Conditional Time Series Imputation for Distinctive Denoising Diffusion Model

Ziliang Wang
490035861

Huicheng Zhang
500244274

Ximeng Chen
520015656

***Abstract*—In this paper, we applied the two types of non-Gaussian noises, including Mixture Gaussian and Gamma noise, into a Conditional-score based denoising diffusion model for the time series imputation tasks. After conducting different experiments and hyper-parameter tuning, we found that Mixture Gaussian underperformed when using the PM2.5 dataset because this noise distribution is not suitable for the heavy tail data with unimodal. Linear, Quadratic and Cosine noise schedules were examined in order to achieve an appropriate noise adding strategy. Experimental results suggest that the Gamma noise which is more flexible to deal with right-skewed data accelerates the model training process in the earlier stage. There is also a slightly better result under different evaluation metrics for Gamma noise compares to Gaussian noise in the later training steps.**

***Index Terms*—DDPM, CSDI, Time series imputation, Gamma Noise**

I. Introduction

Time series are popular when applied to the environment, finance and medical areas [1] [2] [3]. Missing data always arise when dealing with time series, and it is difficult to get access to accurate results from the models without data imputation [4]. However, traditional imputation methods seldom consider the temporal information of data, which is sometimes significant. Therefore, deep learning imputation methods that can acquire temporal relationships in the time series are widely employed.

Score-based generative models, which gradually add noise until the data is fully polluted, then denoising to generate data samples [5], have some huge improvements in many downstream tasks, such as image generation [6] [7]. There is also some research related to sequence modelling, such as TimeGrad, which utilizes the Denoising Diffusion Probabilistic Model (DDPM) for multivariate time series forecasting [8]. However, this method is not suitable for time series imputation because of the characteristics of Recurrent Neural Networks (RNNs). As a result, the Conditional Score-based Diffusion model for Imputation (CSDI) is applied to solve this problem [9].

While CSDI presented satisfactory results in time series imputation tasks, we want to investigate whether there are still any possible improvements. The main methodology of CSDI is adding conditional observations to help the training of the DDPM model for predicting noise. [10] indicates that some non-Gaussian noise, such as Gamma and Mixture Gaussian noise, in the DDPM model performs well compared to traditional Gaussian noise in image generation and speech generation tasks in certain conditions. Therefore, we proposed a new idea to replace the Gaussian noise in CSDI with Gamma noise and Mixture Gaussian noise to analyse the performance and discover if there are any suitable conditions for this replacement. The comparison and analysis under different conditions are presented in detail in our work.

II. Related work

A. Time series imputation

Time series imputations are crucial for real world applications [11]. Traditionally, ARIMA [12] and KNN [13] are utilised for time series imputations. In recent years, RNNs have been widely applied in time series imputation work because they are able to memorize the temporal information of data [14] [15] [16]. Gated Recurrent Units (GRUs) and Generative Adversarial Networks (GANs) are also employed to capture more information and improve accuracy [17] [18]. CSDI [9] as the state-of-the-art models perform significantly better than existing probabilistic and deterministic imputation methods, which add observation conditions to the self-supervised training process.

B. Denoising Diffusion Probabilistic Models

The Denoising Diffusion Probabilistic Model (DDPM) [5] is a recently proposed generative model, which differs from other types of generative models like Auto-Encoding Variational (VAE) [19] and Generative Adversarial Networks (GAN) [20]. For VAE, data is compressed into latent space by an encoder and reconstructed by a decoder. In GAN, generator G continually generates samples to deceive the discriminator D. The discriminator aims to distinguish samples generated by the generator. DDPM consists of a forward process and a backward process. In the forward process, noise $\mathbf{x_0}$ is gradually added to the image in different timestep T until the image follows a Gaussian distribution. In the backward process, the noise is denoised at each stage, resulting in the distribution $q\left(x_t \mid x_{t-1}\right)$ of the image.

1) Forward process: Given data x_0 follows the distribution $q(x)$, the forward diffusion process adds Gaussian noise to x_0 with a series of time step t as the following Equation 1. The Gaussian noise from the time series $x_0 ... x_T$ has variance $\{\beta_t \in (0,1)\}_{t=1}^{T}$.

$$q\left(x_1, \ldots, x_T \mid x_0\right)=\prod_{t=1}^{T} q\left(x_t \mid x_{t-1}\right)$$
$$q\left(x_t \mid x_{t-1}\right)=\mathcal{N}\left(x_t ; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I}\right) \tag{1}$$

The reparameterisation (Equation 2) is used to get x_t by given $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{2}$$

In the given context, x_0 is the original data and ϵ is random noise following Gaussian distribution. By applying reparameterisation, $q\left(x_t \mid x_0\right)$ can be represent as: $q\left(x_t \mid x_0\right) = \mathcal{N}\left(x_t ; \sqrt{\bar{\alpha}_t} x_0,\left(1-\bar{\alpha}_t\right) \mathbf{I}\right)$ where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

In the forward process of the DDPM model, a loss function is computed to facilitate the learning of an improved model. This loss function aids in the optimization process by guiding the model towards better performance. The optimization target is to minimize the noise added to the data and model predicted noise:

$$\min_{\theta} \mathcal{L}(\theta)=\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t} x_0+\sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}, t\right)\right\|^2 \tag{3}$$

2) Backward process: Estimating parameter $q(x_{t-1}|x_t)$ is difficult since it requires the complete dataset, which necessitates the learning of a model p_θ that approximates the conditional probabilities. This model is required for an efficient reverse diffusion process. By using the Bayes's theorem, $\tilde{\mu}_t$ (Equation 5) and $\tilde{\beta}_t$ (Equation 6) of the model p_θ can be calculated by

$$q\left(x_{t-1} \mid x_t, x_0\right)=q\left(x_t \mid x_{t-1}, x_0\right) \frac{q\left(x_{t-1} \mid x_0\right)}{q\left(x_t \mid x_0\right)}. \tag{4}$$

$$\tilde{\boldsymbol{\mu}}_t\left(x_t, x_0\right)=\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} x_t+\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} x_0 \tag{5}$$

$$\bar{\sigma}_t=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \tag{6}$$

By combing equation 2, $\tilde{\mu}_{t-1}$ can be calculated by noisy data x_t:

$$x_{t-1}=\frac{x_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \varepsilon_\theta\left(x_t, t\right)}{\sqrt{\alpha_t}}+\sigma_t z \tag{7}$$

,where z is the random noise follows the $\mathcal{N}(0, \mathbf{I})$.

III. Techniques

A. Conditional score-based diffusion models for imputation

Conditional score-based diffusion models for probabilistic time series imputation models (CSDI) implement an innovative approach for the imputation of time series based on the idea of DDPM. The experimental results demonstrate that CSDI outperforms existing probabilistic methods on healthcare and environmental data, exhibiting an improvement of $40-65\%$ in the continuous ranked probability score (CRPS) and decreasing the mean absolute error (MAE) by $5-20\%$ [7]. The primary difference between CSDI and DDPM is that CSDI takes the observed values into account instead of simply learning x_t and t in the diffusion model. Each time series will be divided into imputation values x_0^{ta} and conditional observations x_0^{co} . In the forward diffusion process, these imputation targets will be added as noise following the concept of DDPM, while the conditional observations remain the same, as shown in Figure 1.

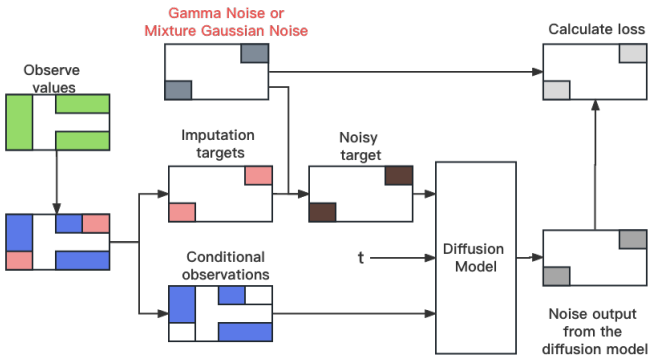

Fig. 1. The training process of the CSDI model

The combined inputs of the noisy targets, conditional observations and timestamps t serve as the inputs for the reverse process of the CSDI model, which means that the reverse process, denoted as p_θ, progressively transforms random noise into plausible time series on the condition of the prior value x_0^{co} [21]. Therefore, it is able to model $p_\theta(x_t^{ta}, t|x_0^{co})$ with a diffusion model by

$$p\left(x_1^{ta}, \ldots, x_T^{ta} \mid x_0^{co}\right)=p_\theta(x_T^{ta}) \prod_{t=1}^{T} p_\theta\left(x_{t-1}^{ta} \mid x_t^{ta}, x_0^{co}\right) \tag{8}$$

and

$$p_\theta(x_{t-1}^{ta} x_t^{ta}, x_0^{co})=\mathcal{N}(x_{t-1}^{ta} ; \mu_\theta(x_t^{ta}, t|x_0^{co}), \sigma_\theta(x_t^{ta}, t|x_0^{co}) \mathbf{I}). \tag{9}$$

The mean, denoted as $\mu_\theta(x_t^{ta}, t|x_0^{co})$, is equal to the estimated mean, represented by $= \tilde{\mu}(x_t^{ta}, t|x_0^{co})$. Similarly, the variance, denoted as $\sigma_\theta(x_t^{ta}, t|x_0^{co})$, is equivalent to the variance from the DDPM, denoted as $\sigma^{DDPM}(x_t^{ta}, t)$. Consequently, the loss of the CSDI model can be computed by minimizing the mean absolute or square error of noise generated in the forward process and outputted from the diffusion model.

B. Mixture Gaussian with CSDI

[22] shows the Fréchet Inception Distance (FID) score was immensely low in the early training stage when the added Gaussian noise is replaced with a mixture of Gaussian and Gamma noise in the DDPM. According to [23], mixture Gaussian distributions are frequently used in data analysis and pattern recognition. Because a mixture Gaussian distribution combines several single Gaussian distributions, which can better fit the shape of the data following multiple distributions.

1) Forward process: When the added noise is replaced with a mixture of Gaussian the equation can be written as

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\left(\sum_{i=0}^{C} z_i \epsilon_t^i\right), \tag{10}$$

where z_i is a dummy variable and C is the number of components of the mixture Gaussian distribution. Considering the two components with the same weight, mu and variance, the forward process equation can be rewritten as

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\left(b\epsilon_t^1 + (1-b)\epsilon_t^2\right), \tag{11}$$

where b follows the Bernoulli distribution with probability p.

2) Backward process: Similar to the backward process in DDPM, the mixture Gaussian distribution is defined as

$$x_{t-1} = \frac{x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta\left(x_t, t\right)}{\sqrt{\alpha_t}} + \sigma_t N_t, \tag{12}$$

where $N_t \sim \mathcal{M}\left(\phi_t^2\right)$ and $\phi_t \in [0,1]$. When extending Mixture Gaussian into CSDI, the sampling steps are modeled as $p\left(x_{t-1}^{\mathrm{ta}} | x_t^{\mathrm{ta}}, x_0^{\mathrm{co}}\right)$. Specifically, the parameterized form of x_{t-1} is obtained through the same step mentioned in Equation 4 and we can obtain data corresponding to the $t-1$ timestamp

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathrm{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(\mathrm{x}_t, t | x_0\right)\right) + \sigma_t N_t, \tag{13}$$

where the $\epsilon_\theta\left(\mathrm{x}_t, |x_0\right))$ conditional predicted noise which consider x_0 as the conditional conditional observations. The loss function becomes

$$\min_\theta \mathcal{L}(\theta) = \left\|N_t - \boldsymbol{\epsilon}_\theta(x_t, t | x_0)\right\|^2. \tag{14}$$

C. Gamma with CSDI

In contrast to the traditional DDPM that utilizes a Gaussian distribution as noise, our noise-based CSDI model also employs a Gamma distribution as noise. The gamma distribution is practical to represent many different real world scenarios [24]. It is also suitable for right-skewed data, which is widely used in meteorology [25]. In the time series area, there are many meteorology or right-skewed data which further firm the ideas of replacing the Gamma noise into the CSDI for time series imputation.

Let us denote c and θ as the shape and scale respectively. The noise of Gamma distribution $\mathcal{G}(c, \theta)$ can be generated by the concentration c and the rate $\frac{1}{\theta}$.

1) Forward process: Since the sum of Gamma distributions with the same scale parameter follows a Gamma distribution, a closed-form equation for x_t can be derived. This equation allows the computation of x_t from x_0 directly in the diffusion procedure after a step-by-step derivation by

$$x_t^{ta} = \sqrt{\bar{\alpha}}x_0^{ta} + (\bar{g}_t - \bar{c}_t\theta_t), \tag{15}$$

where θ_t can be calculated by $\sqrt{\bar{\alpha}}\theta_0$ and c_t can be computed by $\frac{\beta_t}{\alpha_t\theta^2}$. In addition, the formulation of $\bar{c}_t$ and $\bar{g}_t$ are $\bar{c}_t = \sum_{i=1}^{t} k_i$ and $\bar{g}_t \sim \mathcal{G}(\bar{c}_t, \theta_t)$. During the experiment, we will perform hyperparameter tuning by adjusting the values of θ_0 and β.

In the training procedure of our gamma CSDI model, the loss function is represented as the residuals of the gamma noise and noise outputted from the CSDI model. The formulation of the object function can be written as

$$\min_\theta \mathcal{L}(\theta) = \left\|\frac{\bar{g}_t - \bar{c}_t\theta_t}{\sqrt{1-\bar{\alpha}}}_t - \boldsymbol{\epsilon}_\theta(x_t^{ta}, t | x_0^{co})\right\|^2. \tag{16}$$

2) Backward process: The algorithm initiates by sampling a noise, denoted as x_T, with a mean value of zero by $x_T = \gamma - \theta_T * \bar{k}_T$, where γ is sampled from the Gamma distribution $\mathcal{G}(\bar{c}_t, \theta_t)$. Hence, we are able to get access to $\mu_\theta(x_t^{ta}, t | x_0^{co})$ by Equation 7. In the process of parameterization of sampling, x_{t-1} can be written as

$$x_{t-1} = \mu_\theta(x_t^{ta}, t | x_0^{co}) + \sigma_t \frac{z - \theta_{t-1}\bar{c}_{t-1}}{\sqrt{1-\bar{\alpha}}}, \tag{17}$$

where $z \sim \mathcal{G}(\bar{c}_{t-1}, \theta_{t-1})$.

Compared with Gaussian distribution, the changes include substituting the beginning sampling point x_T, changing the sampling noise z, and modifying the equation for calculating x_{t-1}. After gradually integrating the value of time t from T to i, we can eventually obtain the final imputation target denoted as x_0^{ta}.

D. Noise Schedule

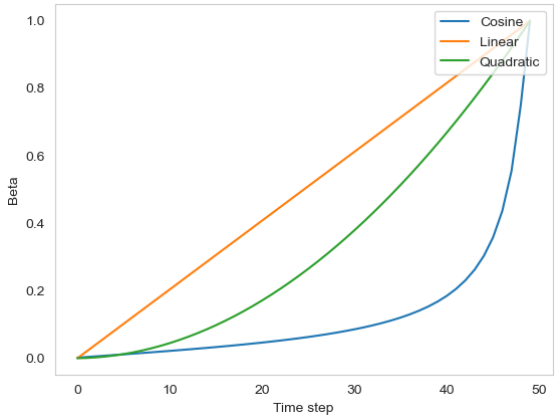

Fig. 2. Three different types of noise schedule

Different noise schedules are suitable for different environments and analyzing the performance of the diffusion model under various conditions requires modifying the noise schedule accordingly. This allows for a comprehensive evaluation of the performance of the model across different scenarios and helps identify the optimal noise schedule for each specific condition [26]. Different noise schedules have various noise levels at each time step as Figure 2 shows. Originally in the CSDI

method, the authors only implemented linear and quadratic schedules, we extended the cosine and sqrt schedules to test the performance of adding Gamma and Mixture Gaussian noise.

1) Linear and Quadratic schedule: [5] used a linear scheduled noise with $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$. Unlike the traditional linear schedule applied in DDPM, with quadratic schedule, noise increases quadratically at each timestamp.

2) Cosine schedule: [27] proposed a cosine schedule noise method because the linear schedule adds too much noise and the sampling quality is not significantly improved. The maximum value is restricted to 0.999. Therefore, the Cosine schedule is used here to investigate whether it can also enhance the sampling quality for Gamma noise in time series interpolation tasks.

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)^2, \beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}} \tag{18}$$

IV. Experiments and Results

A. Dataset

We used the Beijing PM2.5 dataset from 2014/05/01 to 2015/04/30 as the ground truth file. Some data are removed in random timestamps to allow the algorithms to fill these points.

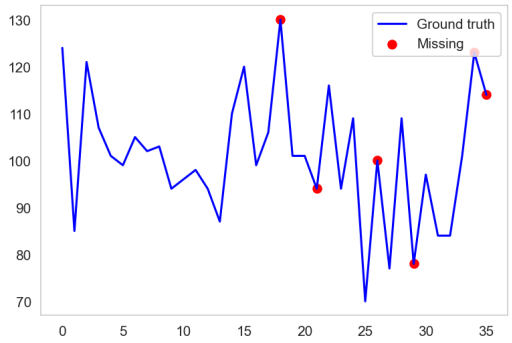

Fig. 3. Data example: PM2.5 data with adding random missing entries

B. Dependencies

Packages	Version
Pytorch	1.9.0
Matplotlib	3.7.1
pickle	4.0.0
Numpy	1.21.2
Pandas	2.0.0

C. Evaluation metrics

1) Root of mean square error: Compared to MAE, the Root of mean square error (RMSE) predicted and observed value. When the difference between the predicted and the observed value is large, the value of RMSE will be amplified.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{19}$$

2) Mean absolute error: The Mean absolute error (MAE) measures the predicted error and observed data in each discrete time step.

$$\text{MAE} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} \tag{20}$$

3) Continuous ranked probability score: Continuous ranked probability score (CRPS) [28] is a measurement to measure the difference between continuous probability distribution with the observed data.

$$CRPS(F, y) = \int \left(F(x) - \mathbf{1}_{\{x \geq y\}}\right)^2 dx, \tag{21}$$

The $F(x)$ is the predicted probability distribution function and $\mathbf{1}_{\{x \geq y\}}$ is the Heaviside step function. The lower CRPS stands for the smaller difference between predicted and observed data.

D. Experiments, comparison & analysis

To test the result of incorporated Gaussian mixture and Gamma noise with CSDI. We selected the following hyper-parameters related to CSDI for experiments:

Learning rates	0.001,0.0001
Added noise type	Gaussian, Mixture Gaussian, Gamma
Imputation model type	DDPM,CSDI
Schedule type	Linear,Quadratic,Consine

1) Mixture Gaussian: After modifying the ϕ_t to make it more suitable for time series imputation, the performance is unsatisfactory under different ϕ_t. We experimented with the linear and quadratic decreasing or increasing ϕ_t with several values, however, after 5 epochs, the model becomes overfitting. This results in the performance of the model after 20 epochs becoming significantly worse than traditional Gaussian and Gamma noise.

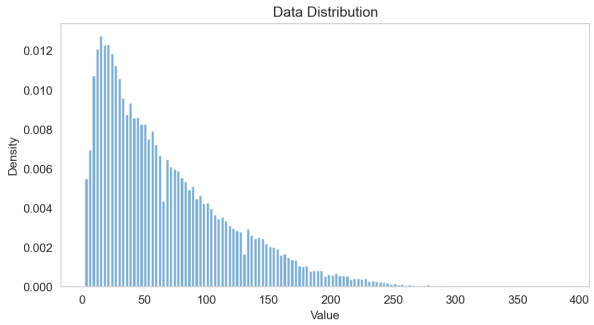

Fig. 4. Data distribution

The reason behind this is ϕ_t is a variable that changes with time steps, when the variable is not well defined, then the shape of the distribution will be a bimodal distribution, which is not appropriate for the training process. From figure 4, the distribution of data we tested is not suitable for this model. Because this is a unimodal distribution with a heavy tail, which is not suitable for Mixture Gaussian distributions [23]. Therefore, even if it converges faster than traditional

TABLE I
GAMMA HYPER-PARAMETER TUNING RESULTS WITH RMSE/MAE/CRPS LOSS

	Epochs				
θ_0	20	40	60	80	100
0.001	19.698/10.623/0.120	19.993/10.173/0.114	19.347/10.052/0.113	19.787/9.988/0.113	19.084/9.703/0.110
0.002	22.867/11.274/0.130	20.994/10.422/0.120	19.317/9.756/0.112	19.377/9.715/0.113	19.803/9.901/0.114
0.004	20.049/10.581/0.119	**18.647/9.790/0.110**	**18.135**/9.733/0.111	18.856/**9.464**/0.109	18.795/**9.428**/0.108
0.006	20.564/10.914/0.124	19.091/10.110/ 0.112	18.894/9.915/ 0.111	18.911/9.673/0.108	19.167/9.646/0.111
0.008	**19.492**/10.748/0.123	20.198/10.948/0.124	18.397/9.881/0.112	**18.175**/9.508/0.108	18.307/9.500/**0.107**
0.0002	20.627/11.038/0.125	18.755/10.099/0.114	18.911/9.931/0.111	18.909/9.991/0.112	19.438/10.033/0.113
0.0004	19.950/**10.487/0.117**	20.284/10.214/0.116	19.073/9.891/0.111	19.499/9.835/0.113	19.341/9.701/0.112
0.0006	21.535/11.291/0.128	19.641/10.167/0.114	19.177/9.844/0.112	18.999/9.714/**0.110**	19.638/9.789/0.112
0.0008	21.858/11.222/0.127	19.984/10.295/0.117	18.967/**9.650/0.110**	19.029/9.632/**0.110**	**18.327**/9.540/0.108

TABLE II
COMPARE DDPM AND CSDI IMPUTATION RESULTS WITH RMSE/MAE/CRPS LOSS UNDER GAUSSIAN AND GAMMA NOISE

	Epochs				
	20	40	60	80	100
Unconditional(Gaussian)	24.201/13.136/0.143	22.685/12.649/0.138	22.846/12.619/0.139	21.393/12.339/0.137	24.929/12.908/0.141
Unconditional(Gamma)	22.513/13.319/0.149	25.143/13.256/0.145	24.235/12.740/0.140	22.955/12.547/0.138	24.450/12.835/0.142
CSDI(Gaussian)	22.451/11.219/0.124	**19.651/10.228/0.113**	19.117/**9.841/0.110**	19.01/9.700/0.110	18.972/9.553/0.108
CSDI(Gamma)	**19.492/10.748/0.123**	20.198/10.948/0.124	**18.397**/9.881/0.112	**18.175/9.508/0.108**	**18.307/9.500/0.107**

TABLE III
COMPARE THE INFLUENCE OF DIFFERENT LEARNING RATES AND NOISE SCHEDULES WITH RMSE/MAE/CRPS LOSS UNDER GAUSSIAN AND GAMMA NOISE

			Epochs				
Noise type	Learning rate	Schedule	20	40	60	80	100
Gaussian	0.0001	Quadratic	**36.216/18.401**/0.285	**22.459/12.569/0.143**	**22.803/12.576/0.138**	**24.338/12.90/0.142**	24.65/12.956/**0.141**
	0.001	Quadratic	20.807/11.266/0.124	**20.137**/10.794/0.121	**18.978/9.857/0.109**	**19.208/9.656/0.107**	19.302/9.744/**0.109**
	0.0001	Cosine	29.026/16.993/0.188	22.997/12.574/0.139	21.948/11.863/0.131	21.356/11.453/0.127	20.253/10.899/0.121
	0.001	Cosine	21.310/11.420/0.127	18.960/10.026/0.111	19.287/9.853/0.111	19.690/9.773/**0.110**	**19.51/9.745/0.109**
Gamma	0.0001	Quadratic	42.161/24.255/**0.279**	25.197/14.281/0.162	24.209/13.283/0.151	24.933/13.204/0.151	**22.952/12.469**/0.142
	0.001	Quadratic	**19.502/10.766/0.122**	20.529/**10.221/0.117**	19.351/9.910/0.112	19.989/9.883/0.114	**18.722/9.700/0.110**
	0.0001	Cosine	**21.684/10.897/0.126**	**21.023/10.467/0.121**	**19.446/10.160/0.116**	**18.887/9.871/0.112**	**19.136/9.727/0.110**
	0.001	Cosine	**20.420/10.494/0.120**	**18.868/9.702/0.110**	**18.819/9.652/0.108**	**19.226/9.548/0.110**	19.728/9.765/0.113

Gaussian in the first few epochs, the model will underperform after because of the over-fitting problem.

Therefore, we decided to abandon this method after thoroughly experiments and the rest of the analyses will focus on Gamma noise versus traditional Gaussian noise.

2) Gamma: To enhance performance, it is necessary to tune the Gamma hyper-parameter θ_0. By adjusting this parameter, we can optimize the performance of the model and achieve better results. Tuning θ_0 allows us to find the optimal value that improves the overall performance of the model. From table I, when θ_0 is 0.004, in every 20 epochs, it has the most stable and satisfied performance. In our subsequent experiments, we expect that the performance should remain relatively stable every 20 epochs. If the performance fluctuates significantly, it can reduce the reliability of evaluating and tuning other general hyperparameters.

3) Compare DDPM and CSDI imputation methods: In order to examine the performance of key components of our model, we conduct a ablation study. In our ablation study, we removed the CSDI module while retaining the DDPM component. Table II shows the median evaluation performance of adding 100 imputation samples to the test. The results indicate that CSDI surpasses all unconditional cases in terms of performance. Furthermore, the utilization of Gamma noise in CSDI outperforms Gaussian noise in the majority of the results, particularly during the initial 20 epochs. Compared with Gaussian, Gamma is improved by 13.2%, 4.2%, and 0.8% in iteration respectively in RMSE, MAE and CRPS three evaluation metrics in the early stage. For the later stage, Gamma noise also has a slightly lower error compared to Gaussian noise. The experiment results display the advancement of our model.

4) General hyper-parameter tuning: Hyperparameter tuning mainly focuses on the schedule strategy and learning rates. We conducted comparative experiments between the Gamma and Gaussian noise. From table III, the quadratic noise schedule performs well under Gaussian noise and the cosine noise schedule performs better under Gamma noise. When the learning rate is low, cosine schedule has relatively stable and better performance. However, when the learning rate is 0.001, the difference can be ignored. A larger learning rate often brings some benefits to the training process of our model in early stage of training, while over-fitting problems arise after 80 epochs. The cosine schedule is more robust to over-fitting problems because it mitigates the information loss during the training. Overall, Gamma noise has satisfied training performance with faster convergence speed in the first

40 epochs and performs slightly better than Gaussian noise in later stages.

V. Conclusion and Discussion

A. Limitations and further improvements

We tried to modify the distribution of adding noise to CSDI; however, we tested the performance on the same dataset that CSDI utilised. This may affect the performance of our methods because the distribution of the dataset significantly affects the final performance of our model. As a result, the mixture Gaussian distribution performs worse than the other two types of noise, no matter how we revise the variance of the distribution. Moreover, we only attempted two different types of basic noise distribution without huge performance improvements, which is not enough to replace the CSDI with traditional noise distribution.

In the future, we will try to apply more generalised distributions to the CSDI model to test the suitability of each distribution under different datasets. We will also investigate the performance of revised noise distributions for time series forecasting under the DDPM model.

B. Conclusion

In this paper, we replaced the noise in the CSDI model with Mixture of Gaussian and Gamma noise. According to the experiment results, we found that the our model is more stable and outperformed when θ_0 is 0.004. Our findings demonstrate a noteworthy improvement in the early training stage when utilizing Gamma noise, while a slight improvement is observed in the later training stage for time series imputation tasks. This suggests that the introduction of Gamma noise is particularly beneficial in the initial phases of training, contributing to enhanced performance in time series imputation, which aligns with the result [10] in the CelebA dataset. The Mixture of Gaussian approach did not perform well in our experiments, indicating that it is not suitable for the PM2.5 dataset. The characteristics of the PM2.5 dataset may not align well with the assumptions and modeling capabilities of the Mixture of Gaussian method.

References

[1] E. Afrifa-Yamoah, U. A. Mueller, S. M. Taylor, and A. J. Fisher, "Missing data imputation of high-resolution temporal climate time series data," *Meteorological Applications*, vol. 27, no. 1, Jan. 2020, doi: https://doi.org/10.1002/met.1873.

[2] W. Pongsena, P. Ditsayabut, N. Kerdprasop, and K. Kerdprasop, "Deep Learning for Financial Time-Series Data Analytics: *An Image Processing Based Approach," International Journal of Machine Learning and Computing*, vol. 10, no. 1, pp. 51–56, Jan. 2020, doi: https://doi.org/10.18178/ijmlc.2020.10.1.897.

[3] N. Wagner, Z. Michalewicz, M. Khouja, and R. R. McGregor, "Time Series Forecasting for Dynamic Environments: The DyFor Genetic Program Model," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 4, pp. 433–452, Aug. 2007, doi: https://doi.org/10.1109/tevc.2006.882430.

[4] P. J. García-Laencina, P. H. Abreu, M. H. Abreu, and N. Afonoso, "Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values," Comput. Biol. Med., vol. 59, pp. 125–133, 2015.

[5] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Advances in *Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[6] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," Advances in *neural information processing systems*, vol. 32, 2019.

[7] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv* [cs.LG], 2020.

[8] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting," *Proceedings of the 38th International Conference on Machine Learning*, Jan. 2021.

[9] Y. Tashiro, J. Song, Y. Song, and Stefano Ermon, "CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation," *35th Conference on Neural Information Processing Systems*, Dec. 2021, doi: https://doi.org/10.48550/arxiv.2107.03502.

[10] Eliya Nachmani, R. San-Roman, and L. Wolf, "Non Gaussian Denoising Diffusion Models," Jun. 2021.

[11] C. Fang, S. Song, Z. Chen, and A. Gui, "Fine-Grained Fuel Consumption Prediction," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2783–2791.

[12] G. Peter. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003, doi: https://doi.org/10.1016/s0925-2312(01)00702-0.

[13] G. Batista and M.-C. Monard, "A study of k-nearest neighbour as an imputation method.," in *Hybrid Intelligent Systems*, ser Front Artificial Intelligence Applications, Jan. 2002, vol. 30, pp. 251–260.

[14] Y. Yan, K. Zhao, J. Cao, and H. Ma, "Prediction research of cervical cancer clinical events based on recurrent neural network," *Procedia Computer Science*, vol. 183, pp. 221–229, 2021, doi: https://doi.org/10.1016/j.procs.2021.02.052.

[15] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," Scientific Reports, vol. 8, no. 1, Apr. 2018, doi: https://doi.org/10.1038/s41598-018-24271-9.

[16] W. Cao, Ben Zhong Tang, J. Li, H. Zhou, L. Li, and Y. Li, "BRITS: Bidirectional Recurrent Imputation for Time Series," *Annual Conference on Neural Information Processing Systems 2018*, no. 3–8, May 2018.

[17] D. Snow, "MTSS-GAN: Multivariate Time Series Simulation Generative Adversarial Networks," *SSRN Electronic Journal*, 2020, doi: https://doi.org/10.2139/ssrn.3616557.

[18] Y. Luo, Y. Zhang, X. Cai, and X. Yuan, "E2GAN: End-to-End Generative Adversarial Network for Multivariate Time Series Imputation," *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Aug. 2019*, doi: https://doi.org/10.24963/ijcai.2019/429.

[19] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," arXiv [stat.ML], 2013.

[20] I. J. Goodfellow et al., "Generative Adversarial Networks," *arXiv* [stat.ML], 2014

[21] Y. Song, C. Durkan, I. Murray, and Stefano Ermon, "Maximum Likelihood Training of Score-Based Diffusion Models," *35th Conference on Neural Information Processing Systems*, Jan. 2021

[22] E. Nachmani, R. S. Roman, and L. Wolf, "Non Gaussian Denoising Diffusion Models," *arXiv* [cs.LG], 2021.

[23] F. Najar, Sami Bourouis, Nizar Bouguila, and Safiya Belghith, "A Comparison Between Different Gaussian-Based Mixture Models," Oct. 2017, doi: https://doi.org/10.1109/aiccsa.2017.108.

[24] Eric, Oti, and Francis, "A study of properties and applications of gamma distribution," *African Journal of Mathematics and Statistics Studies*, vol. 4, no. 2, pp. 52–65, 2021.

[25] L. Yan, "Confid ence interval estimation of the common mean of several gamma populations," PLoS One, vol. 17, no. 6, p. e0269971, 2022.

[26] T. Chen, "On the importance of noise scheduling for diffusion models," *arXiv* [cs.CV], 2023.

[27] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," *arXiv* [cs.LG], 2021.

[28] J. E. Matheson and R. L. Winkler, "Scoring rules for continuous probability distributions," Manage. Sci., vol. 22, no. 10, pp. 1087–1096, 1976.

VI. Appendix

The code has been uploaded to Colab. Note, our experiment is not runed on Colab environment and we conducted the different experiments on different ipnb files for save time purposes.

Please use Chrome open the following link:
https://drive.google.com/drive/folders/1AEjenQ7uQI17OwxMBwaFkIns-szzPMEH?usp=sharing

Our research is based on the CSDI paper https://arxiv.org/abs/2107.03502 and code is referencing from https://github.com/ermongroup/CSDI

The files contain Model, Dataset construction, training, evaluate, experiments, four parts. Please run the code in order.