Interspeech 2024
1-5 September 2024, Kos, Greece

Locally Aligned Rectified Flow Model for Speech Enhancement Toward Single-Step Diffusion

Zhengxiao Li[1], *Nakamasa Inoue*[1]

[1]Tokyo Institute Technology

ri.s.ah@m.titech.ac.jp

Abstract

Diffusion models based on stochastic differential equations have been shown to be effective in speech enhancement, a task of recovering clean speech signals from noisy speech signals. However, these models are limited by computational complexity, mainly due to the large number of function evaluations required in the reverse diffusion process. To address this limitation, we propose the locally aligned rectified flow (LARF) model, a diffusion model based on ordinary differential equations that learns a transport mapping between the distributions of clean and noisy speech features. By introducing global and local flow matching losses, LARF restricts the transport mapping to be as straight as possible, resulting in a reduction in the number of function evaluations. In experiments, we demonstrate the effectiveness of LARF on the two speech enhancement datasets: WSJ0-CHiME3 and VoiceBank-DEMAND. On WSJ0-CHiME3, LARF achieved a PESQ of 2.95 and an SI-SDR of 19.3 with a single step.

Index Terms: speech enhancement, diffusion model, rectified flow model

1. Introduction

Speech enhancement (SE) aims to improve the intelligibility and quality of speech by reducing environmental noise without distorting the original speech signals [1, 2]. SE has been extensively studied in the field of speech and signal processing. Proposed approaches range from statistical methods [3] to deep learning methods [4, 5, 6]. Recently, the effectiveness of deep generative models based on diffusion processes, so-called diffusion models, has been demonstrated [7, 8, 9, 10, 11].

Most diffusion models for SE are based on stochastic differential equations (SDEs) that model transport mappings between distributions of noisy and clean speech signals [7, 8, 9]. An SDE-based diffusion model typically includes forward and reverse SDEs [12]. The forward SDE describes the diffusion process, where Gaussian noise is iteratively added to clean speech. The reverse SDE inverts the forward SDE, thereby reducing noise and enhancing speech. A neural network is trained to solve the reverse SDE. SDE-based diffusion models often achieve high-quality SE. However, solving the reverse SDE is time-consuming and often requires more than ten neural network evaluations. This is mainly because SDEs involve a Wiener process that predisposes the transport mappings to be on-straight flow.

To reduce the computational cost, some recent work has focused on diffusion models based on ordinary differential equations (ODEs). For example, the flow matching model [13] formulates the vector field via an ODE. The rectified flow model [14] more concisely learns an ODE to make the paths that connect the points drawn from two distributions as straight as possible. Several studies have applied ODE-based models to speech processing tasks such as text-to-speech synthesis [15, 16]. Richter *et al.* [8] evaluated a probability flow ODE sampler for SE; however, they concluded that an SDE-based predictor-corrector sampler is better. Although ODE-based models have the potential to reduce computational costs, their application to SE remains challenging.

The present study proposes the locally aligned rectified flow (LARF) model, an ODE-based model for SE. LARF restricts the transport mapping between distributions of clean and noisy speech signals to be as straight as possible by applying global and local flow matching losses. The global flow matching loss is imported from the original rectified flow [14]. It aligns the velocity flow V_t at each time t with the global straight flow $V_{\text{global}} = Z_1 - Z_0^*$, where Z_1 is a noisy speech feature and Z_0^* is the corresponding clean speech feature. The local flow matching loss further facilitates the transport mapping have straight paths by aligning V_t with nearby flow V_{local}. We also develop a neural network architecture for LARF. More specifically, we introduce a velocity flow network that has an efficient two-stream U-Net architecture for SE. To demonstrate the effectiveness of the proposed method, we compare LARF with the SDE-based methods including SGMSE+ [8] in experiments on the WSJ0-CHiME3 and VoiceBank-DEMAN datasets.

2. Related Work

Speech enhancement models. Both discriminative and generative approaches have been proposed for SE models [17]. The former directly learn the mapping from noisy speech signals to clean speech signals. Example models include the DNN-based regression model [18] and Conv-TasNet [4]. The latter learn an a prior distribution over clean speech data using generative models such as generative adversarial networks (GANs) and variational autoencoders (VAEs). For example, SEGAN [19] is a GAN-based fully convolutional network for SE. The stochastic temporal convolutional network [20] is a VAE-based model that incorporates the hierarchy of stochastic latent variables. The dynamical VAE [6] models the temporal dependencies between successive observation and latent variables. Recently, a number of studies have shown the effectiveness of diffusion models for SE. For example, CDiffuse [9] applies a conditional diffusion probabilistic model in the reverse process. SRTNet [21] uses a diffusion model as a module for stochastic refinement. SGMSE and SGMSE+ [7, 8] are score-based diffusion models designed for SE. Some studies have investigated hybrid approaches, including discriminative learning using GANs [5, 22] and predictive learning using diffusion models [10, 11].

10.21437/Interspeech.2024-162

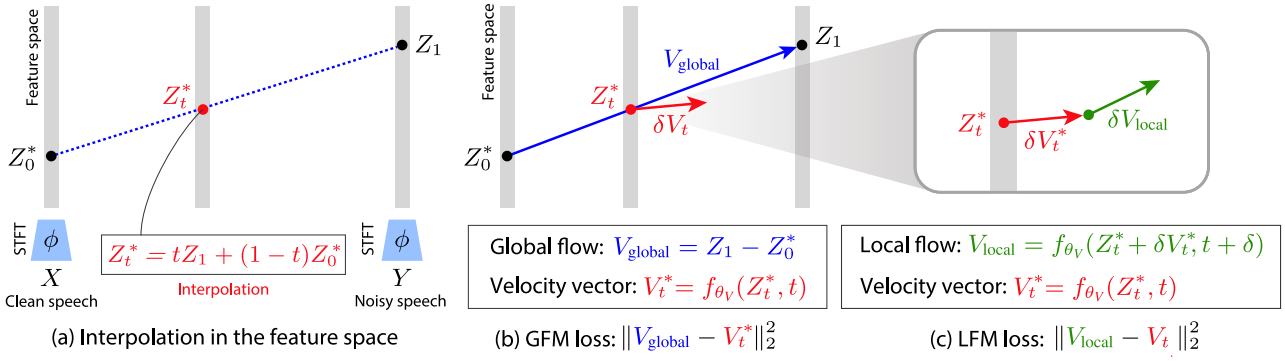

Figure 1: *Framework of LARF. (a) Interpolation point Z_t^* between a clean speech feature $Z_0^* = \phi(X)$ and a noisy speech feature $Z_1 = \phi(Y)$. (b) Global flow matching at Z_t^*, where mean squared error between global straight flow V_{global} (blue) and velocity vector V_t^* (red) is minimized. (c) Local flow matching at Z_t^*, where mean squared error between local flow V_{local} (green) and velocity vector V_t^* (red) is minimized.*

Diffusion formulations. The denoising diffusion probabilistic model (DDPM) [23] for image generation was the first model that formulated generative deep learning using the diffusion process. It was extended to score-based formulations using SDEs [24, 12, 25]. Instead of formulating the reverse diffusion process using SDEs, it is also possible to use ODEs such as the probability flow ODE [12]. ODE-based sampling is often faster than SDE-based sampling [26, 27]. Recent methods have applied ODE-based formulations for training diffusion models. Examples include the flow matching model [13] and the rectified flow model [14].

3. Locally Aligned Rectified Flow Model

This section describes the proposed LARF model for SE, which is an ODE-based model designed to learn the transport mapping between the feature distributions of clean and noisy speech signals. Our method extends the concept of diffusion models, aligning with adaptations in the field that transfer noisy data to clean data across different domains.

3.1. Overview

Problem setting. Following previous studies [9, 8], we assume that a dataset $\mathcal{D}_{\text{train}}$ that consists of pairs of clean and noisy speech signals is given for training. The goal is to train a model that recovers clean speech signals X from audio recordings Y that are affected by environmental noise.

Velocity model. LARF learns velocity model f_θ, where θ is a set of learnable parameters. It estimates the transport mapping from the distribution of noisy speech features to that of clean speech features. More specifically, given a noisy speech signal Y as input, the inference procedure for SE is written as follows:

$$Z_1 = \phi(Y),\ Z_0 = \int_1^0 f_{\hat{\theta}}(Z_t, t)dt,\ \hat{X} = \psi(Z_0), \quad (1)$$

where ϕ is an invertible feature extractor, $\psi = \phi^{-1}$ is the inverse of the feature extractor, $\hat{X}$ is the output enhanced speech, and $\hat{\theta}$ is a set of trained parameters. We use the short-time Fourier transform (STFT) and the inverse STFT for ϕ and ψ, respectively (see Sec. 3.3 for details). Note that in practice, the integration is approximated discretely with multiple steps. The number of these steps, which corresponds to the number of function evaluations, is denoted as T.

3.2. Training method

The differential form of Eq. (1) is the following ODE:

$$dZ_t = f_\theta(Z_t, t)dt. \quad (2)$$

LARF makes the transport mapping as straight as possible during training with two loss functions, namely global flow matching (GFM) loss and local flow matching (LFM) loss. Note that straight transport mapping helps reduce the number of function evaluations because it improves the accuracy of the approximate integration. Figure 1 shows the proposed framework.

1) GFM loss. Following the original rectified flow model [14], LARF aligns the velocity vectors $V_t = f_\theta(Z_t, t)$ with the global straight flow $V_{\text{global}} = Z_1 - Z_0^*$, where $Z_1 = \phi(Y)$ is a noisy speech feature and $Z_0^* = \phi(X)$ is the corresponding clean speech feature. This global alignment is enforced by minimizing the following GFM loss:

$$\mathcal{L}_{\text{GFM}} = \int_0^1 \mathbb{E}_{(X,Y)} \left[\|V_{\text{global}} - V_t^*\|_2^2 \right] dt, \quad (3)$$

where $V_t^* = f_\theta(Z_t^*, t)$ is the velocity vector evaluated at the interpolation point $Z_t^* = tZ_1 + (1-t)Z_0^*$ between the clean and noisy speech features (Figure 1a).

Figure 1b shows a diagram that explains GFM loss. Given a pair $(X, Y) \in \mathcal{D}_{\text{train}}$ of clean and noisy speech signals, the feature extractor ϕ is first applied to each signal to obtain two embeddings, namely Z_0^* and Z_1 (Figure 1a). Then, the velocity vector V_t^* (red arrow) is aligned with the global straight flow V_{global} (blue arrow) by minimizing the GFM loss (Figure1b).

2) LFM loss. To further facilitate the flow alignment, we introduce the following LFM loss:

$$\mathcal{L}_{\text{LFM}} = \int_0^1 \mathbb{E}_{(X,Y)} \left[\|V_{\text{local}} - V_t^*\|_2^2 \right] dt, \quad (4)$$

where $V_t^* = f_\theta(Z_t^*, t)$ is the velocity vector at Z_t^* and V_{local} is the velocity vector in the neighborhood of Z_t^*. We define the vector V_{local} as follows:

$$V_{\text{local}} = f_\theta(Z_t^* + \delta V_t^*, t + \delta), \quad (5)$$

where δ is a randomly sampled small step size.

Figure 1c shows a diagram that explains LFM loss. After calculating the velocity vector V_t^* at Z_t^* (red arrow), we evaluate the velocity vector V_{local} at the expected next position

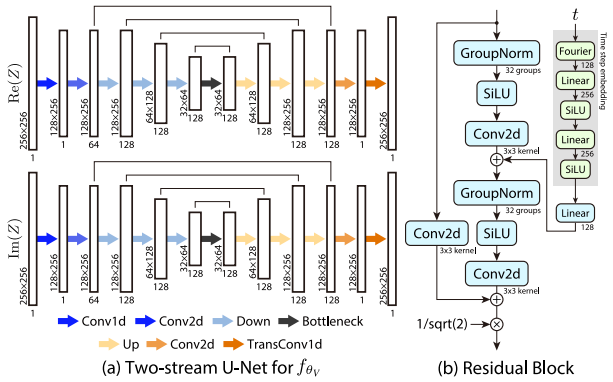

Figure 2: *Network architecture for VFN. (a) Two-stream U-Net for f_{θ_V}. (b) Residual block for Down, Bottleneck and Up layers.*

$Z_t^* + \delta V_t^*$ (green arrow). The red and green arrows are aligned by minimizing the LFM loss. This helps to keep local velocity flows as straight as possible.

Total loss. Finally, the total loss for training is given by

$$\mathcal{L} = \mathcal{L}_{\text{GFM}} + \lambda \mathcal{L}_{\text{LFM}} \tag{6}$$

where $\lambda \geq 0$ is a hyperparameter. The training algorithm is summarized in Algorithm 1.

3.3. Velocity flow network for speech enhancement

We introduce a neural network, referred to as a velocity flow network (VFN), for the LARF model. As shown in Figure 2, VFN is a two-stream architecture that takes as inputs the real and imaginary parts of complex spectrograms Z. Each stream is a U-Net inspired by NCSN++ [12] with 13M parameters.

Input features. Complex spectrograms are used as input features. Specifically, given an input speech signal X, complex spectrograms $Z = \phi(X) \in \mathbb{C}^{T,F}$ are extracted by the STFT with amplitude transformation [8], where T is the number of time frames and F is the number of frequency bins.

Architecture details. Each U-Net consists of 1) a Conv1D layer, 2) a Conv2D input layer, 3) three Down layers, 4) a Bottleneck layer, 5) three Up layers, 6) a Conv2D output layer, and 7) a TransConv1d layer.

1) Conv1D layer. This layer applies a one-dimensional convolution operation to the dimension of frequency bins, followed by a tanh activation function. The kernel size, stride, and padding are set to 3, 2, and 1, respectively. The number of output channels is 1.

2) Conv2D input layer. This layer extends the number of channels from 1 to 64 by applying a two-dimensional convolution operation. The kernel size, stride, and padding are set to $(3, 3)$, $(1, 1)$, and $(1, 1)$, respectively.

3-5) Down, Bottleneck, and Up layers. This part is imported from NCSN++. Each Down layer consists of four residual blocks (Figure 2b). The first and second Down layers have an additional residual block for downsampling followed by a two-dimensional convolution operation. The Bottleneck layer consists of two residual blocks, with an attention module using a single attention head in between. Each Up layer consists of five residual blocks with two-dimensional transposed convolution operations. The first and second Up layers have an additional residual block for upsampling. The time step embedding obtained from a Fourier embedding layer followed by two linear layers each with SiLU is fed into each block. The number of input and output channels and hyperparameters are detailed in Figure 2.

6) Conv2d output layer. This layer consists of a group normalization followed by a two-dimensional convolution operation. The kernel size, stride, and padding are set to $(3, 3)$, $(1, 1)$, and $(1, 1)$, respectively.

7) TransConv1d layer. This layer applies a one-dimensional transposed convolution operation to the dimension of frequency bins. The kernel size, stride, and output padding are set to 2, 2, and 1, respectively.

Algorithm 1 Training algorithm for LARF

Input: Feature extractor ϕ, Velocity model f_θ,
Training dataset $\mathcal{D}_{\text{train}}$, Hyperparameters λ .
Output: Trained parameter $\hat{\theta}$
repeat
Draw a minibatch $\{(X_i, Y_i)\}_{i=1}^B$ from $\mathcal{D}_{\text{train}}$.
$t \sim \text{Uniform}([0, 1])$, $\delta \sim \text{Uniform}([0, 1-t])$
for $i = 1, 2, \cdots B$ **do**
$Z_{i,0}^* \leftarrow \phi(X_i)$, $Z_{i,1} \leftarrow \phi(Y_i)$ # feature extraction
$Z_{i,t}^* \leftarrow tZ_{i,1} + (1-t)Z_{i,0}^*$ # interpolation
$V_{i,t}^* \leftarrow f_\theta(Z_{i,t}^*, t)$ # velocity vector
$V_{i,\text{global}} \leftarrow Z_{i,1} - Z_{i,0}^*$ # global flow for Eq. (3)
$V_{i,\text{local}} \leftarrow f_\theta(Z_{i,t}^* + \delta V_{i,t}^*, t+\delta)$ # local flow for Eq. (4)
end for
$\mathcal{L} = \frac{1}{B}\sum_i (\|V_{i,\text{global}} - V_{i,t}^*\|_2^2 + \lambda \|V_{i,\text{local}} - V_{i,t}^*\|_2^2)$
$\theta_V \leftarrow \theta_V - \eta \partial_{\theta_V} \mathcal{L}$
until convergence

4. Experiments

4.1. Experimental settings

Datasets. For evaluation, we used two datasets: WSJ0-CHiME3 (WSJ0-C3) and VoiceBank-DEMAND (VBD), both sampled at 16kHz. The WSJ0-C3 dataset consists of clean speech utterances from the Wall Street Journal dataset [28] and noise signals from the CHiME3 dataset [29]. The signal-to-noise ratio (SNR) was uniformly distributed between 0 and 20 dB. The training, validation, and test sets consisted of 12,777, 1,206 and 651 utterances, respectively. The VBD dataset [30] consists of clean speech utterances from the VCTK corpus [31]. The noise signals include eight real-recorded noise signals from the DEMAND database [32] and artificial babble and speech-shaped noise. The SNRs were 0, 5, 10, and 15 dB for training and 2.5, 7.5, 12.5, and 17.5 dB for testing. The training and test sets consisted of 11,572 and 824 utterances, respectively. Two speakers in the training set (p226 and p287) were used for validation.

Evaluation metrics. We used five metrics for evaluation, namely the perceptual evaluation of speech quality (PESQ) [33], the extended short-time objective intelligibility (ESTOI) [34], the scale-invariant signal-to-distortion ratio (SI-SDR), the scale-invariant signal-to-interference ratio (SI-SIR), and the scale-invariant signal-to-artifact ratio (SI-SAR) [35]. A single Tesla P100 GPU was used to evaluate computation time.

Baseline. We selected two SDE-based methods, namely CDiffuse [9] and SGMSE+ [8], as baselines. We used the models provided by the authors.

Training. We used the Adam optimizer with an initial learning rate of 0.01 and default hyperparameters in PyTorch for 300k iterations. The batch size was 8. The loss weight was set to $\lambda = 1.0$. The input and output tensor shapes are shown in Figure 2a. The time step embedding has 256 dimensions.

Table 1: *Comparison of LARF with SDE-based diffusion models. Steps: the number of steps. RTF: real-time factor.*

Method	Steps	RTF	WSJ0-C3 PESQ	WSJ0-C3 ESTOI	WSJ0-C3 SI-SDR	WSJ0-C3 SI-SIR	WSJ0-C3 SI-SAR	VBD PESQ	VBD ESTOI	VBD SI-SDR	VBD SI-SIR	VBD SI-SAR
Mixture	–	–	$2.01^{\pm 0.55}$	$0.81^{\pm 0.12}$	$13.5^{\pm 4.7}$	$18.7^{\pm 5.5}$	$15.4^{\pm 4.7}$	$1.98^{\pm 0.76}$	$0.79^{\pm 0.15}$	$8.4^{\pm 5.6}$	$8.5^{\pm 5.6}$	$47.5^{\pm 10.4}$
CDiffuse [9]	200	1.34	$2.27^{\pm 0.51}$	$0.83^{\pm 0.09}$	$9.2^{\pm 2.3}$	$19.8^{\pm 5.9}$	$10.0^{\pm 2.3}$	$2.52^{\pm 0.58}$	$0.79^{\pm 0.10}$	$12.4^{\pm 2.8}$	$19.8^{\pm 6.0}$	$13.8^{\pm 1.8}$
SGMSE+ [8]	1	0.074	$1.13^{\pm 0.22}$	$0.01^{\pm 0.02}$	$-27.7^{\pm 1.3}$	$26.6^{\pm 10.5}$	$-27.7^{\pm 1.3}$	$1.06^{\pm 0.10}$	$0.02^{\pm 0.02}$	$-25.3^{\pm 1.7}$	$23.3^{\pm 9.9}$	$-25.3^{\pm 1.7}$
SGMSE+ [8]	10	0.707	$1.86^{\pm 0.25}$	$0.88^{\pm 0.06}$	$12.9^{\pm 1.9}$	$31.1^{\pm 6.3}$	$13.0^{\pm 1.9}$	$1.63^{\pm 0.23}$	$0.80^{\pm 0.10}$	$14.2^{\pm 2.5}$	$27.5^{\pm 6.4}$	$14.6^{\pm 2.2}$
SGMSE+ [8]	20	1.40	$2.89^{\pm 0.52}$	$\underline{\mathbf{0.92}}^{\pm 0.06}$	$17.9^{\pm 4.0}$	$\underline{\mathbf{31.7}}^{\pm 4.8}$	$18.1^{\pm 4.1}$	$2.73^{\pm 0.50}$	$\underline{\mathbf{0.88}}^{\pm 0.08}$	$\underline{\mathbf{18.5}}^{\pm 3.2}$	$\underline{\mathbf{29.7}}^{\pm 3.8}$	$\mathbf{18.9}^{\pm 3.3}$
SGMSE+ [8]	30	2.10	$\underline{\mathbf{2.96}}^{\pm 0.55}$	$\underline{\mathbf{0.92}}^{\pm 0.06}$	$\mathbf{18.3}^{\pm 4.4}$	$\mathbf{31.1}^{\pm 4.6}$	$\mathbf{18.6}^{\pm 4.5}$	$\mathbf{2.93}^{\pm 0.62}$	$\mathbf{0.87}^{\pm 0.10}$	$17.3^{\pm 3.4}$	$\mathbf{29.1}^{\pm 5.8}$	$18.0^{\pm 3.4}$
LARF (Ours)	1	0.087	$\mathbf{2.95}^{\pm 0.58}$	$\underline{\mathbf{0.92}}^{\pm 0.06}$	$\underline{\mathbf{19.3}}^{\pm 4.5}$	$25.2^{\pm 5.3}$	$\underline{\mathbf{20.7}}^{\pm 4.4}$	$\underline{\mathbf{2.97}}^{\pm 0.70}$	$\mathbf{0.87}^{\pm 0.10}$	$\underline{\mathbf{19.2}}^{\pm 3.7}$	$26.4^{\pm 5.6}$	$\underline{\mathbf{20.7}}^{\pm 3.7}$

Table 2: *Results obtained with various strides for Conv1D and TransConv1D layers.*

Stride	RTF	PESQ	ESTOI	SI-SDR	SI-SIR	SI-SAR
1	0.145	$2.68^{\pm 0.61}$	$0.93^{\pm 0.06}$	$19.7^{\pm 4.4}$	$30.2^{\pm 5.2}$	$20.2^{\pm 4.5}$
2	0.087	$2.95^{\pm 0.58}$	$0.92^{\pm 0.06}$	$19.3^{\pm 4.5}$	$25.2^{\pm 5.3}$	$20.7^{\pm 4.4}$
4	0.060	$2.22^{\pm 0.58}$	$0.79^{\pm 0.12}$	$10.6^{\pm 4.1}$	$22.1^{\pm 6.1}$	$11.0^{\pm 3.9}$

Table 3: *Results obtained with various numbers of steps.*

Steps	RTF	PESQ	ESTOI	SI-SDR	SI-SIR	SI-SAR
1	0.087	$2.95^{\pm 0.58}$	$0.92^{\pm 0.06}$	$19.3^{\pm 4.5}$	$25.2^{\pm 5.3}$	$20.7^{\pm 4.4}$
2	0.131	$2.95^{\pm 0.58}$	$0.92^{\pm 0.06}$	$19.3^{\pm 4.5}$	$25.3^{\pm 5.4}$	$20.7^{\pm 4.4}$
3	0.178	$2.95^{\pm 0.58}$	$0.92^{\pm 0.06}$	$19.3^{\pm 4.5}$	$25.4^{\pm 5.4}$	$20.6^{\pm 4.4}$

4.2. Experimental results

Comparison with SDE-based methods. Table 1 compares LARF with SDE-based methods. As shown, LARF achieves a performance comparable to that of SGMSE+, which has 30 steps, in just one step with an exception of SI-SIR. The computational time is reduced by 95.9%. This result demonstrates the effectiveness and efficiency of the proposed method.

Conv1D layer. The first Conv1D layer of VFN applied to the dimension of frequency bins reduces the dimension from F to F/s, where F is the number of frequency bins and s is the stride hyperparameter. Thus, the stride of this layer is a factor in reducing computation time. Table 2 shows the results for $s = 1, 2$, and 4. As shown, $s = 2$ yields the best performance with the proposed VFN architecture. For $s = 1$, the decrease in performance is probably due to the network size being too small to learn the velocity vector with the full dimension of F. Note that SGMSE+ utilizes NCSN++ with 65M parameters, whereas VFN utilizes two U-Nets, each with 13M parameters. There could be more efficient and effective architectures for stride values other than 2. Optimal architectures will be considered in future work.

Two-stream architecture. Table 4 compares one- and two-stream architectures. As shown, the two-stream architecture outperforms the one-stream architecture. When the transport mapping from noisy speech features to clean speech features is restricted to be as straight as possible with LARF, the two-stream architecture improves performance because it makes two independent paths for the real and imaginary parts.

Number of steps. Table 3 shows the results for few steps. We did not observe any significant improvement by increasing the number of steps because the performance is saturated with a single step.

Ablation study. We performed an ablation study to verify the effectiveness of each loss function. As shown in Table 5, the combination of GFM loss and LFM loss improves performance.

Table 4: *Comparison of one- and two-stream architectures. One: one-stream architecture with two channels for real and imaginary parts. Two: two-stream architecture used in VFN.*

Arc.	RTF	PESQ	ESTOI	SI-SDR	SI-SIR	SI-SAR
One	0.060	$2.19^{\pm 0.61}$	$0.85^{\pm 0.11}$	$14.8^{\pm 5.3}$	$17.9^{\pm 5.9}$	$17.8^{\pm 4.9}$
Two	0.087	$2.95^{\pm 0.58}$	$0.92^{\pm 0.06}$	$19.3^{\pm 4.5}$	$25.2^{\pm 5.3}$	$20.7^{\pm 4.4}$

Table 5: *Results of ablation study with respect to loss.*

Loss	PESQ	ESTOI	SI-SDR	SI-SIR	SI-SAR
LARF	$2.95^{\pm 0.58}$	$0.92^{\pm 0.06}$	$19.3^{\pm 4.5}$	$25.2^{\pm 5.3}$	$20.7^{\pm 4.4}$
w/o $\mathcal{L}_{\text{GFM}}$	$1.70^{\pm 0.50}$	$0.78^{\pm 0.14}$	$9.8^{\pm 5.9}$	$9.8^{\pm 5.9}$	$44.6^{\pm 5.5}$
w/o $\mathcal{L}_{\text{LFM}}$	$2.89^{\pm 0.58}$	$0.92^{\pm 0.07}$	$19.0^{\pm 4.5}$	$25.4^{\pm 5.3}$	$20.2^{\pm 4.5}$

Table 6: *Results for the mismatched condition. SGMSE+ uses 30 steps. LARF uses a single step.*

Loss	PESQ	ESTOI	SI-SDR	SI-SIR	SI-SAR
SGMSE+	$2.48^{\pm 0.58}$	$0.90^{\pm 0.07}$	$16.2^{\pm 4.1}$	$\mathbf{28.9}^{\pm 4.6}$	$16.4^{\pm 4.1}$
LARF	$\mathbf{2.73}^{\pm 0.56}$	$\mathbf{0.91}^{\pm 0.07}$	$\mathbf{17.6}^{\pm 4.5}$	$24.3^{\pm 5.3}$	$\mathbf{18.8}^{\pm 4.4}$

Mismatched condition. Table 6 shows results for the mismatched condition where models are trained on VBD and tested on WSJ0-C3. LARF with a single step outperformed SGMSE+ across the four metrics, especially in PESQ.

5. Conclusion

In this study, we proposed the LARF model, an OED-based model using rectified flow, for SE. With GFM loss and LFM loss, LARF restricts the transport mapping between clean and noisy speech features to be as straight as possible, resulting in a significant reduction in the number of function evaluations. Experiments showed the effectiveness of the proposed method on the WSJ0-C3 and VBD datasets.

Future research direction. This work compared generative approaches based on ODEs with those based on SDEs. A combination of recent predictive approaches [10, 36] has the potential to further reduce the computational cost and improve the quality of SE. The introduction of streaming architectures to apply diffusion models to real-time applications should also be examined.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Number JP23H00490.

6. References

[1] R. C. Hendriks, T. Gerkmann, and J. Jensen, "Dft-domain based single-microphone noise reduction for speech enhancement: A survey of the state-of-the-art." *Morgan & Claypool*, 2013.

[2] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 22, pp. 745–777, 2014.

[3] T. Gerkmann and E. Vincent, "Spectral masking and filtering," in *Audio Source Separation and Speech Enhancement*, E. Vincent, T. Virtanen, and S. Gannot, Eds. John Wiley & Sons, 2018.

[4] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 27, no. 8, pp. 1256–1266, 2019.

[5] S.-W. Fu, C. Yu, K.-H. Hung, M. Ravanelli, and Y. Tsao, "MetricGAN-U: Unsupervised speech enhancement/dereverberation based only on noisy/reverberated speech," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7412–7416.

[6] X. Bie, S. Leglaive, X. Alameda-Pineda, and L. Girin, "Unsupervised speech enhancement using dynamical variational autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 30, pp. 2993–3007, 2022.

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

[8] J. Richter, S. Welker, J.-M. Lemercier, B. Lay, and T. Gerkmann, "Speech enhancement and dereverberation with diffusion-based generative models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 31, pp. 2351–2364, 2023.

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

[10] H. Shi, K. Shimada, M. Hirano, T. Shibuya, Y. Koyama, Z. Zhong, S. Takahashi, T. Kawahara, and Y. Mitsufuji, "Diffusion-based speech enhancement with joint generative and predictive decoders," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[11] B. Lay, S. Welker, J. Richter, and T. Gerkmann, "Reducing the prior mismatch of stochastic differential equations for diffusion-based speech enhancement," in *Proc. Interspeech*, 2023, pp. 3809–3813.

[12] 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. International Conference on Learning Representations (ICLR)*, 2021.

[13] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *Proc. International Conference on Learning Representations (ICLR)*, 2022.

[14] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *International Conference on Learning Representations (ICLR)*, 2023.

[15] Y. Guo, C. Du, Z. Ma, X. Chen, and K. Yu, "Voiceflow: Efficient text-to-speech with rectified flow matching," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[16] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar, and W.-N. Hsu, "Voicebox: Text-guided multilingual universal speech generation at scale," in *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[17] J.-M. Lemercier, J. Richter, S. Welker, and T. Gerkmann, "Analysing diffusion-based generative approaches versus discriminative approaches for speech restoration," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[18] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 23, no. 1, pp. 7–19, 2015.

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

[20] J. Richter, G. Carbajal, and T. Gerkmann, "Speech enhancement with stochastic temporal convolutional networks." in *Proc. Interspeech*, 2020, pp. 4516–4520.

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

[22] R. Cao, S. Abdulatif, and B. Yang, "CMGAN: Conformer-based metric GAN for speech enhancement," in *Proc. Interspeech*, 2022, pp. 936–940.

[23] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 6840–6851.

[24] Y. Song and S. Ermon, "Improved techniques for training score-based generative models," in *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[25] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet, "Diffusion schrödinger bridge with applications to score-based generative modeling," in *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[26] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver: a fast ode solver for diffusion probabilistic model sampling in around 10 steps," in *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[27] Q. Zhang and Y. Chen, "Fast sampling of diffusion models with exponential integrator," in *The Eleventh International Conference on Learning Representations*, 2023.

[28] J. S. Garofolo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) complete," Linguistic Data Consortium, no. LDC93S6A.

[29] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 504–511.

[30] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating RNN-based speech enhancement methods for noise-robust text-to-speech," in *ISCA Speech Synthesis Workshop (SSW)*, 2016, pp. 146–152.

[31] J. Yamagishi, C. Veaux, and K. MacDonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2019, cSTR, University of Edinburgh.

[32] J. Thiemann, N. Ito, and E. Vincent, "The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND): A database of multichannel environmental noise recordings," *The Journal of the Acoustical Society of America*, vol. 133, no. 5, pp. 3591–3591, 2013.

[33] 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 *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2001, pp. 749–752.

[34] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 24, no. 11, pp. 2009–2022, 2016.

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

[36] B. Lay, J.-M. Lemercier, J. Richter, and T. Gerkmann, "Single and few-step diffusion for generative speech enhancement," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.