INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Variance-Preserving-Based Interpolation Diffusion Models for Speech Enhancement

Zilu Guo[1], *Jun Du*[1,*], *Chin-Hui Lee*[2], *Yu Gao*[3], *Wenbin Zhang*[3]

[1]University of Science and Technology of China, Hefei 230027, China
[2]Georgia Institute of Technology, Atlanta, GA. 30332-0250, USA
[3]AI Innovation Center, Midea Group (Shanghai) Co.,Ltd., Shanghai 201702, China

guozl@mail.ustc.edu.cn, ✉jundu@ustc.edu.cn, chl@ece.gatech.edu,
{gaoyu11, zhangwb87}@midea.com

Abstract

The goal of this study is to implement diffusion models for speech enhancement (SE). The first step is to emphasize the theoretical foundation of variance-preserving (VP)-based interpolation diffusion under continuous conditions. Subsequently, we present a more concise framework that encapsulates both the VP- and variance-exploding (VE)-based interpolation diffusion methods. We demonstrate that these two methods are special cases of the proposed framework. Additionally, we provide a practical example of VP-based interpolation diffusion for the SE task. To improve performance and ease model training, we analyze the common difficulties encountered in diffusion models and suggest amenable hyper-parameters. Finally, we evaluate our model against several methods using a public benchmark to showcase the effectiveness of our approach.

Index Terms: speech enhancement, speech denoising, diffusion models, score-based, interpolation diffusion

1. Introduction

Speech enhancement (SE) [1] has been the subject of research for several decades with the goal of diminishing or even eliminating the noise in a noisy speech while minimizing the distortion to speech quality. In recent years, SE has been approached as a supervised regressive task with the assistance of deep learning. The early attempts to deploy deep learning for SE involve employing off-the-shelf models to predict the targets that are typically utilized in traditional approaches. However, it is important to note that these approaches are sub-optimal since they are not able to obtain the clean phase of the speech. Common targets for these methods include the magnitude of the spectrogram [2], the log magnitude (Mapping) [3], the ideal ratio mask (IRM)[4], the spectral magnitude mask (SMM) [2], and etc. The real and imaginary parts of the spectrogram are utilized directly as the target to obtain the clean phase afterward [5, 6, 7]. Meanwhile, another main-stream SE method seeks to predict clean waveforms in an end-to-end (E2E) manner [8], rather than the spectrum. In addition to the regressive methods, some researchers have utilized generative models for the SE, such as, VAE [9, 10], GAN [11, 12, 13], flow [14, 15], and etc.

Recently, Diffusion models have been successful in various generative tasks, including image generation [16, 17, 18], image editing [19], speech synthesis [20] and etc. Diffusion models involve two processes, i.e., diffusion (or forward) and reverse (or backward) processes. Another approach in the discrete domain is the score-based model [18]. Both the diffusion [16] and the score-based [18, 21] model are unified in [17, 22], and generalized to the continuous time domain, endowing the model with more capacity. Moreover, the authors in [17] have classified the diffusion models into two types: the variance-preserving(VP)-based [16] and the variance-exploding(VE)-based [18] method grounded on their intuitive properties. The VE-based approach gradually increases the variance over time while keeping the clean component unaltered. In contrast, VP-based diffusion attempts to preserve the amplitude with fewer fluctuations. Besides their applications in the image processing field, diffusion models have also been explored for SE tasks.

In their work, CDiffSE [23] proposes that a degraded signal is composed of three components, i.e, the clean signal , the noisy and the Gaussian noise. They suggest that the mean of the degraded signal in the diffusion process is obtained through a linear combination of the clean and noisy speech, specifically the linear interpolation of the two. They then apply this interpolation method to Diffwave [20], a model for speech synthesis, for the SE task. SRTNET [24] follows a similar approach, utilizing the interpolation diffusion to estimate the distortion between the clean speech and an enhanced speech predicted by a pre-trained discriminative model. However, the two-stage models commonly have higher computational overhead. Apart from the discrete diffusion models mentioned above, there are also several interpolation-based methods for SE under continuous conditions. In [25, 26], the authors formulate the theoretical foundation for the VE-based interpolation diffusion model (VEIDM) in the continuous time field and generalize it to the continuous time system. However, they leave the VP-based one untouched which showcases better performance in some tasks than the VE-based diffusion. In this paper, we attempt to apply linear interpolation to VP-based diffusion.

The rest of the paper is organized as follows. The proposed method is introduced in Sec. 2. Specifically, we accentuate the proposed signal model, training method, and sampling algorithms in this section. Experiments settings, results, and analyses are presented in Sec. 3. we draw conclusions in Sec. 4.

2. The proposed method

In the forward process of the vanilla VP-based diffusion model, a clean signal (such as an image or speech) is gradually degraded by adding Gaussian noise in steps until it reaches an approximate Gaussian noise level. In this process, the mean of the state t is an affine function of the clean signal. However, in the VP-based interpolation diffusion, the mean is replaced with a linear interpolation of the clean and the noisy.

2.1. The VP-based interpolation diffusion model (VPIDM)

For tasks of SE, image editing, voice conversion and etc, there is an existing condition that holds copious information about the target. In the case of SE, for instance, the off-the-shelf noisy

[*] corresponding author

10.21437/Interspeech.2023-1265

Algorithm 1 Training stage

```
Batch size is $B$, number of training iterations $N_{\text{iter}}$
**for** $i = 1$ to $N_{\text{iter}}$ **do**
    Sample a batch of clean and noisy speech pairs $[(\boldsymbol{x}_0^1, \boldsymbol{y}^1), \dots, (\boldsymbol{x}_0^b, \boldsymbol{y}^b), ..., (\boldsymbol{x}_0^B, \boldsymbol{y}^B)]$, $(\boldsymbol{x}_0^b, \boldsymbol{y}^b)) \sim p(\boldsymbol{x_0}, \boldsymbol{y})$
    Sample a batch of the time indexes $[t^1, t^2, ..., t^B]$, $t^b \sim \mathcal{U}(\epsilon, 1)$
    Sample a batch of Gaussian noises $[\boldsymbol{z}^1, \boldsymbol{z}^2, ..., \boldsymbol{z}^B]$, $\boldsymbol{z}^b \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \boldsymbol{I})$
    Get a batch of $[\boldsymbol{x}^1(t^1), \boldsymbol{x}^2(t^2), ..., \boldsymbol{x}^B(t^2)]$, $\boldsymbol{x}^b(t^b)$ is sampled from Eq.(1)
    Input the $[\boldsymbol{x}^1(t^1), \boldsymbol{x}^2(t^2), ..., \boldsymbol{x}^B(t^2)]$ to the neural network
    Get the output $[\boldsymbol{\theta}(\boldsymbol{x}^1(t^1)), \boldsymbol{\theta}(\boldsymbol{x}^2(t^2)), \dots, \boldsymbol{\theta}(\boldsymbol{x}^B(t^B))]$
    Compute the loss $\mathcal{L} = \{\Sigma_{b=1}^B ||G(t^b)\boldsymbol{\theta}(\boldsymbol{x}(t^b) + \boldsymbol{z}||^2\}/B$ from Eq.(9)
    Update the parameters
**end for**
```

Algorithm 2 Sampling (enhancing) stage

```
Suppose the number of sample steps is $K$, $t_k = \frac{(1-\epsilon)}{K}k + \epsilon$
Sample $\boldsymbol{x}_K = \boldsymbol{x}(t_K) = \boldsymbol{x}(1) = \alpha_1 \boldsymbol{y} + G(1)\boldsymbol{z}$
**for** $k = K-1, K-2, \dots, 1$ **do**
    Input the $\boldsymbol{x}_{k+1}$, get the $\hat{\boldsymbol{\theta}}(\boldsymbol{x}_{k+1})$
    Sample a Gaussian noise $\boldsymbol{z}$, $\boldsymbol{z} \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \boldsymbol{I})$
    Compute the $\boldsymbol{x}_k$ from Eq.(17)
**end for**
Input the $\boldsymbol{x}_1$, get the $\hat{\boldsymbol{\theta}}(\boldsymbol{x}_1)$
$\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_1 - [f(\boldsymbol{x}_1, \boldsymbol{y}) - g_1^2 \hat{\boldsymbol{\theta}}(\boldsymbol{x}_1)]\Delta$
**return** $\hat{\boldsymbol{x}}_0$
```

speech could be implemented to guide the diffusion process of the clean. We refer to this approach as the VP-based interpolation diffusion model (VPIDM). The signal model for VPIDM is defined as

$$\boldsymbol{x}(t) = \alpha_t[\lambda_t \boldsymbol{x}_0 + (1-\lambda_t)\boldsymbol{y}] + \sqrt{1-\alpha_t^2}\boldsymbol{z} \tag{1}$$

where $x(t)$ is the degraded signal at t time index in the diffusion process, $\boldsymbol{x}_0$ is the clean signal, $\boldsymbol{y}$ is the noisy speech, $\boldsymbol{z}$ is the Gaussian noise sampled from normal distribution, α_t determines the diffusion process, λ_t is the slope of the interpolation process, α_t and λ_t are functions of t. The differential of $\boldsymbol{x}(t)$ is

$$d\boldsymbol{x}(t) = [\boldsymbol{x}(t)\ln^{'}(\alpha_t\lambda_t) - \boldsymbol{y}\alpha_t \ln^{'}\lambda_t]dt + \boldsymbol{\Sigma}_t d\boldsymbol{w} \tag{2}$$

where $\boldsymbol{w}$ is the stochastic process, $ln^{'}[\cdot]$ is the derivative of $ln[\cdot]$ with respect to t, $d[\cdot]$ is the operation of differential. When $\boldsymbol{\Sigma}_t$ is a diagonal matrix, suppose that $\boldsymbol{\Sigma}_t = g(t)\boldsymbol{I}$. From Eq.(5.53) in [27], we get

$$\frac{dG^2(t)}{dt} = 2G^2(t)\ln^{'}(\alpha_t\lambda_t) + g^2(t) \tag{3}$$

where $g(t)$ indicates the spread speed of the stochastic process in the derivative of $\boldsymbol{x}(t)$, $g(t) = \sqrt{-2G^2(t)\ln^{'}\lambda_t - 2\ln^{'}\alpha_t}$, $G(t)$ is the coefficient of the Gaussian noise in $\boldsymbol{x}(t)$, $G^2(t) = 1-\alpha_t^2$, $t \in (0,1]$, $\ln^{'}\alpha_t \leq 0$ and $\ln^{'}\lambda_t \leq 0$, which means λ_t and α_t are monotonous decrease functions. In this article, all constant superscripts represent powers, unless otherwise specified. When $t \to 0$, then $\lambda_t \to 1$,$\alpha_t \to 1$.

In principle, we hope $\lambda_1 \to 0$, which implies that the final state is a combination of the noisy signal and the Gaussian noise. Therefore, the larger $-\ln^{'}\lambda_t$ appears to be more favorable. However, empirical evidence suggests that $-\ln^{'}\lambda_t$ cannot be infinitely large. The reason behind this is that when we sample a clean and $-\ln^{'}\lambda_t$ is set to sufficiently large, the linear interpolation tends to change quickly from $\boldsymbol{y}$ to $\boldsymbol{x}_0$ over several steps, it is difficult for neural networks to capture.

In this paper, we adopt the similar α_t schedule of the VP-based diffusion in [17] for the SE, i.e., $\alpha_t = e^{-0.5\int_0^t \beta(\tau)d\tau}$, where $\beta(t) = (\beta_{\max} - \beta_{\min})t + \beta_{\min}$, $\beta_{\min}$ controls the slope of the clean scale when $t \to 0$, $(\beta_{\max} - \beta_{\min})$ controls the changing speed of $\boldsymbol{x}_t$ from the clean to the Gaussian.

$$g(t) = \sqrt{\beta(t) + 2\lambda(1 - e^{-\int_0^t \beta(\tau)d\tau})} \tag{4}$$

where $\lambda_t = e^{-\lambda t}$. From Eq.(2), the $d\boldsymbol{x}(t)$

$$d\boldsymbol{x}(t) = [-(0.5\beta(t) + \lambda)\boldsymbol{x}(t) + \lambda\alpha_t\boldsymbol{y}]dt + g(t)d\boldsymbol{w} \tag{5}$$

In the reverse process, the $\boldsymbol{x}(1)$ is sampled from the distribution $\mathcal{N}(\alpha_1\boldsymbol{y}, \sqrt{1-\alpha_1^2}\boldsymbol{I})$.

2.2. The loss function and the training stage

For unconditional diffusion, the neural network is trained for predicting $\boldsymbol{\nabla_x}\ln(p_t(\boldsymbol{x}))$. This is equivalent to optimizing the following cost function

$$\mathcal{L} = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}(t)}\{W \cdot ||\boldsymbol{\theta}(\boldsymbol{x}(t)) - \boldsymbol{\nabla_x}\ln(p_t(\boldsymbol{x}))||^2\} \tag{6}$$

where $\boldsymbol{\theta}(\boldsymbol{x}(t))$ is the output of the neural network. For interpolation-based diffusion,

$$p_t(\boldsymbol{x}) = p(\boldsymbol{x}(t)|\boldsymbol{x}_0, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{m}(\boldsymbol{x}_0, \boldsymbol{y}); G(t)\boldsymbol{I}) \tag{7}$$

where $p_t(\boldsymbol{x})$ is the conditional probability density function of $\boldsymbol{x}(t)$, $\boldsymbol{m}(\boldsymbol{x}_0, \boldsymbol{y}) = \alpha_t[\lambda_t\boldsymbol{x}_0 + (1-\lambda_t)\boldsymbol{y}]$ is the mean of $p_t(\boldsymbol{x})$.

$$\boldsymbol{\nabla_x}\ln(p_t(\boldsymbol{x})) = \boldsymbol{\nabla_x}[-\frac{||\boldsymbol{x}(t) - \boldsymbol{m}(\boldsymbol{x}_0, \boldsymbol{y})||^2}{2G^2(t)}] = -\frac{\boldsymbol{z}}{G(t)} \tag{8}$$

here $\boldsymbol{x}(t) - \boldsymbol{m}(\boldsymbol{x}_0, \boldsymbol{y}) = G(t)\boldsymbol{z}$. Then we get the loss function

$$\mathcal{L} = \mathbb{E}\{W \cdot ||\boldsymbol{\theta}(\boldsymbol{x}(t)) + \frac{\boldsymbol{z}}{G(t)})||^2\} = \mathbb{E}||G(t)\boldsymbol{\theta}(\boldsymbol{x}(t)) + \boldsymbol{z})||^2 \tag{9}$$

We follow the settings in [17, 18, 25], utilize the weighted loss, and set $W = G^2(t)$ for better performance.

The training algorithm is shown in Alg. 1, where ϵ represents the minimum sample time, where the superscript b in $[\cdot]^b$ denotes the b-th sample of a batch, batch size is B, $p(\boldsymbol{x_0}, \boldsymbol{y})$ denotes the joint probability density function of the clean and noisy pair, i.e., $\boldsymbol{x}_0, \boldsymbol{y}$.

2.3. The reverse process for sampling a clean

From [17, 28], the reverse process is also a diffusion process which can be represented as

$$d\boldsymbol{x}(t) = [f(\boldsymbol{x}(t), \boldsymbol{y}) - g(t)^2\boldsymbol{\theta}(\boldsymbol{x}(t))]dt + g(t)d\bar{\boldsymbol{w}} \tag{16}$$

Table 1: *Comparison of the VEIDM, VPIDM, and IDM.*

	The state equations		The stochastic differential equations	
VEIDM [25]	$\boldsymbol{x}(t) = \lambda_t \boldsymbol{x}_0 + (1-\lambda_t)\boldsymbol{y} + \sqrt{\ln\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)\frac{\sigma_{\min}^2((\sigma_{\max}/\sigma_{\min})^{2t} - e^{-2\lambda t})}{\lambda + \ln(\sigma_{\max}/\sigma_{\min})}}\boldsymbol{z}$	(10)	$d\boldsymbol{x}(t) = \lambda(\boldsymbol{y} - \boldsymbol{x}(t))dt + \sigma_{\min}(\frac{\sigma_{\max}}{\sigma_{\min}})^t \sqrt{2\ln\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)} d\boldsymbol{w}$	(11)
VPIDM	$\boldsymbol{x}(t) = \alpha_t[\lambda_t \boldsymbol{x}_0 + (1-\lambda_t)\boldsymbol{y}] + \sqrt{1-\alpha_t^2}\boldsymbol{z}$	(12)	$d\boldsymbol{x}(t) = [\boldsymbol{x}(t)\ln^{'}(\alpha_t\lambda_t) - \boldsymbol{y}\alpha_t \ln^{'}\lambda_t]dt + \boldsymbol{\Sigma}_t d\boldsymbol{w}$	(13)
IDM	$\boldsymbol{x}(t) = \alpha_t[\lambda_t \boldsymbol{x}_0 + (1-\lambda_t)\boldsymbol{y}] + G(t)\boldsymbol{z}$	(14)	$d\boldsymbol{x}(t) = \boldsymbol{x}(t)d\ln(\alpha_t\lambda_t) - \boldsymbol{y}\alpha_t d\ln\lambda_t + g(t)d\boldsymbol{w}$	(15)

where $f(\boldsymbol{x}(t), \boldsymbol{y}) = \boldsymbol{x}(t)\ln^{'}(\alpha_t\lambda_t) - \boldsymbol{y}\alpha_t \ln^{'}\lambda_t$ for the VPIDM and $\bar{\boldsymbol{w}}$ is another stochastic process but has the same distribution with $\boldsymbol{w}$. Typically, the continuous process is discretized in the sampling stage. $\Delta = \frac{1-\epsilon}{K}$ is set and let $\boldsymbol{x}_k = \boldsymbol{x}(\frac{k(1-\epsilon)}{K} + \epsilon)$ and $g_k = g(\frac{k(1-\epsilon)}{K} + \epsilon)\sqrt{\Delta}$. The sampling stage is elaborated in Alg. 2.

$$\boldsymbol{x}_{k-1} = \boldsymbol{x}_k - [f(\boldsymbol{x}_k, \boldsymbol{y}) - g_k^2 \boldsymbol{\theta}(\boldsymbol{x}_k)]\Delta + g_k \boldsymbol{z} \quad (17)$$

where the subscript k of $\boldsymbol{x}_k$ and g_k denote the discrete sampling time index, $\boldsymbol{x}_k$ and g_k represent the discrete samplings of $\boldsymbol{x}(t)$ and $g(t)$.

2.4. Comparison with the VEIDM

Furthermore, the proposed VPIDM and the VEIDM proposed in [25] can be concluded as

$$d\boldsymbol{x}(t) = \boldsymbol{x}(t)d\ln(\alpha_t\lambda_t) - \boldsymbol{y}\alpha_t d\ln\lambda_t + g(t)d\boldsymbol{w} \quad (18)$$

$$\boldsymbol{x}(t) = \alpha_t[\lambda_t \boldsymbol{x}_0 + (1-\lambda_t)\boldsymbol{y}] + G(t)\boldsymbol{z} \quad (19)$$

The relation between G and g is constrained by Eq.(3) which is referred to as the interpolation diffusion model (IDM). When $G^2(t) = 1 - \alpha_t^2$, the interpolation diffusion becomes a VP-based method. In the case of VE-based interpolation diffusion, α_t in Eq.(18) and (19) is constant 1. Substitute α_t with constant 1 and solve the ordinary differential equation in Eq.(3), we get $G(t)$

$$G^2(t) = \lambda_t^2 G(0)^2 + \lambda_t^2 \int_0^t g^2(\tau)/\lambda_\tau^2 d\tau \quad (20)$$

In [25], The interpolation coefficient λ_t is defined as $e^{-\lambda t}$. It can be verified that the VEIDM, which use $G(t)$ from Eq.(10), and the $g(t)$ from Eq.(11), satisfies Eq.(20) as a special case. Additionally, a comparison is made between the VEIDM, the proposed VPIDM, and IDM in Tab.1.

In the reverse process, the initial sample of the VEIDM in [25] is taken as $\boldsymbol{y} + G(1)\boldsymbol{z}$, rather than the ground truth $\lambda_1\boldsymbol{x}_0 + (1-\lambda_1)\boldsymbol{y} + G(1)\boldsymbol{z}$ because obtaining the clean is not possible at this stage. However, this can result in damage to the enhanced speech. To quantify this effect, we define the initial error (*IE*)

$$\textit{IE}_{\text{VEIDM}} = [\boldsymbol{y} + G(1)\boldsymbol{z}] - [\lambda_1\boldsymbol{x}_0 + (1-\lambda_1)\boldsymbol{y} + G(1)\boldsymbol{z}] \quad (21)$$

$$= \lambda_1(\boldsymbol{y} - \boldsymbol{x}_0) \quad (22)$$

Whereas, the reverse of the proposed start from $\alpha_1\boldsymbol{y} + G(1)\boldsymbol{z}$, and the truth is $\alpha_1(\lambda_1\boldsymbol{x}_0 + (1-\lambda_1)\boldsymbol{y}) + G(1)\boldsymbol{z}$. The *IE* is

$$\textit{IE}_{\text{VPIDM}} = \alpha_1\lambda_1(\boldsymbol{y} - \boldsymbol{x}_0) \quad (23)$$

When the same λ_t is utilized in the Sgmse+ and VPIDM, $\textit{IE}_{\text{VPIDM}} \ll \textit{IE}_{\text{Sgmse+}}$, where $\alpha_1 \to 0$. That is, the VPIDM has a smaller *IE* than the VEIDM.

3. Experiments

3.1. Training settings

We conduct our experiments on the publicly available benchmark, i.e., VoiceBank-DEMAND(VBD) [29] Metrics in [30, 31, 32], i.e., SI-SDR, SI-SIR, SI-SAR, PESQ, CSIG, CBAK, COVL, are adopted to compare the performance to other state-of-the-art methods. 25 speech clips from the test dataset are selected randomly as the validation dataset. We train the neural network for 120 epochs. The best checkpoint is saved when PESQ is in its optimal state during the validation phase.

We use the neural network proposed in [26] as our backbone model, which is originally introduced in [17] for the image generation task. We treat the complex spectrum as a real-valued tensor to circumvent complex-valued computation where the real and imaginary parts of the complex are represented as two channels of the tensor. The tensor is scaled to ensure that its amplitude approximately falls within the range of -1 to 1 before being fed into the neural network. We follow the scaling function described in [25] where given a complex-valued spectrum $\boldsymbol{x}(t) = |\boldsymbol{x}(t)|e^{\angle\boldsymbol{x}(t)}$, the scaled $[\boldsymbol{x}(t)]^s = a|\boldsymbol{x}(t)|^c e^{\angle\boldsymbol{x}(t)}$, here a, c are two hyper-parameters, $[\cdot]^s$ means the operation of the scaling function. We find that $a50^c \approx 1, 0 < c \leq 1$ is of avail for the performance and c can not be set too small, as it compresses the dynamic range of the signal drastically and makes it more difficult to learn the structure of clean spectrum. In this paper, we empirically set $a = 0.15$ and $c = 0.5$, $\beta_{\text{min}} = 0.1$, $\beta_{\text{max}} = 2$ and $\lambda_t = e^{-\lambda t}$, $\lambda = 1.5$, $G(t) = \sqrt{1-\alpha_t^2}$.

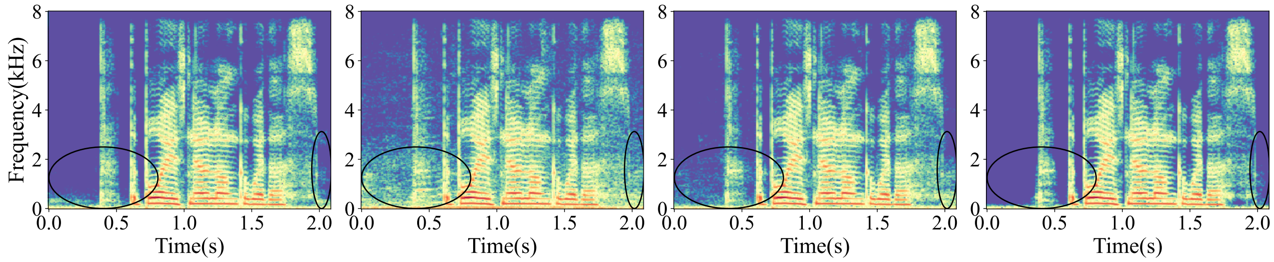
3.2. Results and analyses

For the continuum diffusion model, we typically sample t from the uniform distribution $\mathcal{U}(\epsilon, 1)$, ϵ denotes the time index of the first state after the clean. Ideally, we want to set ϵ as small as possible. However, when ϵ is too small, it can lead to difficulties in achieving convergence during training and may result in unstable fluctuations in the optimization process.

Table 2: *Illustration of several metrics over different settings for VPIDM on the VBD dataset.*

Settings	PESQ ↑	SISDR ↑	SISIR ↑	SISDR ↑
$\epsilon = 1 \cdot 10^{-2}$	3.01	18.3	**31.9**	18.6
$\epsilon = 3 \cdot 10^{-2}$	3.02	**18.9**	30.1	19.4
$\epsilon = 4 \cdot 10^{-2}$	**3.13**	18.7	28.6	19.3
$\epsilon = 5 \cdot 10^{-2}$	2.86	18.6	27.6	19.4
$\epsilon = 6 \cdot 10^{-2}$	2.95	18.7	26.6	**19.7**
$\epsilon = 7 \cdot 10^{-2}$	1.77	16.1	24.5	16.9

In Fig.2, we vary the value of ϵ for the model and observe the training loss with training steps. The results show that when the ϵ is too small, the training loss exhibits lots of fluctuations and becomes difficult to converge. In our view, the reason is



a) The clean magnitude b) The noisy magnitude c) The enhanced (Sgmse+). d) The enhanced (VPIDM).

Figure 1: *The log spectrogram of the STFT magnitudes. a) the clean $\boldsymbol{x}_0$ spectrogram; b) the noisy $\boldsymbol{y}$ spectrogram; c) the spectrogram of the estimated clean from the Sgmse+ (2); d) the spectrogram of the estimated clean from the VPIDM.*

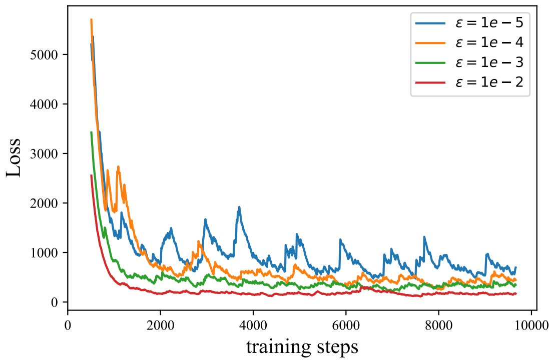


Figure 2: *The training loss when $t \sim \mathcal{U}(\epsilon, 1)$.*

that, when ϵ is too small, the model is required to estimate the target in a wider range of SNR conditions. We conduct experiments to check the model's ability to predict the target in low SNR conditions using only one state ($t = \epsilon$) as the input for the model. However, the model can not predict the target well when $\epsilon \leq 10^{-2}$. That is, when $t \to 0$, $\alpha_t \to 1$, $\lambda_t \to 1$, then $\boldsymbol{x}(t) \approx \boldsymbol{x}_0 + \sqrt{1-\alpha_t^2}\boldsymbol{z}$. From the perspective of Gaussian noise, the SNR $\approx 10\log_{10}(\frac{1-\alpha_t^2}{1}) \approx 10\log_{10}(\beta_{\min} t)$. Therefore, when $t = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$, the model predicts the target at about -60dB, -50dB, -40dB, -30dB SNR, respectively. Based on these results, we choose $\epsilon \geq 10^{-2}$. We treat the ϵ as the minimum resolution of the model, so we set the number of sample steps in the reverse process to $K \approx [\frac{1}{\epsilon}]$. As a result, for $\epsilon = [10^{-2}, 3\cdot10^{-2}, 4\cdot10^{-2}, 5\cdot10^{-2}, 6\cdot10^{-2}, 10^{-1}]$, the number of sampling steps are $100, 30, 25, 20, 15, 10$, respectively. According to Tab.2, the best PESQ is achieved when $\epsilon = 4 \cdot 10^{-2}$, which corresponds to 25 sample steps. The model demonstrates strong performance across various evaluation metrics when the number of sample steps exceeds 15, but when the sample steps are less than 15, the performance drops significantly.

In Tab.3, we compare our model to several SOTA methods. Sgmse+ (1) denotes the vanilla model without the corrector, while Sgmse+ (2) includes the corrector. The proposed model achieves the best CSIG, indicating the least speech distortion. This demonstrates that our generative model has learned the closest clean distribution to the ground truth that can preserve clean speech best. Sgmse+ (1) in [26] is the VE-based interpolation diffusion with 30 sampling steps. The proposed method with fewer steps, i.e., 25 steps ($\epsilon = 4 \cdot 10^{-2}$), achieves 0.3 PESQ increment over Sgmse+ (1). To further improve the performance, Sgmse+ (2) in [26] implements a corrector which requires 60 total steps. The proposed requires less than half steps of the Sgmse+ (2) and obtains a 0.2 PESQ improvement. In fact, the scaling of the signal, i.e., α_t, on the right side of Eq.(1) is a type of data augmentation that is beneficial for the model to learn the intrinsic structure of clean speech. However, as shown in Fig.1, Sgmse+ (2) causes the neural network to consider some noises in $\boldsymbol{y}$ as the clean when starting from $\boldsymbol{y} + g(1)\boldsymbol{z}$ in the sampling stage. Therefore, some noises in the noisy speech are not effaced thoroughly. From the two black ellipses in Fig.1, we can see that the Sgmse+ (2) has residual noise in $0 - 0.6$(s) and $2 - 2.1$(s) time intervals.

Table 3: *The proposed method versus some SOTA methods with respect to different metrics.*

Model	PESQ ↑	CSIG ↑	CBAK ↑	COVL ↑
noisy	1.97	3.35	2.44	2.64
PFP [33]	**3.15**	4.18	**3.60**	3.67
MetricGAN [12]	2.86	3.99	3.18	3.42
MetricGAN+ [13]	3.15	4.14	3.16	3.64
CDiffuSE [23]	2.52	3.72	2.91	3.10
SRTNet [24]	2.69	4.12	3.19	3.39
CDSE [34]	2.77	3.91	3.32	3.33
Sgmse+ (1) [26]	2.80	4.10	3.24	3.44
Sgmse+ (2) [26]	2.93	4.12	3.37	3.51
VPIDM	3.13	**4.63**	3.41	**3.94**

4. Conclusions

In this paper, we present the VP-based interpolation diffusion model in a continuous time system and summarize the VE- and VP-based interpolation diffusion models into a more concise framework called the IDM. The VE- and VP-based interpolation diffusion models serve as examples of the IDM. While we only apply the VPIDM to the SE task to showcase our proposed method, it is worth noting that the approach is a general method that can be used for other tasks as well.

5. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62171427. We also thank Midea Group Co., Ltd for funding this work

6. References

[1] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[2] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[3] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2014.

[4] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7092–7096.

[5] D. S. Williamson, Y. Wang, and D. Wang, "Complex Ratio Masking for Monaural Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 483–492, Mar. 2016.

[6] K. Tan and D. Wang, "Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6865–6869.

[7] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex u-net," in *International Conference on Learning Representations*, 2018.

[8] S.-W. Fu, T.-W. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1570–1584, 2018.

[9] H. Fang, G. Carbajal, S. Wermter, and T. Gerkmann, "Variational autoencoder for speech enhancement with a noise-aware encoder," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 676–680.

[10] Y. Bando, K. Sekiguchi, and K. Yoshii, "Adaptive Neural Speech Enhancement with a Denoising Variational Autoencoder," in *Proc. Interspeech 2020*, 2020, pp. 2437–2441.

[11] S. Pascual, A. Bonafonte, and J. Serrà, "Segan: Speech enhancement generative adversarial network," in *Proc. Interspeech 2017*, 2017, pp. 3642–3646.

[12] S.-W. Fu, C.-F. Liao, Y. Tsao, and S.-D. Lin, "Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2031–2041.

[13] S.-W. Fu, C. Yu, T.-A. Hsieh, P. Plantinga, M. Ravanelli, X. Lu, and Y. Tsao, "MetricGAN+: An Improved Version of MetricGAN for Speech Enhancement," in *Proc. Interspeech 2021*, 2021, pp. 201–205.

[14] A. A. Nugraha, K. Sekiguchi, and K. Yoshii, "A flow-based deep latent variable model for speech spectrogram modeling and enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1104–1117, 2020.

[15] M. Strauss and B. Edler, "A flow-based neural network for time domain speech enhancement," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5754–5758.

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

[17] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

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

[19] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani, "Imagic: Text-based real image editing with diffusion models," in *Conference on Computer Vision and Pattern Recognition 2023*, 2023.

[20] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[21] D. P. Kingma and Y. Cun, "Regularized estimation of image statistics by score matching," in *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[22] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," *Advances in neural information processing systems*, vol. 34, pp. 21 696–21 707, 2021.

[23] Y.-J. Lu, Z.-Q. Wang, S. Watanabe, A. Richard, C. Yu, and Y. Tsao, "Conditional diffusion probabilistic model for speech enhancement," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7402–7406.

[24] Z. Qiu, M. Fu, Y. Yu, L. Yin, F. Sun, and H. Huang, "Srtnet: Time domain speech enhancement via stochastic refinement," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[25] S. Welker, J. Richter, and T. Gerkmann, "Speech enhancement with score-based generative models in the complex STFT domain," in *Proc. Interspeech 2022*, 2022, pp. 2928–2932.

[26] J. Richter, S. Welker, J.-M. Lemercier, B. Lay, and T. Gerkmann, "Speech enhancement and dereverberation with diffusion-based generative models," *arXiv preprint arXiv:2208.05830*, 2022.

[27] S. Särkkä and A. Solin, *Applied Stochastic Differential Equations*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

[28] B. D. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.

[29] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating rnn-based speech enhancement methods for noise-robust text-to-speech." in *SSW*, 2016, pp. 146–152.

[30] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.

[31] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749–752.

[32] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on audio, speech, and language processing*, vol. 16, no. 1, pp. 229–238, 2007.

[33] T.-A. Hsieh, C. Yu, S.-W. Fu, X. Lu, and Y. Tsao, "Improving Perceptual Quality by Phone-Fortified Perceptual Loss Using Wasserstein Distance for Speech Enhancement," in *Proc. Interspeech 2021*, 2021, pp. 196–200.

[34] H. Yen, F. G. Germain, G. Wichern, and J. L. Roux, "Cold diffusion for speech enhancement," *arXiv preprint arXiv:2211.02527*, 2022.