ENSEMBLE INFERENCE FOR DIFFUSION MODEL-BASED SPEECH ENHANCEMENT

Hao Shi, Naoyuki Kamo, Marc Delcroix, Tomohiro Nakatani, Shoko Araki*

NTT Corporation, Japan

ABSTRACT

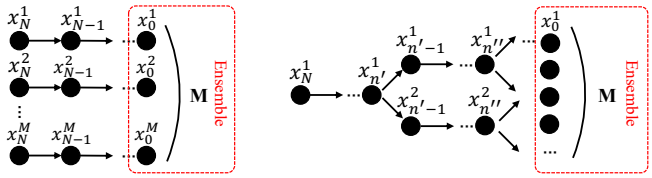
Diffusion-based speech enhancement (SE) is a probabilistic model that provides distribution of enhanced speech. Based on this property, we previously proposed ensemble inference and showed that solving a reverse Stochastic Differential Equation (SDE) multiple times and performing an ensemble over the obtained samples significantly improves the performance. Unfortunately, we failed to sufficiently explore our proposed ensemble inference. First, we only tested it on target speech extraction (TSE) and not on such general SE tasks as denoising. Second, sample generation greatly increased the computational complexity. Finally, the method considered all samples equally without heeding potential outliers. This paper addresses these issues and proposes a computationally efficient sample generation technique called SplitTree and ensemble inference combined with outlier removal to improve the ensemble's effectiveness. We conducted experiments using the WSJ-CHiME3 and LibriMix-2spk datasets for denoising and TSE tasks and confirmed the following: 1) Ensemble inference also helps denoising tasks; 2) SplitTree reduces the complexity of the ensemble inference by about 40% while maintaining a similar level of performance; and 3) Our proposed outlier removal improves the ensemble performance for TSE task.

***Index Terms*—** Ensemble inference, diffusion model, speech enhancement, SplitTree, target speaker extraction

1. INTRODUCTION

Speech enhancement (SE) aims to improve the quality of speech signals corrupted by other speakers or non-speech interferences. This is particularly important in real-world scenarios [1] where noise can significantly degrade the performance of speech-related applications. Deep neural network (DNN)-based SE systems [2] have been shown more powerful than traditional SE systems [3, 4], fueling research interest [2, 5, 6, 7]. These systems can be broadly categorized as either deterministic [6, 7, 8, 9] or probabilistic [10, 11, 12, 13] approaches with different processing ideas. Deterministic SE systems learn optimal deterministic mapping from noisy speech to clean speech [12]. On the other hand, probabilistic SE systems capture the target distribution, either implicitly or explicitly [10, 11, 12, 13].

Among probabilistic systems, diffusion models have received significant attention for their robust performance across various tasks [14, 15]. Diffusion models are inspired by non-equilibrium thermodynamics. The data are gradually transformed into noise, during which a neural network learns to reverse the incremental process of noise addition. The score-based diffusion model stands out with excellent performance for various SE tasks such as speech denoising, dereverberation, blind source separation, and target speech extraction (TSE) [11, 12, 16, 17]. This model is based on a stochastic differential equation (SDE), which makes the training fully probabilistic without any prior noise distribution assumptions [12, 18]. Its reverse diffusion process is also based on SDE [18].



(a) Ensemble inference (b) Proposed SplitTree ensemble inference

Fig. 1: Flowchart of ensemble inference (a) performing M independent reverse diffusion processes in parallel from $n = N$ to 0 and (b) starting a single reverse diffusion process and splitting it into several branches at its intermediate time steps ($n = n'$ and $n = n''$).

Because the reverse diffusion process involves numerous random components, they introduce local variations of the enhanced speech, and often cause the estimate largely deviate from the desirable values. In our previous research [16], we found that rather than performing a single reverse process, performing multiple reverse processes followed by computing their ensemble (Fig.1–(a)) significantly improved the enhancement. We call this approach ensemble inference. However, performing multiple reverse processes significantly increases the computational cost. The ensemble inference proposed for TSE [16] is not fully analyzed and untested for other SE tasks such as denoising.

In this paper, we extend ensemble inference to a denoising task and conduct deeper studies on it. We first confirm the effectiveness of ensemble inference for a denoising task and explore better ways to generate ensemble samples from two aspects:

- We propose SplitTree to more efficiently generate samples for ensemble inference. Instead of running several independent reverse processes to generate samples, SplitTree splits a single diffusion process at its intermediate steps to generate multiple samples. Because SplitTree shares certain parts of computations across different samples, it can largely reduce computational costs.

- Enhanced samples with poor quality unfavorably affect the ensemble inference's performance. We remove such outliers from the ensemble process to improve its performance.

The remainder of this paper is organized as follows. Section 2 provides an overview of diffusion-based SE for denoising and TSE. Section 3 describes ensemble inference with our proposed modifications. Section 4 gives the experimental settings and results. Section 5 gives the conclusion and future work.

*Hao Shi is with Kyoto University. This work was done during an Internship at NTT Corporation.

2. SPEECH ENHANCEMENT USING SCORE-BASED DIFFUSION MODEL

In this paper, we investigate two distinct SE tasks, denoising and TSE. To simplify the derivation, we provide a general formulation for both tasks. The signal received by a microphone is represented as follows:

$$y = x_0 + v, \tag{1}$$

where y, x_0, and $v \in \mathbb{C}^{N_f \times N_t}$ denote complex spectra of the microphone signal, the clean speech, and the interference signal. N_f and N_t are numbers of frequencies and time frames, respectively. For the denoising task, v represents the additive noise, and for the TSE task, it represents the interference speakers. Although these two tasks aim to recover x_0 from y, there are some differences. For the denoising task, only y is needed as the input feature, while for the TSE task, an additional enrollment utterance of target speaker c is needed as a speaker clue to identify the target in the mixture.

2.1. Stochastic Process in Score-based Diffusion Model

A diffusion model is defined by a forward stochastic differential equation (SDE). For the denoising and TSE tasks, it transforms a clean speech x_0 to an observed speech y plus a white Gaussian noise [12, 16]. The SE tasks can then be achieved by reversely solving the forward SDE using a reverse SDE [18, 19]. This requires a score $\nabla_{x_t} \log p_t(x_t|y)$ on a state of the SDE, x_t, at each continuous state index $t \in [0, T]$ [18, 20]. To approximate the score, we use a neural network s_θ with parameter set θ, called a score model.

The reverse SDE can be expressed using the score model:

$$\mathrm{d}x_t = \left[-\mathrm{f}(x_t, y) + \mathrm{g}(t)^2 s_\theta\right] \mathrm{d}t + \mathrm{g}(t)\mathrm{d}\bar{\mathrm{w}}, \tag{2}$$

where $\bar{\mathrm{w}}$ is a standard Wiener process, and f and g are drift and diffusion coefficient functions, respectively. The SE tasks are achieved by solving the reverse SDE from $t = T$ to 0. Different conditions are posed to the score models for the denoising and TSE tasks. The denoising is conditioned only on an observed speech y, and follows conditional probability $p_t(x_t|y)$. The TSE is also conditioned on speaker clue c, and follows conditional probability $p_t(x_t|y, c)$. Thus, the input arguments of the score models for the denoising and TSE are defined as $s_\theta(x_t, y, t)$ and $s_\theta(x_t, y, c, t)$.

2.2. Training Objective for Score-based Diffusion Model

The training loss to determine model parameters θ is defined based on the Mean Square Error (MSE) between true and estimated scores. The overall training objectives for denoising and TSE are derived as follows [12, 16]:

$$\arg\min_\theta \mathbb{E}\left[\left\|\mathrm{s}_\theta + \frac{\mathrm{z}}{\sigma(t)}\right\|_2^2\right], \tag{3}$$

where $z \sim \mathcal{N}_C(\mathrm{z}; 0, I)$ is a sampled white Gaussian noise with I being an identity matrix, and $\sigma(t)^2$ is the variance of the white Gaussian noise included in x_t according to the forward SDE.

2.3. Inference

The inference procedure by the reverse SDE, Eq. (2), starts at $t = T$ and iteratively goes backward to $t = 0$. Starting state x_T of the reverse process at $t = T$ is sampled as follows [12]:

$$x_T \sim \mathcal{N}_c(x_T; y, \sigma(T)^2 \mathrm{I}). \tag{4}$$

To numerically find the solution of the reverse SDE , the interval $[0, T]$ is partitioned into N steps of width $\Delta t = T/N$, and we utilize the discrete reverse process over $\{x_T, x_{T-\Delta t}, ..., x_0\}$. We employ the so-called Predictor-Corrector (PC) samplers [18] to numerically solve the SDE. For each step, the current state is determined using both predictor and corrector methods by referencing the state from the previous step. Each of the predictor and corrector steps requires one call of score model s_θ. Note that at each step, white Gaussian noise is introduced according to Eq. (2) both by the predictor and corrector. Consequently, the inference results depend on the seed of the random generator in the actual implementation.

3. ENSEMBLE INFERENCE

Because the inference is stochastic, running the inference process with different random seeds leads to different enhanced signals sampled from $p_0(x_0|y)$ (or $p_0(x_0|y, c)$ for TSE). In our previous work[16], we exploited this property to improve the estimation by performing ensemble inference and obtained enhanced speech $\bar{x}_0$ by averaging multiple generated samples x_0^m ($1 \le m \le M$):

$$\bar{x}_0 = \frac{\sum_{m=1}^{M} x_0^m}{M}. \tag{5}$$

Although ensemble inference works well with Eq. 5, it is time-consuming to get K instances of x_0. Poor-quality enhanced samples are also ensembled during the inference. To overcome these issues, in the following, we propose two modifications of the ensemble average approach, i.e., SplitTree and outlier removal, to improve the computational efficiency and effectiveness of the ensemble process.

3.1. SplitTree Ensemble Inference

Although ensemble inference significantly improved the quality of the enhanced speech, unfortunately, it increased the decoding time. For example, in our previous work [16], shown in Fig. 1-(a), we generated M samples for ensemble inference by sampling M different initial states, i.e., x_T in Eq. (4), of the reverse SDE process. Therefore, the entire reverse diffusion process must be run M times; It evokes MN "calls" of the PC sampling using deep neural network when discretizing each reverse diffusion process into N steps. We showed that this approach was successful for ensemble inference in terms of enhanced speech quality. However, increasing the number of calls for the ensemble inference significantly lengthens the decoding time.

As explained in Section 2.3, noise is introduced not only at the initial step $t = T$ but also at each step t of the reverse process. This means that we can also generate multiple noise samples at intermediate steps in the reverse process as in Fig. 1-(b). It allows us to share a part of the computations across the reverse processes. This approach is called SplitTree. Even though the interval $[0, T]$ is still divided into N steps, fewer *calls* are needed when splitting the reverse process at intermediate steps. Hereafter, we call a step at which we split the process a split point, and refer to it by an integer from $n = 0$ to N corresponding to the discrete step of the diffusion process from $t = 0$ to T. We may put a single or multiple split points in the process. For example, when we split the process into M branches at a single split point n ($< N$), the number of calls required by SplitTree is $N + (n + 1)(M - 1)$, which can be much smaller when using $n \ll N$ in comparison with MN.

Note also that using sufficiently diverse samples is crucial for ensemble inference. The sample diversity will decrease if the split points are put only at the latter stages of the inference process since

the variance of the added noise decreases in the latter stages. We next examine a tradeoff between performance and efficiency in our experiments.

3.2. Outlier Removal

When some samples generated for ensemble inference have extremely poor quality, they degrade the ensemble inference. We avoid this problem by introducing outlier removal, which detects poor samples as outliers and removes them before calculating the ensemble. We adopt for outlier detection a simple probabilistic thresholding that assumes that generated samples are normally distributed and detects a sample as an outlier when it deviates probabilistically from the mean of the distribution by more than a certain threshold η. Sample x_0^m is detected as an outlier when distance $D[m]$ defined below is $D[m] > \eta$:

$$d[m,l] = \frac{\sum_{n\in S_l} |x_0^m[i] - \bar{x}_0[i]|^2}{\sum_{n\in S_l} \beta^2[i] + \tau}, \tag{6}$$

$$\beta^2[i] = \frac{1}{M}\sum_m |x_0^m[i] - \bar{x}_0[i]|^2, \tag{7}$$

$$D[m] = \frac{1}{L}\sum_l d[m,l], \tag{8}$$

where $x_0^m[i]$ and $\bar{x}_0[i]$ are the time domain signals for x_0^m and $\bar{x}_0$ in Eq. (5) at time index i, $\beta^2[i]$ is a sample variance at i, and $d[m,l]$ is the deviation of the m-th sample within each signal segment S_l ($1 \le l \le L$) with flooring constant τ. Distance $D[m]$ is determined to be the average of $d[m,l]$ over all the segments.

In preliminary experiments, we observed the outliers tended to occur within a relatively short time segment. Thus, we first calculated the deviation within each segment in Eq. (6), and averaged it over a whole utterance in Eq. (8) to determine the outliers. We then performed utterance-wise ensemble.

4. EXPERIMENTS

4.1. Datasets

We performed experiments on both the denoising and TSE tasks. For denoising task, we synthesized the WSJ0-CHiME3 dataset by combining clean speech utterances from the Wall Street Journal (WSJ0) dataset [21] with noise signals from the CHiME3 dataset [22]. Each observed signal was created by randomly selecting a noise file and combining it with clean utterance. Every utterance was employed only once, and the Signal-to-Noise Ratio (SNR) was uniformly sampled within a range of 0 to 20 dB for the training, validation, and test sets.

For the TSE task, we performed experiments using the openly available LibriMix-2spk dataset [23]. We used the 100-hours version of the data and followed the openly available recipe[1] that defines the enrollment utterances used for each mixture in the test set.

4.2. Settings

The network architecture and training strategies were consistent with those of SGMSE+ for both the denoising and TSE tasks[2] [12]. Since the diffusion process is defined in the complex short-time Fourier transform (STFT) domain, we used the concatenation of the real and imaginary parts of the signals for the input and output features of the score model. The Noise Conditional Score Network (NCSN++) architecture was used for the score model. BigGAN architecture was used for the residual blocks in the upsampling and downsampling layers. Two or three residual blocks were in each upsampling or downsampling layer. Global attention was added at a resolution of 16×16 and in the bottleneck layer.

For the TSE task, in addition to the input for the denoising model, we need to feed the enrollment utterance to it. To extract the speaker embedding vector from the enrollment utterance, we used an additional neural network, called clue encoder. The network structure consisted of three BLSTM layers. Speaker embedding was created by averaging the output values from the clue encoder over all the time frames and multiplied with the output of the second resblock in NCSN++. As described in our previous work [16], we also proposed a multi-task (MT) objective model for the TSE task, referred to as Diff-TSE-MT, by combining diffusion model and conventional deterministic TSE model. We used the Diff-TSE-MT model in this paper.

We evaluated enhancement performance using the following schemes: Perceptual Evaluation of Speech Quality (PESQ), Extended Short-Time Objective Intelligibility (ESTOI), Scale-Invariant (SI-) Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR), and Signal-to-Artifact Ratio (SAR).

For an experiment of outlier removal, we investigated multiple values for threshold η but fixed τ to 1×10^{-4} and the segment size to 2048.

Table 1: Performance of Ensemble with different numbers of enhanced samples. ESTOI utilizes a percentage value (%).

Model	PESQ	ESTOI	SI-SDR	SI-SIR	SI-SAR
Denoising					
1 sample	2.84	91.93	16.52	33.53	16.61
En. 2	3.01	92.97	17.32	33.54	17.43
En. 4	3.11	93.49	17.82	33.57	17.95
En. 8	3.16	93.79	18.10	33.56	18.24
En. 10	3.17	93.86	18.16	33.57	18.30
TSE					
1 sample	2.79	77.26	9.40	44.31	9.40
En. 2	2.94	79.17	10.46	44.48	10.45
En. 4	3.04	80.26	11.16	44.60	11.15
En. 8	3.08	80.89	11.58	44.68	11.57
En. 10	3.10	81.04	11.67	44.69	11.66

4.3. Ensemble Analysis

Table 1 shows the enhancement performance for both the denoising and TSE tasks with and without ensemble inference for different numbers of samples. The results show that ensemble inference was effective for both tasks, respectively improving SI-SDR by 1.6 and 2.3 dB for the denoising and TSE tasks. Ensemble inference was effective even when using just two samples; the performance further improved with up to eight samples. After that, the improvement became less significant. In the following, we set the number of samples to eight. By comparing the performances across different ensemble numbers, it becomes evident that the Ensemble method primarily reduces signal distortion and artifacts, although it has less impact on reducing interference, as seen by the SI-SIR values.

[1] https://github.com/butspeechfit/speakerbeam

[2] https://github.com/sp-uhh/sgmse

Table 2: SplitTree performance with a reverse process splitted at different "Split points" to "# Splits" branches. "# Calls" denotes the total number of PC sampling calls.

	Split points	# Splits	#Calls	PESQ	SI-SDR
(1)	-	1	30	2.84	16.52
(2)	30	8	240	3.16	18.10
(3)	21	8	177	3.14	18.10
(4)	11	8	107	3.05	17.73
(5)	30, 21	2, 4	186	3.15	18.08
(6)	30, 21, 11	2, 2, 2	146	3.13	18.06

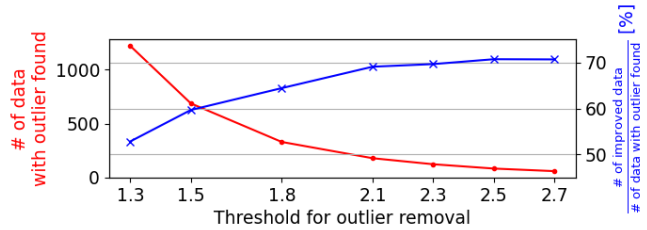


Fig. 2: Effect of threshold η on number of mixtures with outliers found (red line) and effectiveness of Ensemble with outlier removal (blue line) measured as the ratio of the amount of improved test data by outlier removal over the amount of test data with outliers found.

4.4. Evalutation of SplitTree

Next, we analyzed the effect of the proposed SplitTree to improve the efficiency of ensemble inference. Table 2 shows its results for a denoising task with different splitting strategies. We fixed the number of reverse diffusion steps to $N = 30$ and the total number of generated samples to $M = 8$. As shown in the table, (2) corresponds to splitting the process at the initial step ($n = 30$), as we previously proposed in [16]. A single split point was set for (3) and (4), and multiple split points were set for (5) and (6). Comparing experiments (2) and (3), splitting the SDE process at an intermediate step was also very effective because it significantly reduced the number of calls while maintaining consistent performance. Comparing experiments (3) and (4), splitting at a smaller n reduced the number of calls but also lowered the performance. This observation indicates that when n is too small, the added noise power is unable to generate enough variation in the generated samples. Performing splitting at different steps (i.e., experiments (5) and (6)) can further reduce the number of calls by up to 40 % while maintaining performance.

4.5. Evaluation of Outlier removal

Finally, we analyzed the impact of our proposed outlier removal for the TSE task. First, we investigated how to set threshold value η. Fig. 2 shows the amount of test data with outliers found when varying threshold η and the proportion of the improved test data. Setting η to a small value (e.g., $\eta = 1.3$) means that many samples are considered outliers, which reduces the benefit of ensemble inference since many valid samples are discarded. Increasing threshold η reduces the number of outliers found (red line) but ensures that Ensemble is more effective (blue line). In the following, we set η to 2.5.

Figure 3 shows the SI-SDR improvement with outlier removal as a function of the SI-SDR of the original ensemble without removal (for simplicity, we only show test data where outliers were found by our proposed method). Outlier removal successfully improved the SI-SDR in most cases. Note that the test data with low SI-SDR values, e.g., below 0 dB, are degraded. These data points constitute extraction failures for which the target speaker was not correctly identified in the mixture for most samples. In such cases, our proposed outlier removal naturally cannot improve the performance as most of the samples correspond to the interference and not the target speaker.

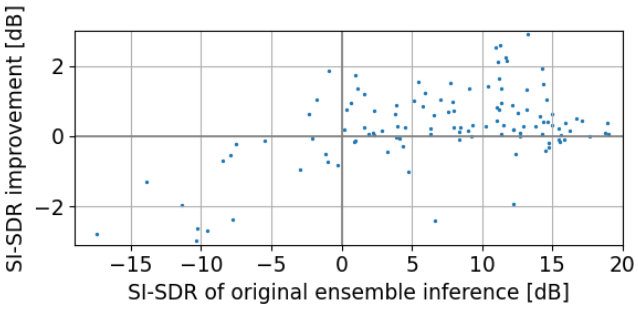
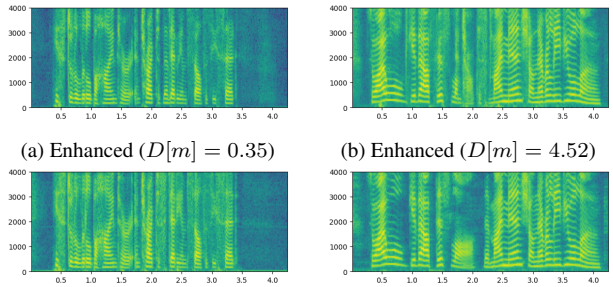


Fig. 3: Effect of outlier removal with $\eta = 2.5$



(a) Enhanced ($D[m] = 0.35$) (b) Enhanced ($D[m] = 4.52$)

(c) Ground truth of target (d) Ground truth of interference

Fig. 4: Spectrograms

Figure 4 shows the spectrograms of the enhanced samples and the ground-truth targets and the interference signals for a test mixture, where the SI-SDR performance was improved by 3.60 dB with outlier removal. With a threshold value of $\eta = 2.5$, the proposed method identified sample (b) as an outlier, which is indeed an extraction failure since its spectrogram resembles the interference.

Note that we also tested outlier removal for a denoising task, although we did not observe any improvement because the identification failure for a target speaker cannot happen with it. This result suggests the potential improvement is limited.

5. CONCLUSIONS

We investigated ensemble inference for diffusion model-based TSE and denoising tasks and proposed two extensions: outlier removal to improve its effectiveness and SplitTree to raise its computational efficiency. Experimental results showed that ensemble inference was effective for both the denoising and TSE tasks and primarily contributed to reducing the artifacts. We also showed that our proposed outlier removal procedure could improve TSE performance and that ensemble inference's efficiency was raised by up to 40% using the proposed SplitTree approach. In the future, we will investigate additional methods to identify outliers and further enhance the sample generation speed.

6. REFERENCES

[1] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An Overview of Noise-Robust Automatic Speech Recognition," *IEEE/ACM TASLP*, vol. 22, no. 4, pp. 745–777, 2014.

[2] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A Regression Approach to Speech Enhancement Based on Deep Neural Networks," *IEEE/ACM TASLP*, vol. 23, no. 1, pp. 7–19, 2015.

[3] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in *Proc. ICASSP*, 2002, vol. 4, pp. IV–4164–IV–4164.

[4] Y. Ephraim and H.L. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 251–266, 1995.

[5] M. Kolbæk, Z.-H. Tan, S. H. Jensen, and J. Jensen, "On Loss Functions for Supervised Monaural Time-Domain Speech Enhancement," *IEEE/ACM TASLP*, vol. 28, pp. 825–838, 2020.

[6] H. Shi, L. Wang, M. Ge, S. Li, and J. Dang, "Spectrograms Fusion with Minimum Difference Masks Estimation for Monaural Speech Dereverberation," in *Proc. ICASSP*, 2020, pp. 7544–7548.

[7] 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," 2021.

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

[9] A. Li, C. Zheng, L. Zhang, and X. Li, "Glance and gaze: A collaborative learning framework for single-channel speech enhancement," *Applied Acoustics*, vol. 187, pp. 108499, 2022.

[10] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech Enhancement Generative Adversarial Network," in *Proc. Interspeech*, 2017, pp. 3642–3646.

[11] 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, pp. 7402–7406.

[12] J. Richter, S. Welker, J.-M. Lemercier, B. Lay, and T. Gerkmann, "Speech Enhancement and Dereverberation With Diffusion-Based Generative Models," *IEEE/ACM TASLP*, vol. 31, pp. 2351–2364, 2023.

[13] 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.

[14] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. NIPS*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, Curran Associates, Inc.

[15] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Proc. NIPS*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, Eds. 2021, vol. 34, pp. 21696–21707, Curran Associates, Inc.

[16] N. Kamo, M. Delcroix, and T. Nakatani, "Target Speech Extraction with Conditional Diffusion Model," in *Proc. Interspeech*, 2023, pp. 176–180.

[17] H. Yen, F. G. Germain, G. Wichern, and J. L. Roux, "Cold diffusion for speech enhancement," in *Proc. ICASSP*, 2023, pp. 1–5.

[18] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. ICLR*, 2021.

[19] C.-W. Huang, J. H. Lim, and A. C. Courville, "A variational perspective on diffusion-based generative models and score matching," in *Proc. NIPS*, 2021, vol. 34, pp. 22863–22876.

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

[21] John S. Garofolo, David Graff, Doug Paul, and David Pallett, "CSR-I (WSJ0) complete," `https://catalog.ldc.upenn.edu/LDC93S6A`.

[22] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *Proc. 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 504–511.

[23] Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent, "Librimix: An open-source dataset for generalizable speech separation," 2020.