END-TO-END SPEECH SYNTHESIS BASED ON DEEP CONDITIONAL SCHRÖDINGER BRIDGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Speech synthesis plays an important role in human-computer interaction. Existing methods mainly employ traditional two-stage pipeline, e.g. text-to-speech and vocoder. In this paper, we propose a system called Schrön, which can generate speech waves in an end-to-end mamaner by solving Schrödinger bridge problems (SBP). In order to make SBP suitable for speech synthesis, we generalize SBP from two aspects. The first generalization makes it possible to accept condition variables, which are used to control the generated speech, and the second generalization allows it to handle variable-size input. Besides these two generalizations, we propose two techniques to fill the large information gap between text and speech waveforms for generating high-quality voice. The first technique is to use a text-mel joint representation as the conditional input of the conditional SBP. The second one is to use a branch network for the generation of mel scores as a regularization, so that the text features will not be degenerated. Experimental results show that Schrön achieves state-of-the-art MOS of 4.52 on public data set LJSpeech. Audio samples are available at https://schron.github.io/.

1 INTRODUCTION

Speech synthesis usually consists of two parts, text-to-speech (TTS) and vocoder. TTS converts text to intermediate feature representations, such as mel-spectrogram, while vocoder converts mel-spectrogram to final waveform. Most of the research focuses on just one part, TTS or vocoder, and then connects these two subsystems to get a complete speech synthesis system. However, such a connected system can lead to the accumulation of errors, e.g. if the mel-spectrogram generated by the TTS is defective, which is difficult for subsequent vocoders to correct. An end-to-end speech generation system can avoid this cumulative error.

At present, there are not many end-to-end speech synthesis systems, and the representative ones are FastSpeech 2s (Ren et al., 2020), EATS (Jeff et al., 2021), and VITS (Kim et al., 2021). In order to overcome the large information gap between text and speech wave, FastSpeech 2s (Ren et al., 2020) uses the implicit representation of the mel-spectrogram as an intermediate variable, which is first generated from text, and then becomes wave. In the process of wave generation, FastSpeech 2s (Ren et al., 2020) uses a network structure similar to WaveNet (Kim et al., 2018) and uses adversarial training to overcome the lack of phase information. Compared to FastSpeech 2s, EATS (Jeff et al., 2021) adopts a different way of method, which uses two equally important technologies, one is the adversarial training loss to force the generation of high-fidelity speech audio, the other is to use the prediction loss on mel-spectrogram and duration as a regularizer. Current state-of-the-art (SOTA) end-to-end system is VITS (Kim et al., 2021), it adopts variational inference augmented with normalizing flows and an adversarial training process.

In this paper, we propose a novel end-to-end speech synthesis called Schrön by solving the Schrödinger bridge problem (Schrödinger, 1932), which can achieve exact diffusion between different distributions in finite time duration, without the need of time going to infinity. The Schrödinger bridge problem was proposed and solved by Schrödinger (1932), and it has profound applications in quantum mechanics (Cruzeiro & Zambrini, 1991) and optimal control (Mikami, 2008). Recently, some researchers have used the Schrödinger bridge for image generation and have achieved good results (Wang et al., 2021; Vargas et al., 2021). Most of these methods based on Schrödinger bridge require the input image to be of the same size, but the natural voice wave is basically of different

lengths. In this paper, we generalize the Schrödinger bridge generative model from two aspects to do speech synthesis. The first is that it can handle input waves of different size. The second is that it can accept conditional inputs as control variables, such as raw text in voice wave generation.

The advantage of Schrön over VITS (Kim et al., 2021), FastSpeech 2s (Ren et al., 2020) and EATS (Jeff et al., 2021) is that it is easy to train, while all of VITS, FastSpeech 2s and EATS use adversarial training, which is more difficult to converge. At the same time, Schrön has no special requirements for the network structure and has more design flexibility.

The main contributions of this paper are as follows

- To the best of our knowledge Schrön is the first end-to-end speech synthesis system based on the Schrödinger bridges.
- Schrön generalized SBP so that it has two new features, the first is that it can accept conditional control variables, and the second is that it can handle data of indefinite length. At the same time, it can achieve exact generation of target distributions in a limited time duration.
- An effective two-stage training algorithm for Schrön is proposed. The text encoder, mel density ratio estimator, and mel score predictor are trained in the first stage; the text decoder, mel-encoder, the wave density ratio estimator, and wave score predictor are trained in the second stage.
- Several insights in the end-to-end speech synthesis based on solving the Schrödinger bridge problems are given, especially the design of the network structure.

2 TRINITY OF CONDITIONAL SCHRODINGER BRIDGES

In this section, we first generalize the standard Schrödinger bridge problem (SBP) to accept conditional input as explicit variables for controllable probability measure transferring. Then we introduce two equivalent forms to the conditional SBP, one is the conditional Schrödinger system, the other is a conditional stochastic control problem. These two equivalent systems are more computable to realize the continuous transformation among the probability measures of wave data in Schrön.

2.1 NOTATIONS AND CONCEPTS

Let $\Omega = C([0,1], \mathbb{R}^n)$ be the set of all continuous functions (also called paths) ω from $[0,1]$ to $\mathbb{R}^n$. Here $\mathbb{R}^n$ is the space where the wave data is located. Let $\mathcal{D}$ be the space of all probability measure on Ω. $\mathcal{D}(\rho_0, \rho_1)$ be the set of all probability measures on Ω with marginal density ρ_0 at $t = 0$ and ρ_1 at $t = 1$. $\Pi(\rho_0, \rho_1)$ denotes the set of all probability distributions on $\mathbb{R}^n \times \mathbb{R}^n$ with marginals ρ_0 and ρ_1. δ_x and δ_y are Dirac's deltas on $\mathbb{R}^n$ concentrated at x and y.

$W_{\epsilon,x} \in \mathcal{D}$ denotes the Wiener measure with variance ϵ starting at $x \in \mathbb{R}^n$ at $t = 0$. Wiener measure is induced by the Riesz representation theorem (Riesz, 1907) from a certain integral functional on $C(\Omega)$ based on the heat kernel with variance ϵ

$$p_\epsilon(x, y; s, t) = \frac{1}{[2\pi\epsilon(t-s)]^{n/2}} e^{-\|x-y\|^2/2\epsilon(t-s)}, \tag{1}$$

where $s < t$. The intuitive idea behind $W_{\epsilon,x}$ is that it assigns

$$W_{\epsilon,x}(E) := \int_{E_1} \cdots \int_{E_m} p_\epsilon(x, x_1; 0, t_1) p_\epsilon(x_1, x_2; t_1, t_2) \cdots p_\epsilon(x_{m-1}, x_m; t_{m-1}, t_m) dx_1 \cdots dx_m \tag{2}$$

to the set of all functions $\omega \in E \subset \Omega$ which start at x and pass through the set E_1 at time t_1, the set E_2 at time t_2 etc.. Let $W_\epsilon(\cdot) := \int W_{\epsilon,x}(\cdot) dx$, then it is an unbounded measure on Ω and has marginals at each time in $[0,1]$ that coincidence with the Lebesgue measure on $\mathbb{R}^n$. The relative entropy between two probability measures on Ω is defined as Leonard (2013)

$$\mathbb{D}(P \parallel Q) = \begin{cases} \mathbb{E}_P \left[\log \frac{\mathrm{d}P}{\mathrm{d}Q} \right], & \text{if} \quad P << Q \\ +\infty \quad \text{otherwise.} \end{cases} \tag{3}$$

2.2 CONDITIONAL SCHRÖDINGER BRIDGE PROBLEM

The SBP is to find a probability measure that has the minimum relative entropy with a reference reversible Brownian motion under the condition of fixed initial and ending marginals (Schrödinger, 1932; Leonard, 2013; Chen et al., 2021). If we set the initial and ending marginals to the prior Dirac's delta distribution and the target wave data distribution corresponding to the text, then the probability measure solution obtained by solving the SBP corresponds to a continuous transformation between the two distributions. Thus the problem is about how to find the continuous transformation between two distributions so that transformation is most similar to a preset reversible Brownian motion. It should be noted that wave generation always needs raw text as conditional input, but there is no conditional variable in standard SBPs. Thus we need to generalize the standard SBPs to accept conditional variables.

The SBP (Schrödinger, 1932) is to find a $P \in \mathcal{D}(\rho_0, \rho_1)$ that minimizes $\mathbb{D}(P \parallel W_\epsilon)$, which is the relative entropy with the prior W_ϵ. We generalized it to the following conditional SBP (cSBP)

$$\arg\min_P \left\{ \mathbb{D}(P(\cdot|j) \parallel W_\epsilon) | P(\cdot|j) \in \mathcal{D}(\rho_0, \rho_1) \right\}, \tag{4}$$

where j is certain middle representations of the raw text as conditional input, and ρ_0 is a simple prior distribution that is easy to sample, such as Dirac's delta distribution, and ρ_1 is the distribution of wave data we need to generate. If $P^*(\cdot|j)$ is the solution of the problem (4), we can start sampling from the Dirac's delta distribution of ρ_0, use $P^*(\cdot|j)$ as the transition distribution between time $[0, 1]$, and finally realize the sampling of the target voice wave data in ρ_1 based on conditional input j. The optimal solution $P^*(\cdot|j)$ is called the conditional Schrödinger bridge between ρ_0 and ρ_1 over W_ϵ.

Although SBP is not easy to solve, it has two equivalent forms that are more suitable for computation Chen et al. (2021); Leonard (2013). Correspondingly, conditional SBP (4) also has two equivalent forms, which will be introduced in the following two sub-sections.

2.3 CONDITIONAL SCHRÖDINGER SYSTEM

Following the work of Föllmer (1988), we can obtained the following simple equivalent form of cSBP (4) through the disintegrations of $P(\cdot|j)$ and W_ϵ with respect to the initial and final positions, and the further decomposition of relative entropy to show that if $\rho^*_{01}(\cdot|j)$ is the solution of minimizing the static cSBP

$$\mathbb{D}(\rho^P_{01}(\cdot|j) \parallel \rho^{W_\epsilon}_{01}) = \int\int \left[\log \frac{\rho^P_{01}(x, y|j)}{\rho^{W_\epsilon}_{01}(x, y)} \right] \rho^P_{01}(x, y|j) dx dy, \tag{5}$$

where ρ^P_{01} satisfies

$$\int \rho^P_{01}(x, y|j) dy = \rho_0(x), \quad \int \rho^P_{01}(x, y|j) dx = \rho_1(y).$$

Then

$$P^*(\cdot|j) = \int W_{\epsilon,xy}(\cdot) \rho^*_{01}(x, y|j) dx dy \tag{6}$$

is the solution of cSBP (4).

Apply the standard 2-step constrained optimization optimality condition derivation on the static cSBP (5), that is to say first forming Lagrangian function, second setting the first variation equal to zero. It is found that the optimal $\rho^*_{01}(x, y)$ has the form

$$\rho^*_{01}(x, y|j) = \hat{\phi}(0, x) p_\epsilon(x, y; 0, 1) \phi(1, y), \tag{7}$$

where $\hat{\phi}(0, x)$ and $\phi(1, y)$ satisfy the conditional Schrödinger system

$$\begin{cases} \hat{\phi}(1, y) = \int \hat{\phi}(0, x) p_\epsilon(x, y; 0, 1) dx, \\ \phi(0, x) = \int p_\epsilon(x, y; 0, 1) \phi(1, y) dy, \end{cases} \tag{8}$$

with the condition

$$\hat{\phi}(0, x) \phi(0, x) = \rho_0(x), \quad \phi(1, y) \hat{\phi}(1, y) = \rho_1(y). \tag{9}$$

Define

$$\phi(t, x) = \int p_\epsilon(x, y; t, 1) \phi(1, y) dy, \quad \hat{\phi}(t, y) = \int \hat{\phi}(0, x) p_\epsilon(x, y; 0, t) dx, \tag{10}$$

then at each time t, the marginal $\rho(t, x)$ is $\hat{\phi}(t, x) \phi(t, x)$.

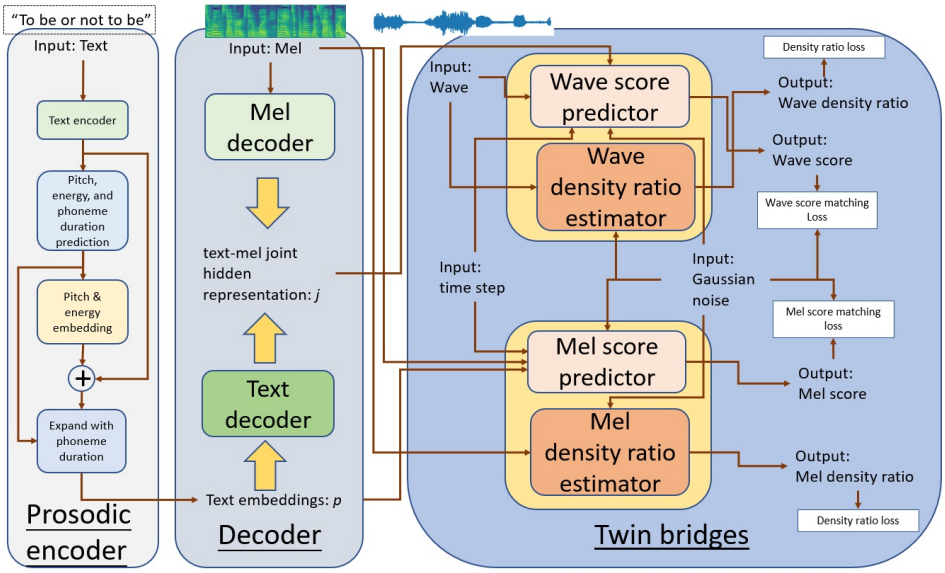


Figure 1: The overall structure of Schrön.

2.4 Conditional Stochastic Control Problem

From another perspective, the process $\mathbf{X}$ under $P \in \mathcal{D}$ can also be viewed as a diffusion governed by a stochastic differential equation(SDE) (Föllmer, 1988)

$$d\mathbf{X} = \beta(\mathbf{X}, t|j)dt + \sqrt{\epsilon}d\mathbf{W}. \tag{11}$$

Then by Girsanov's theorem (Karatzas & Shreve, 1987), (Jamison, 1975) showed that the original cSBP (4) is equivalent to

$$\arg\min_{\beta} \mathbb{E}\left[\int_0^1 \frac{1}{\epsilon} \parallel \beta(\mathbf{X}, t|j) \parallel\right], \tag{12}$$

where $\beta(\mathbf{X}, t|j)$ satisfies

$$\begin{cases} d\mathbf{X} = \beta(\mathbf{X}, t|j)dt + \sqrt{\epsilon}d\mathbf{W}, \\ \mathbf{X}(0) \sim \rho_0(x), \quad \mathbf{X}(1) \sim \rho_1(x), \end{cases} \tag{13}$$

and $\beta(\mathbf{X}, t|j)$ is called the drift coefficient.

Indeed if $\hat{\phi}(t, x)$ and $\phi(t, x)$ are the solutions of Schrödinger system (8) then

$$\beta^*(x, t) = \epsilon\nabla \log \phi(t, x) \tag{14}$$

is the solution of (13) (Leonard, 2013; Chen et al., 2021).

At this point, we obtain the equivalent relationship between the cSBP (4), the conditional Schrödinger system (8) and the conditional stochastic control problem (12). If we pre-set the solution ϕ and $\hat{\phi}$ of the Schrödinger system (8), then with the help of (14), we acquired the explicit conditional SDE (12), which can realize the continuous transformation between any two distributions in a limited time duration. Below we apply this idea to end-to-end speech synthesis, and propose a two-stage wave generation method with conditional variables.

3 Schrön

The overall structure of Schrön is shown in Figure 1, which shows the position and function of each module in the framework. Figure 2 shows the pipeline when Schrön is used for inference sampling. The principle and implementation details of Schrön will be introduced in the following sections. Please keep these two diagrams in mind, and then the following sections will be easy to follow.

3.1 TWO-STAGE WAVE GENERATION

Diffusion-based methods need to calculate the score function, which is the gradient of the log probability distribution. Since voice is a kind of structured data, which is often in a low-dimensional manifold of zero Lebesgue measure in the ambient space like images (Song et al., 2020). Thus the probability distribution is zero in most of the ambient Euclidean space and there will be singular values in score computation. Therefore we propose a two-stage voice generation algorithm to overcome this problem. Figure 3 (in Appendix) shows an example. In the first stage, zero-position Dirac distribution data are diffused into noisy wave, and the second stage diffuses noisy waves into clean waves.

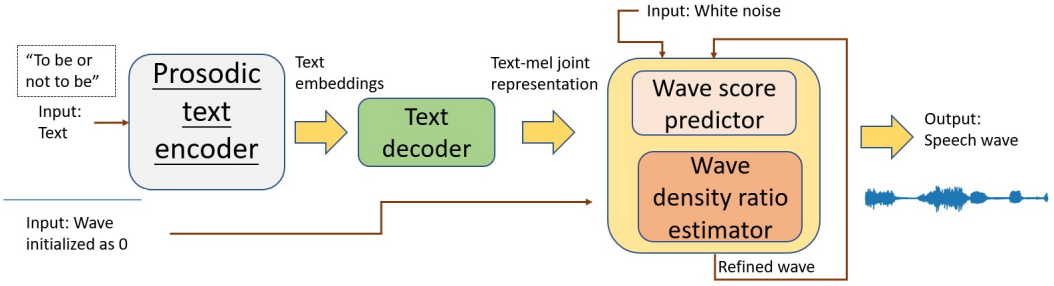


Figure 2: The inference pipeline of Schrön.

In the first stage, based on the conditional input of the text representations, we diffuse the Dirac's delta distribution with support at zero to a noisy wave distribution corresponding to the conditional text representation. Let $p_{wave}(\mathbf{X}|j)$ be the distribution of wave data corresponding to certain conditional text representation input j. After adding a small noise $z \sim \mathcal{N}(\cdot|\mathbf{0}, \sigma^2\mathbf{I})$ to smooth the wave data, the new smooth noisy wave distribution is

$$q(\cdot|j) = \int \mathcal{N}(\cdot|\mathbf{X}, \sigma^2\mathbf{I}) p_{wave}(\mathbf{X}|j) d\mathbf{X}. \tag{15}$$

Set $r(\mathbf{X}|j) = \frac{q(\mathbf{X}|j)}{\mathcal{N}(\mathbf{X}|\mathbf{0},\epsilon\mathbf{I})}$ be the density ratio between the smooth noisy wave data $q(\mathbf{X}|j)$ and the Gaussian noise $\mathcal{N}(\cdot|\mathbf{0}, \epsilon\mathbf{I})$. And let

$$\hat{\phi}(0, x) = \delta_0(x), \tag{16}$$

$$\hat{\phi}(1, y) \int \hat{\phi}(0, x) p_\epsilon(x, y; 0, 1) dx = \mathcal{N}(y|\mathbf{0}, \epsilon\mathbf{I}), \tag{17}$$

$$\phi(1, y) = r(y), \tag{18}$$

$$\phi(0, 0) = \int p_\epsilon(0, y; 0, 1)\phi(1, y) dy = 1, \tag{19}$$

in the Schrödinger system (8). Then ϕ and $\hat{\phi}$ solves the Schrödinger system with $\rho_0(x) = \delta_0(x)$ and $\rho_1(y) = q(\cdot|j)$. Thus

$$\phi(t, x) = \int p_\epsilon(x, y; t, 1)\phi(1, y) dy = \mathbb{E}_{y\sim\mathcal{N}(\cdot|\mathbf{0},(1-t)\epsilon\mathbf{I})} r(x + y) = \sqrt{1-t}\mathbb{E}_{z\sim\mathcal{N}(\cdot|\mathbf{0},\epsilon\mathbf{I})} r(x + \sqrt{1-t}z).$$

Then if we put $\epsilon\nabla_x \log \phi(t, x)$ in the problem (12), result in

$$\begin{cases} d\mathbf{X} = \epsilon\nabla_{\mathbf{x}} \log \mathbb{E}_{z\sim\mathcal{N}(\mathbf{0},\epsilon\mathbf{I})} r(\mathbf{X} + \sqrt{1-t}z|j) dt \qquad + \sqrt{\epsilon} d\mathbf{W} \\ \mathbf{X}(0) = \mathbf{0}, \end{cases} \tag{20}$$

and we have $\mathbf{X}(1) \sim \mathbf{wave} + \sigma z$, where $\mathbf{wave} \sim p_{wave}(\mathbf{X}|j)$ and $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. That is to say SDE (20) realizes the the transformation from Dirac's delta distribution with support at $\mathbf{0}$ to noisy wave corresponding to the conditional text representation j.

In the second stage, the output "$\mathbf{wave} + \sigma z$" of the first stage will be diffused into the clean wave that corresponding to the conditional text representation. Set

$$\hat{\phi}(0, x) = 1, \tag{21}$$

$$\hat{\phi}(1, y) \int \hat{\phi}(0, x) p_\epsilon(x, y; 0, 1) dx = 1, \tag{22}$$

$$\phi(1, y) = p_{wave}(y), \tag{23}$$

$$\phi(0, 0) = \int p_\epsilon(0, y; 0, 1) \phi(1, y) dy = q(\cdot|j), \tag{24}$$

in the Schrödinger system (8). Then ϕ and $\hat{\phi}$ solves the Schrödinger system with $\rho_0(x) = q(\cdot|j)$ and $\rho_1(y) = p_{wave}(y)$. Thus

$$\phi(t, x) = \int p_{\sigma^2}(x, y; t, 1) \phi(1, y) dy = q_{\sqrt{1-t}\sigma}(\cdot|j) \tag{25}$$

Then put $\sigma^2 \nabla_x \log \phi(t, x)$ into the problem (12), we have

$$\begin{cases} d\mathbf{X} = \sigma^2 \nabla_x \log q_{\sqrt{1-t}\sigma}(\mathbf{X}|j) dt + \sigma d\mathbf{W} \\ \mathbf{X}(0) \sim \mathbf{wave} + \sigma z, \quad \mathbf{X}(1) \sim \mathbf{wave} \end{cases} \tag{26}$$

where $\mathbf{wave} \sim p_{wave}(\mathbf{X}|j)$, which is what we want to generate.

3.2 Model Learning and Wave Sampling

Algorithm 1 Wave sampling in Schrön.

Input and initialization: The conditional mel-spectrograms m, the trained density ratio network $\mathfrak{D}_\theta$ and score prediction network $\mathfrak{S}_\vartheta$, the number of diffusion steps T.
//First, generate the noisy wave from $\mathbf{0}$.
1: Let $\mathbf{X} = \mathbf{0}$ be the initialization of the wave corresponding to m. Segment $\mathbf{X}$ into a fixed length, and do padding at the end if it is less than the fixed length.
2: **for** each segment $\mathbf{x}$ in $\mathbf{X}$, conditional segment m' in m.
3: **for** $t = 0, 1, \cdots,$ T
4: Sample $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \sim \mathcal{N}(\cdot|\mathbf{0}, \mathbf{I})$, where $\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$ as random tensors share the same size of $\mathbf{x}$.
5: Let $\mathbf{x}_1 = \mathbf{x} + \sqrt{\epsilon(1 - \frac{t}{T})}\mathbf{z}_1$ and $\mathbf{x}_2 = \mathbf{x} + \sqrt{\epsilon(1 - \frac{t}{T})}\mathbf{z}_2$.
6: Compute the density ratios $r_1 = \exp(\mathfrak{D}_\theta(\mathbf{x}_1))$ and $r_2 = \exp(\mathfrak{D}_\theta(\mathbf{x}_2))$.
7: Update $\mathbf{x}$:

$$\mathbf{b} \leftarrow \frac{r_1}{r_2} \left(-\mathfrak{S}_\vartheta(\mathbf{x}_1, t, m') + \sqrt{\frac{1}{\epsilon}(1 - \frac{t}{T})}\mathbf{z}_1 \right) + \frac{\mathbf{x}}{\epsilon},$$

$$\mathbf{x} \leftarrow \mathbf{x} + \frac{\epsilon}{T}\mathbf{b} + \sqrt{\frac{\epsilon}{T}}\mathbf{z}_3.$$

8: Concatenate all the generated noisy wave segment $\mathbf{x}$ to result in the full noisy wave $\mathbf{X}$ corresponding to the conditional m. The full noisy wave $\mathbf{X}$ will be the input to the next stage.
//Second, generate clean wave from noisy wave
9: **for** $t = 0, 1, \cdots,$ T
10: Let $\mathbf{e} = \mathfrak{S}_\vartheta(\mathbf{X}, t, m)$ and $\mathbf{z} \sim \mathcal{N}(\cdot|\mathbf{0}, \mathbf{I})$
11: Update $\mathbf{X}$: $\mathbf{X} \leftarrow \mathbf{X} - \sigma\frac{\mathbf{e}}{T} + \sqrt{\frac{\sigma}{T}}\mathbf{z}$
Output: $\mathbf{X}$.

It can be seen from the previous sections that once we have the values of the conditional wave density ratio function $r(\mathbf{X}|j) = \frac{q(\mathbf{X}|j)}{\mathcal{N}(\mathbf{X}|\mathbf{0}, \epsilon\mathbf{I})}$ and the wave score function $\nabla_x \log q_{\sqrt{1-t}\sigma}(\mathbf{X}|j)$, we can acquire the SDEs (20) and (26), thus achieve the two-stage wave generation. In this paper, we use neural networks to learn the rules from the data, and then to predict the values of density ratio function and the score function in the unknown environment.

For the conditional density ratio function, let $\mathfrak{D}_\theta$ be the neural network to predict the value of $\log r(\mathbf{X}|j)$. Then we will have $\exp(\mathfrak{D}_\theta(\mathbf{X})) \approx r(\mathbf{X}|j)$. The loss function and training algorithm for $\mathfrak{D}_\theta$ is shown in Algorithm 2. Gaussian noise is added to each wave data in the batch, and then use $\mathfrak{D}_\theta$ to predict the probability of noisy wave data and Gaussian noise respectively. Compute the loss according to Eq. (28) and backward propagated to update the weights of $\mathfrak{D}_\theta$, so that the score of noisy wave becomes higher, and the probability of Gaussian noise becomes lower.

For the score function value, we refer to the method in Wu & Shi (2021). Let $\mathfrak{S}_\vartheta$ be the neural network to predict the wave score value $\nabla \log q_{\sqrt{1-t}\sigma}(\mathbf{X}|j)$, the loss function and training algorithm for $\mathfrak{S}_\vartheta$ is also shown in Algorithm 2. First, Gaussian noise is added to each wave data in the batch, and then the noisy wave, condition text representation input and step information are fed into the score prediction network $\mathfrak{S}_\vartheta$. The distance between the output wave score and the noise will be calculated as loss in Eq. (29). Finally the backward propagation algorithm is used to update the weights of $\mathfrak{S}_\vartheta$ to make the prediction score better.

After we get the optimal wave density ratio estimator $\mathfrak{D}_{\theta_*}$ and wave score predictor $\mathfrak{S}_{\theta_*}$ through Algorithm 2, we can get the full numerical form of the two SDEs (20) and (26). Based on these two SDEs, then we can use the Euler-Maruyama procedure twice to generate high-quality clean speech from nothing (0). For the detail of the sampling algorithm, please refer Algorithm 1.

In the following subsections, we will take $\mathfrak{D}_\theta$ and $\mathfrak{S}_\theta$ from theory to implementations.

3.3 Wave Density Ratio $r(\mathbf{X}|j)$ Predictor

The ratio is used as guidance to add Gaussian noise step-by-step to the diffusion process to accurately change the distribution. The insight we get in designing the structure of the network $\mathfrak{D}_\theta$ is that we should not add conditional variables in the input. The conditional variables will make the network $\mathfrak{D}_\theta$ converge to a trivial saddle point. The reason should be that text information leaks into the diffusion process, causing it to quickly converge to a trivial solution. In fact, the noisy wave data is enough to calculate the density ratio. The network structure is shown in Figure 8 (in Appendix), where PhaseShuffle pDonahue et al. (2018) is used to make the estimation independent of the phase of the input wave or noise, that is to say, it is more robust to the regular blemish noise that sometimes appears in the frequency domain of the input waveform.

$\mathfrak{D}_\theta$ adopts fixed-length wave as input, which means that it can not accept wave input of different lengths. In order to solve this problem, when we have a long wave, we will cut it into several segments with a fixed length. Then we process each segment separately, and splice them together for subsequent processing. For the details please refer to Algorithm 2 and 1.

3.4 Wave Score $\nabla_x \log q_{\sqrt{1-t}\sigma}(\mathbf{X}|j)$ Estimator

The structure of the wave score prediction network $\mathfrak{S}_\vartheta$ is shown in Figure 9 (refer in Appendix). The main input of $\mathfrak{S}_\vartheta$ is the wave or noise, and it also requires condition input, including time information and the text representations. The expected output is the $\nabla_x \log q_{\sqrt{1-t}\sigma}(\mathbf{X}|j)$.

3.5 Fill The Gap

Many studies have shown that it is difficult to generate waves directly from text, such as FastSpeech 2s (Ren et al., 2020), EATS (Jeff et al., 2021), and VITS (Kim et al., 2021). Generally, the mel spectrogram or other middle feature representation of the speech wave is used as a regularizer. In the experiment, we also found such similar phenomenon.

We propose two approaches to fill this information gap between text and wave. The first way is a joint text-mel representation, which is an intermediate express between text and mel-spectrogram, as shown in Figure 1. Both text and mel-spectrogram can be transformed into this joint representation, which thus has both text and spectrogram information. We use this joint text-mel representation as the conditional input data j in the wave conditional Schrödinger bridge.

The second approach is a twin-bridges structure, that is, there are two sets of almost identical conditional Schrödinger bridges in Schrön (to be precise, totally four bridges). One set is used for end-to-end wave generation, and the other set is used for mel-spectrogram generation. The set of

mel Schrödinger bridges are used as regularizations to prevent the prosodic text embeddings p from degeneration.

The density ratio estimation and score estimation networks in the mel conditional Schrödinger bridge are similar to those in the wave conditional Schrödinger bridge, with only two differences. The first difference is that the dimensions of the input are different. The input to the wave Schrödinger bridge is a one-dimensional waveform, so the corresponding convolution operator is also 1D. The input to the mel Schrödinger bridge is a two-dimensional spectrum, so the corresponding convolution operator is 2D. The second difference is that the text conditional input used in the mel score estimation network is different from that used in the wave score predictor. It uses the direct output p of the text encoder, while the wave score network uses the text-mel joint representation feature j. The network structures of mel density ratio estimator and mel score predictor are shown in Appendix A.3

The training algorithm of mel density ratio estimator is exactly the same as the training of wave density ratio estimator, so does score predictors. Thus we won't go into details here.

3.6 Training of Schrön

Combining the modules introduced in the previous subsections, a two-stage approach is proposed to train Schrön. In the first stage, we train the prosodic text encoder $\mathfrak{M}$ and mel density ratio estimator $\mathfrak{d}_\theta$ and mel score predictor $\mathfrak{s}_\theta$; in the second stage, mel decoder $\mathfrak{M}$, text decoder $\mathfrak{T}$, wave density ratio estimator $\mathfrak{D}_\theta$ and wave score predictor $\mathfrak{S}_\theta$ are trained. For the second stage, in addition to the loss mentioned in Algorithm 2, we also need an additional consistency loss, which is

$$loss_c = \| j - \mathfrak{T}(p) \|_2^2 = \| \mathfrak{M}(\text{mel}) - \mathfrak{T}(p) \|_2^2, \tag{27}$$

to make the joint text-mel representation partially filling the information gap between text expression and raw mel-spectrogram.

4 Experiments

Table 1: MOS with 95% confidence in a comparative study between different state-of-the-art system and Schrön.

Models	Model size	MOS
Ground truth	-	4.59± 0.028
Tacotron 2+HiFi-GAN	28.1M+13.9M	3.90± 0.038
Fastspeech 2+HiFi-GAN	35.1M+13.9M	4.26±0.025
ItôTTS+ItôWave	34.1M+2.6M	4.31±0.049
VITS	36.3M	4.4±0.029
Schrön Base	56M	4.47±0.026
Schrön Large	80.2M	4.52±0.029

4.1 Dataset and setup

The data set we use is LJSpeech (Ito & Johnson, 2017), a single female speech database, with a total of 24 hours, 13100 sentences, randomly divided into 13000/50/50 for training/verification/testing. The sampling rate is 22050. For the experiments, in all the places that involve mel-spectrogram, the window length is 1024, the hop length is 256, the number of mel channels is 80.

In order to verify the performance of Schrön, we compared with two types of state-of-the-art systems. One is the end-to-end systems, and the other is connected systems. For end-to-end system, we chose the VITS (Kim et al., 2021) to compare, since it has open-source implementations[1]. For the connected system, we chose three for comparison, namely Tacotron 2 (Wang et al., 2017)[2] with HiFi-GAN (Su et al., 2020)[3], Fastspeech 2 (Ren et al., 2020)[4] with HiFi-GAN (Su et al., 2020), and ItôTTS with ItôWave (Wu & Shi, 2021) (a re-implemented version is used). The parameters and network structure of these comparison systems are the same as the corresponding settings in the respective papers.

For our Schrön, two network size variations were compared: Base and Large, contained 56M and 80.2M parameters respectively. The score prediction networks $\mathfrak{S}_\vartheta$ in the Base and Large are different, and the density ratio networks $\mathfrak{D}_\theta$ are the same. The $\mathfrak{S}_\vartheta$ of Schrön Base and Large uses 30 and 50 residual layers respectively. Adam Kingma & Ba (2014) is used in the training for Schrön, and all experiments were performed on GeForce RTX 3090 GPUs with 24G memory.

4.2 Results and discussion

In order to verify the naturalness and fidelity of the synthesized voice, we randomly select 40 from 50 test data for each subject, and then let the subject give the synthesized sound a MOS score of 0-5.

The results are shown in Table 1. MOS with 95% confidence is used in a comparative study of different state-of-the-art systems on the test set of the LJSpeech dataset. It can be seen that the MOS of Schrön is better than the previous state-of-the-art method, reaching 4.52, which is close to the ground truth.

Figures 4 and 5 show how wave and mel-spectrogram diffuse overtime in the two stages, for example, how to diffuse from $\mathbf{0}$ step by step to the noisy wave (mel-spectrogram) in the first stage, and how to diffuse from noisy wave (mel-spectrogram) step by step to clean wave (mel-spectrogram) in the second stage.

4.2.1 Ablation Study

We conducted ablation studies to prove the effectiveness of Schrön, including the text decoder for extraction of text-mel joint hidden representation and the mel-spectrogram Schrödinger bridges. All models in the ablation study were trained in the same settings as Schrön. The results are shown in Table 2. Removal of the text decoder results in 1.9 MOS lower than the baseline, which indicates that the flexibility of the conditional text-mel joint input significantly affects the synthesis quality. Replacing the dual Schrödinger bridges with a single wave Schrödinger bridge in a decrease in quality (-2.51 MOS), indicating that filling the gap with mel regularization is effective for Schrön to improve the synthesis quality.

Table 2: MOS with 95% confidence in the ablation study.

Model	MOS
Ground truth	4.59± 0.028
Schrön	4.52±0.029
Schrön w/o text decoder	2.62±0.032
Schrön w/o mel denoiser	2.01±0.055

5 Conclusion

In this paper, we propose a speech voice generation system called Schrön, which is the first end-to-end approach based on solving conditional Schrödinger bridge problems (SBP). The strength of Schrön includes 1) the generalization of SBP so that it can accept conditional control variables and also process data of indefinite length; 2) achieve exact generation of target distribution in a limited time. At the same time, since there is no generation of intermediate mel features, thus no cumulative effect of errors. Therefore, Schrön can generate high-quality voice. Our experimental results show that Schrön is superior to both the two-stage TTS system and previous state-of-the-art end-to-end system, achieving a MOS of 4.52 that is close to human's.

References

Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrödinger bridge. *Siam Review*, 63(2):249–313, 2021.

Ana Bela Cruzeiro and Jean-Claude Zambrini. Malliavin calculus and euclidean quantum mechanics. i. functional calculus. *Journal of Functional Analysis*, 96(1):62–95, 1991.

[1]https://github.com/jaywalnut310/vits

[2]https://github.com/NVIDIA/tacotron2

[3]https://github.com/jik876/hifi-gans

[4]https://github.com/ming024/FastSpeech2

Chris Donahue, Julian J. McAuley, and Miller S. Puckette. Adversarial audio synthesis. In *International Conference on Learning Representations*, 2018.

Hans Föllmer. Random fields and diffusion processes. pp. 101–203, 1988.

Keith Ito and Linda Johnson. The lj speech dataset. *Online: https://keithito. com/LJ-Speech-Dataset*, 2017.

Benton Jamison. The markov processes of schrödinger. *Probability Theory and Related Fields*, 32 (4):323–331, 1975.

Donahue Jeff, Dieleman Sander, Bińkowski Mikołaj, Elsen Erich, and Simonyan Karen. End-to-end adversarial text-to-speech. *ICLR*, 2021.

Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. 1987.

Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *ICML 2021: 38th International Conference on Machine Learning*, pp. 5530–5540, 2021.

Sungwon Kim, Sang-Gil Lee, Jongyoon Song, Jaehyeon Kim, and Sungroh Yoon. Flowavenet: A generative flow for raw audio. *arXiv preprint arXiv:1811.02155*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Christian Leonard. A survey of the schrodinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

Toshio Mikami. Optimal transportation problem as stochastic mechanics. 2008.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Swish: a self-gated activation function. *arXiv: Neural and Evolutionary Computing*, 2017.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020.

Frigyes Riesz. *Sur une espèce de géométrie analytique des systèmes de fonctions sommables*. Gauthier-Villars, 1907.

E. Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'institut Henri Poincaré*, 2(4):269–310, 1932.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Jiaqi Su, Zeyu Jin, and Adam Finkelstein. Hifi-gan: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks. *arXiv preprint arXiv:2006.05694*, 2020.

Francisco Vargas, Pierre Thodoroff, Neil D. Lawrence, and Austen Lamacraft. Solving schrödinger bridges via maximum likelihood. *arXiv preprint arXiv:2106.02081*, 2021.

Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via schrödinger bridge. In *ICML 2021: 38th International Conference on Machine Learning*, pp. 10794–10804, 2021.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

Shoule Wu and Ziqiang Shi. Itôtts and itôwave: Linear stochastic differential equation is all you need for audio generation. *arXiv preprint arXiv:2105.07583*, 2021.

A APPENDIX

A.1 SCHEMATIC ILLUSTRATION OF TWO-STAGE WAVE AND MEL DIFFUSION IN SCHRÖN.

As shown in Figure 3, the first stage diffuses the zero-position Dirac distributed data into noise waves, and the second stage diffuses the noise waves into clean waves. Figure 4 and 5 show several steps in the two-stage diffusion of wave and mel, respectively.

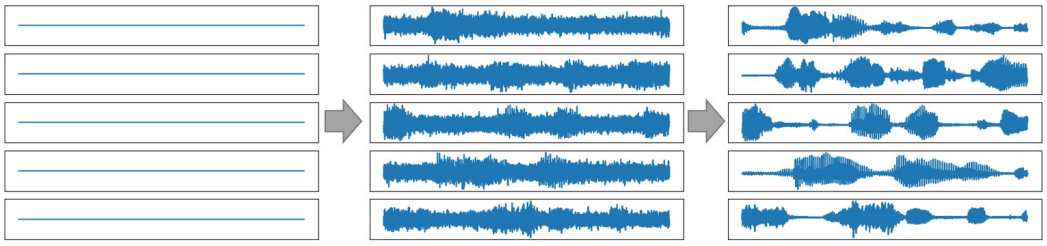
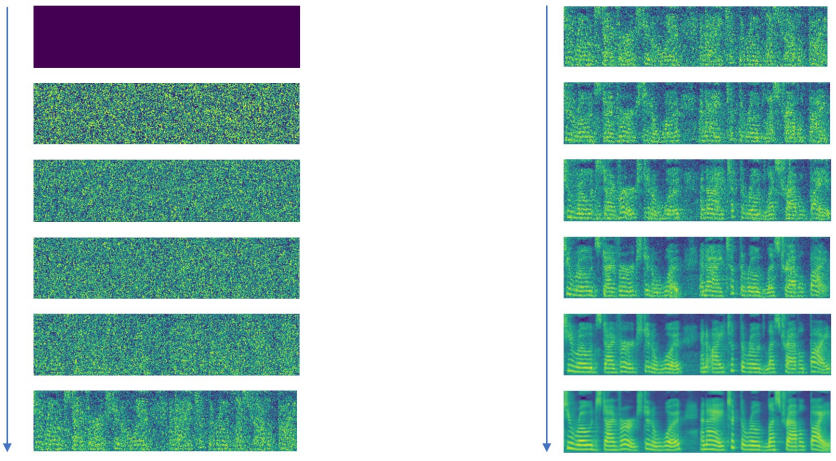


Figure 3: Example of two-stage wave generation in Schrön.

(a) (b)

Figure 4: Two stages of wave generation in Schrön. (a)The generation steps of the noisy wave from Dirac's delta distribution in the first stage. (b)The generation steps of the clean wave from the noisy wave in the second stage.



(a) (b)

Figure 5: Two stages of mel-spectrogram generation in Schrön. (a) The generation steps of noisy mel-spectrogram from Dirac distribution in the first stage. (b) The generation steps of clean mel-spectrogram from noisy mel-spectrogram in the second stage.

A.2 TRAINING ALGORITHM OF SCHRÖN.

Algorithm 2 shows the details of Schrön's full training algorithm.

Algorithm 2 Training of the wave density ratio network $\mathfrak{D}_\theta$ and wave score prediction network $\mathfrak{S}_\vartheta$ in Schrön.

Input and initialization: The voice wave $\mathbf{X}$ and the corresponding text representation j, the number of diffusion steps T. Here ϵ and σ are the same as in the SDEs (20) and (26).
//First do the training of $\mathfrak{D}_\theta$.
1: **for** $k = 0, 1, \cdots$
2: Randomly sample batch of waves $\mathbf{X}$ and text representations j.
3: Sample $\mathbf{e}_1, \mathbf{e}_2 \sim \mathcal{N}(\cdot|\mathbf{0}, \mathbf{I})$, where $\mathbf{e}_1$ and $\mathbf{e}_2$ as random tensors share the same size of $\mathbf{X}$.
4: Let $\mathbf{X}_1 = \mathbf{X} + \sqrt{\sigma}\mathbf{e}_1$ and $\mathbf{z} = \sqrt{\epsilon}\mathbf{e}_2$
5: Do the forward inference $r = \mathfrak{D}_\theta(\mathbf{X}_1)$ and $f = \mathfrak{D}_\theta(\mathbf{z})$.
6: Compute th following loss

$$loss_d = \log(1 + \exp(-r)) + \log(1 + \exp(f)), \tag{28}$$

and do the back-propagation for the updating of $\mathfrak{D}_\theta$.
7: $k \leftarrow k + 1$.
8: **Until** stopping conditions are satisfied.
//Then do the training of $\mathfrak{S}_\vartheta$.
9: **for** $k = 0, 1, \cdots$
10: Randomly sample batch of waves $\mathbf{X}$ and text representations j, and uniformly sample t from $[0, T]$.
11: Sample $\mathbf{e} \sim \mathcal{N}(\cdot|\mathbf{0}, \mathbf{I})$, where $\mathbf{e}$ as a random tensor shares the same size of $\mathbf{X}$.
12: $\mathbf{X}_1 = \mathbf{X} + \frac{t}{T}\sigma\mathbf{e}$.
13: $\tilde{\mathbf{e}} = \mathfrak{S}_\theta(\mathbf{X}_1, t, p)$
14: Compute the following loss

$$loss_r = \frac{1}{2} \parallel \mathbf{e} - \frac{t}{T}\tilde{\mathbf{e}} \parallel^2, \tag{29}$$

and do the back-propagation for the parameter updating of $\mathfrak{S}_\vartheta$.
15: $k \leftarrow k + 1$.
16: **Until** stopping conditions are satisfied.
Output: $\mathfrak{D}_\theta$ and $\mathfrak{S}_\vartheta$,

A.3 THE NETWORK STRUCTURE OF MEL DENSITY RATIO PREDICTOR AND MEL SCORE ESTIMATOR

The network structures of mel density ratio estimator and mel score predictor are shown in Figure 6 and 7 respectively.

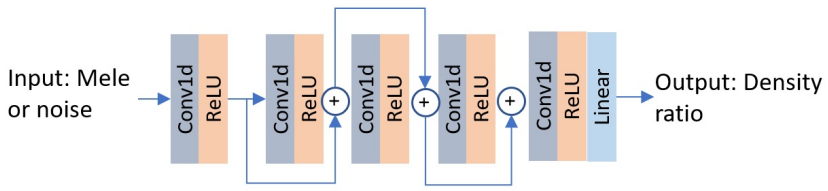


Figure 6: The mel density ratio prediction network in Schrön.

A.4 THE NETWORK STRUCTURE OF WAVE DENSITY RATIO PREDICTOR AND WAVE SCORE ESTIMATOR

The network structures of wave density ratio estimator and wave score predictor are shown in Figure 8 and 9 respectively.

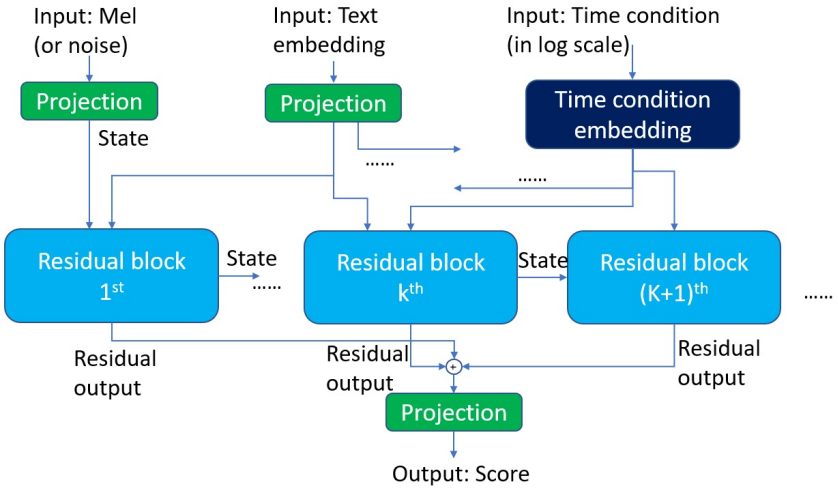


Figure 7: The mel score prediction network in Schrön.

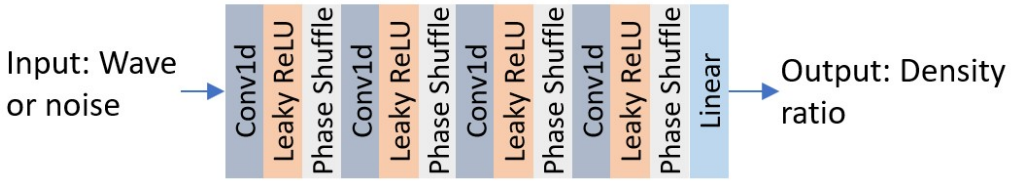


Figure 8: The wave density ratio prediction network in Schrön.

A.5 THE MODULES IN SCORE PREDICTION NETWORK.

In $\mathfrak{S}_\vartheta$, a Gaussian Fourier projection (GFP) of $[\sin(t \cdot \alpha), \cos(t \cdot \alpha)]$ with following linear and SIgmoid Linear Unit module (SILU) Ramachandran et al. (2017) are used to encode the time step t, where α is scalar Gaussian random number and SILU $:= x \cdot \sigma(x) = x \cdot \frac{1}{1+\exp\{(-x)\}}$. The time step embedding module is shown in Figure 10(a). At the same time, the text representations need to be up-sampled to the same size as the wave (noise) input. The up-sampling module is shown in Figure 10(c). The up-sampled text representations will be sent to the first residual block together with the time step embedding and the wave (noise). The residual block is a critical module in the score prediction network. The detail of the residual block is shown in Figure 10(b), where the CHUNK layer is to divide the input tensor into two parts along a channel. We use several residual blocks, and each residual block has three inputs, which are the current state, and the time embedding and up-sampled text representations. Each residual block will output a temporary residual signal, and then all the temporary residual signals will be summed and averaged. The final score will be obtained through a fully connected layer.

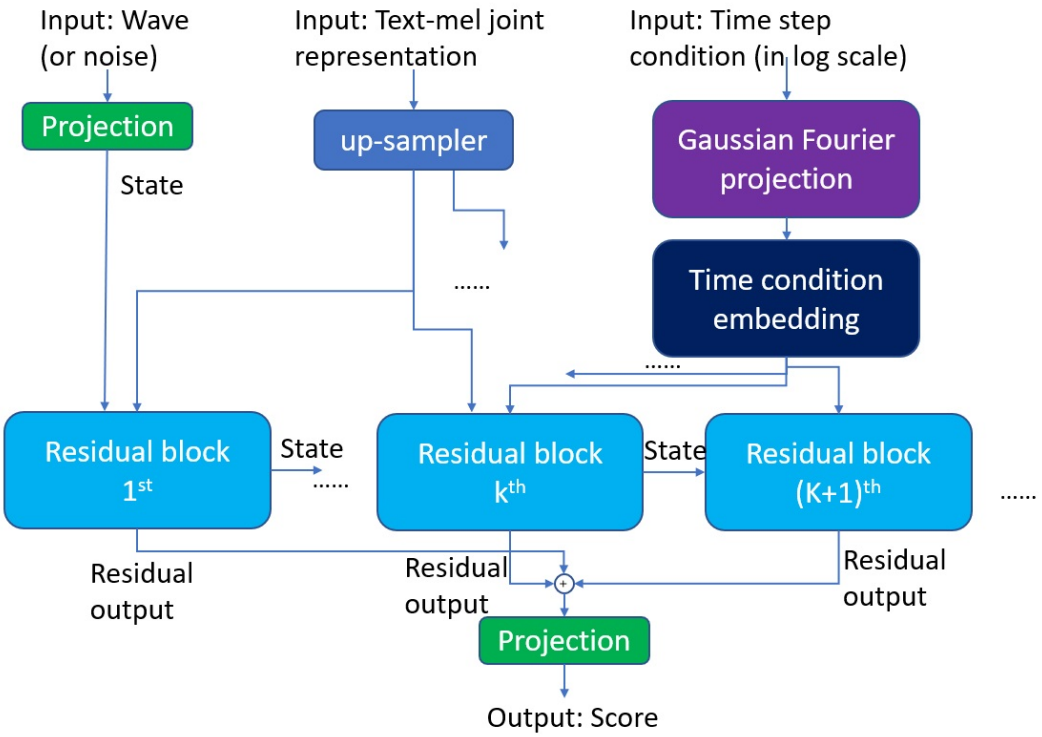


Figure 9: The wave score prediction network in Schrön.

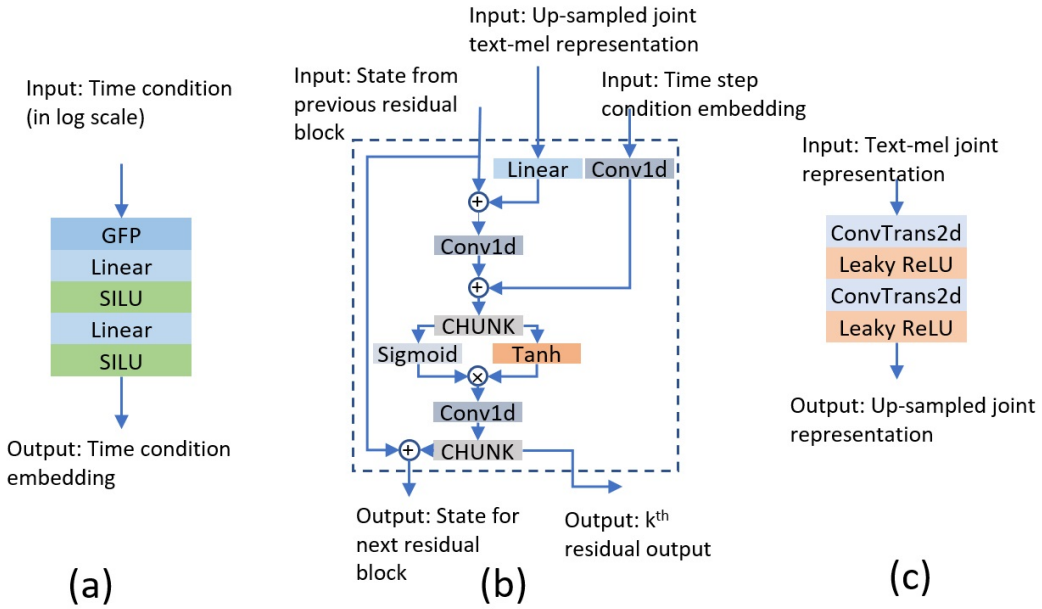


Figure 10: Modules in score prediction network. (a) Time step embedding; (b) the residual block; (c) up-sampling of text representations.