

# DISCRETE-TIME DIFFUSION-LIKE MODELS FOR SPEECH SYNTHESIS

Xiaozhou Tan, Minghui Zhao, Anton Ragni

Department of Computer Science, University of Sheffield, UK  
 {xtan30, mzhao39, a.ragni}@sheffield.ac.uk

## ABSTRACT

Diffusion models have attracted a lot of attention in recent years. These models view speech generation as a continuous-time process. For efficient training, this process is typically restricted to additive Gaussian noising, which is limiting. For inference, the time is typically discretized, leading to the mismatch between continuous training and discrete sampling conditions. Recently proposed discrete-time processes, on the other hand, usually do not have these limitations, may require substantially fewer inference steps, and are fully consistent between training/inference conditions. This paper explores some diffusion-like discrete-time processes and proposes some new variants. These include processes applying additive Gaussian noise, multiplicative Gaussian noise, blurring noise and a mixture of blurring and Gaussian noises. The experimental results suggest that discrete-time processes offer comparable subjective and objective speech quality to their widely popular continuous counterpart, with more efficient and consistent training and inference schemas.

**Index Terms**— diffusion models, flow matching, iterative process, speech synthesis

## 1. INTRODUCTION

Diffusion and diffusion-like models have recently garnered significant attention in the areas of image [1], language [2], and speech generation [3]. A diffusion model consists of both a noising and a denoising process, forming a data trajectory between clean data and fully corrupted, noisy, data. One of the most popular diffusion models (DM) is the score-based generative model [4], where scores are derivatives of log-likelihoods of noised data in a continuous-time  $t$  space. These scores are at the core of stochastic differential equations (SDEs) that describe how to denoise noised data into clean data [1]. Another popular diffusion-like model is the flow matching model (FM) [5]. The core idea behind FM is to learn a velocity field that describes a probability path between a source distribution and a target distribution. This probability path defines the data trajectory over a continuous-time, and the velocity field corresponds to the time derivative of this trajectory. Flow matching [6] generalizes diffusion models by interpreting the data trajectories generated through the noising process — modeled by SDEs in diffusion models — as probability paths between source and target distributions. These two models are similar in that they both perform a noising process in a continuous-time space in training and generate samples using discretized differential equation solvers [5]. During inference, these models start from a fully corrupted data/initial sample and iteratively refine it to obtain a clean sample.

Despite their successful application to text-to-speech (TTS) [7, 8], the majority of these works [4, 8] only choose additive Gaussian noise as a noising method. This choice implicitly assumes that the noised data are locally well-modelled by isotropic Gaussians, which have covariance matrices proportional to the identity matrix. However, the covariance matrix of the data, such as the Mel spectrogram, is non-isotropic and depends on the underlying signal energy. Unfortunately, a more complex noising process can render core components — scores (in DM) and velocity fields (in FM) — analytically unsolvable, which is limiting, thus making training intractable. In addition, DM and FM training and inference methods are inconsistent. Training is performed in a continuous-time space, whereas inference is performed in a discrete-time space as continuous-time inference would require an infinite number of inference steps. Another discrepancy between training and inference in score-based generative models [4] is that the initial sample for inference is assumed to be drawn from the forward noising process at a time step approaching infinity, but during training, the model is never exposed to these extreme time steps.

In contrast, diffusion-like models that have a fully discrete training and inference process have not been evaluated for TTS. Like DMs and FMs, they have an iterative refinement process during inference. Unlike DMs and FMs, they preserve the consistency between training and inference by performing both processes in discrete-time space  $n$  and by using exactly the data exposed in training as the initial sample during inference. In addition, more diverse noise types, for example, multiplicative Gaussian noise, can be explored with these models. Unlike in DM or FM cases, no substantial changes are needed. The application of such discrete-time models in speech synthesis has lagged behind that in image generation, where blurring [9], snowification [10], and noise mixture [11] have been explored. This paper aims to address this gap by exploring discrete-time diffusion-like models with the following noising methods:

- Additive Gaussian noise. This allows comparison between fully discrete and partially discretized processes.
- Multiplicative Gaussian noise. While it is commonly used to model signal-dependent variability in natural systems [12, 13], it is underexplored in DM and FM. Compared with commonly used additive Gaussian noise, it has a non-isotropic covariance matrix. Because real-world data’s covariance matrix is highly likely non-uniform, multiplicative Gaussian noise might generalize better for real-world data, such as speech.
- Blurring noise is an example of fully deterministic noise. Using blurring as a noising process examines the need for any randomness in diffusion-like models in speech synthesis.
- A mixture of blurring and Gaussian noise. This mixture of noise is explored for leveraging the structured dependencies

Thanks to Mattias Cross from University of Sheffield for answering some theoretical questions.

of the deterministic noising process and stochastic variability in the stochastic process [11].

The rest of the paper is organized as follows: Section 2 presents the preliminaries of this work, and Section 3 describes different noising processes, training, and inference methods explored in this work. Section 4 presents and discusses the experimental results. Section 5 provides the conclusion.

## 2. PRELIMINARIES

### 2.1. Continuous-time diffusion-like models

In continuous-time DMs [4], noised data at each time step can be obtained in closed form without simulation. Clean data can be recovered by integrating the score function over a discretized stochastic differential equation (SDE) [1], which also has a corresponding non-stochastic formulation as an ordinary differential equation (ODE). Models can be trained to predict score [4], noise [14], or a mixture of clean data and noise [15]. In all these models, scores are either directly predicted [4] or computed [14, 15], and then used by discretized SDE or probability-flow ODE samplers for denoising.

FM learns a time dependent velocity field that describes a probability path between a source distribution and a target distribution. By viewing sample from source distribution as clean data and sample from target distribution as fully corrupted data, moving along the velocity field can be regarded as noising/denoising process. Inference in FM can be performed by numerically integrating the velocity field from the target (fully corrupted) distribution back to the source (clean) distribution. The common methods for defining the velocity field in flow matching model are optimal transport conditional flow matching (OT-CFM) [16] and rectified flows [17]. Both methods define a straight path between clean data  $\mathbf{X}_0$  and fully corrupted data  $\mathbf{X}_1$ . In particular, a data trajectory between  $\mathbf{X}_0$  and  $\mathbf{X}_1$  in rectified flows is given by

$$\mathbf{X}_t = (1 - t)\mathbf{X}_0 + t\mathbf{X}_1, \quad (1)$$

where  $t$  belongs to  $[0, 1]$ . Both methods have also been successfully applied to speech synthesis [18, 8, 19, 20]. Interestingly, although they predict a velocity field, the clean data can be computed through the deterministic link between velocity fields and clean data at any denoising step rather than just the terminal value  $t = 0$ . For instance, in rectified flow [17],  $\mathbf{X}_0$  can be computed by simply subtracting the predicted velocity field at any time  $t$  from  $\mathbf{X}_1$ . Thus any such noise prediction network could be viewed as a clean data prediction network rather than a noise prediction network as commonly described in the literature.

### 2.2. Discrete-time diffusion-like models

Interestingly, the popularity of diffusion models started from a discrete-time diffusion model — denoising diffusion probabilistic model (DDPM) [1, 21] rather than continuous. DDPMs are Markov chains that progressively add Gaussian noise by transitioning from the start state,  $n = 1$ , to the end state,  $n = T$ . The noise added at discrete-time index  $n$  is predicted and used to compose an inverse Markov chain. Given noised data in state  $n$ , the inverse Markov chain is used to predict less noised data in state  $n - 1$ . Although DDPMs could achieve high-fidelity results [1], sampling typically requires simulating a Markov chain for many (thousands of) steps. While DDPM states are Markov, they generally do not need to be Markov. Non-Markovian noising and denoising processes have been shown to lead to more efficient models [22], capable of reducing

the number of inference steps to 100s rather than 1000s. Deterministic Markov chains, implemented through deterministic noising processes such as blurring, have been explored in image processing [10], but their application in speech synthesis remains largely underexplored.

## 3. METHOD

This section presents noising, training and inference methods used in this work. Four noising processes are explored for the first time in a discrete-time diffusion-like model in speech synthesis.

### 3.1. Noising process

In all noising processes, the corrupted data  $\mathbf{X}_n$  in step  $n$  can be calculated directly without simulation by

$$\text{noising}(\mathbf{X}_0, \mathbf{U}, n) = \mathbf{X}_n, \quad n \in \{0, 1, 2, \dots, N\}, \quad (2)$$

where  $\mathbf{X}_0$  is clean data,  $\mathbf{U} = E_\theta(\mathbf{c})$  is the text embedding, where  $\mathbf{c}$  is the input text. Although this restricts the range of possible noising methods, it provides an efficient training approach similar to the continuous-time methods of DM and FM.

#### 3.1.1. Additive Gaussian noise

Two types of additive Gaussian noising processes are explored.

**In the first additive Gaussian noising process**, a discretized noising process adapted from a popular model Grad-TTS [7] is employed.

$$\begin{aligned} \mathbf{X}_n = & \left( I - e^{-\frac{1}{2} \int_0^n \beta_s ds} \right) \mathbf{U} + e^{-\frac{1}{2} \int_0^n \beta_s ds} \mathbf{X}_0 \\ & + \int_0^n \sqrt{\beta_s} e^{\frac{1}{2} \int_s^n \beta_u du} d\mathbf{W}_s, \end{aligned} \quad (3)$$

where  $\mathbf{W}_t$  is a Brownian motion (BM), which is integrated over time and evaluated at discrete-time step  $n$ ,  $\beta_s$  is a linearly increasing function with respect to  $s$ . Discretizing the noising process in this model allows comparison between fully discrete and partially discretized processes. In the following, this adapted system is called ‘Grad-TTS-DT (discrete-time)’.

**In the second additive Gaussian noising process**, a straight path between clean data and corrupted data is explored

$$\mathbf{X}_n = \left(1 - \frac{n}{N}\right)\mathbf{X}_0 + \frac{n}{N}(\epsilon_1 + \mathbf{U}), \quad (4)$$

where  $\epsilon_1 \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$  and  $\sigma$  is a hyperparameter. This system can be regarded as a discretized version of the rectified flow (1) [17] or discrete-time flow matching. In a continuous noising process, straight paths are believed to be preferred [17] because they correspond to deterministic couplings that do not increase transport cost under convex cost functions. This motivates exploring straight paths in a discrete-time space. A straight path is also applied in the following noising method. In the following, this system is called ‘RFAG (rectified flow with additive Gaussian noise)’.

#### 3.1.2. Multiplicative Gaussian noise

In this case, additive Gaussian noise  $\epsilon_1 + \mathbf{U}$  in (4) is replaced with multiplicative noise  $\epsilon_2 \cdot \mathbf{U}$ , where  $\epsilon_2 \sim \mathcal{N}(\mathbf{I}, \sigma^2 \mathbf{I})$ , and

$$\mathbf{X}_n = \left(1 - \frac{n}{N}\right)\mathbf{X}_0 + \frac{n}{N}(\epsilon_2 \cdot \mathbf{U}). \quad (5)$$

The denoising process in this case starts with a multiplication between Gaussian noise and the prior. Multiplicative Gaussian noise is commonly used to model signal-dependent variability in natural systems, such as biological vision [23], auditory perception [24], and sensor imaging [25]. Compared with linear diffusion paths of additive Gaussian noising process, multiplicative Gaussian introduces non-linear distortions. This leads to richer and more varied transformations potentially leading to more diverse and expressive outputs. In the following, this system is called ‘RFMG (rectified flow with multiplicative Gaussian noise)’.

### 3.1.3. Blurring noise

The blurring process is performed by applying a heat equation [9]

$$\text{blurring}(\mathbf{X}_0, n) = \mathbf{V} \exp(\mathbf{\Lambda} n) \mathbf{V}^\top \mathbf{X}_0, \quad (6)$$

where  $n \in \{0, 1, 2, \dots, N\}$ ,  $\mathbf{V}^\top$  is the cosine basis projection matrix [9].  $\mathbf{\Lambda}$  is a negative matrix with the same shape  $(W, H)$  as  $\mathbf{X}_0$  and  $\lambda_{i,j}$  given by  $-\pi^2 \left( \frac{i^2}{W^2} + \frac{j^2}{H^2} \right)$ , where  $i = 0, \dots, W - 1$  and  $j = 0, \dots, H - 1$ . This operation smooths/averages out  $\mathbf{X}_0$  in the noised samples

$$\mathbf{X}_n = (1 - \frac{n}{N}) \text{blurring}(\mathbf{X}_0, n) + \frac{n}{N} \mathbf{U}. \quad (7)$$

This noising process is fully deterministic. Deterministic noising methods leverage structured dependencies between localized spectrotemporal features and global spectral patterns to enhance signal representation. In the following, this system is called ‘Blurring’.

### 3.1.4. Mixture of blurring and Gaussian noises

The mixture noise is drawn from a Gaussian distribution whose mean is the blurring noise (7)

$$\text{mixture\_noise}(\mathbf{X}_0, n) \sim \mathcal{N}(\text{blurring}(\mathbf{X}_0, n), \frac{1}{2}(-\mathbf{\Lambda}) \odot \mathbf{I}), \quad (8)$$

where  $\mathbf{\Lambda}$  is the same as the negative matrix in the blurring process (6). The constant value  $\frac{1}{2}$  is chosen based on the best results achieved in a related image generation work [11]. This mixture noise is applied to  $\mathbf{X}_0$  to yield noised samples as follows

$$\mathbf{X}_n = (1 - \frac{n}{N}) \text{mixture\_noise}(\mathbf{X}_0, n) + \frac{n}{N} \mathbf{U}. \quad (9)$$

This noising process [11] mixes blurring (6) and Gaussian noise. Although blurring can exploit the structured dependencies, it disregards the role of noise (randomness) in structuring the data manifold [26]. Therefore, the above process takes advantage of the deterministic and stochastic noising processes by controlling blurring and noise jointly. In the following, this system is called ‘Mixture’.

## 3.2. Training

The discrete-time diffusion-like models in this work are implemented following Grad-TTS [7]. In particular, the Monotonic Alignment Search (MAS) method followed by [27] is used to train a duration predictor. The duration predictor is part of a text encoder which produces prior  $\mathbf{U} = E_\theta(\mathbf{c})$ . Using this prior, a discrete-time training process is applied to the decoder.

The decoder is a non-causal residual convolutional network  $R_\theta$  adapted from Grad-TTS. The adaptation includes the removal of continuous-time embeddings and the prediction of clean data instead of scores. As is discussed in Section 2, the majority of diffusion-like

models can be viewed as predicting the clean data. Also, predicting clean data directly is more consistent [28] than predicting noise in different time steps and it generalizes across all noise levels. The training aims to minimize the mean square error (MSE) between predicted  $\hat{\mathbf{X}}_0$  and clean data  $\mathbf{X}_0$ .

$$\mathcal{L}_{\text{clean\_data}} = \text{MSE}(R_\theta(F(\mathbf{X}_0, E_\theta(\mathbf{c}), n), E_\theta(\mathbf{c})), \mathbf{X}_0). \quad (10)$$

The training alternates between alignment optimization (by finding alignment using MAS and optimizing  $\mathcal{L}_{\text{enc}}$ ) and total loss optimization (by optimizing  $\mathcal{L}_{\text{dur}} + \mathcal{L}_{\text{enc}} + \mathcal{L}_{\text{clean\_data}}$ ), where  $\mathcal{L}_{\text{dur}}$  and  $\mathcal{L}_{\text{enc}}$  are losses in the neural network  $E_\theta(\mathbf{c})$  adopted from Glow-TTS [27] and Grad-TTS [4].

## 3.3. Inference

The inference process is presented in Algorithm 1 and is applied to all systems except the system with blurring noise whose inference process applying Algorithm 2 follows [10]. Compared with Algorithm 1, Algorithm 2 is better for smooth/differentiated noising because it corrects restoration errors using a first-order approximation of the smooth degradation process [10].  $\text{Corrupt}(E_\theta(\mathbf{c}), N)$  in Algorithm 2 is a function that provides fully corrupted data / initial sample to start inference.

---

### Algorithm 1: First Sampling Method

---

**Input:** Input text  $\mathbf{c}$   
 $\mathbf{X}_N \leftarrow \text{Corrupt}(E_\theta(\mathbf{c}), N)$ ;  
**for**  $n = N, N - 1, \dots, 1$  **do**  
     $\hat{\mathbf{X}}_0 \leftarrow R_\theta(\mathbf{X}_n, E_\theta(\mathbf{c}))$ ;  
     $\mathbf{X}_{n-1} \leftarrow \text{noising}(\hat{\mathbf{X}}_0, E_\theta(\mathbf{c}), n - 1)$ ;  
**Return**  $\mathbf{X}_0$

---



---

### Algorithm 2: Second Sampling Method

---

**Input:** Input text  $\mathbf{c}$   
 $\mathbf{X}_N \leftarrow \text{Corrupt}(E_\theta(\mathbf{c}), N)$ ;  
**for**  $n = N, N - 1, \dots, 1$  **do**  
     $\hat{\mathbf{X}}_0 \leftarrow R_\theta(\mathbf{X}_n, E_\theta(\mathbf{c}))$ ;  
     $\mathbf{X}_{n-1} = \mathbf{X}_n - \text{noising}(\hat{\mathbf{X}}_0, E_\theta(\mathbf{c}), n)$ ;  
     $\quad + \text{noising}(\hat{\mathbf{X}}_0, E_\theta(\mathbf{c}), n - 1)$ ;

---

## 4. EXPERIMENTS

The dataset used in this work is LJSpeech [29] which contains approximately 13,100 clips totaling 24 hours of American English female voice recordings sampled at 22.05kHz. We follow [4] and split the data into training (around 12,000 clips), validation (around 100 clips), test (around 500 clips) set. 30 randomly selected texts from test set are used for Mean Opinion Score (MOS) evaluation.

Our work follows the pipeline of Grad-TTS, which includes an encoder  $E_\theta(\mathbf{c})$ , a decoder  $R_\theta(\cdot, E_\theta(\mathbf{c}))$ , and a fixed HIFI-GAN [30] vocoder. Each of our systems is trained by using an encoder  $E_\theta(\mathbf{c})$  checkpoint from Grad-TTS, and training the decoder from scratch for 500 epochs with a frozen encoder. Then, the encoder is unfrozen and trained for another 100 epochs. The baseline model uses a checkpoint provided by Grad-TTS with either 5 or 10 inference steps [7]. Specifically, Grad-TTS is a score-based model which learns scores and integrates the score over a discretized SDE for inference. On the other hand, our implemented models are non-score

based, where clean data is predicted (Section 3.2) and used to refine the result (Algorithm 1, Algorithm 2) in each iterative step during inference (Section 3.3). The training and inference conditions are consistent in our models, meaning there is no discretization error.

#### 4.1. Objective evaluation

MCD,  $\log f_0$ , UTMOSv2 [31] are used as objective metrics. The objective evaluation is performed on the whole test set. The evaluation result is presented in table 1, where  $\sigma$  represents the standard deviation of additive Gaussian noise (4) and multiplicative Gaussian noise (5) during noising process. The initial evaluation is conducted on results produced using 10 inference steps.

**Table 1:** Objective evaluation for the whole test set (10 steps)

Systems	MCD ↓	$\log f_0$ ↓	UTMOSv2 ↑
Baseline	$5.71 \pm 0.46$	$0.33 \pm 0.08$	$4.03 \pm 0.22$
Grad-TTS-DT	$5.53 \pm 0.52$	$0.31 \pm 0.08$	$3.95 \pm 0.21$
RFAG ( $\sigma = 0.2$ )	$5.55 \pm 0.55$	$0.32 \pm 0.08$	$3.96 \pm 0.23$
RFAG ( $\sigma = 0.4$ )	$5.43 \pm 0.56$	$0.32 \pm 0.09$	$3.86 \pm 0.22$
RFAG ( $\sigma = 0.6$ )	$5.49 \pm 0.52$	$0.31 \pm 0.08$	$3.86 \pm 0.22$
RFMG ( $\sigma = 0.2$ )	$5.65 \pm 0.51$	$0.32 \pm 0.07$	$3.92 \pm 0.21$
RFMG ( $\sigma = 0.4$ )	$5.63 \pm 0.50$	$0.32 \pm 0.07$	$3.94 \pm 0.23$
RFMG ( $\sigma = 0.6$ )	$5.62 \pm 0.50$	$0.32 \pm 0.08$	$3.87 \pm 0.22$
Blurring	$5.52 \pm 0.54$	$0.30 \pm 0.08$	$3.71 \pm 0.25$
Mixture	$5.58 \pm 0.50$	$0.30 \pm 0.08$	$3.87 \pm 0.22$

As is shown in table 1, discretized baseline (Grad-TTS-DT) performs closely to the baseline, which shows fully discrete process can achieve comparable results to its partially discretized counterparts. Also, there is no significant difference between performance of DM style additive Gaussian noise (in Grad-TTS-DT) and rectified flow style additive Gaussian noise (in RFAG) in discrete-time space. Multiplicative Gaussian noise (in RFMG) performs similarly to additive Gaussian noise (in RFAG) in a discretized rectified flow data trajectory (5) (4). The Mixture system and Blurring system achieved the best  $\log f_0$  result. The Mixture system’s UTMOSv2 score is similar to RFAG’s and RFMG’s in certain  $\sigma$  range, it is likely that Mixture system can perform better with different ratio between blurring and noise (8) [11]. Blurring’s UTMOSv2 is the lowest, likely due to blurring smooths critical acoustic details without introducing sufficient randomness or variability, which limits its ability to improve generalization in speech naturalness prediction.

**Table 2:** Objective evaluation for the whole test set (5 steps)

Systems	MCD ↓	$\log f_0$ ↓	UTMOSv2 ↑
Baseline	$5.69 \pm 0.53$	$0.34 \pm 0.08$	$3.98 \pm 0.22$
Grad-TTS-DT	$5.52 \pm 0.54$	$0.31 \pm 0.08$	$3.93 \pm 0.22$
RFAG ( $\sigma = 0.2$ )	$5.57 \pm 0.53$	$0.32 \pm 0.08$	$4.01 \pm 0.20$
RFAG ( $\sigma = 0.4$ )	$5.44 \pm 0.52$	$0.33 \pm 0.09$	$3.88 \pm 0.24$
RFAG ( $\sigma = 0.6$ )	$5.51 \pm 0.53$	$0.32 \pm 0.08$	$3.84 \pm 0.23$
RFMG ( $\sigma = 0.2$ )	$5.56 \pm 0.53$	$0.33 \pm 0.08$	$3.87 \pm 0.22$
RFMG ( $\sigma = 0.4$ )	$5.55 \pm 0.52$	$0.33 \pm 0.08$	$3.91 \pm 0.24$
RFMG ( $\sigma = 0.6$ )	$5.55 \pm 0.52$	$0.33 \pm 0.08$	$3.86 \pm 0.23$
Blurring	$5.53 \pm 0.56$	$0.31 \pm 0.08$	$3.66 \pm 0.24$
Mixture	$5.52 \pm 0.53$	$0.31 \pm 0.08$	$3.87 \pm 0.22$

Objective evaluation on results from 5 steps inference is performed to check how our systems generalize to fewer time steps.

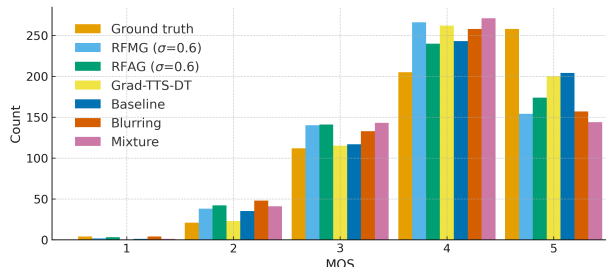
Our systems have better MCD and  $\log f_0$ . In the UTMOSv2 evaluation, all systems performed similarly to the baseline and shows good generalization, except the Blurring system which might lack some randomness.

#### 4.2. Subjective evaluation

The subjective evaluation is carried out on the Amazon Mechanical Turk (AMT) platform. 30 randomly selected texts from test set are used throughout the subjective evaluation. Each system is evaluated on 30 distinct speech samples, each rated by 20 native speakers. The ground truth result is generated by the fixed HIFI-GAN vocoder.

**Table 3:** Subjective scores for 30 randomly selected speech

Systems	MOS ↑
Ground truth	4.15
Baseline	4.02
Grad-TTS-DT	4.07
RFAG ( $\sigma = 0.6$ )	3.90
RFMG ( $\sigma = 0.6$ )	3.89
Blurring	3.86
Mixture	3.86



**Fig. 1:** Detailed breakdown of MOS score counts

All our systems achieved similar MOS scores to the baseline. It is also shown by Fig. 1 that, aside from Ground truth system, the MOS results are distributed similarly across the different MOS levels. In MOS evaluation, Grad-TTS-DT slightly outperformed the baseline, demonstrating the effectiveness of a fully discrete process compared with the partially discretized process. In addition, the encoder used in our systems is suboptimal and we believe that with tuning most of our systems could perform better.

## 5. CONCLUSION

Due to the difficulty of implementing more complex noising processes in continuous-time diffusion-like models and the inconsistency between training and inference in these models, this work proposes discrete-time diffusion-like models. This work presented discrete-time diffusion-like models with four different noising processes. This is the first work to implement multiplicative Gaussian noise in a diffusion-like model, investigate blurring and a combination of blurring with Gaussian noise for speech synthesis using a diffusion-like model, and examine popular continuous-time diffusion-like models within a consistent, fully discrete-time framework. The results demonstrate that discrete-time diffusion models can perform comparably well to a popular continuous diffusion model. The performance of blurring suggests that randomness might still be needed for such discrete-time models. Additionally, noising methods such as multiplicative Gaussian noise, a mixture of blurring and additive Gaussian noise, can have a performance similar to that of the widely used additive Gaussian noise in speech generation tasks.

## 6. REFERENCES

- [1] Jonathan Ho, Ajay Jain, and Pieter Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 6840–6851.
- [2] Xiang Li, Yizhe Qian, Jingjing Liu, Graham Neubig, and Sean Welleck, “Diffusion-lm improves controllable text generation,” *arXiv preprint arXiv:2205.14217*, 2022.
- [3] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro, “Diffwave: A versatile diffusion model for audio synthesis,” in *International Conference on Learning Representations*, 2021.
- [4] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole, “Score-based generative modeling through stochastic differential equations,” in *International Conference on Learning Representations (ICLR)*, 2021.
- [5] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le, “Flow matching for generative modeling,” *arXiv preprint arXiv:2210.02747*, 2022.
- [6] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat, “Flow matching guide and code,” 2024.
- [7] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, “Grad-tts: A diffusion probabilistic model for text-to-speech,” in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8599–8608.
- [8] Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter, “Matcha-tts: A fast tts architecture with conditional flow matching,” 2024.
- [9] Severi Rissanen, Markus Heinonen, and Arno Solin, “Generative modelling with inverse heat dissipation,” 2023.
- [10] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein, “Cold diffusion: Inverting arbitrary image transforms without noise,” 2022.
- [11] Hao-Chien Hsueh, Wen-Hsiao Peng, and Ching-Chun Huang, “Warm diffusion: Recipe for blur-noise mixture diffusion models,” in *The Thirteenth International Conference on Learning Representations*, 2025.
- [12] José M. Bioucas-Dias and Mário AT Figueiredo, “Multiplicative noise removal using variable splitting and constrained optimization,” *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1720–1730, 2010.
- [13] Bernt Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, Springer, 6th edition, 2003.
- [14] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho, “Variational diffusion models,” in *Advances in Neural Information Processing Systems*, 2021.
- [15] Tim Salimans and Jonathan Ho, “Progressive distillation for fast sampling of diffusion models,” in *International Conference on Learning Representations*, 2022.
- [16] Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio, “Improving and generalizing flow-based generative models with minibatch optimal transport,” *Transactions on Machine Learning Research*, 2024, Expert Certification.
- [17] Xingchao Liu, Chengyue Gong, and Qiang Liu, “Flow straight and fast: Learning to generate and transfer data with rectified flow,” 2022.
- [18] Yiwei Guo, Chenpeng Du, Ziyang Ma, Xie Chen, and Kai Yu, “Voiceflow: Efficient text-to-speech with rectified flow matching,” in *ICASSP 2024 – 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11121–11125.
- [19] Ryan Prenger, Rafael Valle, and Bryan Catanzaro, “Waveglow: A flow-based generative network for speech synthesis,” 2018.
- [20] Rafael Valle, Kevin Shih, Ryan Prenger, and Bryan Catanzaro, “Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis,” 2020.
- [21] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” 2015.
- [22] Jiaming Song, Chenlin Meng, and Stefano Ermon, “Denoising diffusion implicit models,” 2022.
- [23] Soo-Young T Jeon, Daphne Maurer, and Terri L Lewis, “Developmental mechanisms underlying improved contrast thresholds for discriminations of orientation signals embedded in noise,” *Frontiers in Psychology*, vol. 5, pp. 977, 2014.
- [24] Chiara F Angeloni, Diego Lozano-Soldevilla, and Neil C Rabinowitz, “Dynamics of cortical contrast adaptation predict perception,” *Nature Communications*, vol. 14, no. 1, pp. 4879, 2023.
- [25] Guillem Riutort-Mayol, Virgilio Gómez-Rubio, Álvaro Marqués-Mateu, José Luis Lerma, and Antonio López-Quílez, “A bayesian multilevel random-effects model for estimating noise in image sensors,” *arXiv preprint arXiv:2004.11849*, 2020.
- [26] Emiel Hoogeboom and Tim Salimans, “Blurring diffusion models,” *arXiv preprint arXiv:2209.05557*, 2022.
- [27] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon, “Glow-tts: A generative flow for text-to-speech via monotonic alignment search,” 2020.
- [28] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever, “Consistency models,” 2023.
- [29] Keith Ito and Linda Johnson, “The lj speech dataset,” <https://keithito.com/LJ-Speech-Dataset/>, 2017.
- [30] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *Advances in Neural Information Processing Systems*, 2020, vol. 33 of *NeurIPS*, *arXiv preprint arXiv:2010.05646*.
- [31] Kaito Baba, Wataru Nakata, Yuki Saito, and Hiroshi Saruwatari, “The t05 system for the VoiceMOS Challenge 2024: Transfer learning from deep image classifier to naturalness MOS prediction of high-quality synthetic speech,” in *IEEE Spoken Language Technology Workshop (SLT)*, 2024.