ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888705

Modifying Flow Matching for Generative Speech Enhancement

Roman Korostik, Rauf Nasretdinov, Ante Jukić
NVIDIA
{rkorostik, rnasretdinov, ajukic}@nvidia.com

***Abstract*—Diffusion-based generative models have been shown to be highly effective in various speech enhancement tasks. This work presents an analysis of a flow matching-based framework for generative speech enhancement as a simpler alternative to diffusion. Four different modifications to flow matching are proposed, employing an informed prior, a data prediction loss, deterministic inference, and early stopping. The proposed variants are evaluated on speech denoising, demonstrating performance comparable to a previous state-of-the art model using the same data setup. Through ablation studies, an efficient deterministic one-step inference configuration is proposed, which does not require any advanced training techniques such as pre-training or distillation. The proposed variants are also evaluated on speech dereverberation, demonstrating that stochastic inference without informed prior is preferable for this task.**

***Index Terms*—flow matching, diffusion, speech enhancement, speech restoration**

I. Introduction

The task of speech enhancement is to recover the clean speech signal from a potentially corrupted recording. Corruptions present in the recorded signal may include, for example, environmental noise, reverberation, clipping or lossy encoding artifacts. Reducing the undesired components and extracting the high-quality speech can be useful for humans (increasing intelligibility, reducing listener fatigue) [1] as well as computers (e.g. increased performance of speech recognition [2], speaker verification [3], text-to-speech [4]).

Recent developments in deep learning have demonstrated superiority of the data-driven approach to speech enhancement [5] over handcrafted algorithms [6], [7]. Moreover, recent developments have demonstrated high effectiveness of generative models in tasks such as denoising and dereverberation. Several generative models for speech enhancement have been proposed recently, including diffusion-based models [8]–[15], a hybrid of a predictive and a diffusion-based model [12], Schrödinger bridge for paired data [16]. While allowing for impressive audio quality, mentioned models are rather inefficient and typically require 20 or more evaluations of the potentially large neural backbone estimator. This limits their practical applicability, especially for on-device, low-power and low-latency applications.

This work proposes to use the flow matching framework for the task of speech enhancement. Compared to diffusion models [17], [18] and Schrödinger bridges for paired data [19], flow matching [20] provides a simpler and easier to understand formulation. In particular, it does not require a rather complicated machinery of stochastic differential equations. This results in a simpler implementation that is less error-prone, and has greater practical potential for modifications. There has been work on unsupervised pre-training of large flow matching models with fine-tuning for speech enhancement (e.g. SpeechFlow [21]). However, in this work we choose to focus on modifications to the base flow matching model not involving large scale pre-training, for a direct comparison with previously-proposed generative models [11], [12], [16].

We propose two training-time and two inference-time modifications to the baseline flow matching model. The proposed approaches are evaluated on two independent tasks of speech denoising and speech dereverberation. We demonstrate that flow matching models provide good performance with as little as five inference steps. Moreover, we discover a simple and efficient one-step inference regime for the denoising task providing audio quality and machine intelligibility comparable to prior work. For dereverberation, we demonstrate that the baseline model is preferrable to proposed modifications. A one-step inference setup that highly improves machine intelligibility is presented.

II. Conditional Flow Matching

We present a brief overview of Conditional Flow Matching (CFM). A more detailed overview of CFM can be found [20].

Model definition. Let $p_{\text{data}} = \{x_1^{(i)}\}_{i=1}^N$ be our dataset of size N. For every datapoint x_1, consider a continuous probability path p_t: a family of distributions describing how noise transforms into this datapoint over continuous time $t \in [0, 1]$:

$$p_t(x|x_1) \quad \text{s.t.} \quad p_0 = p_{\text{noise}}, \quad p_1 \approx \delta(x_1). \tag{1}$$

We call p_t a *conditional* path because its marginals are conditioned on x_1. Next, consider a conditional flow $\phi_t(x|x_1)$, a deterministic function which describes where a sample from p_0 would end up at time t according to the path. Assume ϕ_t is a solution for the following ordinary differential equation (ODE) boundary value problem:

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi_t(x|x_1) = v_t\left(\phi_t(x|x_1)\right), \quad \phi_0(x|x_1) = x. \tag{2}$$

A path p_t is generated by v_t if the noise distribution p_0 pushed through the flow ϕ_t is equal to p_t for all time points. If we restrict p_t to paths with Gaussian marginals $p_t(x|x_1) = \mathcal{N}\left(\mu_t(x_1), \sigma_t^2(x_1)\right)$, and ϕ_t to affine flows $\phi_t(x|x_1) = \mu_t(x_1) + x\sigma_t(x_1)$, the conditional vector field

$u_t(x|x_1)$ generating the conditional path $p_t(x|x_1)$ has the following form

$$u_t(x|x_1) = \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}\left(x - \mu_t(x_1)\right) + \mu_t'(x_t) \tag{3}$$

Given the mean and the variance functions μ_t and σ_t^2, the conditional vector field u_t generating conditional paths p_t can be derived. Therefore, the generative process for each of the known datapoints can be simulated.

Training. The goal is to sample from $p(x_1|x_0)$. This can be done by simulating (2). However, the conditional vector field cannot be easily computed using (3) since x_1 is unknown at inference time. The solution is to train a conditional vector field estimator for arbitrary point in space and time. For a neural network $v_\theta(x,t)$ with parameter set θ, the objective is

$$L_{\text{CFM}}(\theta) = \mathbb{E}_{x_1}\int_0^1 \mathbb{E}_{x\sim p_t}\left|v_\theta(x,t) - u_t(x|x_1)\right|^2 \,\mathrm{d}t, \tag{4}$$

which is regression to the conditional vector field computed at samples from the conditional path. This objective is also called $\mathbf{v}$-prediction [22] or velocity prediction [19].

In practice, (4) is optimized using stochastic gradient descent. For the inner expectation and for the integral over time we use single sample Monte-Carlo approximations, with $x_1 \sim p_{\text{data}}, t \sim U[0,1], x \sim p_t(x|x_1)$.

Inference. Given an estimator v_θ, the sampling is performed by numerically solving a boundary value ODE problem on the time interval $[0,1]$

$$\mathrm{d}x = v_\theta(x,t)\,\mathrm{d}t, \quad \hat{x}_0 \sim p_0 \tag{5}$$

The final state $\hat{x}_1$ is the sample from the model.

III. Flow Matching for Speech Enhancement

Let $\mathbf{y}$ denote the observed speech signal, possibly corrupted by noise, reverberation, and other undesired components, and let $\mathbf{s}$ denote its clean counterpart. Building a generative speech enhancement model means to provide a sampling procedure for $p(\mathbf{s}|\mathbf{y})$.

A. Flow matching modifications

Baseline model. Let $p_{\text{data}} = \{(\mathbf{s},\mathbf{y})^{(i)}\}_{i=1}^N$ denote a paired speech dataset of size N. We use the flow matching framework to model $p(\mathbf{s}|\mathbf{y})$. Following the optimal transport CFM (OT-CFM) [20] variant, μ_t and σ_t are defined as

$$\begin{aligned}\mu_t(\mathbf{x}|\mathbf{s}) &= t\mathbf{s},\\ \sigma_t(\mathbf{x}|\mathbf{s}) &= (1-t)\sigma_{\max} + t\sigma_{\min}\end{aligned} \tag{6}$$

Hyperparameters $\sigma_{\min}$ (typically small, e.g. 10^{-8}) and $\sigma_{\max}$ are standard deviations at the start and end of the path. The corresponding conditional vector field can be expressed as

$$u_t(\mathbf{x}|\mathbf{s}) = \frac{\sigma_{\max}\mathbf{x} - \sigma_{\min}(\mathbf{x}-\mathbf{s})}{(1-t)\sigma_{\max} + t\sigma_{\min}}. \tag{7}$$

A neural estimator v_θ with parameters θ accepts the corrupted signal $\mathbf{y}$ as an additional input, i.e., $v_\theta = v_\theta(\mathbf{x},t,\mathbf{y})$. This results in the following training objective

$$L_{\text{SE-OT-CFM}}(\theta) = \mathbb{E}_{(\mathbf{s},\mathbf{y}),t,\mathbf{x}\sim p_t}\left\|v_\theta(\mathbf{x},t,\mathbf{y}) - u_t(\mathbf{x}|\mathbf{s})\right\|_2^2. \tag{8}$$

Informed prior (IP). The described baseline model does not directly incorporate the corrupted speech signal $\mathbf{y}$ into the assumed generative process, using it merely as a conditional input to the neural estimator v_θ. However, the flow matching framework allows us to easily introduce this information into the prior distribution p_0, and into the whole conditional path, as

$$\begin{aligned}\mu_t(\mathbf{x}|\mathbf{y},\mathbf{s}) &= (1-t)\mathbf{y} + t\mathbf{s},\\ \sigma_t(\mathbf{x}|\mathbf{y},\mathbf{s}) &= (1-t)\sigma_{\max} + t\sigma_{\min}\end{aligned} \tag{9}$$

The corresponding conditional vector field is

$$u_t(\mathbf{x}|\mathbf{y},\mathbf{s}) = \frac{\sigma_{\max}(\mathbf{x}-\mathbf{y}) - \sigma_{\min}(\mathbf{x}-\mathbf{s})}{(1-t)\sigma_{\max} + t\sigma_{\min}} \tag{10}$$

This results in the following training objective

$$L_{\text{SE-OT-CFM}_{\text{IP}}}(\theta) = \mathbb{E}_{(\mathbf{s},\mathbf{y}),t,\mathbf{x}\sim p_t}\left\|v_\theta(\mathbf{x},t,\mathbf{y}) - u_t(\mathbf{x}|\mathbf{y},\mathbf{s})\right\|^2. \tag{11}$$

This can be considered a simpler flow matching counterpart to previously proposed SGMSE+ [11] and BBED [15] diffusion-based models for speech enhancement: these models include $\mathbf{y}$ in the drift coefficient of SDEs used, which, in the above sense, introduces an informed prior.

Data prediction (DP). The original definition of the flow matching model [20] uses the ODE velocity field as the regression target for the estimator. We propose training an estimator x_θ to directly predict the output data (this is also known as $\mathbf{x}$-prediction [22]). Speech signals have a natural structure, so it is reasonable to expect better prediction from a neural network when compared to predicting an abstract vector field. This also means that regression targets have the same magnitude for all timesteps, which should aid the optimization process. Switching from predicting the ODE velocity field to predicting data can also be interpreted as re-weighing loss components for different time steps, giving more weight to lower-noise time regions [22]. The corresponding training objective can be written as

$$L_{\text{SE-OT-CFM}_{\text{DP}}}(\theta) = \mathbb{E}_{(\mathbf{s},\mathbf{y}),t,\mathbf{x}\sim p_t}\left\|x_\theta(\mathbf{x},t,\mathbf{y}) - \mathbf{s}\right\|^2 \tag{12}$$

The corresponding ODE (5) can then be written as

$$dx = u_t\left(x|x_\theta(x|\mathbf{y},t)\right) \tag{13}$$

Therefore, at inference time the velocity field is estimated indirectly, by taking the neural estimate x_θ as the desired target $\mathbf{s}$ and plugging it in (7) or (10), depending on the chosen prior.

B. Inference-time modifications

Starting from the mean (SfM). A sensible inference procedure should start from p_0 to match the training-time setup. This means starting from a noise sample

$$\hat{\mathbf{x}}_0 = \mu_0 + \sigma_0\mathbf{z}, \quad z \sim \mathcal{N}(\mathbf{0},\mathbf{I}),$$

with μ_0 and σ_0 in (6) or (9). However, adding the noise sample $\mathbf{z}$ could be omitted, starting the inference from the path mean rather than from a path sample as $\hat{\mathbf{x}}_0 = \mu_0$. Note that this procedure is fully deterministic. Initial experiments showed that in case of the baseline flow matching model starting from

the mean, i.e., zero in (6), harmed performance tremendously. However, it is a simple and interesting modification to the inference procedure for flow matching with informed prior, and its performance is evaluated in the experimental section.

Early stopping (ES). Not adding noise forces the speech enhancement model to work in a regime it has not been trained in. Initial experiments demonstrated that this can cause the model to over-suppress speech, causing significant artifacts. However, for $t \in [0.8, 0.9]$, the output audio is sufficiently enhanced, even if at times leaving some original noise intact, e.g., for examples with large negative SNR values. We thus suggest early stopping, defined as specifying the end time for inference process to be less than 1. For experimental evaluations in this work we choose stopping time to be 0.85 as a middle point between 0.8 and 0.9. In practice this hyperparameter can be tuned on a separate validation set.

Late start. Previous work on diffusion models for speech enhancement [15] suggested starting late, even as late as 0.5. In initial experiments, we observed inconsistent quality changes when starting at time 0.05 compared to starting at time 0. Moreover, we observed major degradation of all metrics when starting at time 0.1 and later. We did not consider late start for further experiments.

IV. Experimental Setup

Two tasks are considered in the experimental section: speech denoising and speech dereverberation.

Data. Our data setup matches [12], [16]. We use WSJ0 [23] as the clean 16 kHz speech dataset. Denoising dataset is prepared by mixing clean speech with noise from the CHiME3 dataset [24] at SNR values uniformly distributed in $[-6, 14]$ dB. Dereverberation dataset is prepared by convolving clean speech with simulated room responses [25] and reverberation time in $[0.4, 1.0]$s. Utterances are randomly split into train ($\sim$25 hours), validation ($\sim$2 hours), and test ($\sim$1.5 hours) subsets. Audio sampling rate is 16 kHz.

Signal representation. We represent speech signals as complex spectrograms, computed using STFT with 510-sample window ($\approx$32 ms) with and 128-sample hop (8 ms). From an utterance uniformly sampled from the training set we sample 256 frames ($\approx$2s), resulting in training examples $\mathbf{y}_c \in \mathbb{C}^{256\times256}$. We perform element-wise magnitude compression with $a = 0.5$ and $b = 0.33$ as $\mathbf{y} = b|\mathbf{y}_c|^a \mathrm{e}^{j\angle \mathbf{y}_c}$. Real and imaginary parts are stacked, interpreting $\mathbf{y}$ as a two-channel image in $\mathbb{R}^{2\times256\times256}$. The paired clean data is constructed analogously. When the generative processing is finished, we obtain the enhanced time-domain signal by inverting the (reversible) input transformations.

Neural model. We use NCSN++ [17], which is a U-net-like multi-resolution convolutional network, conditioned on the process time t, with approximately 25 million parameters. The main input is current spectrogram estimate stacked together with input spectrogram $\mathbf{y}$, and the output has the shape of the spectrogram estimate. We use bilinear upsampling and downsampling for changing resolutions. For the baseline model, we use $\sigma_{\text{min}} = 10^{-8}$ and $\sigma_{\text{max}} = 1$. For the model with informed prior we switch to $\sigma_{\text{max}} = 0.3$ to approximately match prior variance from [11]. A concurrent work investigated this hyperparameter in detail [26] for diffusion models.

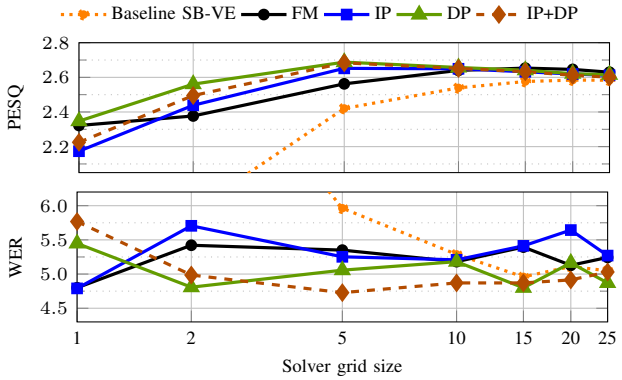

Fig. 1. Audio quality and WER after denoising. Some of the results for the baseline are out of range.

Inference. Sampling is done using a first-order Euler ODE solver on a uniformly spaced time grid. While higher order ODE solvers and non-uniform time grids could possibly improve performance, they are out of scope of this paper. We start inference at time 10^{-8}.

Metrics. Performance is evaluated using perceptual evaluation of speech quality (PESQ) [27], extended short-term objective intelligibility (ESTOI) [28], scale-invariant signal-to-distortion ratio (SI-SDR) [29] and word error rate (WER). WER is computed using NVIDIA's FastConformer-Transducer-Large English ASR model [30] with greedy decoding.

Training. All models were trained using Adam [31] on 8 NVIDIA V100 GPUs with global minibatch of size 64 and learning rate 10^{-4} [12], [16]. Early stopping triggered when SI-SDR averaged over the first 50 utterances in the validation set failed to improve for 100 epochs with validation performed every 5 epochs. The best model was chosen based on validation PESQ. Based on the initial experiments, we settled on ten inference steps for computing validation and test metrics.

Baselines. We consider Schröedinger Bridge with paired data [16], [19] to be a good baseline for comparison since it typically outperforms SGMSE+ [11] and StoRM [12] in both tasks of denoising and dereverberation [16]. SB-VE variant with SDE inference is evaluated at different grid sizes for both denoising and dereverberation. Evaluation metrics for both SB and SGMSE+ [11] for fifty inference steps are taken from [16], to set illustrative baselines for one-step FM.

V. Results

Audio examples can be found on the demo page[1].

A. Denoising

Fig. 1 shows WER and PESQ performance of different flow matching model modifications on denoising task for different number of solver steps. With ten or more steps, all variants reach the same audio quality plateau, whereas for $1, 2, 5$ steps

[1]https://racoiaws.github.io/icassp2025-fm-for-se-samples/

TABLE I
ONE-STEP DENOISING PERFORMANCE

Model	PESQ	SI-SDR	ESTOI	WER
Clean	–	–	–	3.03
Noisy	1.35 ± 0.30	4.0 ± 5.8	0.63 ± 0.18	12.2
Previous work [16]				
SGMSE+, 50 steps	2.28 ± 0.60	13.1 ± 4.9	0.85 ± 0.11	9.52
SB-VE, 50 steps	2.58 ± 0.53	14.7 ± 4.2	0.88 ± 0.07	5.10
One-step FM	2.32 ± 0.66	15.8 ± 4.3	0.88 ± 0.09	4.80
+ IP	2.17 ± 0.50	14.9 ± 3.8	0.88 ± 0.09	4.79
+ SfM	1.51 ± 0.31	8.8 ± 4.1	0.69 ± 0.16	10.32
+ ES	1.85 ± 0.39	12.6 ± 3.9	0.79 ± 0.12	7.48
One-step DP	2.35 ± 0.66	16.1 ± 4.3	0.89 ± 0.08	5.45
+ IP	2.22 ± 0.64	**16.5 ± 4.3**	0.89 ± 0.08	5.76
+ SfM	2.20 ± 0.63	16.3 ± 4.1	0.89 ± 0.08	5.72
+ ES	**2.68 ± 0.58**	16.2 ± 4.2	0.89 ± 0.08	**4.24**

models trained with data prediction (DP) outperform their velocity prediction counterparts. In terms of WER, however, we see that for five or more steps the data prediction model with informative prior (IP+DP) outperforms all other variants. When compared to baseline (SB-VE), we see that all proposed models outperform it in PESQ, but achieve a worse WER when the number of steps rises above five.

Table I displays all evaluation metrics for denoising models evaluated using one inference step. We also provide metrics for diffusion and SB baselines evaluated with fifty inference steps. Data prediction performs comparably to the baseline FM model. Starting from mean (SfM) causes major drop in performance, which could be explained by over-suppression of the speech signal. Early stopping (ES) improves the performance, with the corresponding data prediction model (DP+IP+SfM+ES) outperforming all other setups in WER and PESQ. This setup is also efficient and performs comparably to the SB-VE model evaluated at fifty times more steps. We attribute this result to the fact that modern ASR models are trained on large datasets recorded in imperfect conditions, introducing bias towards data that is a bit noisy. Starting from mean takes out the random uncorrelated noise from the inference process. Early stopping leaves some of the original noise intact, and also stops the model from over-suppressing and introducing extra artifacts.

B. Dereverberation

Fig. 2 shows WER and PESQ performance of different flow matching model modifications on the dereverberation task for different number of solver steps. In terms of audio quality, the default flow matching model outperforms models with informed prior (except in a one-step setup), with velocity prediction leaving more and data prediction performing similarly. However, in terms of ASR performance, the basic flow matching model (FM) either matches or outperforms other flow matching models (IP, DP, IP+DP). All proposed models outperform the SB-VE baseline in PESQ and WER. This makes all considered flow matching models a great low-compute alternative.

Table II displays all evaluation metrics for the same dereverberation models evaluated using one inference step. We also

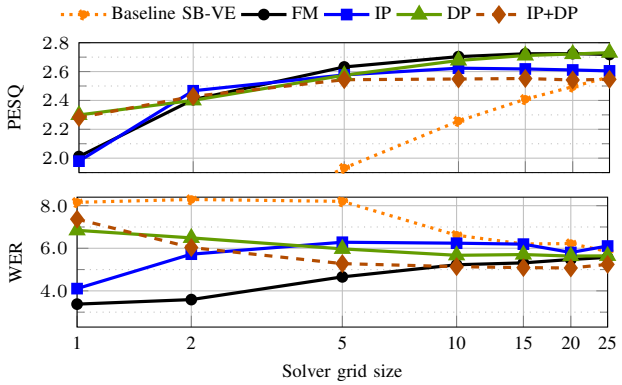

Fig. 2. Audio quality and WER after dereverberation. Some of the results for the baseline are out of range.

TABLE II
ONE-STEP DEREVERBERATION PERFORMANCE

Model	PESQ	SI-SDR	ESTOI	WER
Clean	–	–	–	3.64
Reverberant	1.29 ± 0.13	-9.5 ± 6.3	0.44 ± 0.11	8.29
Previous work [16]				
SGMSE+, 50 steps	2.34 ± 0.43	0.0 ± 8.9	0.82 ± 0.07	5.84
SB-VE, 50 steps	2.68 ± 0.41	6.6 ± 3.7	0.87 ± 0.05	5.91
One-step FM	2.01 ± 0.40	5.2 ± 4.6	0.83 ± 0.07	**3.38**
+ IP	1.98 ± 0.39	4.5 ± 3.8	0.77 ± 0.08	4.11
+ SfM	1.22 ± 0.14	-5.0 ± 6.5	0.42 ± 0.14	12.26
+ ES	1.43 ± 0.26	-3.0 ± 6.5	0.57 ± 0.14	10.53
One-step DP	**2.30 ± 0.48**	**7.6 ± 4.2**	**0.87 ± 0.05**	6.84
+ IP	2.28 ± 0.47	5.1 ± 4.2	0.84 ± 0.07	7.37
+ SfM	1.75 ± 0.32	-5.7 ± 6.9	0.70 ± 0.09	9.01
+ ES	1.73 ± 0.31	-6.2 ± 7.2	0.67 ± 0.10	8.76

provide metrics for diffusion and SB baselines evaluated with fifty inference steps. The results show that velocity (FM) and data prediction (DP) models have opposite trends in PESQ and in WER, revealing a quality-intelligibility tradeoff. Starting inference process from the mean (SfM) results in WER even worse than the unprocessed input (greyed out). Introducing early stopping (ES) allows to recover some performance, but results are still worse than unprocessed data. We conclude that starting from mean (SfM) should be avoided in case of dereverberation. An intriguing setup is the default flow matching: while leaving a lot of noise from the initial step, it pushes WER to a level even lower than that of clean data.

VI. CONCLUSION

In this paper, we present flow matching for speech enhancement, as well as several training- and inference-time modifications. We demonstrate that FM can be an effective alternative to diffusion and SB when applied to task of speech enhancement. The proposed model and its modifications are simpler than their diffusion and SB counterparts, while perform comparably well. For both tasks, we describe efficient and performant one-step inference that does not require advanced training procedures (pre-training, distillation). For dereverberation, we conclude that the basic flow matching approach without informed prior and fully deterministic inference is preferable. Confirming these results on larger and more challenging tasks is a promising direction for future work.

REFERENCES

[1] Rainer Beutelmann and Thomas Brand, "Prediction of speech intelligibility in spatial noise and reverberation for normal-hearing and hearing-impaired listeners," *The Journal of the Acoustical Society of America*, vol. 120, no. 1, pp. 331–342, 2006.

[2] Takuya Yoshioka et al., "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Process. Magazine*, vol. 29, no. 6, pp. 114–126, 2012.

[3] Y. Wu et al., "A fused speech enhancement framework for robust speaker verification," *IEEE Signal Processing Letters*, vol. 30, pp. 883–887, 2023.

[4] Y. Koizumi et al., "LibriTTS-R: A Restored Multi-Speaker Text-to-Speech Corpus," in *Proc. Interspeech*, 2023.

[5] Yong Xu et al., "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. on Audio, Speech, and Language Process.*, vol. 23, no. 1, pp. 7–19, 2014.

[6] Richard C Hendriks, Timo Gerkmann, and Jesper Jensen, *DFT-domain Based Single-microphone Noise Reduction for Speech Enhancement: A Survey of the State-of-the-art*, vol. 11, Morgan & Claypool Publishers, 2013.

[7] Timo Gerkmann and Emmanuel Vincent, "Spectral masking and filtering," in *Audio source separation and speech enhancement*, Emmanuel Vincent, Tuomas Virtanen, and Sharon Gannot, Eds., pp. 65–85. 2018.

[8] Yen-Ju Lu, Yu Tsao, and Shinji Watanabe, "A study on speech enhancement based on diffusion probabilistic model," in *Proc. Asia-Pacific Signal and Inform. Process. Assoc. Annual Summit and Conf. (APSIPA ASC)*, 2021, pp. 659–666.

[9] Yen-Ju Lu et al., "Conditional diffusion probabilistic model for speech enhancement," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, 2022, pp. 7402–7406.

[10] Simon Welker, Julius Richter, and Timo Gerkmann, "Speech enhancement with score-based generative models in the complex STFT domain," in *Proc. Interspeech*, 2022.

[11] J. Richter et al., "Speech Enhancement and Dereverberation with Diffusion-Based Generative Models," *IEEE/ACM Trans. on Audio, Speech, and Language Process.*, vol. 31, pp. 2351–2364, 2023.

[12] Jean-Marie Lemercier et al., "StoRM: A Diffusion-based Stochastic Regeneration Model for Speech Enhancement and Dereverberation," *IEEE/ACM Trans. on Audio, Speech, and Language Process.*, vol. 31, pp. 2724–2737, 2023.

[13] Jean-Marie Lemercier et al., "Analysing diffusion-based generative approaches versus discriminative approaches for speech restoration," in *Proc. ICASSP*, 2023.

[14] Zilu Guo et al., "Variance-preserving-based interpolation diffusion models for speech enhancement," in *Proc. Interspeech*, 2023.

[15] Bunlong Lay, Simon Welker, Julius Richter, and Timo Gerkmann, "Reducing the prior mismatch of stochastic differential equations for diffusion-based speech enhancement," in *Proc. Interspeech*, 2023.

[16] Ante Jukić, Roman Korostik, Jagadeesh Balam, and Boris Ginsburg, "Schrödinger bridge for generative speech enhancement," in *Proc. Interspeech*, 2024.

[17] Y. Song et al., "Score-based generative modeling through stochastic differential equations," in *Proc. Int. Conf. Learning Representations (ICLR)*, May 2021.

[18] Ling Yang et al., "Diffusion models: A comprehensive survey of methods and applications," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–39, 2023.

[19] Z. Chen, G. He, K. Zheng, X. Tan, and J. Zhu, "Schrodinger Bridges Beat Diffusion Models on Text-to-Speech Synthesis," *arXiv preprint arXiv:2312.03491*, Dec. 2023.

[20] Yaron Lipman et al., "Flow matching for generative modeling," in *Proc. ICLR*, 2023.

[21] Alexander H. Liu et al., "Generative pre-training for speech with flow matching," in *Proc. ICLR*, 2024.

[22] Diederik P Kingma and Ruiqi Gao, "Understanding diffusion objectives as the elbo with simple data augmentation," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023, pp. 65484–65516.

[23] J. S. Garofolo et al., "CSR-I (WSJ0) Complete," https://catalog.ldc.upenn.edu/LDC93S6A, [Online].

[24] J. Barker et al., "The third 'CHiME' speech separation and recognition challenge: Analysis and outcomes," *Computer Speech & Language*, vol. 46, pp. 605–626, 2017.

[25] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulations and array processing algorithms," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, 2018.

[26] Bunlong Lay and Timo Gerkmann, "An analysis of the variance of diffusion-based speech enhancement," in *Interspeech 2024*, 2024, pp. 2205–2209.

[27] A. Rix et al., "Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, 2001.

[28] J. Jensen and C. H. Taal, "An Algorithm for Predicting the Intelligibility of Speech Masked by Modulated Noise Maskers," *IEEE/ACM Trans. on Audio, Speech, and Language Process.*, vol. 24, no. 11, pp. 2009–2022, Dec. 2016.

[29] Jonathan Le Roux et al., "SDR - half-baked or well done?," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, 2019.

[30] NVIDIA, "STT En Fast Conformer-Transducer Large," https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_en_fastconformer_transducer_large, 2023, [Online; accessed Sep-2024].

[31] Diederik P Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.