PB-DDPM: A Pontoon Bridge Denoising Diffusion Probabilistic Model for Missing Data Reconstruction

Peng Zhang[1,2], *Zhenjiang Zhang*[1,2*], *Yang Zhang*[1,2]

[1] *Department of Electronic and Information Engineering, Beijing Jiaotong University, China*
[2] *Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, China*
zhangpeng11@bjtu.edu.cn, zhjzhang1@bjtu.edu.cn, zhang.yang@bjtu.edu.cn

Abstract

The passive and active data missing in data acquisition reduces data availability and brings difficulties in subsequent data processing. To solve this problem, a data reconstruction method based on pontoon bridge diffusion model has been proposed. This method transformed the missing position into a 0-1 mask tensor. By element-wise multiplication, missing values were replaced by the Gaussian noise to construct new data for model inputting according to the mask. And then, the missing data were reconstructed under the condition of observed data. Compared to other methods that divide the original data into missing and observed parts while the data dimensions remaining unchanged to make up inputs, this approach reduced the dimension of input data and simplified the correlation between inputs. Meanwhile, using the original parameters of noise strategy as the anchors and adjusting the uncertainly coefficients by in-situ replacement, the pontoon bridge mode reduced the difference between distributions of practical reconstruction and theoretical calculation to enhance the relevance of training process and generating process. Furthermore, the input data of the training process is augmented with various masks to simulate data missing modes. Experimental results demonstrated that data augmentation techniques enhanced the model's ability to handle missing data, and the pontoon bridge diffusion model could effectively improve the quality of reconstructed missing data.

Keywords: Diffusion model, Missing data reconstruction, Noise schedule adjustment, Targeted training

1 Introduction

Data reconstruction, to reconstruct original data or its approximate value from processed data, is one of the primary domains of researches in information field [1]. It has played important roles in almost every stage of data processing, especially when transmitting data or encrypting data. Data can be varied while spreading among entities for the sake of security or efficiency, or even corrupted due to human factors and non-human factors. For instance, removing information that is insignificant in following processes before sending conserves time and resources at the price of impossible 100% recovery. Conversely, adding redundant information to overlay raw data protects privacy-sensitive information taking the risk of misunderstanding. According to the match-up between reconstructed data and processed data, data reconstruction contains four categories called spatial reconstruction, temporal reconstruction, precision reconstruction and characteristic reconstruction [2], corresponding to difference in spatial structure, temporal dimension, degree of precision and characteristic index. During the past decades, researchers focused on imputing the missing data to reconstruct data. In a sense, image super resolution is a kind of missing data processing, in which missing data or lost details are imputed based on the remaining data.

In the information age, everything can be described with data points consisting of several separate values in order. Almost inevitably, data missing will occur for the reason of data discretization and environmental disturbance. According to the relationship between observed values and the probability of missing data, missing data are classified into three categories, Missing Completely At Random (MCAR), Missing At Random (MAR) and Missing Not At Random (MNAR) [3]. In this paper, we distinguish missing mechanisms from the perspective of missing values' positions. Missing completely at random refers to underlying mechanism that missing values' positions generated by surrounding non-human factors which are unrelated to other measured variables and the missing values themselves. It is purely random and can only be simulated approximately. When it comes to data missing not at random, missing values' positions are chosen based on pre-designed strategies to meet special requirements, for which the human factor is decisive. The missing is organized and the corresponding value can be calculated based on exact calculation. Besides, lacking of precision can be treated as a special case of the generalized MNAR, and improving precision is harder to achieve comparing with fitting approximate data from individual perspective. Lastly, data missing at random when missing values' positions are in connection with other measured variables only, and is not related to the missing variables while others have been already observed. Missing data reconstruction will obtain new values at those missing positions that is inverse to the law of entropy

*Corresponding Author: Zhenjiang Zhang; E-mail: zhjzhang1@bjtu.edu.cn

DOI: https://doi.org/10.70003/160792642025012601008

increase, which means extra information is required. The source of those extra information can be the surrounding data, part of the entire data, the entire data, attributes of the object, and additional data empirically. The stronger the correlativity between missing data and extra information provided, the more the obtained values approximate to the ground-truth in theory.

Diffusion probabilistic models (DPM), the shoulders of this research, are a kind of generative models proposed in recent years, which have been widely used in large-scale model applications [4]. Originally, this model was formed to model probability distributions flexibly but tractable to train [5]. Researchers subsequently adjusted the model to produce new samples based on the obtained distributions and its powerful generalization and creation ability have been demonstrated in many domains. One crude explanation is that raw data are mapped to a real-valued random vector of the same size step by step, and each data point is corresponding to distribution throughout the whole probability space with central value (mean) and variance respectively, combining into multivariate normal distribution under ideal conditions. Conversely, once determined value is assigned to the random vector, reverse process will produce sample that is similar to one or more raw data points to a certain extent according to probability values of respective distributions at that determined value. Thus, the model can generate new samples different from all data points in the raw dataset, meanwhile it is hard to replicate the training data points. Furthermore, the implementation of those processes is stepwise, which can share the burden and ensures a smooth transition between dataset and the random vector.

In this paper, we focus on the situation that missing values' position can be simulated by covering with pseudo random mask or fixed mask, which is basically belonged to MCAR or MNAR while mask using to simulate MAR is conditional and much more complex. Values are generated from random by method based on DPMs. DPMs are supposed to be complex and hard to train in practice. But two genius ideas, series to parallel and parameters sharing, increase the utility and feasibility of training. However, the piecewise training strategy cause foreseen obstacle that connections between segments suffer additional loss of log-likelihood for the potential biases. To handle this, we propose a model adding 'sealing gaskets' at the connection points called Pontoon Bridge Denoising Diffusion Probabilistic Model (PB-DDPM). We show that our model has better performance on data reconstruction and get lower value with the same loss function. In addition, we discuss about the problem whether anchored points is needed and finally recognize its necessity. Moreover, we summarize the schedules used to establish the sequence of variances. Furthermore, we discovered that results rely on the conditions when concatenating conditional entries with raw data as the input when reconstructing missing data.

2 Related Work

Missing data was first mentioned while conducting questionnaire surveys and investigator found the existence of non-response to part of questions [6]. Since the main purpose was to collect the samples from which one can make inferences or extrapolations to the population, the primary approach was re-designing the survey or modifying the weight of samples based on the Probability Theory [7]. Afterward, researchers found that filling value calculated by special methods in the positions of missing data got comparable results in the following data analysis [8]. Thus, 'filling in' became one of the main approaches to handle missing data, and can involve either empirical strategy, where missing values are calculated using exact mathematical functions, or estimation, in which model parameters are estimated before generating the missing values [9]. As shown in Table 1, the basic features of representative methods have been listed, and these methods are introduced below.

Table 1. Summary of related works

Method	Type	Training	Characteristics
Rubin [10]	Fixed value	No	Low percentage
Bashir [11]	Regression	Little	Fixed position
Ghahramani [12]	Expectation Maximization	Little	Global optimum
Li [15]	CNN-based	Much	Feature reasoning
Zhang [16]	LSTM-based	Much	Sequence-to -sequence
Kingma [18]	AE-based	Much	Variational lower bound
Sønderby [19]	AE-based	Much	Hierarchical representation
Rezende [20]	Flows-based	Much	Invertible transformations
Ho [21]	DDPM	Much	Simplified objective
Song [22]	DDPM-based	Much	Deterministic generation
Nichol [23]	DDPM-based	Much	Hybrid objective
Kingma [24]	DDPM-based	Much	Learnable noise schedule

The empirical strategies contain using fixed values, such as zero impute, mean impute, median impute, and mode impute, or finding values utilizing special methods, for example, previous value impute, subsequent value impute, imputing with the average of previous value and subsequent value, and most closely resembled point based algorithms [10]. Whereas estimation methods are complex and including training process, for instance, regression model [11] to compute a hyperplane that minimizes the sum of squared differences between the true data and the hyperplane, and expectation maximization model [12] to find maximum likelihood parameters of a model that can be used to calculate the missing values.

With the development of neural network and deep learning, there are more ways to make full use of entire dataset including the incomplete data, which display more implicit correlations between variables to identify the missing values [13]. Meanwhile, the positional encoding of input embedding makes it tolerant to the absence of missing values and can handle complex and irregular missing mechanism [14]. Basic components of deep learning methods include the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Li et al. [15] inpainted images by utilizing a shared feature reasoning module to infer intermediate results generated by method based on multi-layer CNN in the latent space recurrently. And Zhang et al. [16] proposed a sequence to sequence imputation model based on Long Short Term Memory (LSTM) network with a variable-length sliding window algorithm implemented to enrich the potential correlations of time series data.

In those methods, missing positions are padded with zero value which reduces impact of missing data. Furthermore, planned noise has been added to either the input or the intermediate result of training process to improve generalization performance. As an illustration, de-noising auto-encoder [17] is trained to reconstruct inputs from partially destroyed data while variational auto-encoder [18] attempts to reconstruct each input from a series of intermediate points generated from the latent variable. When it comes to generative model, the initial values of generating process are sampled from random noise. Similarly, data reconstruction methods based on generative model adopt random noise samples to initialize the missing values.

As generative model, Variational Auto-Encoder (VAE) generates latent variables with the mean and variance calculated in the encoding process of auto-encoder, from which original data will be regenerated in the decoding process. Moreover, its objective is to maximize the Evidence Lower Bound (ELBO), which is a proxy for maximum likelihood model while minimizing the Kullback-Leibler (KL) divergence between the approximate posterior distribution and the true posterior distribution over latent variables together with the KL divergence between the same approximate posterior distribution and an identical prior distribution. However, it is hard to satisfy those two conditions simultaneously. Thus, some researchers [19-20] find a solution that stacking multiple encoding-decoding processes to sequentially generate multiple latent variables to approach the ideal one.

Whereas sequential generation costs more time and makes it complex to optimize model parameters, several models has been proposed to circumvent piecemeal training. Among which the DPM provides basic complete model and reserves space for self-design. This model regards the latent variables as scaled noisy data with the same dimension and the encoding processes are pre-defined as linear Gaussian models while the decoding processes is designed as a Markov chain. Thus, it is only the decoding processes needed to be trained and can be trained in parallel over different layers. Based on which, Denoising Diffusion Probabilistic Models (DDPM) [21] sets the variances of latent variables in reverse process to constants and trains the mean only, resembling denoising score matching with conditional parameter timestep t to deal with multiple noise scales. Moreover, Denoising Diffusion Implicit Models (DDIM) [22] generates competitive high quality samples within fewer timesteps by adjusting the proportion of stochasticity emanating from input noisy data in reverse process while training model with the same objective used to train DDPM. Besides, Nichol et al. [23] constructs a cosine noise schedule and adopts a weighted combination method to predict the actual variances of arbitrary timesteps to obtain better log-likelihoods called Improved Denoising Diffusion Probabilistic Models (IDDPM).

Furthermore, to obtain adequate default parameters of linear Gaussian models, Variational Diffusion Model (VDM) [24] utilizes a monotonic neural network with respect to the number of steps, of which the weights are restricted to be positive, to learn the noise schedule based on the definition of signal-to-noise ratio (SNR) in variance-preserving diffusion process.

3 Background

Before introducing the model adjusted with pontoon bridge strategy, we review the construct of DDPM and describe the reconstruction problem and its simulation.

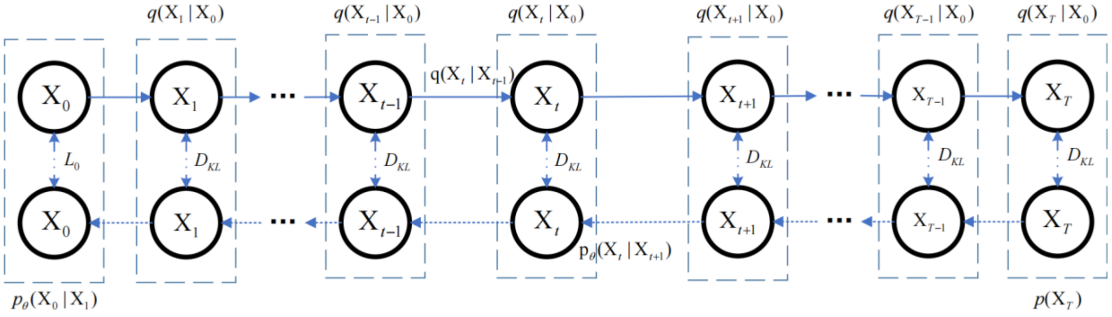


Figure 1. The generative process and forward process of Diffusion Model

3.1 Denoising Diffusion Probabilistic Model

Diffusion models establish a transition between the input data x_0 and the deepest latent variables x_T, which consist of one noise schedule, one bidirectional Markov chain and three processes over multi-steps. The noise schedule is a vector of the same length as diffusion steps T which controls the spacing of each step. Since the noise data involved in each step are sampled from independent identically normal distributions and the probability of output depends only on the input of the same step, the ideal distributions of each point in both forward process and reverse process can be calculated based on the properties of Markov chain. The structure of DDPM is shown in Figure 1.

In the forward process, signal power of input data gradually decreases while Gaussian noise power increasing according to pre-designed noise schedule $\alpha_t \in (0,1]$ (t=1,2,…,T) as shown in formula 1:

$$\begin{aligned} q\left(x_{1:T} \middle| x_0\right) &:= \prod_{t=1}^{T} q\left(x_t \middle| x_{t-1}, x_0\right) = \prod_{t=1}^{T} q\left(x_t \middle| x_{t-1}\right) \\ q\left(x_t \middle| x_{t-1}\right) &:= N\left(x_t; \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t) I\right) \end{aligned} \quad \textbf{(1)}$$

where x_1, x_2, …, x_T are latent variables, and $q(x_t|x_0)$ refers to the conditional distribution of corresponding variable whereas $q(x_T|x_0)$ converging to a standard Gaussian distribution for all x_0. For each step, the output data combines two pieces of information, the weighted input data with attenuation coefficient α_t and the additional Gaussian noise data with standard deviation $(1-\alpha_t)$. By Bayes rule, the reverse process conditioned on x_0 can be described as formula 2 and formula 3:

$$q\left(x_{t-1} \middle| x_t, x_0\right) := N\left(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I\right) \quad \textbf{(2)}$$

where,

$$\begin{aligned} \tilde{\mu}_t(x_t, x_0) &:= \frac{\sqrt{\tilde{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t \\ \tilde{\beta}_t &:= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}(1-\alpha_t) \end{aligned} \quad \textbf{(3)}$$

among which $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. The input and output of forward process are interchanged with each other to establish the input and output of reverse process. Moreover, the output data also consist of weighted input data and additional noise data. It is necessary to mention that each variable in those two processes is a random variable, thus those processes are theoretical and cannot be implemented with certain algorithm. The last process is the generative process that starts at $p_\theta(x_T)=N(x_T;0,I)$ to approximate the reverse process with trainable parameters θ, training to minimize the expectation of negative log likelihood on dataset which is described as formula 4:

$$\begin{aligned} & E_{q(x_0)}\left[-\log p\left(x_0\right)\right] \\ & \leq E_{q(x_0, x_1, \ldots, x_T)}\left[-\log p\left(x_T\right) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q\left(x_t \middle| x_{t-1}\right)}\right] \\ & = \underbrace{E_q(x_0)\left[D_{KL}\left(q\left(x_T \middle| x_0\right) \middle\| p_\theta(x_T)\right)\right]}_{L_T} \\ & + \sum_{t\geq 1} \underbrace{E_{q(x_0, x_t)}\left[D_{KL}\left(q\left(x_{t-1} \middle| x_t, x_0\right) \middle\| p_\theta(x_{t-1}|x_t)\right)\right]}_{L_{t-1}} \\ & - \underbrace{E_{q(x_0, x_1)}\left[\log p_\theta\left(x_0 \middle| x_1\right)\right]}_{L_0} \end{aligned} \quad \textbf{(4)}$$

This training objective has been simplified by DDPM under specify preconditions as shown in formula 5:

$$L_{simple}\left(\theta\right) := E_{t, x_0, \epsilon}\left[\left\| \epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right] \quad \textbf{(5)}$$

A sample x_0 can be generated from x_T by gradually computing $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t$ where $\eta \sim N(0,I)$ and parameters σ_t denote the degree of additional stochasticity at timestep t, distinguishing from the stochasticity inherited from x_t. Adjusting the proportion of those two items result in new samples. $\tilde{\beta}_t$ and $(1-\alpha_t)$ are empirical values of σ_t^2 , and the additional stochasticity is set to zero when t = 1.

3.2 Missing Data Reconstruction

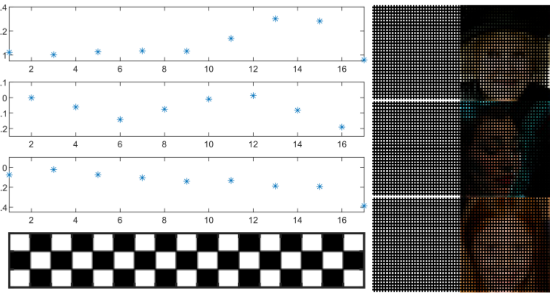
In this paper, we focus on the situation that the missing position M is already known. Thus, the missing data can be described as X_b^M and M, corresponding to the value and position respectively. In the case of data with two dimensions, the value at location i,j of M is 0 if the corresponding value in X_b^M is observed, otherwise the value is 1. The empty spaces in X_b^M have been labeled with Not a Number (NaN) to retain the structure during collection, which must be replaced with real value for further calculation.

The mathematical relation between the replaced data X_T^M and the output data X_0^M is denoted as f, that is to say, $X_0^M = f(X_T^M, M)$. There are various of ways to combine input X_b^M and M, such as concatenating on channel, processing in specified order or performing specific function. Specifically, we treat M as a mask to concentrate on the missing data in each step of diffusion model, and the initial values of missing data are sampled from standard normal distribution and finally converge to a neighborhood around the true values. Which means that observed data only need to be transformed under linear function while noise of different powers is added to the missing position according to the noise schedule. The object function is to reduce the difference between real data X and output data X_0^M that is equivalent to maximizing the probability of missing values, and can be written as formula 6.

$$
\begin{aligned}
L^{M}_{simple}(\theta) &:= E_{t,M,x_0,x_T^M,\epsilon}\left[\left\|\epsilon * M-\epsilon_\theta\right\|^2\right]\\
\epsilon_\theta &= g_\theta\left[(1-M)*x_T^M+M*\left(\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\epsilon\right),t\right]
\end{aligned}
\qquad \textbf{(6)}
$$

where $x_T^M = (1-M) * x_0 + M * \eta$, $\eta \sim N(0,I)$ and I denotes the all-ones matrix with the same size as M. g is a function different from f.

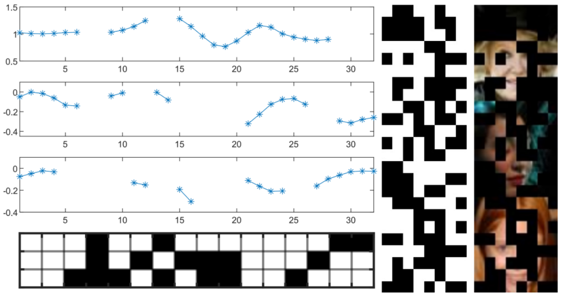
As shown in Figure 2 and Figure 3, two kind of masks, fixed missing position and random missing position, are adopted to simulate the common situations of missing mechanisms. The former is proactive caused by the sampling strategy while the latter is generated by pseudo-random sequence. Missing position is regularly under the situation of fixed missing position that is diversified in form and easy to implement, which can be regarded as special case of MNAR. It is the granularity of required data that affects the sampling strategy, manifesting itself in sampling interval of time series data or alternate acquisition of space data. Under the other situation, missing position is random and the missing proportion fluctuates around the expected value that affects the recovery of each sample. When it comes to model training, although some randomness has been introduced by data augmentation, the MNAR data reconstruction can be treated as regression tasks that the probability distribution at fixed missing position is inferred from surrounding known data. And it is more plausible to assume that the MCAR data reconstruction is a task to mine the intrinsic correlation, based on which the missing data are estimated according to global consistency.



(a) (b)

Figure 2. Examples of missing regularly

(The black and white grid indicates missing marks, and the black patch represents the location of missing data. (a) The upper part consists of three sequence data on the same timeline, corresponding to the three rows of the lower part. (b) The right part is image data of the same size as the left part.)



(a) (b)

Figure 3. Examples of missing completely at random

(The black and white grid indicates missing marks as in Figure 2.)

In order to simplify the problem, we introduce two parameters to describe the missing position, missing unit and missing rate. The former reflects the quantitative relation between minimum missing block and sampling resolution while the latter indicate the uniform proportion of missing position for each sample. For example, Figure 2(a) represents the regular missing with unit of one sampling point and rate of nearly 50% in time dimension. Figure 2(b) represents the alternation missing with unit of 1×1 and rate of nearly 50% in space dimension. Figure 3(a) represents the random missing with unit of two sampling points and rate of nearly 1/3 in time dimension while Figure 3(b) represents the same kind of missing with unit of 8×8 and rate of 1/2 in space dimension. Furthermore, it is obvious that the amount of data points of interest belonging to MNAR varies in a smaller range compared to MCAR while the values of those two parameters are changing.

4 Model

Diffusion model requires large number of steps to ensure the quality of generated samples at the cost of tremendous time and resources, which is confined to the general application. In this paper, we focus on the resources constrained scenario and restrict the amount of steps.

4.1 Additional Uncertainty in Generative Process

In reverse process of diffusion model, the conditional distribution $q(x_{t-1}|x_t, x_0)$ is a linear combination of x_t and x_0 together with an additional uncertainly item. Moreover, the lower and upper bounds on the coefficient of additional uncertainly are $(1-\alpha_t)(1-\bar{\alpha}_{t-1})/(1-\bar{\alpha}_t)$ and $(1-\alpha_t)$ respectively. Obviously, the difference between those bounds decreases with the increasing of α_t. Thus, DDPM chooses the lower bounds as the variances of conditional distribution and selects a large amount of steps to make α_t close to 1.

Furthermore, the training stage is independent to the generating stage under the condition of step-by-step instructions in generative process. During the training stage, the mean of input data is exact and the variance is fixed, model is trained to reveal the variable value added to the input data as much as possible. However, the mean of input data of current step depends on the output of last step, which leads to bias fluctuating around the value designed in training stage. The bias decreases with the increasing of generating step number, for the reason that the weight of underlying value increases and the underlying value x_0 itself takes shape gradually, resulting in better performance of denoising model.

Since the purpose is to minimize the KL-divergence between conditional distributions $q(x_{t-1}|x_t, x_0)$ of reverse process and conditional distributions $p_\theta(x_{t-1}|x_t)$ of generative process at each step, DDPM assumes that $p_\theta(x_{t-1}|x_t)$ are Gaussian distributions and fixes the variances to train the shared parameters θ to predict $\tilde{\mu}_t$ only. In consideration of the objective condition that the difference between predicted value and theoretical value always exists, the actual distributions in sequential generation stage are differ from those presupposed in training stage. In addition, lesser amount of steps has a negative effect on the validity of utilizing the lower bounds as the variances. Adjusting the values of variances can decrease the KL-divergence and give occasion to reasonable weights of the linear combination.

There are two potentially viable approaches to adjust those values based on the premise that variances are not directly trained in generative process. One approach is to introduce a vector to construct new variances by using exact calculating formula. As utilized in IDDPM, the new variances are the weighted logarithmic mean of the lower and upper bounds that converted back to the original scale. The other approach is to obtain new variances individually, which has been used to build the PB-DDPM.

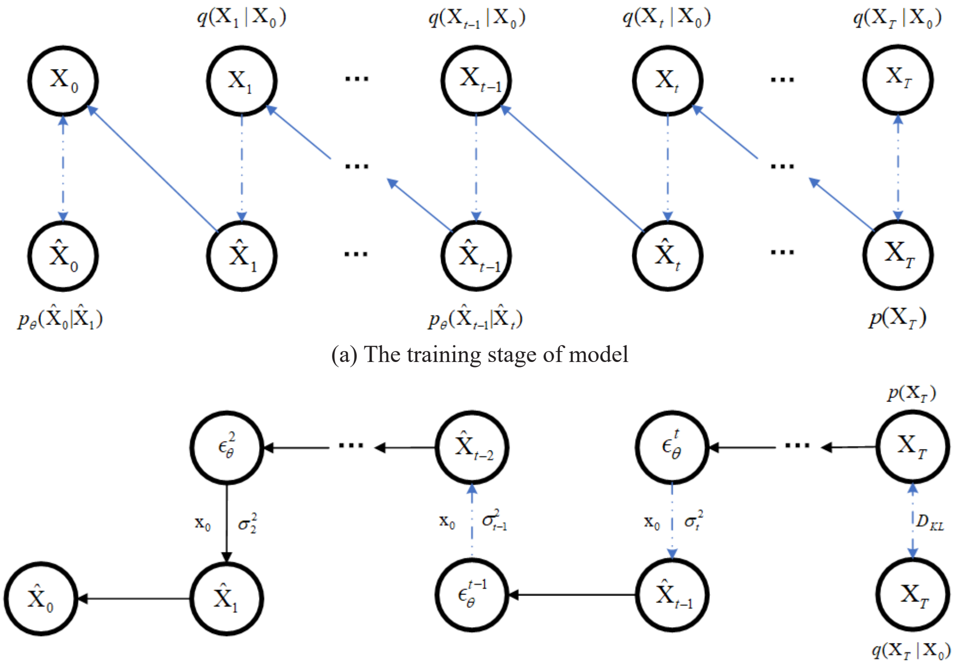
4.2 PB-DDPM

Although the domain of each conditional distribution covers the entire range of values, the accuracy of means and variances affects the quality of generated data. Making connection between the input of training stage and output of generative stage provides a feasible way for information exchange, which is beneficial to handle the local minimum problem. The DPMs bridge the gap between exact data x_0 and random variable x_T by dividing one big problem into several connected smaller problems. The difference between the information entropy of x_0 and x_T can be analogous to elevation inconsistency of ends of the bridge.

Thus, the presupposed parameters in reverse process are likened to piers in the middle, and interstices between beams upon piers have an impact on the smoothness of deck. The PB-DDPM treats the piers as anchors and retains the adjustability of connecting points. Besides, it is important to notice that anchors are indispensable which prevents the model from being washed out. Figure 4 shows the flow chart of PB-DDPM.

Two X_T denote the final output of forward process and the initial input of generative process separately, and the means and variances are fixed for both of them. $\hat{X}_t$ denote the revised input of each step. The solid blue line in training stage corresponds to the denoising procedure that need to be trained, and the blue dotted line represents the underlying restriction to the prospective distribution of each output. When it comes to the generative stage, the solid black line refers to the execution of trained model while the blue dotted line refers to calculation procedure used to connect adjacent steps. δ_t^2 denotes the presupposed variances and x_0 denotes the inferred complete data while ϵ_θ^t denotes the noise estimated by denoising procedure of step t. $\hat{\mu}_\theta(x_t,t)$ and $\hat{\Sigma}_\theta(x_t,t,\sigma_t^2) = \sigma_t^2 + f(\epsilon_\theta^t)$ denotes the mean and variance of conditional distribution $p_\theta(x_t|x_{t+1})$ in generative stage. $f(\epsilon_\theta^t)$ refers to fluctuation caused by training error of denoising procedure. The mean and variance of conditional distribution $\hat{p}_\theta(x_t|x_{t+1},x_0)$ in training stage are denoted as $\mu_\theta(x_t,t)$ and σ_t^2 . The means are equal in value for the reason that they are obtained through the same procedures. Thus, the KL-divergence between those two conditional distributions can be written as formula 7.

$$D_{KL}\left(p_\theta\left(x_t\,|x_{t+1}\right)\|\hat{p}_\theta\left(x_t\,|x_{t+1},x_0\right)\right)$$
$$=\frac{1}{2}\left[\log\frac{\left|\sigma_t^2+f(\epsilon_\theta^t)\right|}{\left|\sigma_t^2\right|}+tr\left(\frac{f(\epsilon_\theta^t)}{\sigma_t^2}\right)\right] \tag{7}$$



(a) The training stage of model

(b) The generative stage of model

Figure 4. The flow chart of Pontoon-Bridge-DDPM

$\hat{\Sigma}_\theta$ is a statistical magnitude whose value is determined by variable ϵ_θ^t, thus, $f(\epsilon_\theta^t)$ is a constant function and do not perform any arithmetic operation. Under the condition that σ_t^2 is fixed, we need to adjust $f(\epsilon_\theta^t)$ to decrease the KL-divergence. The value of the variable depends on the input and model of denoising procedure. Modifying the model structure introduces new parameters and increases the computational complexity, therefore, we choose to adjust the input x_t by updating the variance to turn this into reality, and denote the new variance as γ_t.

In training stage, the value of complete data x_0 is known from the beginning, and the means and variances of input and target at each step is certain. Thus, the presupposed conditional distribution of x_{t-1} and x_T on x_0 are referred to as $N(x_{t-1};\sqrt{\bar{\alpha}_{t-1}}x_0,(1-\bar{\alpha}_{t-1})I)$ and $N(x_t;\sqrt{\bar{\alpha}_t}x_0,\gamma_t I)$ respectively. ε_θ^t is the estimated value to the sample point ε^t acquired in reparameterization of x_t as formula 8.

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{\gamma_t}\varepsilon^t \tag{8}$$

Hence, the estimated value to the sample x_{t-1} can be calculated with formula 9 derived from formula 3.

$$\begin{aligned}\hat{x}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\left[x_t - \varepsilon_\theta^t \frac{\gamma_t - \alpha_t + \bar{\alpha}_t}{\sqrt{\gamma_t}}\right] + \hat{\sigma}_t\eta \\ \hat{\sigma}_t &= \frac{\left(\gamma_t - \alpha_t + \bar{\alpha}_t\right)\left(1-\bar{\alpha}_{t-1}\right)}{\gamma_t}\end{aligned} \tag{9}$$

where η is new sample point of standard normal deviate, $\hat{\sigma}_t$ is the revised coefficient of additional uncertainty and γ_t is initialized to $\left(1-\bar{\alpha}_t\right)$. γ is a new vector of length T and we estimate it by algorithm based on Iterative Method (IM) and Method of Moments (MoM).

The vector is a parameter of denoising model, and the algorithm is an iterative method to find optimal value of the vector to minimize the loss function of model. The algorithm alternates between performing two steps in each iteration, the training step which optimizes parameters of diffusion model while γ remains unchanged and the estimating step which finds (local) maximum likelihood estimates of γ once the loss of model has stabilized.

The loss reflects the denoising performance and its stabilization means that the gaussian component of inputs are recovered approximately. Based on which, the reconstructed input of next step in generative stage can be assumed to be a normal deviate, whose mean is directly proportional to x_0 while the variance γ_t is at variance with the default value. Thus, we attempt to estimate this parameter by maximizing a likelihood function to achieve that the reconstructed data is most probable.

$$\begin{aligned}\log L_t(\gamma_t) &= \sum_{i=1}^{N}\log\left[f_t\left(x_t^i,\gamma_t\right)\right] \\ &= -\frac{N}{2}\log(2\pi\gamma_t) - \frac{1}{2\gamma_t}\sum_{i=1}^{N}\left(x_t^i - \sqrt{\bar{\alpha}_t}x_0\right)^2\end{aligned} \tag{10}$$

As shown in formula 10, L_t denotes the likelihood function of reconstructed data x_t^i that contains N samples, and t ranges from 1 to $T-1$ for the reason that the parameters of x_T is fixed. It is noticeable that maximum likelihood estimation is equivalent to method of moments when the predefined probability density function $f_t(\cdot|\gamma_t)$ is corresponding to the family of normal distribution. Furthermore, there exists bias in the estimation of the population variance, and the estimated value calculated from samples should be corrected by the Bessel's correction that using $N-1$ to instead N in the formula. Since variance is the only parameter to be estimated for each $\hat{x}_t$, the mean of normal deviate x_t can be shifted to 0 by making a subtraction to avoid loss of significance in calculations, such that the estimator is solved by formula 11. V_n denotes the *nth*-order moment about the origin, $\mu(x_t)$ = $\sqrt{\bar{\alpha}_t}x_0$ is the presupposed mean of normal deviate x_t and N is the amount of samples to estimate variance. The estimated value becomes valid for large value of N at the cost of vast amount of calculation. Since the denoising procedure is trained to deal with series of noise data and the deviation of means are restricted, whether subtracting the square of first order moment about the origin or not does not affect the adjustment of variances.

$$\begin{cases} V_1(t) = E_{x_0,\varepsilon^{t+1},\eta}\left[\hat{x}_t - \mu(x_t)\right] \approx \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}(\widehat{x_t^i} - \sqrt{\bar{\alpha}_t}x_0) \\ V_2(t) = E_{x_0,\varepsilon^{t+1},\eta}\left[\hat{x}_t - \mu(x_t)\right]^2 \approx \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}(\widehat{x_t^i} - \sqrt{\bar{\alpha}_t}x_0)^2 \\ \hat{\gamma}_t = \dfrac{N}{N-1}\{V_2(t) - [V_1(t)]^2\} \end{cases} \tag{11}$$

The estimated parameter γ is supposed to converge to a smooth curve close to the original curve under the influence of the anchor points. Moreover, the values are mostly in the range of 0 to 1, even though them can be arbitrary positive number. Besides, a few values near to the beginning of generative process can be specially large as a result of outliers with small probability when it comes to high percentage missing. And the training process and the reconstructing process are shown in Algorithm 1.

As designed in DDPM, this model contains two stage, and the generative stage is called reconstructing stage since the purpose is to recover complete data from the missing. Besides, θ is trained by batch gradient descent while γ is iterative updated to a convergence result. Hence, we can revise formula 6 to formula 12.

Algorithm 1. Missing data reconstruction based on PB-DDPM

```
  Training stage:
Require: Missing position $M$; complete data $x_0$; step number
    $1 \le t \le T$; sample point of standard normal deviate $\eta$;
    the amount of samples $N$; pre-designed noise schedule $\overline{\alpha}_t$;
    presupposed variances $\sigma_t^2$;
Ensure: New parameters $\gamma_{t-1}$.
While Estimated parameters $\gamma_{t-1}$ do not stabilize do
    Initialize parameters; initialize $\gamma_{t-1}$ to $(1-\overline{\alpha}_t)$;
    Train denoising model;
    if Revise the parameters $\gamma_{t-1}$ then
        for $i = 1; i \le N; i++$ do
            Random sample $x_0^i$ and $t^i$;
            Calculate $x_t^i$ and obtain the output $\varepsilon_\theta^{t,i}$;
            Calculate $\hat{x}_{t-1}$ and $\mu(x_{t-1})$;
            Calculate the difference and its square;
        end for
        Calculate $\hat{\gamma}_{t-1}$ and $\hat{\sigma}_t^2$ to update $\gamma_{t-1}$ and $\sigma_t^2$;
    else if Do not revise the parameters $\gamma_{t-1}$ then
        Train denoising model;
    end if
end while
  Reconstructing stage:
Require: Obtain missing position $M$ and missing data $x_T^M$;
Ensure: Get the final reconstructed data $x_0^M$.
for $i = T; i \le N; i--$ do
    Obtain the output $\varepsilon_\theta^t$ and calculate the next input $x_{T-1}^M$;
end for
```

$$L_{simple}^{M}(\theta) := E_{t,M,x_0,x_T^M,\epsilon}\left[\left\|\epsilon * M - \epsilon_\theta^k\right\|^2\right] + E_t\left(\gamma_{t+1}^k - \gamma_{t+1}^{k-1}\right)^2 \qquad \textbf{(12)}$$
$$\epsilon_\theta^k = g_\theta\left[(1-M) * x_T^M + M * \left(\sqrt{\overline{\alpha}_t}x_0 + \sqrt{\gamma_{t+1}^k}\epsilon\right), t\right]$$

where γ_{t+1}^k denotes the th approximation of γ_{t+1} and ϵ_θ^k is the corresponding output of denoising model. Pontoon Bridge strategy is 'hot-swappable' with DDPM, and DDPM-based model can be translated into PB-DDPM after making changes on partial formulas.

4.3 Noise Schedules

With regard to the implementation of DPM-based model, noise schedule is a variable sequence and affects the performance of the entire model directly. The only constraint on the schedule is that the coefficient of uncertainty of ending point $(1-\overline{\alpha}_T)$ should approximate to 1 while the multiplier to complete data x_0 is approximate to 0. Thus, the coefficient of uncertainly adding to each step $(1-\alpha)$ is also a sequence and the values can be arbitrary positive numbers less than 1. There are three common methods to generate this sequence, the first method is that the values is inversely proportion to the amount of remaining steps which is mainly applied to the scenario that the original samples are made up of binary numbers. The second method is that the values increase linearly as designed in DDPM. The last method is that the values are generated based on trigonometric function, and the domain of which is the interval [0, $\pi/2$] as designed in IDDPM, resulting in two sequences α and $(1-\alpha)$ that are symmetrical to each other.

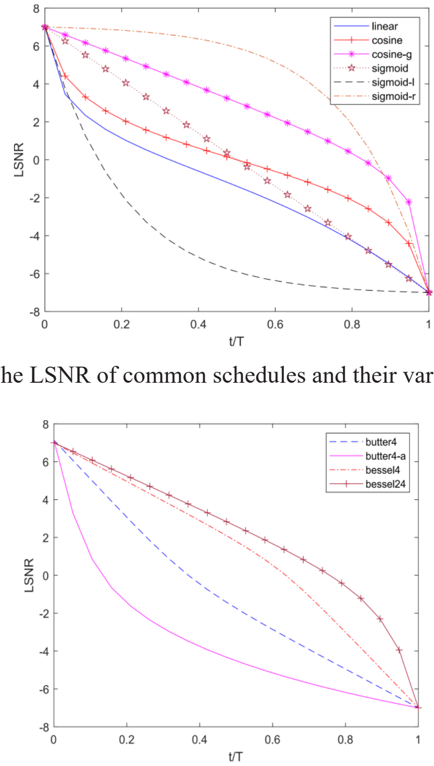
In general, $(1-\overline{\alpha}_t)$ is the parameter applied to numerical computation while $(1-\alpha_t)$ acting as a restriction, and arbitrary increasing sequence that starts from value 0 and ends with value 1 can be regarded as one kind of strategy. By scaling the step number to [0,1], the distance between adjacent points decrease with the increase of T, and the sequence turns into a monotonically increasing function satisfying that $f(0) = 0$ and $f(1) = 1$ in the limit T→∞ . Conversely, if we design a such function, then noise schedule can be generated by taking a series of input-output pairs within the function. Influenced by the idea put forward in VDM, we use the same factor that is the logarithm of the quotient of $\overline{\alpha}_t$ to $(1-\overline{\alpha}_t)$, written as LSNR:= ln $\overline{\alpha}_t/(1-\overline{\alpha}_t)$, to reflect the trend of different noise schedules. On account of the situation that $(1-\overline{\alpha}_t)$ approaches 0 makes the quotient get bigger quickly, which affects the numerical value recorded in computer system. Thus, there is a truncation exists in the actual parameter used in the model. In this paper, we limit the range of LSNR to [−7, 7], that is LSNR(1) = 7 and LSNR(T) = −7 , and the KL-divergence at step T written as $D_{KL}(q(x_T|x_0)\|N(0,I)) \approx 6*10^{-4}$ bits per dimension is a very small value under this assumption.

Suppose that LSNR is part of a linear function $L(z) = kz$ where $k < 0$ and $z \in (-\infty, +\infty)$. $\overline{\alpha}_t$ is defined as $m_z \in [0,1]$, then we have $\ln[m_z/(1-m_z)] = kz$, which is solved by $1-\overline{\alpha}_t = 1-m_z = 1/(1+e^{kz}) = \text{sigmoid}(-kz)$. If LSNR $\in [-7, +7]$, we can get that $z \in [7/k, -7/k]$, $x = t/T(1/14) * (kz+7)$. In particular, $(1-\overline{\alpha}_t)$ is the sigmoid function within a certain range when $k = -1$.

Suppose that LSNR is a convex function, the value of $\overline{\alpha}_t/(1-\overline{\alpha}_t)$ decreases quickly at the beginning and then decreases slowly when t is large. We adopt the frequency response of nth-order low-pass Butterworth filter $f(z)=1/\sqrt{1+z^{2*n}}$ to generate noise schedule $\overline{\alpha}_t$ for the reason that trend of $\overline{\alpha}_t$ is in accord with the physical meaning of low-pass filter. Besides, because of the inherent characteristic of $f(z)$, the geometric sequence sampled from the frequency response of different order low-pass Butterworth filters are the same when the range has been determined.

Suppose that LSNR is a concave function, in contrast to the trend of convex function, $\overline{\alpha}_t/(1-\overline{\alpha}_t)$ decreases slowly at the beginning and then decreases quickly when t is large. Accordingly, we adopt the frequency response of nth-order low-pass Bessel filter to generate noise schedule, such as the frequency response of 2nd-order low-pass Bessel filter $f(z)=1/\sqrt{1+z^2+z^4}$. Since the filter focuses on the properties of harmonic, we sample points by establishing arithmetic sequence within the logarithmic range of the independent variable.

Suppose that LSNR contains multiple stages, the descending speed is non-monotonic, which is influenced by both the trend of the source function and the sampling strategy. For example, noise schedule $\bar{\alpha}_t$ and $(1-\bar{\alpha}_t)$ generated by sampling from the trigonometric function based on arithmetic sequence lead to two kind of trends, that is, $\bar{\alpha}_t/(1-\bar{\alpha}_t)$ decreases quickly when t is small or large but decreases slowly in the middle.



(a) The LSNR of common schedules and their variations

(b) The LSNR corresponding to different filtering schedules

Figure 5. Diffusion strategies comparison

As shown in Figure 5, different schedules correspond to different curves. There are 6 schedules in Figure 5(a), 'linear' refers to linear increase $(1-\alpha_t)$ schedule mentioned before, 'cosine' refers to schedule sampled from trigonometric function based on arithmetic sequence, 'sigmoid' refers to schedule sampled from linear function based on arithmetic sequence when k = −1, and the others adopt different sampling modes. 'cosine-g' refers to schedule sampled from trigonometric function based on geometric sequence, 'sigmoid-l' refers to schedule sampled from linear function based on geometric sequence with common ratio less than 1 while 'sigmoid-r' refers to schedule sampled from linear function based on geometric sequence with common ratio greater than 1. In Figure 5(b), 'butter4' refers to schedule sampled from frequency response of 4th-order low-pass Butterworth filter based on geometric sequence, 'bessel4' refers to schedule sampled from frequency response of 4th-order low-pass Bessel filter based on geometric sequence, 'butter4-a' refers to schedule sampled from frequency response of 4th-order low-pass Bessel filter based on arithmetic sequence and 'bessel24' refers to schedule sampled from frequency response of 24th-order low-pass Bessel filter based on geometric sequence. Among which, the curve of 'sigmoid', 'butter4' and 'bessel4' are linear, convex and concave respectively, while the common schedules 'linear' and 'cosine' are multi-stage. Besides, revising the sampling strategy changes the trend of curve, and arbitrary function can be generated by combining multiple functions. Furthermore, cumulative distribution function of any distribution can be regarded as a feasible noise schedule, although there is no clear physical meaning.

5 Experiments

In this paper, we focus on the situation that resource is constrained and set total step T = 20 and maximum training epoch Maxepoch = 150 for all experiments. The experiments have been conducted on two datasets, called the HAC dataset [25] and the CelebA dataset [26]. The former records the data acquired by Inertial Measurement Unit (IMU) corresponding to 6 human activities performed by 30 volunteers. The capture rate is 50Hz and each point has 6 dimensions that are 3-axial linear acceleration and 3-axial angular velocity. Then we obtain samples with a sliding window of 128 points and 50% overlap. Besides, the samples are reshaped into 2×3×128 and scaled to [-1,1] before inputting it into the model. The latter is a common dataset of image processing that contains 202 599 face images and is resized into 64×64 and scaled to [-1,1] in the preprocessing stage. The samples of the HAC dataset consist of a series of time series data points, each of which is an observation of the target object and has a direct relationship with other points within the same sample to varying degrees. However, the samples of the CelebA dataset comprise foreground and background that are much more complex, and the size and position of the foreground which is of interest exhibit significant heterogeneity across samples.

The model is designed to reconstruct missing data, thus, the difference between complete data and reconstructed data must be displayed. With regard to models that have the same input and a unified objective function, the change of loss reflects the quality of the reconstructed data. Therefore, 5 indicators are selected to measure the performance of models, which are Mean Square Error (MSE) between complete data and reconstructed data, Classification Score (CS) obtained by predesigned classifier, Loss Result (LR) of the objective function, Peak Signal-to-Noise Ratio (PSNR) which is the ratio between the maximum possible power of a signal and the power of corrupting noise that indicates sensitivity to differences, and Structural SIMilarity (SSIM) which compares multiple features from different parts of the images to measure the similarity. The higher CS (or PSNR, SSIM) and lower MSE (or LR) corresponds to higher quality of reconstructed samples.

5.1 Effects of the Proposed Model

At first, it is the missing rate that need to be considered affects the quality of reconstructed images. Since DPM-based models generate samples similar to those in the training set, it is naturally endowed with the ability to reconstruct missing data, and the generative procedure is equivalent to processing 100% missing data. We score on both the HAC and the CelebA dataset evaluated on indicators mentioned before. The denoising model for ϵ_θ follows that in [21], which is a U-Net with skip-connection based on a revised ResNet revitalized by self-attention. The channel of each layer in the encoder of U-Net is increasing by a factor of 2 while it is decreasing in the decoder. The kernel size of CNN is modified to 3×4 or 3×6 for the HAC dataset.

Table 2. Results of handling different rate data missing with DDPM

Missing rate	HAC		
	MSE↓	CS↑	LR↓
10%	0.0729	0.9751	1.8686
20%	0.0751	0.9746	1.8686
50%	0.1123	0.8547	1.8686
80%	0.3876	0.6433	1.8686

Missing rate	CelebA			
	MSE ↓	LR ↓	PSNR ↑	SSIM↑
10%	0.1611	1.326	30.6944	0.9622
20%	0.1875	1.326	26.7207	0.9194
50%	0.3438	1.326	19.5692	0.7264
80%	0.7429	1.326	14.07	0.4394

As shown in Table 2, the indicators for the HAC dataset are MSE, CS and LR, among which the CS refers to the accuracy of multi-class classifier and LR is the product of total steps and unweighted average noise recovering loss, and the indicators for the CelebA dataset are MSE, LR, PSNR and SSIM. When it comes to different missing rates, for example, large-scale missing that lost 80% data, half-missing that lost 50% data and small-scale missing that lost 10% or 20% data, MSE is increasing with the missing rate for both dataset while CS is decreasing for the HAC and PSNR (or SSIM) is decreasing for the CelebA. The change is negligible when missing rate increases from 10% to 20%. Moreover, changes on MSE (or CS) between 50% and 80% are much more significant than that between 20% and 50%, and the difference between MSEs are more pronounced. However, changes on PSNR (or SSIM) between half-missing and large-scale missing is close to that between small-scale missing and half-missing, and PSNR changes a lot from 10% missing to 20% missing while change on SSIM is minor, for the reason that PSNR is inversely proportion to reconstruction loss and the numerator and the denominator of SSIM are of the same order as reconstruction loss. Since model does not change, LR remains consistent despite varying missing rates. Results of PB-DDPM are shown in Table 3.

The indicators to measure the performance of PB-DDPM is the same as those in Table 3, and the total training epoch remains 150. The first half of training stage is the same for the both models, and PB-DDPM adjusts noise schedule in the second half while DDPM serves as the control group, continuing with initial noise schedule.

Table 3. Results of handling different rate data missing with PB-DDPM

Missing rate	HAC		
	MSE↓	CS↑	LR ↓
10%	0.0801	0.9763	1.0327
20%	0.0823	0.9748	1.0327
50%	0.1256	0.8574	1.0327
80%	0.4075	0.6472	1.0327

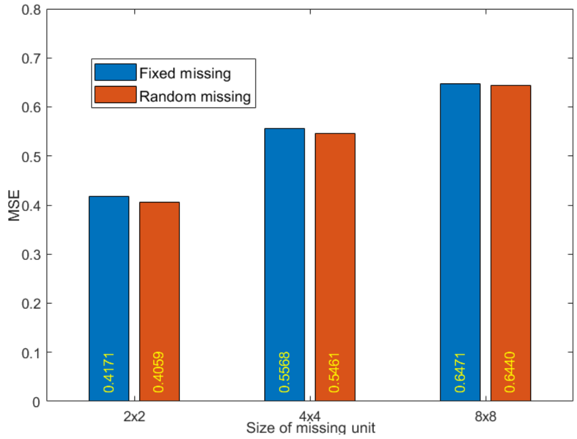
Missing rate	CelebA			
	MSE↓	LR↓	PSNR↑	SSIM↑
10%	0.2070	0.8178	30.8451	0.9680
20%	0.2265	0.8178	26.7463	0.9312
50%	0.3696	0.8178	19.6203	0.7544
80%	0.7216	0.8178	14.2905	0.4783

The overall trend of change in all indicators in Table 3 is roughly the same with those in Table 2. Moreover, the value of LR decreases after adjusting noise schedule, and there are slight increases in CS and SSIM. Besides, we have observed a slight reverse fluctuation in MSE. All components of samples in the dataset are treated equally when we calculate the MSE, it can gauge the reconstructed data's resemblance to the ground truth, despite its limitations in fully reflecting the availability and quality of the data. Since the model is utilized to reconstruct data identical to the complete data, reconstructed data is similar to a newly generated sample when the missing rate is large, which is of limited usability, and when the missing rate is small, data can be reconstructed with PB-DDPM exactly.

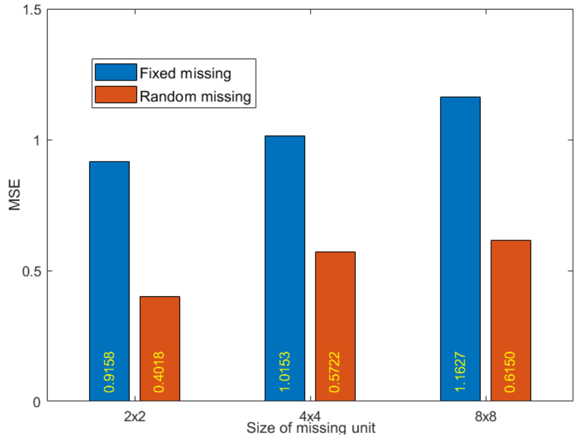
Targeted training conducted by applying data augmentation leads to significant improvements in the quality of reconstructed data, we employ various strategies that training the model with different kind of masks or different sizes of missing unit, and testing trained model with unmatched missing styles. Taking the CelebA dataset as an example, the performance under strategies are measured by MSE.

Figure 6 shows MSE for reconstructed data under situations that sizes of missing unit are 2×2, 4×4 and 8×8 respectively. Missing rate is fixed to large-scale missing to make the distinctions more prominent. Figure 6(a) corresponds to fixed missing 80% data with different sizes of missing unit while Figure 6(b) corresponds to the random missing. Besides, the blue rectangle refers to testing with fixed missing data and the red rectangle refers to testing with random missing data. In other words, model is trained with fixed missing data but tested with both fixed missing and random missing data in Figure 6(a) while model is trained with random missing data but tested with both fixed missing and random missing data in Figure 6(b). Comparing the value of MSE corresponding to blue rectangle in Figure 6(a) and value of MSE corresponding to red rectangle in Figure 6(b) with the value of MSE on the CelebA dataset that lost 80% data in Table 2 and Table 3, the ability to handle large-scale missing is enhanced after applying data augmentation. For different sizes of missing unit, MSE is increasing with the sizes, and so is the task difficulty for the reason that the correlation between missing data and the others is more complex.

Moreover, the values of MSE are close to each other for different kind of masks with the same size of missing units in Figure 6(a), the results of random missing are even better than the fixed missing. Conversely, noticeable differences exist between those values in Figure 6(b), and the results of random missing are much better than the fixed missing. Furthermore, to deal with random missing, model trained with random missing data achieves better performance, and training model with fixed missing data acquires more favorable results when tested with fixed missing data. Thus, it is confirmed that reconstructing data missing at fixed positions are more difficult than data missing at random.



(a) Data augmentation with fixed missing



(b) Data augmentation with random missing

Figure 6. The evaluation result of different data augmentation strategies

Figure 7 shows MSE for reconstructed data under situations that missing ratios are 30%, 50% and 80%. The model is trained with 80% random missing data that the size of missing unit is 8×8, and tested with fixed missing or random missing data of different missing rates. MSE of both fixed missing and random missing are decreasing with the decrease of missing rate, and the differences between those two kinds of masks are diminishing, although the value corresponding to random missing is always smaller than the other one. Comparing with the results in Table 3, training with large-scale missing data improves the quality of data reconstructed from large-scale missing and the other missing rates, which is effective for both fixed missing and random missing. A possible explanation is that the applying of large-scale data missing reinforces the learning of intrinsic associations of data, and the increase in the proportion of true data reduces the difficulty of data reconstruction.

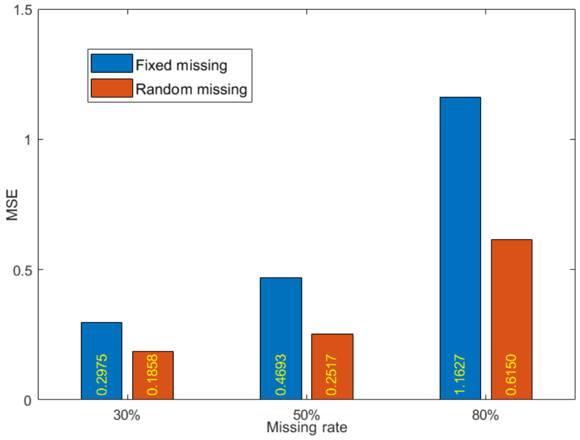


Figure 7. The evaluation result of different missing data ratios

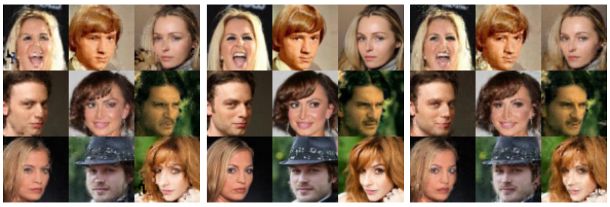


Figure 8. Comparison between data reconstructed by DDPM and PB-DDPM

Figure 8 shows the samples reconstructed from half-missing data with a missing unit of 2×2 by vanilla DDPM (a) and PB-DDPM (c) trained with large-scale missing data, and (b) refers to the ground truth. Whereas DDPM produces samples consistent to the ground truth, despite distortion in the margin of the missing part and deviation in facial features and expressions, data reconstructed by PB-DDPM are closer to the complete data and can exactly recover the basic characteristics.

Comparing the results of recent models on the CelebA dataset in Table 4, the PB-DDPM performs better than the other models when reconstructing data with 50% random missing after training with 80% random missing data that the size of missing unit is 8×8.

Table 4. Comparison of methods on the CelebA dataset

Method	MSE↓	PSNR↑	SSIM↑
LVAE [19]	0.2719	15.1153	0.5526
DDPM [21]	0.2543	15.9428	0.5786
DDIM [22]	0.2556	16.0319	0.5806
IDDPM [23]	0.2533	16.1817	0.5941
Ours	0.2517	16.5213	0.6057

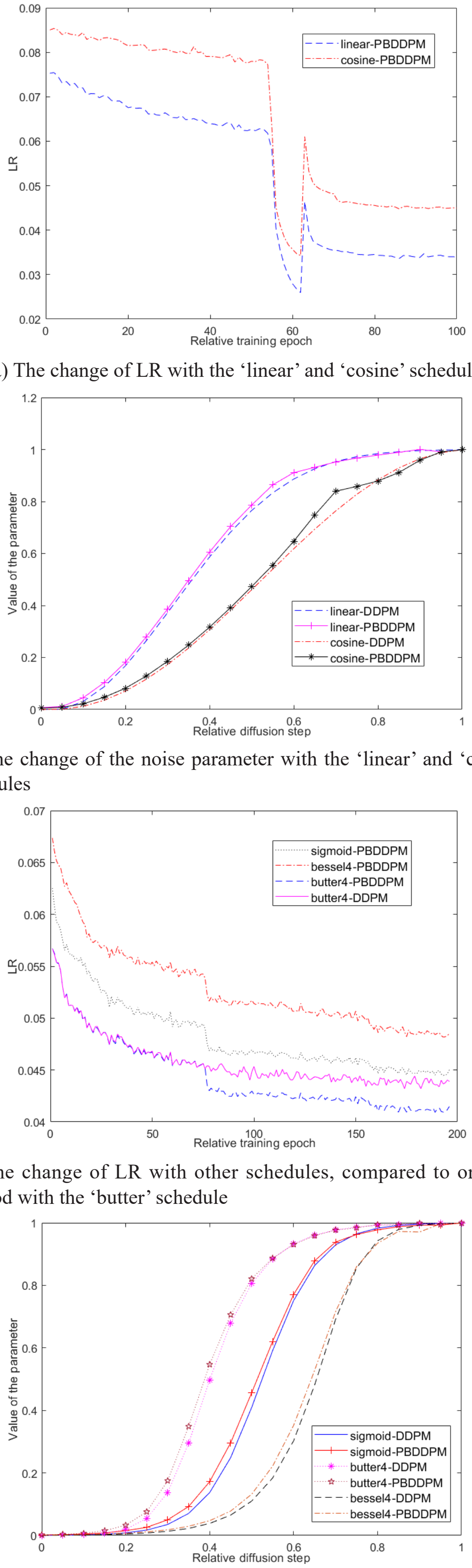
5.2 Results of Different Schedules

Various schedules exhibit varying trends, which may cause different results, especially when the amount of total steps cannot ensure that the value of are close to 1. Moreover, the changes caused by pontoon bridge strategy should be taken into account. The same experiment has been conducted on 5 representative schedules mentioned before to analyze the impact of pontoon bridge strategy and various schedules respectively, are 1) the ‘linear’ schedule, 2) the ‘cosine’ schedule, 3) the ‘sigmoid’ schedule, 4) the ‘butter4’ schedule and 5) the ‘bessel4’ schedule. In order to reduce the probability of extreme values in training procedure, models that adopt ‘linear’ or ‘cosine’ schedule are trained on the CelebA dataset with 80% random missing data that the size of missing unit is 8×8, and models that adopt other schedules are trained on the HAC dataset with 30% random missing data that the size of missing unit is 4×2. In fact, different kind of training data together with the same result in similar γ and $\hat{x}_0$. The convergence of dataset affects the total epoch cost for training.

Figure 9 shows the results of loss function and the values of the parameter γ before and after the utilizing of pontoon bridge strategy. Figure 9(a) and Figure 9(b) correspond to the ‘linear’ and ‘cosine’ schedule while Figure 9(c) and Figure 9(d) represent the others. The x-axis on Figure 9(a) and Figure 9(c) represent the relative training step which means the training curve displayed in the figure is incomplete in order to demonstrate the changes, and the strategy takes effect at the 75th epoch for the CelebA dataset but at the 85th epoch for the HAC dataset. The x-axis on Figure 9(b) and Figure 9(d) represent the relative diffusion step applying a normalization process to ensure consistency.

LR re-stabilizes after several fluctuations for the ‘linear’ and ‘cosine’ schedules, and the value corresponding to the ‘linear’ is always smaller than those belonging to the ‘cosine’. The trend of stable states after fluctuations are approximately consistent. There exists a difference between the value of parameter before and after the adjustment, and values increase in general, despite the decreasing at few points approaching the ending step. Values tend to become flat when the number of steps is large while the trend remains consistent with the original at the beginning of the diffusion process, and values at the junction between those two areas have shown a notable increase.

The curves of LR corresponding to the ‘sigmoid’, ‘butter4’ and ‘bessel4’ schedules are similar to each other, and the value of ‘butter4’ is the smallest while the value of ‘bessel4’ is the largest. After the adjustment, the values of those three schedules have decreased and returned to stability again. Moreover, LR values of the adjusted ‘sigmoid’ and ‘bessel4’ schedule are larger than those of the un-adjusted ‘butter4’ schedule. The fast-changing region of the parameters’ values corresponding to those three schedules are similar too, and the ‘butter4’ takes more steps to increase from 0.8 to 1 while ‘bessel4’ takes more steps to increase from 0 to 0.2. As a result, the adjusted values are closer to the original values in the region that takes more steps, and values also increase in general, except for few points.



(a) The change of LR with the ‘linear’ and ‘cosine’ schedules

(b) The change of the noise parameter with the ‘linear’ and ‘cosine’ schedules

(c) The change of LR with other schedules, compared to original method with the ‘butter’ schedule

(d) The change of the noise parameter with other schedules

Figure 9. Results of Pontoon Bridge method with different noise strategies

The values of the parameter belonging to ‘linear’ and ‘cosine’ schedule increase more smoothly than the others, resulting in fewer epochs cost to reach a stable state. Thus, we are inclined to apply those three schedules in simple tasks that are easy to converge. Furthermore, the noise schedules with parameters $(1-\overline{\alpha}_t)$ that takes more steps within the narrow range close to 1 and smoothly increase from 0 to 1 are supposed to improve the quality of the reconstructed data.

6 Conclusion

On the basis of the denoising diffusion probability model, we introduced a parameter to establish relationship between the training process and the generation process, and proposed a DDPM-based model called Pontoon Bridge Denoising Diffusion Probabilistic Model to reconstruct missing data. At the cost of fewer additional computation, some parameters of the noise schedule were adjusted to enhance the ability to handle data missing without changing the main structure of the model. Meanwhile, targeted training has been conducted by applying multiple masks to construct missing samples as the input, which improved the quality of data reconstructed by the proposed model. Moreover, we compared the results of different data augmentation strategies, and tested the trained model with different kind of missing data to analyze the model’s applicability. Furthermore, the results of models with different noise schedules have been compared and analyzed. The results of experiments proved the effectiveness of PB-DDPM, the MSE result has been reduced by 0.02 at most, the CS result has increased by 0.039 at most, the PSNR result has increased by 1.6% at most, the SSIM result has increased by 8.9% at most while the LR result has reduced by 0.83 at most, and training model with 80% random missing data enhanced the ability to handle multiple kind of missing data whereas trained with fixed missing data would take effect in special scenarios. Besides, different schedules could be applied to handle different tasks, and schedules with special characteristics may improve the quality of the reconstructed data. Future works will include seeking out more effective noise schedules while increasing the total steps of diffusion model and observing the differences under various data distributions. Train the model with multiple kind of missing data at the same time efficiently may enhance the practicality of the model.

Acknowledgment

This research was funded by National Natural Science Foundation under Grant 62173026, the Academic Discipline and Postgraduate Education Project of Beijing Municipal Commission of Education and the Fundamental Research Funds for the Central Universities 2017YJS006.

References

[1] L. Li, Y. Fang, L. Liu, H. Peng, J. Kurths, Y. Yang, Overview of compressed sensing: Sensing model, reconstruction algorithm, and its applications, *Applied Sciences*, Vol. 10, No. 17, Article No. 5909, September, 2020.

[2] Q. Wang, Y. Tang, Y. Ge, H. Xie, X. Tong, P. M. Atkinson, A comprehensive review of spatial-temporal-spectral information reconstruction techniques, *Science of Remote Sensing*, Vol. 8, Article No. 100102, December, 2023.

[3] A. N. Baraldi, C. K. Enders, An introduction to modern missing data analyses, *Journal of School Psychology*, Vol. 48, No. 1, pp. 5–37, February, 2010.

[4] P. Dhariwal, A. Nichol, Diffusion models beat GANs on image synthesis, *2021 35th International Conference on Neural Information Processing Systems*, Virtual Event, Canada, 2021, pp. 8780–8794.

[5] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, *2015 32nd International Conference on Machine Learning*, Lile, France, 2015, pp. 2256–2265.

[6] W. E. Deming, On errors in surveys, *American Sociological Review*, Vol. 9, No. 4, pp. 359–369, August, 1944.

[7] A. Politz, W. Simmons, An attempt to get the “not at homes” into the sample without callbacks, *Journal of the American Statistical Association*, Vol. 44, No. 245, pp. 9–16, March, 1949.

[8] J. W. Graham, Missing data analysis: Making it work in the real world, Annual review of psychology, Vol. 60, pp. 549-576, January, 2009.

[9] D. Bertsimas, C. Pawlowski, Y. Zhuo, From predictive methods to missing data imputation: an optimization approach, *Journal of Machine Learning Research*, Vol. 18, No. 196, pp. 1–39, April, 2018.

[10] R. J. Little, D. B. Rubin, *Statistical Analysis with Missing Data*, John Wiley & Sons, 2019.

[11] F. Bashir, H. Wei, Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm, *Neurocomputing*, Vol. 276, pp. 23–30, February, 2018.

[12] Z. Ghahramani, M. Jordan, Supervised learning from incomplete data via an EM approach, *1993 7th International Conference on Neural Information Processing Systems*, Denver, Colorado, USA, 1993, pp. 120–127.

[13] Y. Sun, J. Li, Y. Xu, T. Zhang, X. Wang, Deep learning versus conventional methods for missing data imputation: A review and comparative study, *Expert Systems with Applications*, Vol. 227, Article No. 120201, October, 2023.

[14] T. Liu, J. Fan, Y. Luo, N. Tang, G. Li, X. Du, Adaptive data augmentation for supervised learning over missing data, *Proceedings of the VLDB Endowment*, Vol. 14, No. 7, pp. 1202–1214, March, 2021.

[15] J. Li, N. Wang, L. Zhang, B. Du, D. Tao, Recurrent feature reasoning for image inpainting, *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 7760–7768.

[16] Y. Zhang, P. J. Thorburn, W. Xiang, P. Fitch, SSIM-a deep learning approach for recovering missing time series sensor data, *IEEE Internet of Things Journal*, Vol. 6, No. 4, pp. 6618–6628, August, 2019.

[17] P. Vincent, H. Larochelle, Y. Bengio, P. A. Manzagol, Extracting and composing robust features with denoising

autoencoders, *2008 25th International Conference on Machine learning*, Helsinki, Finland, 2008, pp. 1096–1103.

[18] D. P. Kingma, M. Welling, Auto-encoding variational bayes, *2014 2nd International Conference on Learning Representations*, Banff, Canada, 2014, pp. 1-14.

[19] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, O. Winther, Ladder variational autoencoders, *2016 30th International Conference on Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 3745–3753.

[20] D. Rezende, S. Mohamed, Variational inference with normalizing flows, *2015 32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 1530–1538.

[21] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, *2020 34th International Conference on Neural Information Processing Systems*, Virtual Event, 2020, pp. 6840–6851.

[22] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models, *2021 9th International Conference on Learning Representations*, Virtual Event, Austria, 2021, pp. 1-22.

[23] A. Q. Nichol, P. Dhariwal, Improved denoising diffusion probabilistic models, *2021 38th International Conference on Machine Learning*, Virtual Event, 2021, pp. 8162–8171.

[24] D. P. Kingma, T. Salimans, B. Poole, J. Ho, Variational diffusion models, *2021 35th International Conference on Neural Information Processing Systems*, Virtual Event, Canada, 2021, pp. 21696–21707.

[25] J. L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, D. Anguita, Transition-aware human activity recognition using smartphones, *Neurocomputing*, Vol. 171, pp. 754–767, January, 2016.

[26] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, *2015 15th IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 3730–3738.

Biographies

Peng Zhang received the bachelor's degree from Beijing Jiaotong University, China, in 2015, where he is currently pursuing the Ph.D. degree with the School of Electronics and Information Engineering. His current research fields include data fusion, deep learning, and internet of things.

Zhenjiang Zhang received the Ph.D. degree from Beijing Jiaotong University, China, in 2013. He is currently a Professor in Beijing Jiaotong University. His research interests include sensor networks, deep learning, and edge computing. Dr. Zhang has been the guest editor of a number of journals, including IET Communications and Sensors.

Yang Zhang received the Ph.D. degree from Beijing Jiaotong University, China, in 2019. Now he is an associate professor in the School of Electronic and Information Engineering, Beijing Jiaotong University. His current research fields include edge computing, information theory, internet of things and wireless sensor network.