A Study on Speech Enhancement Based on Diffusion Probabilistic Model

Yen-Ju Lu∗, Yu Tsao∗ and Shinji Watanabe†
∗ Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
† Language Technology Institute, Carnegie Mellon University, Pittsburgh, PA, United States
E-mail: {neil.lu, yu.tsao}@citi.sinica.edu.tw
E-mail: shinjiw@cmu.edu

Abstract—Diffusion probabilistic models have demonstrated an outstanding capability to model natural images and raw audio waveforms through a paired diffusion and reverse processes. The unique property of the reverse process (namely, eliminating non-target signals from the Gaussian noise and noisy signals) could be utilized to restore clean signals. Based on this property, we propose a diffusion probabilistic model-based speech enhancement (DiffuSE) model that aims to recover clean speech signals from noisy signals. The fundamental architecture of the proposed DiffuSE model is similar to that of DiffWave—a high-quality audio waveform generation model that has a relatively low computational cost and footprint. To attain better enhancement performance, we designed an advanced reverse process, termed the supportive reverse process, which adds noisy speech in each time-step to the predicted speech. The experimental results show that DiffuSE yields performance that is comparable to related audio generative models on the standardized Voice Bank corpus SE task. Moreover, relative to the generally suggested full sampling schedule, the proposed supportive reverse process especially improves the fast sampling, taking few steps to yield better enhancement results over the conventional full step inference process.

I. INTRODUCTION

The goal of speech enhancement (SE) is to improve the intelligibility and quality of speech, by mapping distorted speech signals to clean signals. The SE unit has been widely used as a front-end processor in various speech-related applications, such as speech recognition [1]–[3], speaker recognition [4], assistive hearing technologies [5], [6], and audio attack protection [7]. Recently, deep neural network (DNN) models have been widely used as fundamental tools in SE systems, yielding promising results [8]–[14]. Compared to traditional SE methods, DNN-based methods can more effectively characterize nonlinear mapping between noisy and clean signals, particularly under extremely low signal-to-noise (SNR) scenarios and/or non-stationary noise environments [15]–[17].

Traditional SE methods calculates the noisy-clean mapping through the discriminative methods in time-frequency (T-F) domain or time domain. For the T-F domain methods, the time-domain speech signals are first converted into spectral features through a short-time Fourier transform (STFT). The mapping function of noisy to clean spectral features is then formulated by a direct mapping function [8], [11], or a masking function [9], [18], [19]. The enhanced spectral features are reconstructed to time-domain waveforms with the phase of the noisy speech based on the inverse STFT operation [20]. As compared with T-F domain methods, it has been shown that the time-domain SE methods can avoid the distortion caused by inaccurate phase information [21], [22]. To date, several audio generation models have been directly applied to or moderately modified to perform SE, estimating the distribution of the clean speech signal, such as generative adversarial networks (GAN) [23]–[25], autoregressive models [26], variational autoencoders (VAE) [27], and flow-based models [28].

The diffusion probabilistic model, proposed in [29], has shown strong generation capability. The diffusion probabilistic model includes a diffusion/forward process and a reverse process. The diffusion process converts clean input data to an isotropic Gaussian distribution by adding Gaussian noise to the original signal at each step. In the reverse process, the diffusion probabilistic model predicts a noise signal and subtracts the predicted noise signal from the noisy input to retrieve the clean signal. The model is trained by optimizing the evidence lower bound (ELBO) during the diffusion process. Recently, the diffusion probabilistic models have been shown to provide outstanding performance in generative modeling for natural images [30], [31], and raw audio waveforms [32], [33]. As reported in [32], the DiffWave model, formed by the diffusion probabilistic model, can yield state-of-the-art performance on either conditional or unconditional waveform generation tasks with a small number of parameters.

In this study, we propose a novel diffusion probabilistic model-based SE method, called DiffuSE. The basic model structure of DiffuSE is similar to that of Diffwave. Since the target task is SE, DiffuSE uses the noisy spectral features as the conditioner, rather than the clean Mel-spectral features used in Diffwave. Meanwhile, different from the derived equation of the diffusion model, we combine the noisy speech signal into the reverse process instead of the isotropic Gaussian noise. To further improve the quality of the enhanced speech, we pretrained the model using clean Mel-spectral features as a conditioner. After pretraining, we replaced the conditioner with noisy spectral features, reset the parameters in the conditioner encoder, and preserved other parameters for the SE training.

The contributions of this study are three-fold: (1) It is the first study to apply the diffusion probabilistic model to
the SE tasks. (2) We propose a novel supportive reverse process, specifically for the SE task, which combines the noisy speech signals during the reverse process. (3) The experimental results confirm the effectiveness of DiffuSE, which provides comparable or even better performance as compared to related time-domain generative SE methods.

The remainder of this paper is organized as follows. We present the diffusion models in Section II and introduce the DiffuSE architecture in Section III. We provide the experimental setting in Section IV, report the results in Section V, and conclude the paper in Section VI.

II. DIFFUSION PROBABILISTIC MODELS

This section introduces the diffusion and the reverse procedures of the diffusion probabilistic model. A detailed mathematical proof of the model’s ELBO can be found in [30], and we only discuss the diffusion and reverse processes with their algorithm in this section.

\[ q(x_0) \rightarrow q(x_t|x_{t-1}) \rightarrow p_{latent}(x_T) = N(0,I) \]

\[ p_{latent}(x_T) = N(0,I) \]

\[ p_{latent}(x_T) = N(0,I) \]

\[ p_{latent}(x_T) = N(0,I) \]

In Figure 1, the solid arrows are the diffusion process from data \( x_0 \) to the latent variable \( x_T \), represented as:

\[ q(x_1, \ldots, x_T|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \] (1)

where \( q(x_t|x_{t-1}) \) is formulated by a fixed Markov chain, \( N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \), with a small positive constant ratio \( \beta_t \), and the Gaussian noise is added to the previous distribution \( x_{t-1} \). The overall process gradually converts data \( x_0 \) to a latent variable with an isotropic Gaussian distribution of \( p_{latent}(x_T) = N(0,I) \), according to the predefined schedule \( \beta_1, \ldots, \beta_T \).

The sampling distribution at the \( t \)-th step, \( x_t \), can also be derived from the distribution of \( x_0 \) in a closed form by marginalizing \( x_1, \ldots, x_{t-1} \) as:

\[ q(x_t|x_0) = N(x_t; \sqrt{1-\alpha_t}x_0, \alpha_t I), \] (2)

where \( \alpha_t = 1 - \beta_t \) and \( \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s \). Empirically, we can sample the \( t \)-th step distribution \( x_t \) from the initial data \( x_0 \) directly. In contrast, The dashed arrows in Figure 1 are the reverse process, converting the latent variable \( x_T \) to \( x_0 \), which is also defined by a Markov chain:

\[ p_{0}(x_0, \ldots, x_{T-1}|x_T) = \prod_{t=1}^{T} p_{0}(x_{t-1}|x_t), \] (3)

where \( p_{0}(\cdot) \) is the distribution of the reverse process with learnable parameter \( \theta \). Because the marginal likelihood \( p_{0}(x_0) = \int p_{0}(x_0, \ldots, x_{T-1}|x_T) \cdot p_{latent}(x_T)dx_{1:T} \) is intractable for calculations in general, the model should be trained using ELBO. Recently, [30] showed that under a certain parameterization, the ELBO could be calculated using a closed-form solution.

B. Training through Parameterization

1) Parameterization: The transition probability in the reverse process \( p_{0}(x_{t-1}|x_t) \) in Eq. 3 can be represented by two parameters, \( \mu_\theta \) and \( \sigma_\theta \), as \( N(x_{t-1}; \mu_\theta(x_t), \sigma_\theta(x_t,t)^2 I) \), with a learnable parameter \( \theta \). \( \mu_\theta \) is an \( L \)-dimensional vector, that estimates the mean of the distribution of \( x_{t-1} \). \( \sigma_\theta \) denotes the standard deviation (a real number) of the \( x_{t-1} \) distribution. Note that both values take two inputs: the diffusion step \( t \in \mathbb{N} \), and variable \( x_t \in \mathbb{R}^L \). Further, Eq. 2 can also be reparameterized as \( x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \) for \( \epsilon \sim N(0,I) \). \( \sigma_\theta(x_t, t) \) was set to \( \bar{\sigma}_t \) as a time-dependent parameter.

\[ \mu_\theta(x_t) \]

\[ \sigma_\theta(x_t, t) \]

\[ \mu_\theta(x_t) \]

\[ \sigma_\theta(x_t, t) \]

\[ \mu_\theta(x_t) \]

\[ \sigma_\theta(x_t, t) \]
2) Training and Sampling: In the reverse process, \( p_\theta(x_{t-1}|x_t) \) in Eq. 3 aims to predict the previous distribution by the current mixed data with extra Gaussian noise added in the diffusion process. Therefore, the predicted mean \( \mu_\theta \) is estimated by eliminating the Gaussian noise \( \epsilon \) in the mixed data \( x_t \). According to the derivations in [30], \( \mu_\theta \) can be predicted by a given \( x_t \) and \( t \) as Eq. 4:

\[
\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \alpha_t}} \epsilon_\theta(x_t, t) \right),
\]

(4)

Note that the real Gaussian noise added in the diffusion process \( \epsilon \) is unknown in the reverse process. Therefore, the model \( \epsilon_\theta \) should be designed to predict \( \epsilon \). In contrast, \( \sigma_t \), the standard deviation of the \( x_{t-1} \), can be fixed to a constant for every step \( t \) as Eq. 5:

\[
\sigma_t = \tilde{\sigma}_t^{'2}, \quad \tilde{\sigma} = \left\{ \begin{array}{ll} \frac{1 - \alpha_{t-1}}{1 - \alpha_t} \beta_t & \text{for } t > 1, \\ \beta_0 & \text{for } t = 0. \end{array} \right.
\]

(5)

Therefore, for predicting \( \mu_\theta(x_t, t) \) in the reverse process, the model parameters \( \theta \) aim to estimate the Gaussian noise \( \epsilon_\theta(x_t, t) \) by input \( x_t \) and \( t \). During the diffusion process, the training loss of the model is defined to reduce the distance of the estimated noise \( \epsilon_\theta(x_t, t) \) and the Gaussian noise \( \epsilon \) in the mixed data \( x_t \), as shown in Eq. 6.

\[
\nabla_\theta \| \epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_t + \sqrt{1 - \alpha_t}\epsilon, t) \|^2_2
\]

(6)

After the training process, \( x_{t-1} \) was computed using Eq. 7 where \( z \sim N(0, I) \).

\[
x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \alpha_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z,
\]

(7)

To summarize, the model is trained during the diffusion process by estimating the Gaussian noise \( \epsilon \) inside the mixed-signal \( x_t \), and samples the data \( x_0 \) through the reverse process. We describe the diffusion and reverse processes in Algorithms 1 and 2, respectively. Table I lists the parameters of the diffusion probabilistic models.

### Table I

<table>
<thead>
<tr>
<th>Process</th>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diffusion Process</td>
<td>( \alpha_t )</td>
<td>ratio of ( x_{t-1} ) in ( x_t )</td>
</tr>
<tr>
<td></td>
<td>( \beta_t )</td>
<td>ratio of noise added in ( x_t )</td>
</tr>
<tr>
<td></td>
<td>( \bar{\alpha}_t )</td>
<td>ratio of ( x_{t-1} ) in ( x_t )</td>
</tr>
<tr>
<td></td>
<td>( \epsilon )</td>
<td>isotropic Gaussian noise</td>
</tr>
<tr>
<td>Reverse Process</td>
<td>( \epsilon_\theta )</td>
<td>predicted noise from model ( \theta )</td>
</tr>
<tr>
<td></td>
<td>( \mu_\theta )</td>
<td>predicted mean from model ( \theta )</td>
</tr>
<tr>
<td></td>
<td>( \sigma_t )</td>
<td>standard deviation</td>
</tr>
</tbody>
</table>

### III. Diffuse Architecture

In the proposed DiffuSE model, we derive a novel supportive reverse process to replace the original reverse process, to eliminate noise signals from the noisy input more effectively.

A. Supportive Reverse Process

In the original diffusion probabilistic model, the Gaussian noise is applied in the reverse process. Since the clean speech signal was unseen during the reverse process, the calculated speech signal, \( x_t \), may be distorted during the reverse process from step \( T, \ldots, t + 1 \). To address this issue, we proposed a supportive reserve process, starting the sampling process from the noisy speech signal \( y \), and combining \( y \) at each reverse step while reducing the additional Gaussian signal.

The noisy speech signal \( y \in \mathbb{R}^L \) can be considered as a combination of the clean speech signal \( x_0 \) and background noise \( n \in \mathbb{R}^L \), as \( y = x_0 + n \). In the supportive reserve process, we define a new valuable \( \tilde{\mu}_\theta(x_t, t) \), which is a combination of noisy speech \( y \) and the predicted \( \mu_\theta(x_t, t) \) as shown in Eq. 8:

\[
\tilde{\mu}_\theta(x_t, t) = (1 - \gamma_t)\mu_\theta(x_t, t) + \gamma_t \sqrt{\bar{\alpha}_{t-1}} y
\]

(8)

where \( \mu_\theta(x_t, t) \) can be formulated as \( \mu_\theta(x_t, t) = \sqrt{\bar{\alpha}_{t-1}}(x_0 + \gamma_t n) \) from the mean of \( x_{t-1} \) is known as \( \sqrt{\bar{\alpha}_{t-1}}x_0 \) in the diffusion process. Therefore, we filled the remaining part of noise by the Gaussian signal with the independent assumption as Eq. 9:

\[
\bar{\sigma}_t = \sqrt{\bar{\sigma}_t^{'2} - \gamma_t^2 \bar{\alpha}_{t-1}}
\]

(9)

In diffusion models, \( \epsilon_\theta(x_t, t) \) is used to predict the noise signal \( \epsilon \) from \( x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon \). For the SE task, instead of following the original reverse equations derived from the diffusion process, the objective of \( \epsilon_\theta(x_t, t) \) could also be considered as predicting the non-speech part \( \epsilon \), which is then used to recover the clean speech signal \( x_0 \) from the mixed-signal \( x_t \). Therefore, although the supportive reverse process replaces the combination of predicted mean and Gaussian noise by the noisy signal, \( \epsilon_\theta \) still has the ability to predict the non-speech components from the noisy signal \( x_t \) at the \( t \)-th step based on the learned knowledge about different speech-noise combinations during the diffusion process. In addition, because \( x_t \) is a combination of the clean speech signal \( x_0 \) and the Gaussian noise \( \epsilon \), to reach a more efficient clean speech recovery, the supportive reverse process directly uses the noisy speech signal \( y \) as the input of the reverse process rather than the Gaussian noise. Meanwhile, at each reverse step, the supportive reverse process combines \( \mu_\theta(x_t, t) \) with the noisy speech \( y \) and the Gaussian noise \( z \) to form the input \( x_t \) of \( \epsilon_\theta(x_t, t) \). After the overall reverse process is completed, we follow the suggestion in [34], [35] to combine the enhanced and original noisy signal to obtain the final enhanced speech. The detailed procedure of the supportive reverse process is shown in Algorithm 3.

B. Model Structure

1) DiffWave Architecture: The model architecture of DiffWave is similar to that of WaveNet [36]. Without an autoregressive generation constraint, the dilated convolution is replaced with a bidirectional dilated convolution (Bi-DilConv). The non-autoregressive generation property of DiffWave yields
**Algorithm 3 Supportive Reverse Sampling**

\[ x_T = y, \]

\[
\text{for } t = T, T - 1, \ldots, 1 \text{ do}
\]

\[
\text{Compute } \hat{\mu}_\theta(x_t, t) \text{ and } \sigma_t
\]

\[
\text{Sample } z \sim N(0, I) \text{ if } t > 1, \text{ else } z = 0
\]

\[
x_{t-1} = \hat{\mu}_\theta(x_t, t) + \sqrt{\sigma_t^2 - \gamma_t^2} \alpha_{t-1} z
\]

\[
\text{according to Eq. 8 and 9}
\]

\[\text{end for}\]

\[\text{return } x_0\]

---

**Algorithm 4 Fast Sampling**

\[ x_T \sim \Phi_{\text{arent}} = N(0, I), \]

\[
\text{for } s = T_{\text{infer}}, T_{\text{infer}} - 1, \ldots, 1 \text{ do}
\]

\[
\text{Compute } \mu_{\text{fast}}(x_s, s) \text{ and } \sigma_{\text{fast}}^2
\]

\[
\text{Sample } x_{s-1} \sim p_{0}(x_{s-1}|x_s) = N(x_{s-1}; \mu_{\text{fast}}(x_s, s), \sigma_{\text{fast}}^2 I)
\]

\[\text{end for}\]

\[\text{return } x_0\]

---

A major advantage over WaveNet in that the generation speed is much faster. The network comprises a stack of \(N\) residual layers with residual channel \(C\). These layers were grouped into \(m\) blocks, and each block had \(n = \frac{N}{m}\) layers. The kernel size of Bi-DilConv is 3, and the dilation is doubled at each layer within each block as \([1, 2, 4, \ldots, 2^{n-1}]\). Each of the residual layers has a skip connection to the output, which is the same as that used in WaveNet.

2) **DiffuSE Architecture:** Figure 2 shows the model structure of the DiffuSE. As Diffwave, the conditioner in DiffuSE aims to keep the output signal similar to the target speech signal, enabling \(\epsilon_\theta(x_t, t)\) to separate the noise and clean speech from the mixed data. Thus, we replace the input of the conditioner from clean Mel-spectral features to noisy spectral features. We set the parameter of DiffuSE, \(\epsilon_\theta : \mathbb{R}^L \times \mathbb{N} \rightarrow \mathbb{R}^L\), to be similar to those used in the DiffWave model [32].

**C. Pretraining with Clean Mel-spectral Conditioner**

To generate high-quality speech signals, we pretrained the DiffuSE model with the clean Mel-spectral features. In DiffWave, the conditional information is directly adopted from the clean speech, allowing the model \(\epsilon_\theta(x_t, t)\) to separate the clean speech and noise from the mixed signals. After pretraining, we changed the conditioner from clean Mel-spectral features to the noisy spectral features, reset the parameters in the conditioner encoder, and preserved other parameters for the SE training.

**D. Fast Sampling**

Given a trained model from Algorithm 1, the authors in [32] discovered that the most effective denoising steps in sampling occur near \(t = 0\) and accordingly derived a fast sampling algorithm. The algorithm collapses the \(T\)-step in the diffusion process into \(T_{\text{infer}}\)-step in the reverse process with a proposed variance schedule. This motivates us to apply the fast sampling into DiffuSE to reduce the number of denoising steps. In addition, by changing \(\mu_{\text{fast}}(x_t, t)\) and \(\sigma_{\text{fast}}^2\) to \(\mu_{\text{fast}}(x_t, t)\) and \(\sigma_{\text{fast}}^2\) using Eq. 8 and Eq. 9, respectively, the fast sampling schedule can be combined with the supportive reverse process.

**IV. EXPERIMENTS**

**A. Data**

We evaluated the proposed DiffuSE on the VoiceBank-DEMAND dataset [37]. The dataset contains 30 speakers from the VoiceBank corpus [38], which was further divided into a training set and a testing set with 28 and 2 speakers, respectively. The training utterances were mixed with eight real-world noise samples from the DEMAND database [39] and two artificial (babble and speech shaped) samples at SNR levels of 0, 5, 10, and 15 dB. The testing utterances were mixed with different noise samples, according to SNR values of 2.5, 7.5, 12.5, and 17.5 dB to form 824 utterances (0.6 h). Additionally, utterances from two speakers were used to form a validation set for model development, resulting in 8.6 h and 0.7 h of data for training and validation, respectively. All of the utterances were resampled to 16 kHz sampling rates.

**B. Model Setting and Training Strategy**

The DiffuSE model was constructed using 30 residual layers with three dilation cycles \([1, 2, \ldots, 512]\) and a kernel size of 3. Based on the design of DiffWave in [32], we set the number of diffusion steps and residual channels as \([T, C] \in [50, 63], [200, 128]\) for Base and Large DiffuSE, respectively. The training noise schedule was linearly spaced as \(\beta_t \in [1 \times 10^{-4}, 0.05]\) for Base DiffuSE, and \(\beta_t \in [1 \times 10^{-4}, 0.02]\) for Large DiffuSE. The learning rate was \(2 \times 10^{-4}\) for both pretraining (using clean Mel-spectrum) and fine-tuning the DiffuSE model. The dimension for the Mel-spectrum was 80, and the dimension of the noisy spectrum was 513 for the same window size of 1024 with 256 shifts. The \(\gamma_t\) parameter in the supportive reverse process was set to \(\gamma_t = \frac{1}{\sqrt{\sigma_t^2}}\) for \(t\) larger than 1, and \(\gamma_t\) was set to 0.2 as the combination ratio of noisy signal to the enhanced output. During pretraining, we followed the instructions in [32], where the vocoder model was trained for one million iterations, and the large model for three hundred thousand iterations for better initialization. In the training of the SE model, we trained the
C. Evaluation Metrics

We report the standardized evaluation metrics for performance comparison, including perceptual evaluation of speech quality (PESQ) [40], (the wide-band version in ITU-T P.862.2), prediction of the signal distortion (CSIG), prediction of the background intrusiveness (CBAK), and prediction of the overall speech quality (COVL) [41]. Higher scores indicated better SE performance for all of evaluation scores.

V. EXPERIMENTAL RESULTS

In this section, we first present the DiffuSE results with the original reverse process and the proposed supportive reverse process. Next, we compare DiffuSE with other state-of-the-art (SOTA) time-domain generative SE models. Finally, we justify the effectiveness of DiffuSE by visually analyzing the spectrogram and waveform plots of the enhanced signals.

A. Supportive Reverse Process Results

In the supportive reverse process, we adopted two sampling schedules, namely a fast sampling schedule and a full sampling schedule. For the fast sampling schedule, the variance schedules were [0.0001, 0.001, 0.01, 0.05, 0.2, 0.5] for Base DiffuSE and [0.0001, 0.001, 0.01, 0.05, 0.2, 0.7] for Large DiffuSE, as suggested in [32]. The full sampling schedule used the same β, as that used in the diffusion process.

Tables II (a) and (b) list the results of the Base DiffuSE model and the Large DiffuSE model, respectively. In the tables, the results of DiffuSE using the original reverse process and the supportive reverse processes are denoted as “RP” and “SRP” respectively. The table reports the results of both fast and full sampling schedules. To investigate the effectiveness of the supportive reverse process, we further tested performance by including noisy speech signal at the input, output, and both input and output of the DiffuSE model with the original reverse process; the results are denoted by “RP-”, “RP-”, and “RP-”, respectively.

From Tables II (a) and (b), we notice that for “RP” and “SRP,” the fast sampling schedule yielded better enhancement results than the full sampling schedule, which is consistent with the findings reported in DiffWave [32]. In contrast, for “RP-,” “RP-,” and “SRP,” the fast sampling schedule yielded better results than the full sampling schedule. A possible reason is that the noisy speech signal is a combination of clean speech and noise signals and presents clearly different properties from the pure Gaussian noise. Therefore, when including noisy speech in the input, it is more suitable to apply a fast sampling schedule than the full sampling schedule.

<table>
<thead>
<tr>
<th>(a) Evaluation results of the Base DiffuSE model. Base DiffuSE Schedule</th>
<th>PESQ</th>
<th>CSIG</th>
<th>CBAK</th>
<th>COVL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td>3.97</td>
<td>3.35</td>
<td>2.44</td>
<td>2.63</td>
</tr>
<tr>
<td>RP</td>
<td>Fast</td>
<td>1.96</td>
<td>3.13</td>
<td>2.22</td>
</tr>
<tr>
<td>Full</td>
<td>1.97</td>
<td>3.21</td>
<td>2.22</td>
<td>2.57</td>
</tr>
<tr>
<td>RP-</td>
<td>Fast</td>
<td>2.07</td>
<td>3.21</td>
<td>2.57</td>
</tr>
<tr>
<td>Full</td>
<td>2.05</td>
<td>3.27</td>
<td>2.48</td>
<td>2.64</td>
</tr>
<tr>
<td>RP-N</td>
<td>Fast</td>
<td>2.05</td>
<td>3.21</td>
<td>2.21</td>
</tr>
<tr>
<td>Full</td>
<td>2.12</td>
<td>3.38</td>
<td>2.25</td>
<td>2.72</td>
</tr>
<tr>
<td>RP-N</td>
<td>Fast</td>
<td>2.29</td>
<td>3.47</td>
<td>2.67</td>
</tr>
<tr>
<td>Full</td>
<td>2.31</td>
<td>3.51</td>
<td>2.61</td>
<td>2.88</td>
</tr>
<tr>
<td>SRP</td>
<td>Fast</td>
<td>2.41</td>
<td>3.61</td>
<td>2.81</td>
</tr>
<tr>
<td>Full</td>
<td>2.38</td>
<td>3.60</td>
<td>2.79</td>
<td>2.97</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b) Evaluation results of the Large DiffuSE model. Large DiffuSE Schedule</th>
<th>PESQ</th>
<th>CSIG</th>
<th>CBAK</th>
<th>COVL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td>1.97</td>
<td>3.35</td>
<td>2.44</td>
<td>2.63</td>
</tr>
<tr>
<td>RP</td>
<td>Fast</td>
<td>2.09</td>
<td>3.29</td>
<td>2.31</td>
</tr>
<tr>
<td>Full</td>
<td>2.16</td>
<td>3.39</td>
<td>2.31</td>
<td>2.75</td>
</tr>
<tr>
<td>RP-N</td>
<td>Fast</td>
<td>2.18</td>
<td>3.45</td>
<td>2.60</td>
</tr>
<tr>
<td>Full</td>
<td>2.20</td>
<td>3.42</td>
<td>2.48</td>
<td>2.78</td>
</tr>
<tr>
<td>RP-N</td>
<td>Fast</td>
<td>2.16</td>
<td>3.42</td>
<td>2.30</td>
</tr>
<tr>
<td>Full</td>
<td>2.17</td>
<td>3.45</td>
<td>2.29</td>
<td>2.78</td>
</tr>
<tr>
<td>RP-N</td>
<td>Fast</td>
<td>2.37</td>
<td>3.45</td>
<td>2.69</td>
</tr>
<tr>
<td>Full</td>
<td>2.33</td>
<td>3.55</td>
<td>2.56</td>
<td>2.91</td>
</tr>
<tr>
<td>SRP</td>
<td>Fast</td>
<td>2.43</td>
<td>3.63</td>
<td>2.81</td>
</tr>
<tr>
<td>Full</td>
<td>2.39</td>
<td>3.63</td>
<td>2.75</td>
<td>2.99</td>
</tr>
</tbody>
</table>
TABLE III
EVALUATION RESULTS OF DIFFUSE WITH COMPARATIVE TIME-DOMAIN GENERATIVE SE MODELS. DIFFUSE WITH THE BASE AND LARGE MODELS ARE DENOTED AS DIFFUSE(BASE) AND DIFFUSE(LARGE), RESPECTIVELY. ALL OF THE METRIC SCORES FOR THE COMPARATIVE METHODS ARE TAKEN FROM THEIR SOURCE PAPERS.

<table>
<thead>
<tr>
<th>Method</th>
<th>PESQ</th>
<th>CSIG</th>
<th>CBAK</th>
<th>COVL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td>1.97</td>
<td>3.35</td>
<td>2.44</td>
<td>2.63</td>
</tr>
<tr>
<td>SegAn</td>
<td>2.16</td>
<td>3.48</td>
<td>2.94</td>
<td>2.80</td>
</tr>
<tr>
<td>DsegAn</td>
<td>2.39</td>
<td>3.46</td>
<td>3.11</td>
<td>2.90</td>
</tr>
<tr>
<td>se-flow</td>
<td>2.28</td>
<td>3.70</td>
<td>3.03</td>
<td>2.97</td>
</tr>
<tr>
<td>Diffuse(base)</td>
<td>2.41</td>
<td>3.61</td>
<td>2.81</td>
<td>2.99</td>
</tr>
<tr>
<td>Diffuse(large)</td>
<td>2.43</td>
<td>3.63</td>
<td>2.81</td>
<td>3.01</td>
</tr>
</tbody>
</table>

In addition to quantitative evaluations, we present spectrogram and waveform plots to qualitatively analyze the enhanced speech signals obtained from the Diffuse models. Figures 3 and 4, respectively, show the spectrogram and waveform plots of (a) clean, (b) noisy, (c) enhanced speech using Diffuse with the original reverse process (denoted as Diffuse+RP), and (d) enhanced speech using Diffuse with the supportive reverse process (denoted as Diffuse+SRP). From Figure 3, we first note that both of the original and supportive reverse processes can effectively remove noise components from a noisy spectrogram. Next, we observe notable speech distortions in (c) Diffuse+RP, especially in the high-frequency regions (marked with red rectangles). For (d) Diffuse+SRP, although some noise components remained, the speech structures were better preserved as compared to (c) Diffuse+RP. From Figure 4, the waveform plots present similar trends to the spectrogram plots: the waveform of (d) Diffuse+SRP preserves speech structures better than that of (c) Diffuse+RP (please compare the two waveforms around 0.8 and 1.3 (s)). The observations in Figures 3 and 4 better explain the results obtained using the supportive reverse process over the original reverse process, as reported in Table II. The samples of the Diffuse-enhanced signals can be found online1.

![Fig. 3. Spectrogram plots of (a) Clean speech, (b) Noisy signal, (c) Enhanced speech by Diffuse with the original reverse process (Diffuse+RP) (d) Enhanced speech by Diffuse with the supportive reverse process (Diffuse+SRP).](https://github.com/neillu23/DiffuseSE)

![Fig. 4. Waveform plots of (a) Clean speech, (b) Noisy signal, (c) Enhanced speech by Diffuse with the original reverse process (Diffuse+RP) (d) Enhanced speech by Diffuse with the supportive reverse process (Diffuse+SRP).](https://github.com/neillu23/DiffuseSE)

B. Comparing Diffuse with Related SE Methods

The proposed Diffuse model is a time-domain generative SE model. For comparison, we selected three SOTA baselines that are also based on time-domain generative SE models, namely SegAn [23], SE-Flow [28], and improved deep SegAN (DsegAn) [42]. The experimental results of the three comparative SE methods are presented in Table III. The results of the Diffuse with the supportive reverse process are also listed, where Diffuse(base) and Diffuse(large) denote the results of using the base and large models, respectively. Compared with the three baselines, the PESQ scores of Diffuse(base) and Diffuse(large) are 2.41 and 2.43, respectively, both of which are much higher than those obtained from the comparative methods. The CSIG scores of Diffuse(base) and Diffuse(large) are 3.61 and 3.63, respectively, again notably higher than those achieved by SegAN and DsegAN. The results confirm that the proposed Diffuse method provides a competitive performance against SOTA generative SE models.

VI. Conclusions

In this study, we have proposed Diffuse, the first diffusion probabilistic model-based SE method. To enable an efficient sampling procedure, we proposed modifying the reverse equation to a supportive reverse process, specially designed for the SE task. Experimental results show that the supportive reverse

---

1https://github.com/neillu23/DiffuseSE
process can improve the quality of the generated speech with few steps to obtain better performance than that of the full reverse process. The results also show that DiffuSE achieves SE performance comparable to that of other SOTA time-domain generative SE models. The results of DiffuSE are reproducible and the code of DiffuSE will be released online³. We believe that the results will shed light on further extensions of using the diffusion probabilistic model for the SE task. In future work, we will further improve the DiffuSE model through different network structures.

VII. ACKNOWLEDGEMENT

This work was supported in part by the grants AS-GC-109-05 and AS-CDA-106-M04 and we would like to thank Alexander Richard at Facebook for his valuable comments about this work.

REFERENCES


