block_idx, block in enumerate(self.downs):
            x = block(x, t)
            h.append(x)

        x = self.mid(x, t)


        for block_idx, block in enumerate(self.ups):
            x = torch.cat([x, h.pop()], dim=1)
            x = block(x, t)

        x = torch.cat([x, r], dim=1)

        return self.mlp_out(x)
```

# 국문초록

이 논문에서는 Diffusion 모델로 ising data 를 증강하여 Inverse Ising problem 성능 향상 가능성에 대해 연구하였다. Diffusion model 은 Boltzmann machine 과는 다르게 data distribution 의 score function 을 학습하기 때문에 계산량이 많은 normalization term 을 계산하지 않고도 data 의 분포를 학습할 수 있고 이를 통해 Ising data augmentation 이 가능하다. 우리는 Analog Bit 방식의 Diffusion model 에 의해 생성된 샘플이 Ising 모델 매개변수 추정의 정확도를 향상시키는 것을 확인하였고 다양한 시스템 크기와 주어진 데이터와 증강된 데이터 양이 성능에 미치는 영향을 논하였다. 또한 neuronal data 로 missing value imputation task 에서도 성능 향상을 보여 현실 세계의 문제에도 적용 가능함을 확인했다. 본 연구는 Ising data augmentation 의 가능성을 확인하였으며, Diffusion 모델의 물리 기반의 기계학습에서의 더 넓은 적용 가능성에 대한 통찰을 제시한다.

**주요어** : Diffusion Model, Inverse Ising problem, Data Augmentation

**학 번** : 2022-27415