Deep Generative Models for Text-to-Speech Synthesis

Xu Tan/谭旭
Microsoft Research Asia
Principal Research Manager
xuta@microsoft.com
Outline

• Background
  • Text-to-Speech Synthesis
  • Deep Generative Models

• Deep Generative Models for TTS
  • AR/Flow/GAN/VAE/Diffusion based TTS Models
  • Comparisons and Analyses

• Summary and Outlook
Speech Signal Processing

- Speech Recognition
- Speaker Recognition
- Speech Synthesis
- Speech Enhancement
- Speech Separation
- Speech Translation
- ... (Other categories)
- Text-to-Speech Synthesis
  - from text
- Singing Voice Synthesis
  - from music score
- Voice Conversion
  - from speech
- EEG based Speech Synthesis
  - from brain signal
- ... (Other methods)
Text-to-Speech Synthesis

- Text-to-speech (TTS): generate intelligible and natural speech from text

“speech synthesis is awesome”

Text → TTS System → Speech

- Enabling machine to speak is an important part of AI
  - TTS (speaking) is as important as ASR (listening), NLU (reading), NLG (writing)
  - Human beings tried to build TTS systems dating back to the 12th century

<table>
<thead>
<tr>
<th>Year</th>
<th>Synthesis Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950s</td>
<td>Articulatory Synthesis</td>
</tr>
<tr>
<td>1970s</td>
<td>Formant Synthesis</td>
</tr>
<tr>
<td>1990s</td>
<td>Concatenative Synthesis</td>
</tr>
<tr>
<td>2010s</td>
<td>Statistical Parametric Synthesis</td>
</tr>
<tr>
<td>2016</td>
<td>Neural Speech Synthesis</td>
</tr>
<tr>
<td></td>
<td>(Deep) Neural Speech Synthesis</td>
</tr>
</tbody>
</table>

Neural TTS

WaveNet (DeepMind)
Text-to-Speech Mapping is One-to-Many

• Speech contains much information that not exists in text
  • What to say: content
  • Who to say: speaker/timbre
  • How to say: prosody/emotion/style
  • Where to say: noisy environment
  • …

Text [duration, pitch, sound volume, prosody, speaker, style, emotion, etc] Speech

• Text-to-speech mapping
  • Not point-wise, but distribution-wise
  • Usually not single-modal, but multi-modal
Typical Methods to Handle One-to-Many Mapping in TTS

- Split text-to-speech conversion into multiple stages

---

Text

Linguistic Features

Acoustic Features

Waveform

<table>
<thead>
<tr>
<th>Stage</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SPSS [418, 358, 417, 427, 359]</td>
</tr>
<tr>
<td>1</td>
<td>ARST [377]</td>
</tr>
<tr>
<td>2</td>
<td>WaveNet [255], DeepVoice 1/2 [8, 88], Par. WaveNet [256], WaveRNN [151], HiFi-GAN [23]</td>
</tr>
<tr>
<td>3</td>
<td>DeepVoice 3 [271], Tacotron 2 [304], FastSpeech 1/2 [291, 293], WaveGlow [280], FloWaveNet [164]</td>
</tr>
<tr>
<td>4</td>
<td>Char2Wav [316], Clarinet [270], FastSpeech 2s [293], EATS [70], VITS [161], NaturalSpeech [346]</td>
</tr>
</tbody>
</table>
Typical Neural TTS Pipeline

- Text analysis, acoustic model, and vocoder
- Text analysis: text → linguistic features
- Acoustic model: linguistic features → acoustic features
- Vocoder: acoustic features → speech

One-to-many mapping is alleviated, but not eliminated!
How to Model One-to-Many Mapping (Multimodal Distribution)

- Providing more variance information
  - Providing pitch/duration/speaker ID
    - Autoregressive models ($x_0 \rightarrow x_{0:1} \rightarrow \ldots \rightarrow x_{0:t} \rightarrow \ldots \rightarrow x_{0:T}$)
    - Diffusion models ($x_T \rightarrow \ldots \rightarrow x_t \rightarrow x_{t-1} \rightarrow \ldots \rightarrow x_0$)

- Advanced loss function
  - L1/L2 loss
    - Distribution-wise loss (e.g., SSIM, GMM)
    - GAN loss (match any distribution)

- Synthesis-by-analysis
  - $X \rightarrow Z \rightarrow X$
    - VAE, Flow, etc
Outline

• Background
  • Text-to-Speech Synthesis
  • Deep Generative Models

• Deep Generative Models for TTS
  • AR/Flow/GAN/VAE/Diffusion based TTS Models
  • Comparisons and Analyses

• Summary and Outlook
Deep Learning and Generative Learning

1950s /1960s (Computer) → 2022
CV/NLP/Speech/Machine Learning

2012 (AlexNet) → 2022
Deep Learning (Representation Learning)

2013/2014/2015 (VAE/GAN/Flow/Diffusion) → 2022
Deep Learning (Generative Learning)
Generative Models

- Generative models are learnt to estimate the likelihood of data $P_\theta$ to be close to the true data distribution $P_D$
  - **Data generation**: sample new data from $P_\theta$
  - **Density estimation**: predict the density/probability of a data point

- Taxonomy of deep generative models
Deep Generative Models—GAN

• Generative Adversarial Networks

\[
\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim p_{\text{data}}} \log D(x; \phi) + \mathbb{E}_{z \sim p_z} \log(1 - D(G(z; \theta); \phi))
\]

• Not to find a corresponding \( z \) for \( x \), but to directly match the distribution of \( x \)
Deep Generative Models—Flow

• Normalizing Flows: finding a $z$ for $x$, and convert $z$ back to $x$
  - $z = f_k^{-1} f_{k-1}^{-1} ... f_0^{-1}(x)$
  - $x = f_0 f_1 ... f_k(z), z \sim N(0, 1)$

• Training: maximizing the log likelihood $p(x)$
  - $\log p(x) = \log p(z) + \log \det \left( \frac{dz}{dx} \right) = \log p(z) + \sum_{i=1}^{k} \log |\det(J(f^{-1}_i(x)))|$
  - Flow can estimate the data likelihood exactly, as in autoregressive models

• The transformation function $f$ should satisfy two requirements
  - It is easily invertible
  - Its Jacobian determinant is easy to compute
Deep Generative Models—Flow

- Two types: **Coupling (bipartite)** and **Autoregressive (AR)** technologies

<table>
<thead>
<tr>
<th>Flow</th>
<th>Evaluation $z = f^{-1}(x)$</th>
<th>Synthesis $x = f(z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>$z_t = \frac{x_t - \mu_t(x_{&lt;t})}{\sigma_t(x_{&lt;t})}$</td>
<td>$x_t = z_t \cdot \sigma_t(x_{&lt;t}) + \mu_t(x_{&lt;t})$</td>
</tr>
<tr>
<td>IAF [38]</td>
<td>$z_t = x_t \cdot \sigma_t(z_{&lt;t}) + \mu_t(z_{&lt;t})$</td>
<td>$x_t = \frac{z_t - \mu_t(z_{&lt;t})}{\sigma_t(z_{&lt;t})}$</td>
</tr>
<tr>
<td>Bipartite</td>
<td>$z_a = x_a$,</td>
<td>$x_a = z_a$,</td>
</tr>
<tr>
<td>RealNVP [36]</td>
<td>$z_b = x_b \cdot \sigma_b(x_a; \theta) + \mu_b(x_a; \theta)$</td>
<td>$x_b = \frac{z_b - \mu_b(x_a; \theta)}{\sigma_b(x_a; \theta)}$</td>
</tr>
</tbody>
</table>

- It is easily invertible
  - See table above
- Its Jacobian determinant is easy to compute
  - The invertible functions have triangular Jacobians
  - It’s easy to calculate from the diagonal elements
Deep Generative Models—VAE

• Why Variational Autoencoders?
  • Naïve AE: \( ||x - dec(enc(x))||^2 \)
  • No regularization: \( z \) is irregular and non-smoothing, generalization is poor

• Maximizing the log likelihood \( p(x) \)

\[
\log p(x) = \log \int p(x|z)p(z)dz = \log \int q(z|x) \frac{p(x|z)p(z)}{q(z|x)} dz \\
= \log \mathbb{E}_{z \sim q(z|x)} \frac{p(x|z)p(z)}{q(z|x)} \geq \mathbb{E}_{z \sim q(z|x)} \log \frac{p(x|z)p(z)}{q(z|x)} \\
= \mathbb{E}_{z \sim q(z|x)} \log p(x|z) - KL(q(z|x)||p(z)),
\]

• Maximize the ELBO

\[
L(x; \theta, \phi) = -\mathbb{E}_{z \sim q(z|x; \phi)} \log p(x|z; \theta) + KL(q(z|x; \phi)||p(z))
\]
Deep Generative Models—DDPM

- Denoising Diffusion Probabilistic Models

\[ q(x_1:T|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \quad q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \]

- Forward process

\[ p_\theta(x_0:T) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \]
Deep Generative Models—DDPM

• Maximizing the log likelihood $p(x_0)$

$$
\log p(x_0) = \log \int p(x_{0:T}) dx_{1:T} = \log \int q(x_{1:T}|x_0) \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} dx_{1:T}
$$

$$
= \log \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} \geq \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \log \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} = ELBO
$$

• Maximize the ELBO

$$
ELBO = \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \log \frac{p(x_{0:T})}{q(x_{1:T}|x_0)}
$$

$$
= -\mathbb{E}_q \left[ KL(q(x_T|x_0)||p(x_T)) \sum_{t=2}^{T} KL(q(x_{t-1}|x_t,x_0)||p_\theta(x_{t-1}|x_t)) - \frac{\log p_\theta(x_0|x_1)}{L_0} \right]
$$

$$
L_{\text{simple}}(\theta) := \mathbb{E}_{t,x_0,\epsilon} \left[ ||\epsilon - \epsilon_\theta(x_t,t)||^2 \right]
$$
Deep Generative Models—DDPM

• Training and inference pipeline

<table>
<thead>
<tr>
<th>Algorithm 1 Training</th>
<th>Algorithm 2 Sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>repeat</strong></td>
<td>Sample $x_T \sim \mathcal{N}(0, I)$</td>
</tr>
<tr>
<td>Sample $x_0 \sim q_{\text{data}}$, $\epsilon \sim \mathcal{N}(0, I)$</td>
<td><strong>for</strong> $t = T, T - 1, \ldots, 1$ <strong>do</strong></td>
</tr>
<tr>
<td>Sample $t \sim \mathcal{U}({1, \ldots, T})$</td>
<td>Sample $z \sim \mathcal{N}(0, I)$ if $t &gt; 1$; else $z = 0$</td>
</tr>
<tr>
<td>$\mathcal{L} = |\epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon, t)|^2$</td>
<td>$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, t)) + \sigma_t z$</td>
</tr>
<tr>
<td>Update $\theta$ with $\nabla_\theta \mathcal{L}$</td>
<td><strong>end for</strong></td>
</tr>
<tr>
<td><strong>until</strong> converged</td>
<td><strong>return</strong> $x_0$</td>
</tr>
</tbody>
</table>
Deep Generative Models—SMLD

• Score Matching with Langevin Dynamics (SMLD)
  • Score: the score of a probability density $p(x)$ is $\nabla x \log p(x)$

• Training: score matching for score estimation

\[ \mathbb{E}_{p(x)} \left[ \| s_\theta(x) - \nabla \log p(x) \|_2^2 \right] \quad \arg \min_{\theta} \sum_{t=1}^{T} \lambda(t) \mathbb{E}_{p_{\sigma_t}(x_t)} \left[ \| s_\theta(x, t) - \nabla \log p_{\sigma_t}(x_t) \|_2^2 \right] \]

• Inference: sampling with Langevin dynamics

\[ x_{i+1} \leftarrow x_i + c \nabla \log p(x_i) + \sqrt{2c} \epsilon, \quad i = 0, 1, ..., K \]

\[ \nabla \log p(x_t) = -\frac{1}{\sqrt{1 - \alpha_t}} \epsilon \]
Deep Generative Models—SDE

- Stochastic Differential Equation (SDE)
  - Extend discrete time to continuous time

\[
\begin{align*}
\text{Forward SDE (data }\rightarrow\text{ noise)} & : \quad \mathbf{x}(0) \rightarrow \mathbf{x}(T) \\
\quad dx &= f(x, t)dt + g(t)d\mathbf{w} \\
\text{Reverse SDE (noise }\rightarrow\text{ data)} & : \quad \mathbf{x}(0) \rightarrow \mathbf{x}(T) \\
\quad dx &= [f(x, t) - g^2(t)\nabla_x \log p_t(x)] dt + g(t)d\mathbf{w}
\end{align*}
\]

\[
\theta^* = \arg \min_{\theta} \mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t) | \mathbf{x}(0)} \left[ \left\| \mathbf{s}_\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) | \mathbf{x}(0)) \right\|^2 \right] \right\}.
\]
Deep Generative Models—VE-SDE, VP-SDE

- VE-SDE (Variance-Exploding Stochastic Differential Equation) and SMLD

\[
x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}, \quad i = 1, \ldots, N,
\]

\[
dx = \sqrt{\frac{d[\sigma_i^2(t)]}{dt}} dw
\]

- VP-SDE (Variance-Preserving Stochastic Differential Equation) and DDPM

\[
x_i = \sqrt{1 - \beta_i} x_{i-1} + \sqrt{\beta_i} z_{i-1}, \quad i = 1, \ldots, N.
\]

\[
dx = -\frac{1}{2} \beta(t) x dt + \sqrt{\beta(t)} dw
\]

Algorithm 2 PC sampling (VE SDE)

1: \( x_N \sim \mathcal{N}(0, \sigma_{mu}^2 I) \)
2: for \( i = N - 1 \) to 0 do
3: \( x'_i \leftarrow x_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2) s_{\theta^*}(x_{i+1}, \sigma_{i+1}) \)
4: \( z \sim \mathcal{N}(0, I) \)
5: \( x_i \leftarrow x'_i + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z \)
6: for \( j = 1 \) to \( M \) do
7: \( z \sim \mathcal{N}(0, I) \)
8: \( x_i \leftarrow x_i + \epsilon_t s_{\theta^*}(x_i, \sigma_i) + \sqrt{2\epsilon_i} z \)
9: return \( x_0 \)

Algorithm 3 PC sampling (VP SDE)

1: \( x_N \sim \mathcal{N}(0, I) \)
2: for \( i = N - 1 \) to 0 do
3: \( x'_i \leftarrow (2 - \sqrt{1 - \beta_{i+1}}) \beta_{i+1} s_{\theta^*}(x_{i+1}, i + 1) \)
4: \( z \sim \mathcal{N}(0, I) \)
5: \( x_i \leftarrow x'_i + \sqrt{\beta_{i+1}} z \)
6: for \( j = 1 \) to \( M \) do
7: \( z \sim \mathcal{N}(0, I) \)
8: \( x_i \leftarrow x_i + \epsilon_t s_{\theta^*}(x_i, i) + \sqrt{2\epsilon_i} z \)
9: return \( x_0 \)
Deep Generative Models—Probability Flow ODE

- A corresponding deterministic process to SDE: ODE (Ordinary Differential Equation)

\[
\frac{dx}{dt} = \left[ f(x, t) - \frac{1}{2} q(t)^2 \nabla_x \log p_t(x) \right] dt,
\]

\[
x_i = x_{i+1} + \frac{1}{2} (\sigma_i^2 - \sigma_{i+1}^2) s_{\theta^*}(x_{i+1}, \sigma_{i+1}), \quad i = 0, 1, \ldots, N - 1.
\]

\[
x_i = (2 - \sqrt{1 - \beta_{i+1}}) x_{i+1} + \frac{1}{2} \beta_{i+1} s_{\theta^*}(x_{i+1}, i+1), \quad i = 0, 1, \ldots, N - 1.
\]
Outline

• Background
  • Text-to-Speech Synthesis
  • Deep Generative Models

• Deep Generative Models for TTS
  • AR/Flow/GAN/VAE/Diffusion based TTS Models
  • Comparisons and Analyses

• Summary and Outlook
Deep Generative Models—Examples in Acoustic Model

- **Autoregressive models**
  - Tacotron 1/2, DeepVoice 3, TransformerTTS

- **Non-autoregressive models**: FastSpeech 1/2

- **Flow**
  - Glow-TTS

- **VAE**
  - Para. Tacotron 1/2

- **GAN**
  - Diff-TTS, Grad-TTS, DiffGAN-TTS, PriorGrad

- **Diffusion**
  - Diff-TTS, Grad-TTS, DiffGAN-TTS, PriorGrad
Deep Generative Models—Examples in Vocoder

- **Autoregressive models**
  - WaveNet, SampleRNN, WaveRNN
- **Flow**
  - Par. WaveNet, WaveGlow, FloWaveNet
- **GAN**
  - MelGAN, Para. WaveGAN, HiFiGAN
- **VAE**
  - WaveVAE
- **Diffusion**
  - DiffWave, WaveGrad, PriorGrad, SpecGrad

<table>
<thead>
<tr>
<th>Vocoder</th>
<th>Input</th>
<th>AR/NAR</th>
<th>Modeling</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>WaveNet [260]</td>
<td>Linguistic Feature</td>
<td>AR</td>
<td>/</td>
<td>CNN</td>
</tr>
<tr>
<td>SampleRNN [239]</td>
<td>/</td>
<td>AR</td>
<td>/</td>
<td>RNN</td>
</tr>
<tr>
<td>WaveRNN [151]</td>
<td>Linguistic Feature</td>
<td>AR</td>
<td>/</td>
<td>RNN</td>
</tr>
<tr>
<td>LPCNet [370]</td>
<td>BFCC</td>
<td>AR</td>
<td>/</td>
<td>RNN</td>
</tr>
<tr>
<td>Univ. WaveRNN [221]</td>
<td>Mel-Spectrogram</td>
<td>AR</td>
<td>/</td>
<td>RNN</td>
</tr>
<tr>
<td>SC-WaveRNN [271]</td>
<td>Mel-Spectrogram</td>
<td>AR</td>
<td>/</td>
<td>RNN</td>
</tr>
<tr>
<td>MB WaveRNN [426]</td>
<td>Mel-Spectrogram</td>
<td>AR</td>
<td>/</td>
<td>RNN</td>
</tr>
<tr>
<td>FFTNet [146]</td>
<td>Cepstrum</td>
<td>AR</td>
<td>/</td>
<td>CNN</td>
</tr>
<tr>
<td>iSTFTNet [153]</td>
<td>Mel-Spectrogram</td>
<td>/</td>
<td>/</td>
<td>CNN</td>
</tr>
<tr>
<td>Par. WaveNet [261]</td>
<td>Linguistic Feature</td>
<td>NAR</td>
<td>Flow</td>
<td>CNN</td>
</tr>
<tr>
<td>WaveGlow [285]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>Flow</td>
<td>Hybrid/CNN</td>
</tr>
<tr>
<td>FloWaveNet [166]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>Flow</td>
<td>Hybrid/CNN</td>
</tr>
<tr>
<td>WaveFlow [277]</td>
<td>Mel-Spectrogram</td>
<td>AR</td>
<td>Flow</td>
<td>Hybrid/CNN</td>
</tr>
<tr>
<td>SqueezeWave [441]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>Flow</td>
<td>CNN</td>
</tr>
<tr>
<td>WaveGAN [69]</td>
<td>/</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN</td>
</tr>
<tr>
<td>GELP [150]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN</td>
</tr>
<tr>
<td>GAN-TTS [23]</td>
<td>Linguistic Feature</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN</td>
</tr>
<tr>
<td>MelGAN [182]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN</td>
</tr>
<tr>
<td>Par. WaveGAN [410]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN</td>
</tr>
<tr>
<td>HiFi-GAN [178]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>GAN</td>
<td>Hybrid/CNN</td>
</tr>
<tr>
<td>VocGAN [416]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN</td>
</tr>
<tr>
<td>GED [97]</td>
<td>Linguistic Feature</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN</td>
</tr>
<tr>
<td>Fre-GAN [164]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN</td>
</tr>
<tr>
<td>Wave-VAE [274]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>VAE</td>
<td>CNN</td>
</tr>
<tr>
<td>WaveGrad [41]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>Diffusion</td>
<td>Hybrid/CNN</td>
</tr>
<tr>
<td>DiffWave [180]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>Diffusion</td>
<td>Hybrid/CNN</td>
</tr>
<tr>
<td>PriorGrad [189]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>Diffusion</td>
<td>Hybrid/CNN</td>
</tr>
<tr>
<td>SpecGrad [176]</td>
<td>Mel-Spectrogram</td>
<td>NAR</td>
<td>Diffusion</td>
<td>Hybrid/CNN</td>
</tr>
</tbody>
</table>
Deep Generative Models—Examples in End-to-End TTS

• Autoregressive models
  • Char2Wav

• Flow
  • ClariNet, Wave-Tacotron

• GAN
  • FastSpeech 2s, EATS

• Diffusion
  • WaveGrad 2

• VAE+Flow+GAN
  • VITS, NaturalSpeech

<table>
<thead>
<tr>
<th>Model</th>
<th>One-Stage Training</th>
<th>AR/NAR</th>
<th>Modeling</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Char2Wav [321]</td>
<td>N</td>
<td>AR</td>
<td>Seq2Seq</td>
<td>RNN</td>
</tr>
<tr>
<td>ClariNet [275]</td>
<td>N</td>
<td>AR</td>
<td>Flow</td>
<td>CNN</td>
</tr>
<tr>
<td>FastSpeech 2s [298]</td>
<td>Y</td>
<td>NAR</td>
<td>GAN</td>
<td>Self-Att/CNN</td>
</tr>
<tr>
<td>EATS [70]</td>
<td>Y</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN</td>
</tr>
<tr>
<td>Wave-Tacotron [392]</td>
<td>Y</td>
<td>AR</td>
<td>Flow</td>
<td>CNN/RNN/Hybrid</td>
</tr>
<tr>
<td>EfficientTTS-Wav [241]</td>
<td>Y</td>
<td>NAR</td>
<td>GAN</td>
<td>CNN/Self-Att/Hybrid</td>
</tr>
<tr>
<td>VITS [163]</td>
<td>Y</td>
<td>NAR</td>
<td>VAE+Flow+GAN</td>
<td>CNN/Self-Att/Hybrid</td>
</tr>
<tr>
<td>NaturalSpeech [351]</td>
<td>Y</td>
<td>NAR</td>
<td>VAE+Flow+GAN</td>
<td>CNN/Self-Att/Hybrid</td>
</tr>
</tbody>
</table>
Autoregressive Model for TTS

• WaveNet: autoregressive model with dilated causal convolution

• Other works
  • Acoustic model: Tacotron 1/2, DeepVoice 3, TransformerTTS
  • Vocoder: SampleRNN, WaveRNN
Flow for TTS

• Parallel WaveNet (AR)
  • Knowledge distillation: Student (IAF), Teacher (AF)
  • Combine the best of both worlds
    • Parallel inference of IAF student
    • Parallel training of AF teacher

• Other works
  • ClariNet
Flow for TTS

- WaveGlow (Bipartite)
  - Flow based transformation

\[ z = f_k^{-1} \circ f_{k-1}^{-1} \circ \ldots f_0^{-1}(x) \quad x = f_0 \circ f_1 \circ \ldots \circ f_k(z) \quad z \sim \mathcal{N}(z; 0, I) \]

- Affine Coupling Layer

\[ x_a, x_b = \text{split}(x) \]
\[ (\log s, t) = WN(x_a, \text{mel-spectrogram}) \]
\[ x_b' = s \circ x_b + t \]
\[ f_{\text{coupling}}^{-1}(x) = \text{concat}(x_a, x_b') \]

- Other works
  - FloWaveNet, WaveFlow
Flow for TTS

• Glow-TTS (Bipartite) for acoustic model
  • Log likelihood
    \[ \log P_X(x|c) = \log P_Z(z|c) + \log \left| \det \frac{\partial f_{dec}^{-1}(x)}{\partial x} \right| \]
  • Prior is learnt from phoneme text
    \[ \log P_Z(z|c; \theta, A) = \sum_{j=1}^{T_{mel}} \log \mathcal{N}(z_j; \mu_A(j), \sigma_A(j)) \]
  • Alignment A is obtained by monotonic alignment search

• Other works
  • FlowTTS, Flowtron
GAN for TTS

- With specific designs on generators, discriminators, and loss functions
  - Multi-scale discriminator in MelGAN
  - Multi-period discriminator in HiFiGAN

- Other works
  - Para. WaveGAN, BigVGAN
  - FastSpeech 2s, EATS
VAE + Flow + GAN for TTS

- NaturalSpeech for fully end-to-end TTS
  - Reconstruction: $z \sim q(z|x)$, $x \sim p(x|z)$
  - Prior prediction: $z \sim p(z|y)$
  - Solutions in NaturalSpeech
    - Phoneme encoder with phoneme pre-training
    - Differentiable durator
    - Bidirectional prior/posterior
    - Memory based VAE

- Other works
  - VITS, Glow-WaveGAN

---

<table>
<thead>
<tr>
<th>Human Recordings</th>
<th>NaturalSpeech</th>
<th>Wilcoxon p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.58 ± 0.13</td>
<td>4.56 ± 0.13</td>
<td>0.7145</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Human Recordings</th>
<th>NaturalSpeech</th>
<th>Wilcoxon p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>−0.01</td>
<td>0.6902</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>MOS</th>
<th>CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastSpeech 2 [18] + HiFiGAN [17]</td>
<td>4.32 ± 0.15</td>
<td>−0.33</td>
</tr>
<tr>
<td>Glow-TTS [13] + HiFiGAN [17]</td>
<td>4.34 ± 0.13</td>
<td>−0.26</td>
</tr>
<tr>
<td>Grad-TTS [14] + HiFiGAN [17]</td>
<td>4.37 ± 0.13</td>
<td>−0.24</td>
</tr>
<tr>
<td>VITS [15]</td>
<td>4.43 ± 0.13</td>
<td>−0.20</td>
</tr>
<tr>
<td>NaturalSpeech</td>
<td>4.56 ± 0.13</td>
<td>0</td>
</tr>
</tbody>
</table>

11/27/2022

Deep Generative Models for TTS, Xu Tan
Diffusion for TTS

- **Vocoder**: DiffWave, WaveGrad
- **Acoustic model**: Diff-TTS, Grad-TTS
Diffusion—Speedup

- Sampling steps, latency

<table>
<thead>
<tr>
<th>System</th>
<th>RTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastSpeech 2 [18] + HiFiGAN [17]</td>
<td>0.011</td>
</tr>
<tr>
<td>Grad-TTS [14] (1000) + HiFiGAN [17]</td>
<td>4.120</td>
</tr>
<tr>
<td>Grad-TTS [14] (10) + HiFiGAN [17]</td>
<td>0.082</td>
</tr>
<tr>
<td>VITS [15]</td>
<td>0.014</td>
</tr>
<tr>
<td>NaturalSpeech</td>
<td>0.013</td>
</tr>
</tbody>
</table>
**Diffusion—Speedup**

- **Prior distribution**: standard Gaussian $\rightarrow$ non-standard, e.g., PriorGrad, SpecGrad, Grad-TTS, DDGM

- **Forward process**: fixed $\rightarrow$ learnable, e.g., Variational diffusion models

- **Diffusion + X**
  - Diffusion + GAN: e.g., DiffusionGAN
  - Diffusion + VAE: e.g., Latent Diffusion
  - Diffusion + KD: e.g., Progressive Distillation

- **Diffusion assumption**: Markovian $\rightarrow$ non-Markovian: e.g., DDIM

- **Reverse process** (noise levels, schedule, or variance): fixed $\rightarrow$ learnable, e.g., BDDM, Improved DDPM

- **SDE/ODE solver**: e.g., Euler-Maruyama, Runge-Kutta, adaptive-size SDE, PNDM, DPM-Solver, DPM-Solver++
Outline

• Background
  • Text-to-Speech Synthesis
  • Deep Generative Models

• Deep Generative Models for TTS
  • AR/Flow/GAN/VAE/Diffusion based TTS Models
  • Comparisons and Analyses

• Summary and Outlook
Deep Generative Models—Comparisons

• Find a $z$ and transform it into $x$
Deep Generative Models—Comparisons

- Pros and cons

<table>
<thead>
<tr>
<th>Generative Models</th>
<th>GAN</th>
<th>AR</th>
<th>Flow</th>
<th>VAE</th>
<th>Diffusion/SDE</th>
<th>Diffusion/ODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-Quality</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Fast Sampling</td>
<td>Y</td>
<td>N</td>
<td>Y*</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Mode Diversity</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Likelihood Estimation</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y*</td>
<td>Y*</td>
<td>Y</td>
</tr>
<tr>
<td>Error Propagation</td>
<td>N</td>
<td>Y</td>
<td>N*</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Stable Training</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N*</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Latent Manipulation</td>
<td>Y*</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>
Outline

• Background
  • Text-to-Speech Synthesis
  • Deep Generative Models

• Deep Generative Models for TTS
  • AR/Flow/GAN/VAE/Diffusion based TTS Models
  • Comparisons and Analyses

• Summary and Outlook
Summary

- Text-to-speech synthesis is a typical conditional data generation task
  - Suffer from one-to-many mapping

Text \(\xrightarrow{\text{duration, pitch, sound volume, prosody, speaker, style, emotion, etc}}\) Speech

- Usually handled by deep generative models
  - AR/Flow/GAN/VAE/Diffusion models
Outlook—Exploiting Generative Models

• Considering the pros and cons of deep generative models, can we fully exploit them in different scenarios?

<table>
<thead>
<tr>
<th>Generative Models</th>
<th>GAN</th>
<th>AR</th>
<th>Flow</th>
<th>VAE</th>
<th>Diffusion/SDE</th>
<th>Diffusion/ODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-Quality</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Fast Sampling</td>
<td>Y</td>
<td>N</td>
<td>Y*</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Mode Diversity</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Likelihood Estimation</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y*</td>
<td>Y*</td>
<td>Y</td>
</tr>
<tr>
<td>Error Propagation</td>
<td>N</td>
<td>Y</td>
<td>N*</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Stable Training</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N*</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Latent Manipulation</td>
<td>Y*</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

• Find a killer application for each generative model?
• Will a specific kind of generative model take all? e.g., diffusion model
Outlook—Exploiting Generative Models

• Understanding diffusion models
  • Why diffusion models are better than other models?
  • Difference between hierarchical VAEs and continuous normalizing flows

• Improving diffusion models
  • What is the limit of sampling steps? Is one step meaningful?
  • New diffusion or denoising process? e.g., non-diffusion
  • New training procedure?
Outlook—Exploring Generative Models

• Considering the pros and cons of deep generative models, can we design brand-new models that inherit the advantages and avoid the disadvantages?

<table>
<thead>
<tr>
<th>Generative Models</th>
<th>GAN</th>
<th>AR</th>
<th>Flow</th>
<th>VAE</th>
<th>Diffusion/SDE</th>
<th>Diffusion/ODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-Quality</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Fast Sampling</td>
<td>Y</td>
<td>N</td>
<td>Y*</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Mode Diversity</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Likelihood Estimation</td>
<td>N</td>
<td>Y</td>
<td>Y*</td>
<td>Y*</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Error Propagation</td>
<td>N</td>
<td>Y</td>
<td>N*</td>
<td>N</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Stable Training</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N*</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Latent Manipulation</td>
<td>Y*</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

• e.g., AR + Flow, VAE + GAN, VAE + Flow, Diffusion + GAN, Diffusion + VAE
• Can we stop borrowing models from computer vision, invent something new for speech?
Reference

See the references in:

A Survey on Neural Speech Synthesis


---

A Survey on Neural Speech Synthesis

Xu Tan, Tao Qin, Frank Soong, Tie-Yan Liu
{xuta,taoqin,frankkps,tyliu}@microsoft.com
Microsoft Research Asia

https://speechresearch.github.io/

---

Speech Research

This page lists some speech related research at Microsoft Research Asia, conducted by the team led by Xu Tan. The research topics cover text to speech, singing voice synthesis, music generation, automatic speech recognition, etc. Some research are open-sourced via NeuralSpeech and Muzic.

We are hiring researchers on speech, NLP, and deep learning at Microsoft Research Asia. Please contact xuta@microsoft.com if you have interests.

- Machine Translation with Speech-Aware Length Control for Video Dubbing
  August 30, 2022

- BinauralGrad: A Two-Stage Conditional Diffusion Probabilistic Model for Binaural Audio Synthesis
  May 29, 2022

- NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality
  May 03, 2022

- Mixed-Phoneme BERT: Improving BERT with Mixed Phoneme and Sup-Phoneme Representations for Text to Speech
  April 02, 2022

- AdaSpeech 4: Adaptive Text to Speech in Zero-Shot Scenarios
  March 06, 2022

- Speech-T: Transducer for Text to Speech and Beyond
  October 06, 2021

- TeleMelody: Lyric-to-Melody Generation with a Template-Based Two-Stage Method
A book on TTS

A book on “Neural Text-to-Speech Synthesis”, by Xu Tan will be published soon!

Watch this repo for update: https://github.com/tts-tutorial/book
We are hiring

• Research FTE (social/campus hire)
  • Speech/Audio/Music Generation, Machine Translation, etc
  • Digital Human Generation (Talking Face Generation, 3D Synthesis, etc)
  • Generative Models (AR, GAN, Flow, VAE, Diffusion, etc)
  • Machine Learning, Deep Learning

• Research Intern
  • Speech, Music, Machine Translation, Digital Human Generation, Machine Learning

Machine Learning Group, Microsoft Research Asia
Xu Tan xuta@microsoft.com

11/27/2022
Thank You!

Xu Tan/谭旭
Principal Research Manager @ Microsoft Research Asia
xuta@microsoft.com
tan-xu.github.io
https://www.microsoft.com/en-us/research/people/xuta/
https://speechresearch.github.io/