SRTNET: TIME DOMAIN SPEECH ENHANCEMENT VIA STOCHASTIC REFINEMENT

Zhibin Qiu[1,⋆], *Mengfan Fu*[1,⋆], *Yinfeng Yu*[1,2,†], *Lili Yin*[1], *Fuchun Sun*[2], *Hao Huang*[1,3,†]

[1]School of Information Science and Engineering, Xinjiang University, Urumqi, China
[2]Department of Computer Science and Technology, Tsinghua University, China
[3]Xinjiang Key Laboratory of Multi-lingual Information Technology, Urumqi, China

huanghao@xju.edu.cn

ABSTRACT

Diffusion model, as a new generative model which is very popular in image generation and audio synthesis, is rarely used in speech enhancement. In this paper, we use the diffusion model as a module for stochastic refinement. We propose SRTNet, a novel method for speech enhancement via **S**tochastic **R**efinement in complete **T**ime b domain. Specifically, we design a joint network consisting of a deterministic module and a stochastic module, which makes up the "enhance-and-refine" paradigm. We theoretically demonstrate the feasibility of our method and experimentally prove that our method achieves faster training, faster sampling and higher quality. Our code is available at https://github.com/zhibinQiu/SRTNet.git

Index Terms— speech enhancement, time domain, diffusion model, enhance-and-refine, joint training

1. INTRODUCTION

Speech enhancement (SE) using generative models has great potential. Typical generative methods for SE are GAN-based methods [1, 2, 3, 4, 5, 6, 7], flow-based methods [8, 9] and VAE-based methods [10, 11, 12, 13, 14, 15]. Diffusion model is another generative model which is very popular recently [16, 17]. However, it is rarely used for speech enhancement. The SE methods in time domain not only avoid the distortions caused by inaccurate phase information [18], but also avoid the extra overhead of computing the T-F representation. [19] proposed a method for SE in time domain using the diffusion model, but it relies on the speech spectrogram of noisy, so it is not a complete time-domain method. Moreover, in [19], the diffusion model directly estimates the distribution of clean speech by optimizing the evidence lower bound (ELBO), which puts a large computational pressure on the diffusion model and leads to a lot of time consumption during the training phase. [20], a method of image deblurring, proposed the predict-and-refine approach to reduce the computational pressure of the diffusion model while guaranteeing the quality of the image generation. Inspired by this, a joint network paradigm is designed, namely **"enhance-and-refine"** which is comprised of two sub-modules, deterministic module and stochastic module, respectively. The two modules are connected by the residual structure, and the noisy speech is initially enhanced after passing through the deterministic module. Afterward, the initial enhanced result passes through the residual structure and into the stochastic module for detailed refinement. We refer to the network consisting of these two modules as SRTNet. SRTNet allows the diffusion model to act as a stochastic module to learn the residual distribution instead of the distribution of data directly, which significantly reduces the computational overhead of the diffusion model. In addition, SRTNet is a network entirely in time domain and does not depend on any Fourier transform. Our main contributions are as follows:

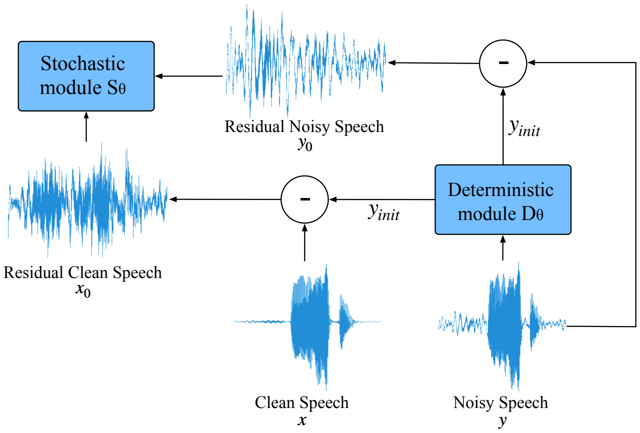

Fig. 1. Overall structure of SRTNet.

- We innovatively apply the diffusion model as a stochastic refinement module for SE task and possess theoretical correctness.
- We introduce an "enhance-and-refine" paradigm and use a joint network to implement it, which results in faster convergence and better speech quality.

2. PROPOSED METHODS

Existing diffusion model based SE methods generally trained directly on the original speech. Motivated by [20], it is also reasonable for the SE task to learn the residual data. In the idea of "enhance-and-refine", we add a deterministic module D_θ to the original conditional diffusion model. The block diagram of the overall structure is illustrated in Fig. 1. The noisy speech y is initially enhanced by D_θ and we call its output y_{init}. Then the residual operations are enforced on the clean speech x and the noisy speech y with y_{init} and the residuals x_0, y_0 are fed to stochastic module S_θ. Because the output of the noisy speech passing through D_θ during sampling is deterministic when the parameters are determined, we call the D_θ deterministic module. Whereas in S_θ, we need to sample from Gaussian distribution in diffusion model, which could produce different outputs, so we call the S_θ stochastic module.

[⋆] Equal contributions. [†] Correspondence authors.

Algorithm 1 SRTNet Training.

```
repeat
    Sample $(x, y) \sim q_{\text{data}}, \epsilon \sim \mathcal{N}(0, \boldsymbol{I})$,
    $s \sim$Uniform$(\{1, \cdots, S\})$ , and $\sqrt{\bar{\alpha}} \sim$Uniform$(l_{s-1}, l_s)$
    Get $y_{init}$ through the deterministic module $D_\theta$,
    $y_{init} = D_\theta(y)$
    Get two Residual: $x_0 = x - y_{init}$ and $y_0 = y - y_{init}$
    Get $x_t$ according to Eq. (12)
    Take gradient descent step on
    $\nabla_\theta \parallel \frac{1}{\sqrt{1-\bar{\alpha}}}(m\sqrt{\bar{\alpha}}(y_0 - x_0) + \sqrt{\delta_t}\epsilon) - \epsilon_\theta(x_t, y_0, \sqrt{\bar{\alpha}}) \parallel_2^2$
until converged
```

Algorithm 2 SRTNet Sampling.

```
Get $y_{init}$ through the deterministic module $D_\theta$,
$y_{init} = D_\theta(y)$
Get Residual $y_0 = y - y_{init}$
Sample $x_T \sim \mathcal{N}(x_T, \sqrt{\bar{\alpha}_T} y_0, \delta_T \boldsymbol{I})$,
for $t = T, T-1, \cdots, 1$ do
    Compute $c_t^x, c_t^y$ and $c_t^\epsilon$ using Eq. (6), (7), and (8)
    Sample $x_{t-1} \sim p_\theta(x_{t-1}|x_t, y_0) =$
    $\mathcal{N}(x_{t-1}; c_t^x x_t + c_t^y y_0 - c_t^\epsilon \epsilon_\theta(x_t, y_0, \sqrt{\bar{\alpha}_t}), \tilde{\delta}_t I)$
end for
return $x_0 + y_{init}$
```

2.1. Diffusion and Reverse process of SRTNet

Diffusion. The diffusion process takes y_0 as a condition, and the noise in diffusion process contains not only Gaussian noise but also non-Gaussian noise in y_0 which is the residual of y and y_{init}. The conditional diffusion process $q(x_t|x_0, y_0)$ is defined as follows:

$$q(x_t|x_0, y_0) = \mathcal{N}(x_t; (1 - m_t)\sqrt{\bar{\alpha}_t} x_0 + m_t \sqrt{\bar{\alpha}_t} y_0, \delta_t I), \quad (1)$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, the noise schedule $\{\alpha_t\}_{t=1}^{T}$ is given. Additionally, m_t is a interpolation ratio between the residual clean speech x_0 and the residual noisy speech y_0. The value of m_t is defined as:

$$m_t = \sqrt{(1 - \bar{\alpha}_t)/\sqrt{\bar{\alpha}_t}}, \quad (2)$$

where $m_0 = 0$ and $m_T \approx 1$. Therefore, the interpolation parameter m_t gradually shifts the mean of Eq. (1) from x-correlated to y-correlated with the diffusion process which satisfies a Markov chain, with details in [19]. And the variance δ_t is defined as:

$$\delta_t = (1 - \bar{\alpha}_t) - m_t^2 \bar{\alpha}_t. \quad (3)$$

Reverse. In the reverse process, we start from x_T, with the condition y_0 and the variance δ_T:

$$p(x_T|y_0) = \mathcal{N}(x_T, \sqrt{\bar{\alpha}_t} y_0, \delta_T I). \quad (4)$$

The reverse process also follows a Markov chain, so we can gradually obtain x_0 from x_T by continuously executing the reverse process. The parameterised conditional reverse process $p_\theta(x_{t-1}|x_t, y_0)$ is denoted as:

$$\mathcal{N}(x_{t-1}, c_t^x x_t + c_t^y y_0 + c_t^\epsilon \epsilon_\theta(x_t, y_0, t), \tilde{\delta}_t I). \quad (5)$$

Note that the mean is parametrized as a linear combination of x_t, residual noisy speech y_0, and estimated noise ϵ_θ. The coefficients c_t^x, c_t^y and c_t^ϵ are derived as follows:

$$c_t^x = \frac{1 - m_t}{1 - m_{t-1}} \frac{\delta_{t-1}}{\delta_t} \sqrt{\alpha}_t + (1 - m_{t-1}) \frac{\tilde{\delta}_t}{\delta_{t-1}} \frac{1}{\sqrt{\alpha}_t}, \quad (6)$$

$$c_t^y = (m_{t-1}\delta_t - \frac{m_t(1 - m_t)}{1 - m_{t-1}} \alpha_t \delta_{t-1}) \frac{\sqrt{\bar{\alpha}_{t-1}}}{\delta_t}, \quad (7)$$

$$c_t^\epsilon = (1 - m_{t-1}) \frac{\tilde{\delta}_t}{\delta_{t-1}} \frac{\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\alpha_t}}. \quad (8)$$

The variance $\tilde{\delta}_t$ in Eq. (5) can be derived from reverse diffusion process $p(x_{t-1}|x_t, x_0, y_0)$, and the detailed derivation is given in [19]:

$$\tilde{\delta}_t = \delta_{t-1} - \Big(\frac{1 - m_t}{1 - m_{t-1}}\Big)^2 \alpha_t \frac{\delta_{t-1}^2}{\delta_t}. \quad (9)$$

2.2. Training and Sampling of SRTNet

Training. The training process of the SRTNet is described in Algorithm 1. During the training phase, all parameters of diffusion process will depend on the noise level $\bar{\alpha}$ which is obtained by hierarchical sampling rather than time steps used in [19]. Specifically, a segment (l_{s-1}, l_s) is sampled from $s \sim U(\{1, \ldots, S\})$ where S is the length of the noise level schedule. Then the noise level $\bar{\alpha}$ is obtained from this segment by sampling from the uniform distribution. The benefit of the diffusion model relying on the noise level in the training phase is that it allows us to sample with an arbitrary noise level schedule. The effectiveness of this approach is verified in [16]. Therefore, in Eq. (1), the interpolation ratio m is denoted as:

$$m = \sqrt{(1 - \bar{\alpha})/\sqrt{\bar{\alpha}}}, \quad (10)$$

and the variance δ is:

$$\delta = (1 - \bar{\alpha}) - m^2 \bar{\alpha}. \quad (11)$$

Therefore, the expression of x_t through reparameterizing the diffusion process Eq. (1) can be denoted as:

$$x_t = (1 - m)\sqrt{\bar{\alpha}} x_0 + m\sqrt{\bar{\alpha}} y_0 + \sqrt{\delta}\boldsymbol{\epsilon}. \quad (12)$$

To optimize the ELBO, we can directly model the mean of reverse process Eq. (5). But in practice, we generally model the noise added at a certain noise level, which is a unweighted variant of ELBO [21]. The objective function is defined as:

$$\mathbb{E} \parallel \epsilon_* - \epsilon_\theta(x_t, y_0, \sqrt{\bar{\alpha}}) \parallel_2^2, \quad (13)$$

where ϵ_* is the noise from the combination of Gauss noise ϵ and the noise in y_0:

$$\epsilon_* = \frac{m\sqrt{\bar{\alpha}}}{\sqrt{1 - \bar{\alpha}}}(y_0 - x_0) + \frac{\sqrt{\delta}}{\sqrt{1 - \bar{\alpha}}}\epsilon. \quad (14)$$

Sampling. The sampling process of the SRTNet is described in Algorithm 2. Through training, our model is able to efficiently model the noise at different noise levels. The noisy speech y is first fed into the deterministic model D_θ to obtain y_{init}. Unlike the training process, here we obtain the noise level from the noise level schedule rather than hierarchical sampling. By iterating over the noise level schedule, we can run the reverse process through the reverse Markov chain. Afterwards, we obtain the residual clean speech x_0. The final enhanced speech will be obtained by summing x_0 and y_{init}.

2.3. Structure of deterministic and stochastic module

The deterministic module and the stochastic module have a similar structure as in [22, 19]. Note that the conditioner and the noise level encoding here are only for the stochastic module. Moreover, we have made some other modifications to the structure. Unlike [22, 19], we have modified the structure with two main changes:

Firstly, we directly use the waveform of the noisy speech y as the conditioner instead of spectrogram. As a result, a complete time-domain SE is achieved. Moreover, it is also experimentally found that our model can converge faster, the detail in Sec. 3.

Secondly, we replace the time step encoding with a noise level

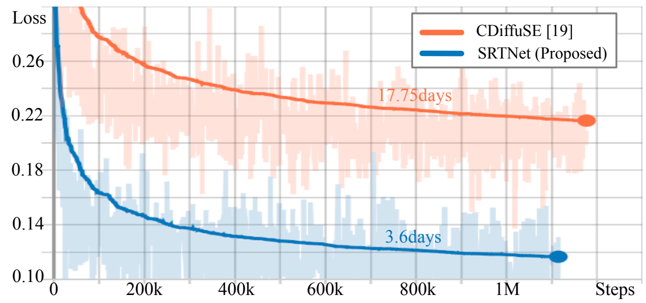

Fig. 2. Training curves of SRTNet and baseline [19] (Base). The numbers (17.75, 3.6) are the time consumption to train the two models for 800k steps.

encoding. But our encoding method differently from both in [17, 16], the encoding method can be expressed by:

$$\sqrt{\bar{\alpha}}_{\text{encoding}} = \Big[\sin\left(10^{\frac{0\times 4}{63}}\sqrt{\bar{\alpha}}\right), \ldots, \sin\left(10^{\frac{63\times 4}{63}}\sqrt{\bar{\alpha}}\right), \cos\left(10^{\frac{0\times 4}{63}}\sqrt{\bar{\alpha}}\right), \ldots, \cos\left(10^{\frac{63\times 4}{63}}\sqrt{\bar{\alpha}}\right) \Big]. \quad (15)$$

Through this, we can obtain different encoding results with different noise level conditions as part of the input to the diffusion model.

3. EXPERIMENTS

3.1. Experimental setup

3.1.1. Datasets

The VoiceBank-DEMAND corpus [23], which includes 30 speakers from different accent regions in the UK and the US, is selected to evaluate the proposed method, with 28 speakers selected for training and 2 others for testing. The training set consists of 11, 572 single channel speech samples, while the test set contains 824 utterances from 2 speakers (one male and one female). The signal-to-noise ratios (SNRs) of the training set are 0 dB, 5 dB, 10 dB, 15 dB. The test set mixes with five unseen test noise types (all from the DEMAND database [24]) selected at 2.5 dB, 7.5 dB, 12.5 dB, 17.5 dB. For the experiments, the original waveform is sub-sampled from 48 kHz to 16 kHz. CHiME-4 [25] is another dataset commonly used for SE tasks. The test set in CHiME-4 is synthesized from noises recorded from four real-life scenes and four speakers, and as in [19], we also take the signal from the fourth microphone to evaluate the generalization performance of our model.

3.1.2. Performance metrics

Four common SE evaluation metrics are used: perceptual evaluation of speech quality (PESQ) [26], background intrusiveness (CBAK), prediction of the signal distortion (CSIG), and overall speech quality (COVL) [27]. The PESQ score ranges from -0.5 to 4.5, and the rest of the metrics range from 1 to 5. Higher score means better speech enhancement performance.

3.1.3. Training and sampling

We train SRTNet 800k steps on two NVidia 3090 GPUs with Adam optimizer, learning rate is 2×10^{-4} and the batch size is set to 32. The inference noise level schedule is same to [19]. In order to recover the high frequency speech we combine the sampling results with the noisy speech with a ratio of 0.2 at the end of the reverse process [28, 29]. Moreover, to avoid randomness, we infer the results several times and take the average value as the final result. For other models, we follow the training setups as in the original papers.

3.2. Results

3.2.1. Results from generative models in matched condition

SRTNet and other recent generative models are trained on the VoiceBank-DEMAND dataset, and tested on the matched condition (training and testing on the same dataset). We can find that our method is the strongest generative model in Table 1. Compared to other generative methods, we further narrow the gap with the regression-based discriminative models. In addition, compared with our baseline [19], not only do we achieve better performance, but also greatly improve the convergence speed of the model. For example, when we train them 800k steps, the two models have almost converged. SRTNet consumes only one-fifth of the training time of [19] as shown in Fig. 2.

Table 1. SRTNet v.s. generative models (matched condition)

Method	PESQ(↑)	CSIG(↑)	CBAK(↑)	COVL(↑)
unprocessed	1.97	3.35	2.44	2.63
SEGAN [1]	2.16	3.48	2.94	2.80
SASEGAN [6]	2.36	3.54	3.08	2.93
DSEGAN [30]	2.39	3.46	3.11	2.90
SE-Flow [9]	2.28	3.70	3.03	2.97
DiffuSE [22]	2.41	3.61	2.81	2.99
CDiffuSE [19](Base)	2.44	3.66	2.83	3.03
CDiffuSE [19](Large)	2.52	3.72	2.91	3.10
SRTNet (Ours)	**2.69**	**4.12**	**3.19**	**3.39**

3.2.2. Results on generalizability to mismatched condition

To verify that our model as a generative model has greater potential in generalizability than the discriminative model, we have done experiments on mismatched condition. The test set is obtained from CHiME-4 test set. Table 2 shows the results. The performance of the discriminative models degrade greatly and our model has a significant advantage. The main reason is that the generative methods learn information about the data distribution rather than a mapping relationship learned by reducing a certain distance such as $L_p - loss$. This feature allows SRTNet as a generative model to perform better.

Table 2. SRTNet v.s. discriminative models (mismatched condition). The numbers in parentheses indicate the relative change in performance under mismatched condition and matched condition.

Method	PESQ(↑)	CSIG(↑)	CBAK(↑)	COVL(↑)
Unprocessed	1.27	2.61	1.93	1.88
WaveCRN [31]	1.43(-1.20)	2.53(-1.42)	2.03(-1.03)	1.91(-1.38)
Demucs [28]	1.38(-1.27)	2.50(-1.49)	2.08(-1.25)	1.88(-1.44)
Conv-TasNet [32]	1.63(-1.21)	1.70(-0.63)	1.82(-0.80)	1.54(-0.97)
CDiffuSE(Large) [19]	1.66(-0.86)	2.98(-0.74)	2.19(-0.72)	2.27(-0.83)
SRTNet (Ours)	**1.87**(-0.82)	**3.37**(-0.75)	**2.49**(-0.70)	**2.67**(-0.72)

3.2.3. Speech waveform and spectrogram analysis

Fig. 3 illustrates the waveforms and spectrograms of SRTNet output at different phases. By observing the waveforms, we can find that

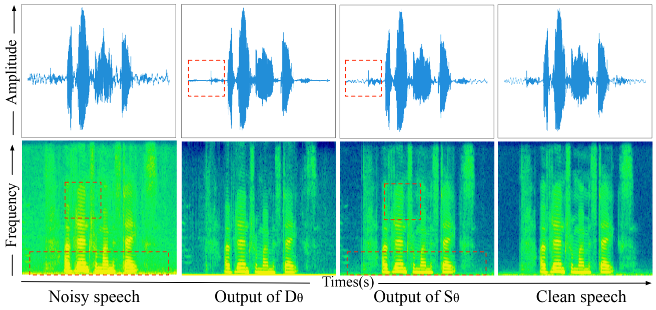

Fig. 3. A test example (matched condition) enhanced by SRTNet and the output at different phases. The output of each phase contains two parts: speech waveform and spectrogram.

the noisy speech is more aggressively erased after the deterministic module, and then the information is refined by the stochastic module. From the two different phase of speech spectrograms, we can find that in the first phase most of the noise has been removed and our model can effectively deals with both high-frequency and low-frequency noise as shown in the red box. Although our model is a complete time domain approach, it performs well in the frequency domain, which is one of the strengths of our model.

3.2.4. Ablation experiments

Here we carry out ablation tests to achieve deeper investigation into the contribution of each individual controlling factor. When the best PESQ is obtained, we record the corresponding time consumption of training and sampling. Table 3 demonstrates the experimental results. The first experiment replaces the conditioner noisy waveform with noisy spectrogram as in [19]. The enhanced speech quality reduces from 2.69 to 2.59, indicating the effectiveness of the time-domain conditioning. We also see significant fast convergence and sampling speed in this setup. This is partly due to the absence of Fourier transform, and partly due to the time of up-sampling the speech spectrogram within the diffusion model. The second experiment replaces the noise level with time step compare to Eq. (5) and the objective function expression is:

$$\mathbb{E} \parallel \epsilon_* - \epsilon_\theta(x_t, y_0, t) \parallel_2^2 . \tag{16}$$

By the results we can find that using continuous noise level can effectively reduce the sampling time. The third experiment removes the deterministic module and we see a significant degradation of the enhanced speech quality, indicating the gain by "enhance-and-refine". However, we also see the deterministic module increases the convergence time to some extent, because it adds an additional generation process. In summary, the deterministic module and time-domain conditioning contribute the most to the performance improvement.

Table 3. Ablation experiments. The PESQ and the relative convergence/Sampling time as metrics for our experiments. The original SRTNet convergence and sampling time to 1.0.

Method	PESQ(↑)	Convergence/Sampling time(↓)
SRTNet	2.69	1.0 / 1.0
−waveform conditioner	2.59	4.86 / 1.22
−continuous noisy level	2.63	1.34 / 3.66
−deterministic module	2.58	0.83 / 0.88

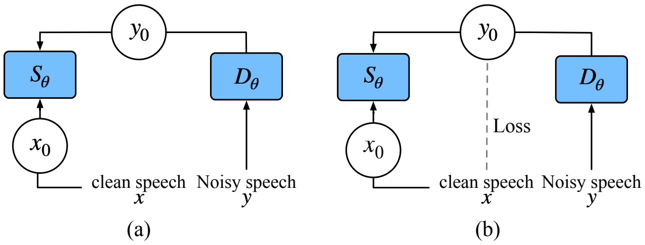

Fig. 4. Two variants of SRTNet. (a) Residual-free SRTNet, (b) Residual-free SRTNet with an additional loss.

3.2.5. Variants of SRTNet

Under the proposed "enhance-and-refine" paradigm, SRTNet uses two residuals as input to the diffusion model. It is natural to directly use clean speech and the initially enhanced noisy speech as input. Therefore, we design a variant of SRTNet, namely, Residual-free SRTNet as show in Fig. 4(a). In this structure, it is intuitive to assume that the final enhanced speech depends heavily on the output of the deterministic module. Therefore, another variant of the experiment is designed to ensure that the output of the deterministic model is as close to clean speech as possible by adding a loss between the output of the deterministic model and clean speech, as shown in Fig. 4(b). The results of the two variant experiments are shown in Table 4. The experiments demonstrate that Residual-free SRTNet decreases in all metrics but still higher than other baselines. This reflects the effectiveness of proposed "enhance-and-refine" structure. When an additional loss function is added, the performance degrades dramatically. We attribute the reason to the additional loss causing the original generative model to be no longer pure and thus unable to learn the true data distribution. In the previous models (SRTNet, Residual-free SRTNet), although there is not an individual loss function for the deterministic module, it still learns some useful information through the unified loss function.

Table 4. Variant experiments of SRTNet.

Method	PESQ(↑)	CSIG(↑)	CBAK(↑)	COVL(↑)
Residual-free SRTNet	2.61	3.73	3.01	3.04
Residual-free SRTNet+loss	2.25	3.59	2.90	2.90
SRTNet	**2.69**	**4.12**	**3.19**	**3.39**

4. CONCLUSION AND FUTURE DIRECTIONS

We propose SRTNet, a joint network for complete time domain speech enhancement. Our model achieves state-of-the-art performance for SE task in generative models while significantly reducing the time consumption of training. However, our model also has some limitations, which we will investigate in our next work. When the noise situation is more complex and the signal-to-noise ratio is extremely low, our model may appear to be under-exerted, i.e., the initial enhancement of the noise interference in the deterministic module is not very thorough, resulting in the stochastic module that does not recover the clean speech well.

5. ACKNOWLEDGEMENTS

This work was supported by Opening Project of Key Laboratory of Xinjiang, China (2020D04047), the National Key R&D Program of China (2020AAA0107902) and NSFC (61663044, 61761041).

6. REFERENCES

[1] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech Enhancement Generative Adversarial Network," in *Proc. Interspeech 2017*, 2017, pp. 3642–3646. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1428

[2] M. H. Soni, N. Shah, and H. A. Patil, "Time-Frequency Masking-Based Speech Enhancement Using Generative Adversarial Network," in *Proc. ICASSP*, 2018, pp. 5039–5043.

[3] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition," in *Proc. ICASSP*, 2018, pp. 5024–5028.

[4] 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 *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 2031–2041.

[5] D. Baby and S. Verhulst, "SERGAN: Speech Enhancement Using Relativistic Generative Adversarial Networks with Gradient Penalty," in *Proc. ICASSP*, 2019, pp. 106–110.

[6] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 7354–7363.

[7] G. Liu, K. Gong, X. Liang, and Z. Chen, "CPGAN: Context Pyramid Generative Adversarial Network for Speech Enhancement," in *Proc. ICASSP*, 2020, pp. 6624–6628.

[8] A. A. Nugraha, K. Sekiguchi, and K. Yoshii, "A Flow-Based Deep Latent Variable Model for Speech Spectrogram Modeling and Enhancement," *IEEE/ACM Trans. Audio, speech, Lang. Process.*, vol. 28, pp. 1104–1117, 2020.

[9] M. Strauss and B. Edler, "A Flow-Based Neural Network for Time Domain Speech Enhancement," in *ICASSP*, 2021, pp. 5754–5758.

[10] S. Leglaive, L. Girin, and R. Horaud, " A Variance Modeling Framework Based On Variational AutoEncoders For Speech Enhancement," in *Proc. MLSP*, 2018, pp. 1–6.

[11] Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, and T. Kawahara, "Statistical Speech Enhancement Based on Probabilistic Integration of Variational Autoencoder and Non-Negative Matrix Factorization," in *Proc. ICASSP*, 2018, pp. 716–720.

[12] S. Leglaive, L. Girin, and R. Horaud, "Semi-supervised Multichannel Speech Enhancement with Variational Autoencoders and Non-negative Matrix Factorization," in *Proc. ICASSP*, 2019, pp. 101–105.

[13] S. Leglaive, U. Şimşekli, A. Liutkus, L. Girin, and R. Horaud, "Speech Enhancement with Variational Autoencoders and Alpha-stable Distributions," in *Proc. ICASSP*, 2019, pp. 541–545.

[14] S. Leglaive, X. Alameda-Pineda, L. Girin, and R. Horaud, "A Recurrent Variational Autoencoder for Speech Enhancement," in *Proc. ICASSP*, 2020, pp. 371–375.

[15] M. Sadeghi and X. Alameda-Pineda, "Mixture of Inference Networks for VAE-Based Audio-Visual Speech Enhancement," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1899–1909, 2021.

[16] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating Gradients for Waveform Generation," in *International Conference on Learning Representations*, 2020.

[17] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A Versatile Diffusion Model for Audio Synthesis," in *International Conference on Learning Representations*, 2020.

[18] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, no. 4, pp. 465–494, 2011.

[19] Y.-J. Lu, Z.-Q. Wang, S. Watanabe, A. Richard, C. Yu, and Y. Tsao, "Conditional Diffusion Probabilistic Model for Speech Enhancement," in *Proc. ICASSP*, 2022.

[20] J. Whang, M. Delbracio, H. Talebi, C. Saharia, A. G. Dimakis, and P. Milanfar, "Deblurring via Stochastic Refinement," in *Proc. CVPR*, June 2022, pp. 16 293–16 303.

[21] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.

[22] Y.-J. Lu, Y. Tsao, and S. Watanabe, "A Study on Speech Enhancement Based on Diffusion Probabilistic Model," in *Proc. APSIPA ASC*, 2021, pp. 659–666.

[23] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *Proc. CASLRE 2013*.

[24] J. Thiemann, N. Ito, and E. Vincent, "The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND): A database of multichannel environmental noise recordings," in *Proceedings of Meetings on Acoustics*, vol. 19, no. 1, 2013, pp. 035–081.

[25] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.

[26] A. Rix, J. Beerends, M. Hollier, and A. 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, 2001, pp. 749–752 vol.2.

[27] 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.

[28] A. Défossez, G. Synnaeve, and Y. Adi, "Real Time Speech Enhancement in the Waveform Domain," in *Proc. Interspeech 2020*, 2020, pp. 3291–3295. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2409

[29] M. Abd El-Fattah, M. I. Dessouky, S. Diab, and F. Abd El-Samie, "Speech enhancement using an adaptive wiener filtering approach," *Progress In Electromagnetics Research M*, vol. 4, pp. 167–184, 2008.

[30] H. Phan, I. V. McLoughlin, L. Pham, O. Y. Chén, P. Koch, M. De Vos, and A. Mertins, "Improving GANs for speech enhancement," *IEEE Signal Processing Letters*, vol. 27, pp. 1700–1704, 2020.

[31] T.-A. Hsieh, H.-M. Wang, X. Lu, and Y. Tsao, "Wavecrn: An efficient convolutional recurrent neural network for end-to-end speech enhancement," *IEEE Signal Processing Letters*, vol. 27, pp. 2149–2153, 2020.

[32] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.