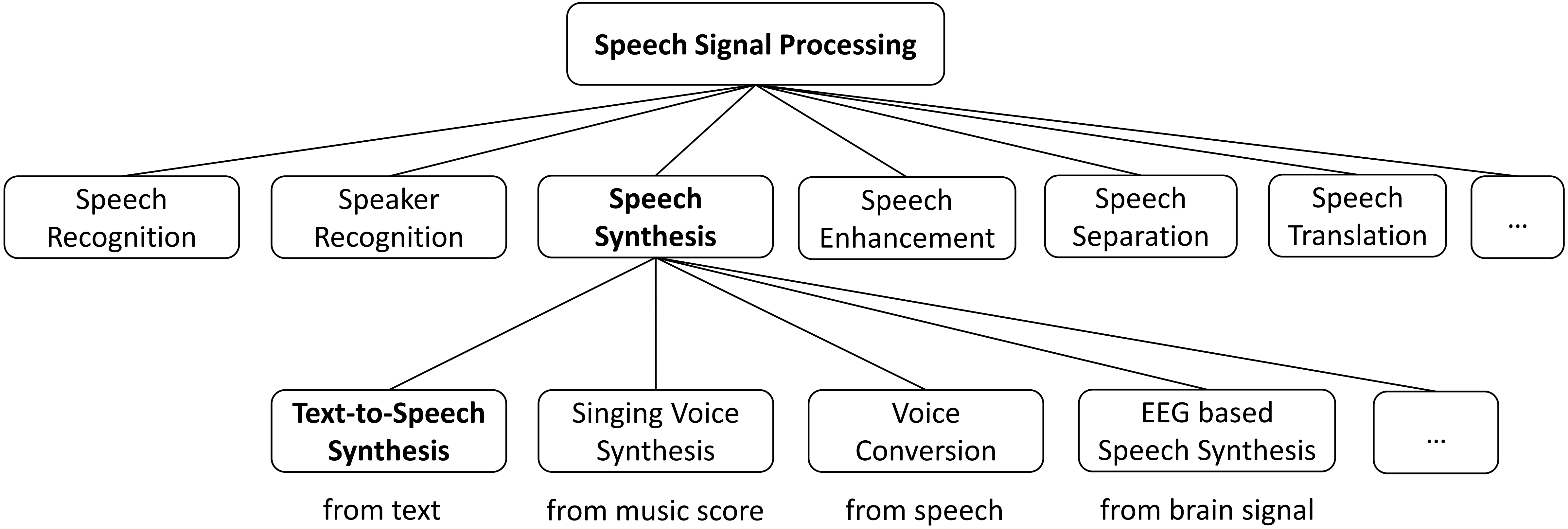
Deep Generative Models for Text-to-Speech Synthesis

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
…
Text-to-Speech Synthesis
Singing Voice Synthesis
Voice Conversion
EEG based Speech Synthesis
…
from text
from music score
from speech
from brain signal

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**

Neural TTS

WaveNet (DeepMind)

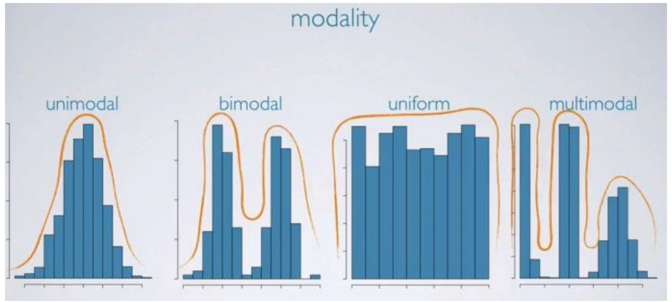
1950s	1970s	1990s		2010s	2016
Articulatory Synthesis	Formant Synthesis	Concatenative Synthesis	Statistical Parametric Synthesis	Neural Speech Synthesis	(Deep) Neural Speech Synthesis

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
  - …

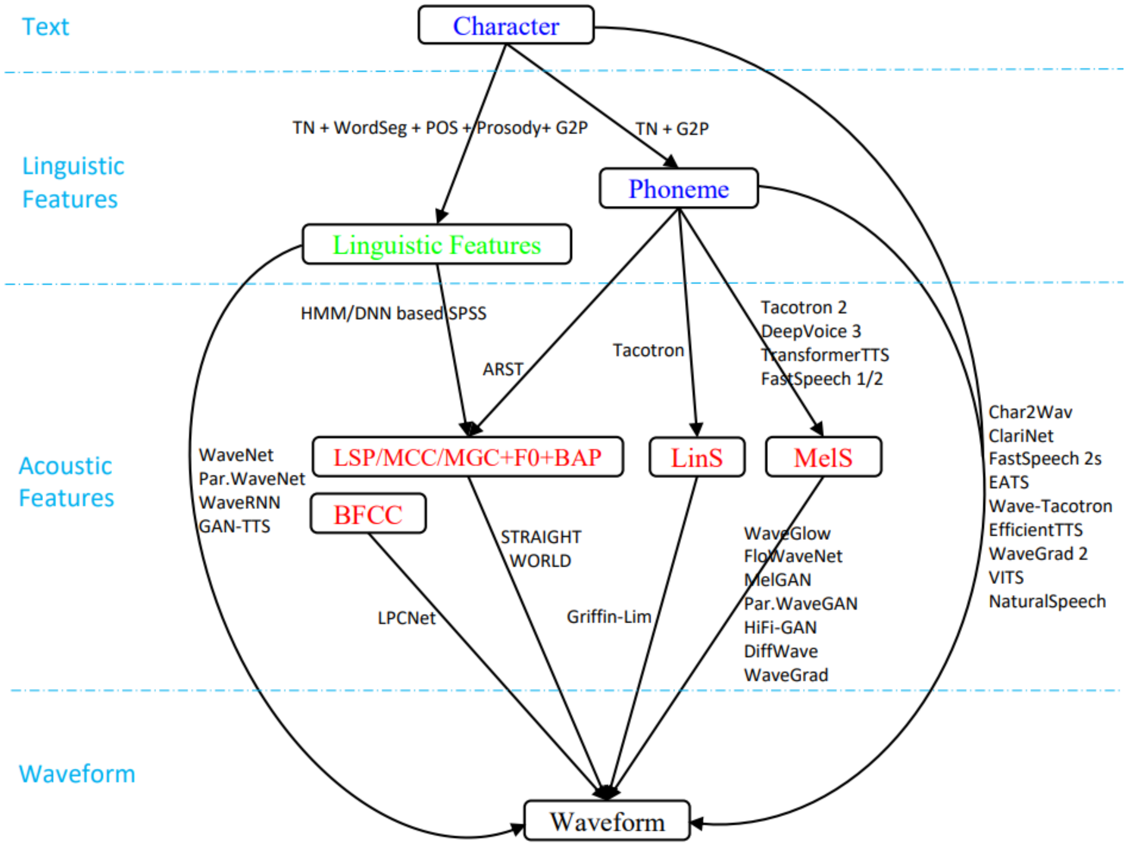
Text $\xrightarrow{\text{duration, pitch, sound volume, prosody, speaker, style, emotion, etc}}$ Speech

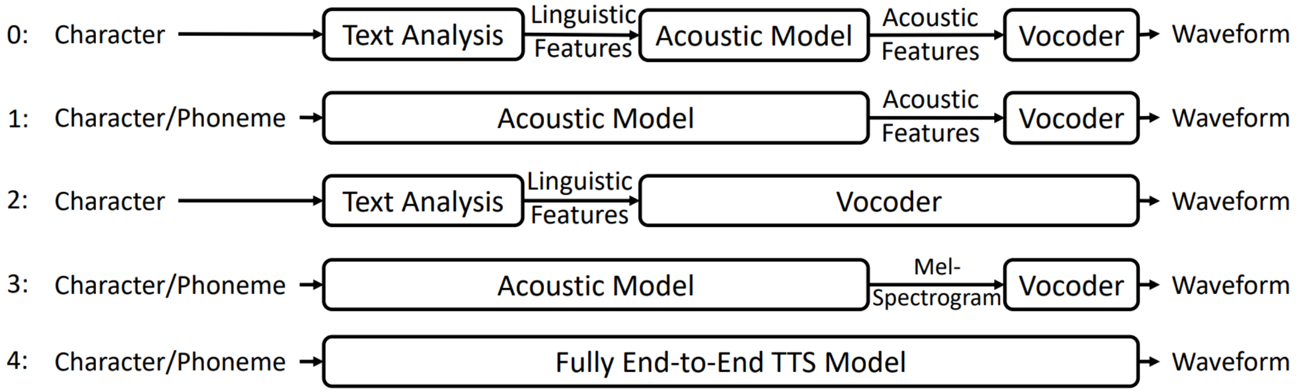
- Text-to-speech mapping
  - Not point-wise, but **distribution-wise**
  - Usually not single-modal, but **multi-modal**



Typical Methods to Handle One-to-Many Mapping in TTS

- Split text-to-speech conversion into **multiple stages**

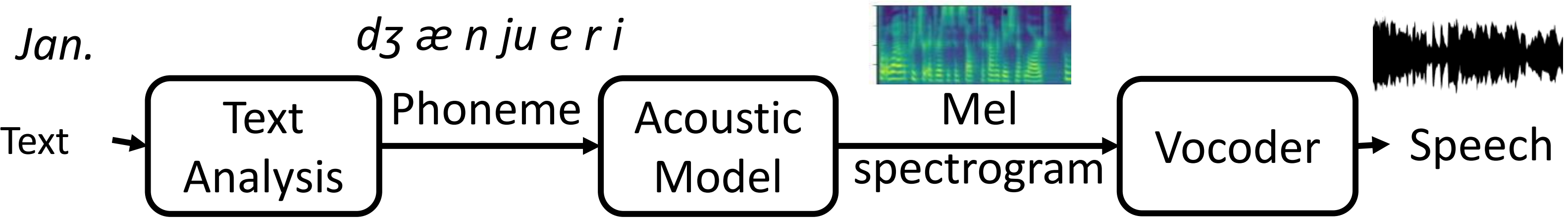




Stage	Models
0	SPSS [418, 358, 417, 427, 359]
1	ARST [377]
2	WaveNet [255], DeepVoice 1/2 [8, 88], Par. WaveNet [256], WaveRNN [151], HiFi-GAN [23]
3	DeepVoice 3 [271], Tacotron 2 [304], FastSpeech 1/2 [291, 293], WaveGlow [280], FloWaveNet [164]
4	Char2Wav [316], ClariNet [270], FastSpeech 2s [293], EATS [70], VITS [161], NaturalSpeech [346]

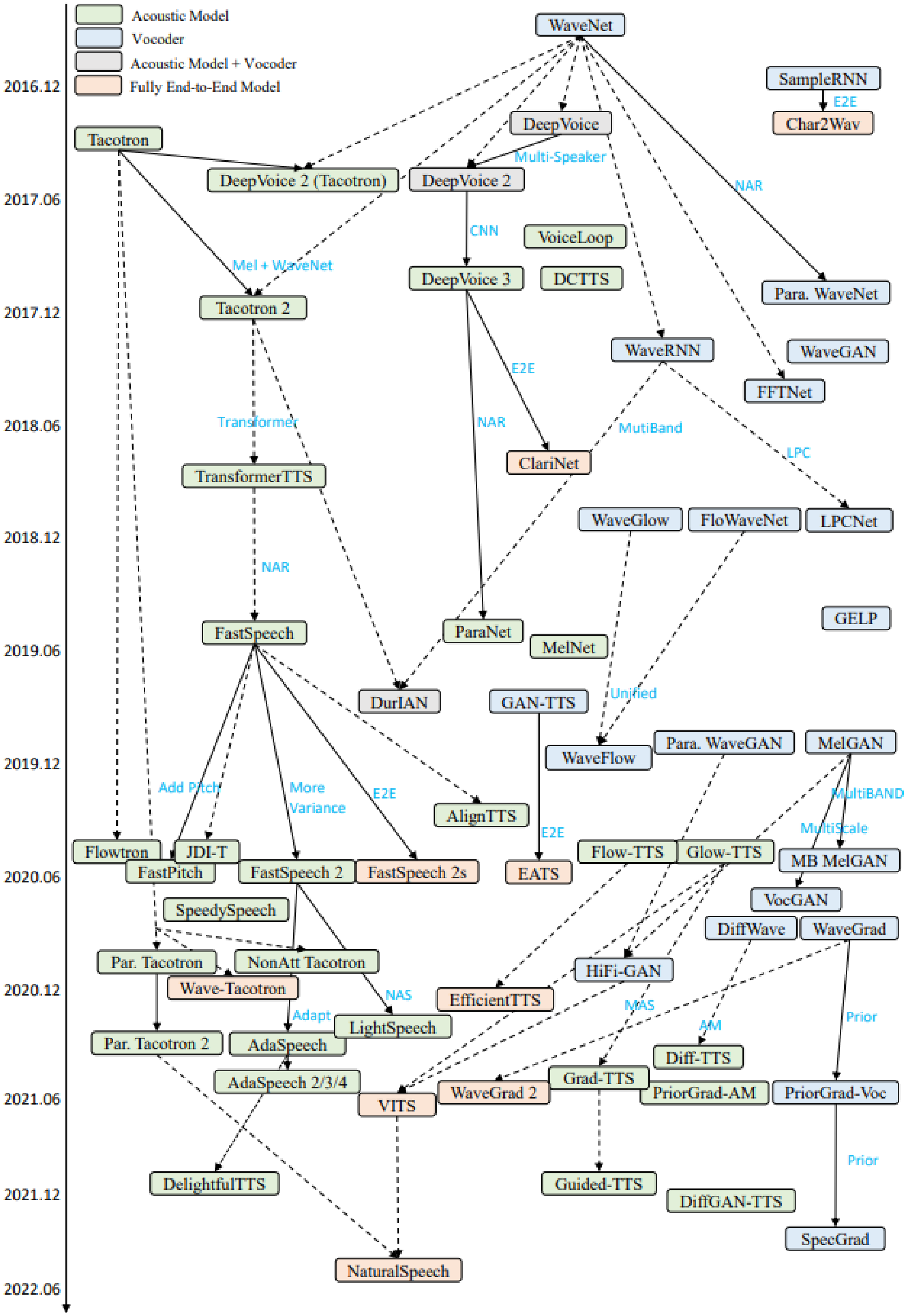
Typical Neural TTS Pipeline

- Text analysis, acoustic model, and vocoder



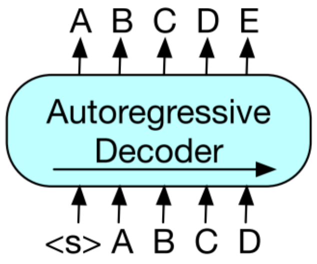
- Text analysis: text → linguistic features
- Acoustic model: linguistic features → acoustic features
- Vocoder: acoustic features → speech

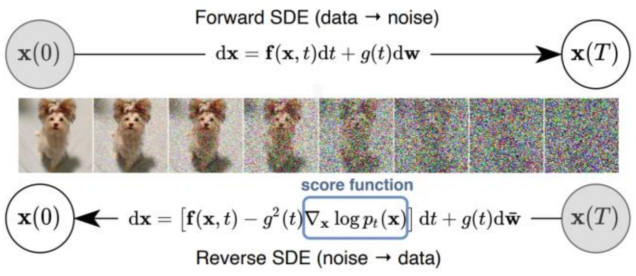
One-to-many mapping is alleviated, but not eliminated!



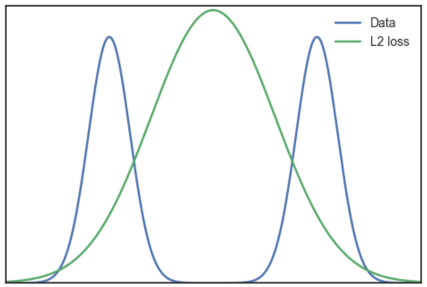
How to Model One-to-Many Mapping (Multimodal Distribution)

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

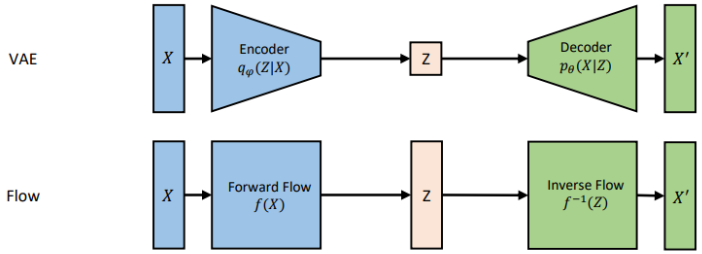




- Advanced loss function
  - L1/L2 loss
    - → Distribution-wise loss (e.g., SSIM, GMM)
    - → **GAN loss** (match any distribution)



- Synthesis-by-analysis
  - X → Z → 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**

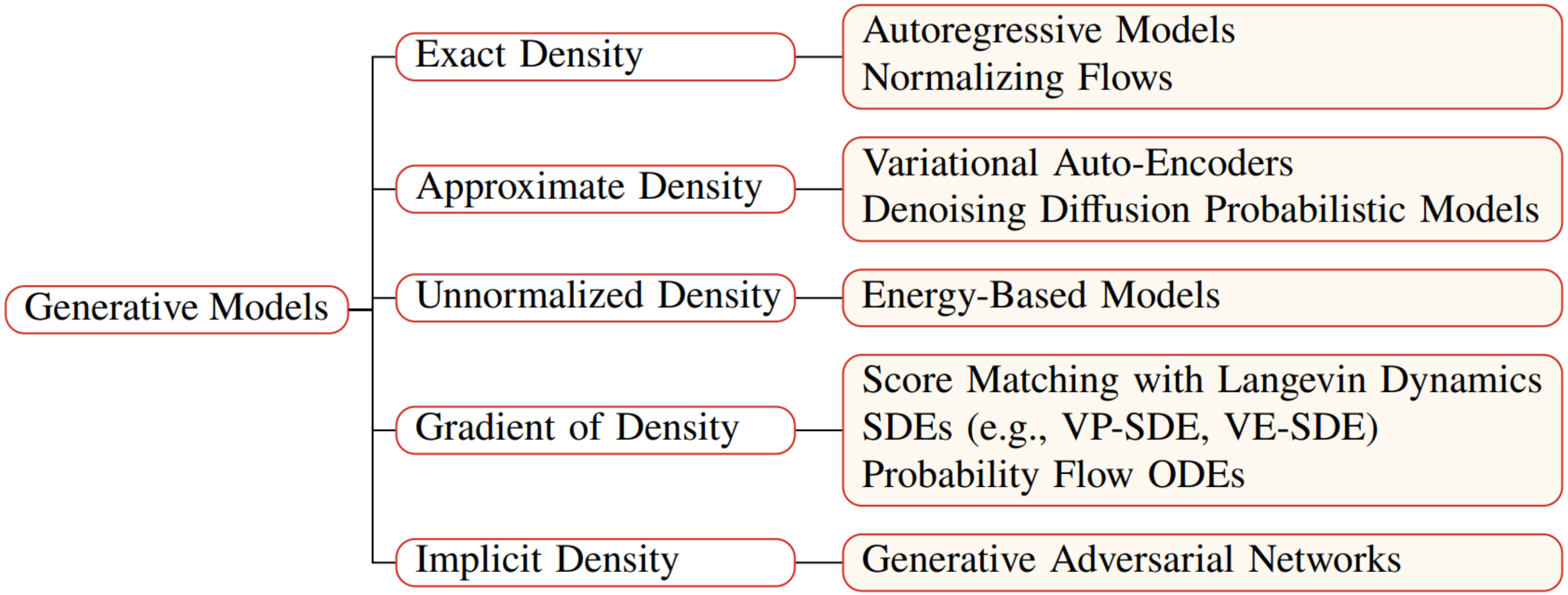
Deep Learning (Representation Learning)

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_θ to be close to the true data distribution P_D
  - **Data generation**: sample new data from P_θ
  - **Density estimation**: predict the density/probability of a data point
- Taxonomy of deep generative models



Deep Generative Models—GAN

- Generative Adversarial Networks

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

  - **Not to find a corresponding z for x**, but to directly **match the distribution of x**

Deep Generative Models—Flow

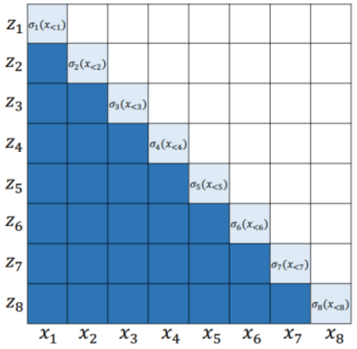
- Normalizing Flows: finding a z for x, and convert z back to x
  - $z = f_k^{-1} f_{k-1}^{-1} \dots f_0^{-1}(x)$
  - $x = f_0 f_1 \dots f_k(z), z \sim N(0, 1)$
- Training: maximizing the log likelihood $p(x)$
  - $\log p(x) = \log p(z) + \log\det\left(\frac{dz}{dx}\right) = \log p(z) + \sum_{i=1}^{k} \log|\det(J(f_i^{-1}(x)))|$
  - 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**

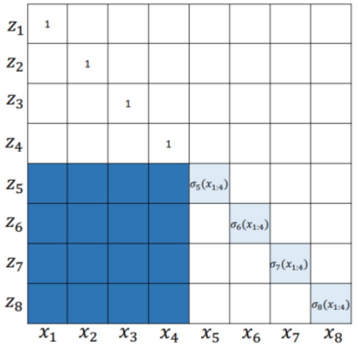
Deep Generative Models—Flow

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

Flow		Evaluation $z = f^{-1}(x)$	Synthesis $x = f(z)$
AR	AF [42]	$z_t = \frac{x_t - \mu_t(x_{<t})}{\sigma_t(x_{<t})}$	$x_t = z_t \cdot \sigma_t(x_{<t}) + \mu_t(x_{<t})$
	IAF [38]	$z_t = x_t \cdot \sigma_t(z_{<t}) + \mu_t(z_{<t})$	$x_t = \frac{z_t - \mu_t(z_{<t})}{\sigma_t(z_{<t})}$
Bipartite	RealNVP [36]	$z_a = x_a,$	$x_a = z_a,$
	Glow [39]	$z_b = x_b \cdot \sigma_b(x_a;\theta) + \mu_b(x_a;\theta)$	$x_b = \frac{z_b - \mu_b(x_a;\theta)}{\sigma_b(x_a;\theta)}$

- 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





[Ping, 2019]

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 $p(x)$

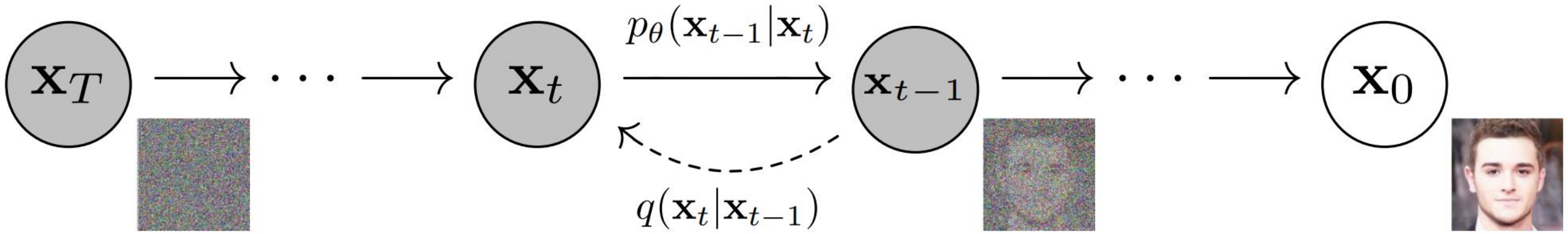
$$
\begin{aligned}
\log p(x) &= \log \int p(x|z)p(z)dz = \log \int q(z|x)\frac{p(x|z)p(z)}{q(z|x)}dz \\
&= \log \mathbb{E}_{z\sim q(z|x)} \frac{p(x|z)p(z)}{q(z|x)} \geq \mathbb{E}_{z\sim q(z|x)} \log \frac{p(x|z)p(z)}{q(z|x)} \\
&= \mathbb{E}_{z\sim q(z|x)} \log p(x|z) - KL(q(z|x)||p(z)),
\end{aligned}
$$

  - 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



  - Forward process

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

  - Backward 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

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

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

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

Deep Generative Models—DDPM

- Training and inference pipeline

Algorithm 1 Training

repeat
  Sample $x_0 \sim q_{data}, \epsilon \sim \mathcal{N}(0, I)$
  Sample $t \sim \mathcal{U}(\{1, \cdots, T\})$
  $\mathcal{L} = \| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t) \|^2$
  Update θ with $\nabla_\theta \mathcal{L}$
until converged

Algorithm 2 Sampling

Sample $x_T \sim \mathcal{N}(0, I)$
for $t = T, T-1, \cdots, 1$ **do**
  Sample $z \sim \mathcal{N}(0, I)$ if $t > 1$; else $z = 0$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)) + \sigma_t z$
end for
return x_0

Deep Generative Models—SMLD

- Score Matching with Langevin Dynamics (SMLD) [Song, 2020]
  - Score: the score of a probability density p(x) is ∇x log p(x)
- Training: score matching for score estimation

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

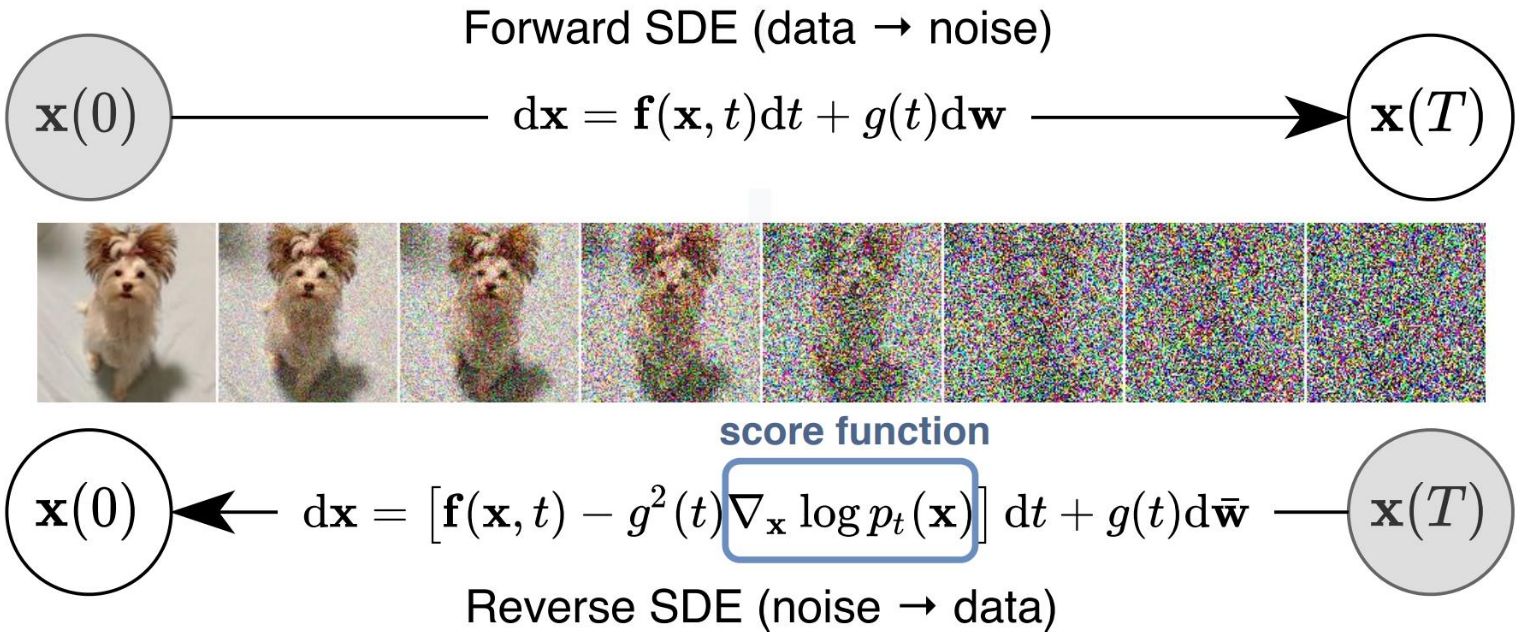
- Inference: sampling with Langevin dynamics

$$\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + c\nabla \log p(\boldsymbol{x}_i) + \sqrt{2c}\boldsymbol{\epsilon}, \quad i = 0, 1, ..., K$$

$$\nabla \log p(\boldsymbol{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}$$

Deep Generative Models—SDE

- Stochastic Differential Equation (SDE) [Song, 2020]
  - Extend discrete time to continuous time



$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_t \Big\{ \lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)} \Big[\big\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) \big\|_2^2 \Big] \Big\}.$$

Deep Generative Models—VE-SDE, VP-SDE

- VE-SDE (Variance-Exploding Stochastic Differential Equation) and SMLD [Song, 2020]

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\,\mathbf{z}_{i-1}, \quad i = 1, \cdots, N, \qquad \mathrm{d}\mathbf{x} = \sqrt{\frac{\mathrm{d}\left[\sigma^2(t)\right]}{\mathrm{d}t}}\,\mathrm{d}\mathbf{w}$$

- VP-SDE (Variance-Preserving Stochastic Differential Equation) and DDPM [Song, 2020]

$$\mathbf{x}_i = \sqrt{1-\beta_i}\,\mathbf{x}_{i-1} + \sqrt{\beta_i}\,\mathbf{z}_{i-1}, \quad i = 1, \cdots, N. \qquad \mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}$$

$$p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) = \begin{cases} \mathcal{N}\left(\mathbf{x}(t); \mathbf{x}(0), [\sigma^2(t) - \sigma^2(0)]\mathbf{I}\right), & \text{(VE SDE)} \\ \mathcal{N}\left(\mathbf{x}(t); \mathbf{x}(0)e^{-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s}, \mathbf{I} - \mathbf{I}e^{-\int_0^t \beta(s)\mathrm{d}s}\right) & \text{(VP SDE)} \\ \mathcal{N}\left(\mathbf{x}(t); \mathbf{x}(0)e^{-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s}, [1 - e^{-\int_0^t \beta(s)\mathrm{d}s}]^2\mathbf{I}\right) & \text{(sub-VP SDE)} \end{cases}.$$

Algorithm 2 PC sampling (VE SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}_i' \leftarrow \mathbf{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2)\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_{i+1}, \sigma_{i+1})$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i' + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\,\mathbf{z}$
  for $j = 1$ **to** M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_i, \sigma_i) + \sqrt{2\epsilon_i}\,\mathbf{z}$
return $\mathbf{x}_0$

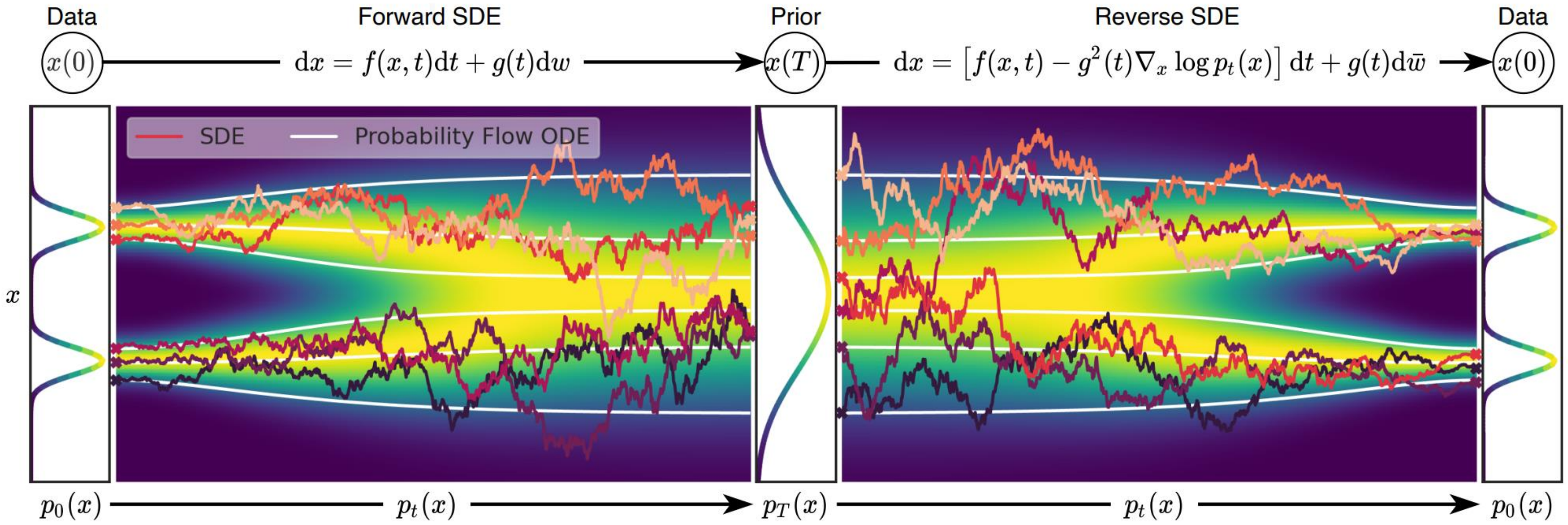
Algorithm 3 PC sampling (VP SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}_i' \leftarrow (2 - \sqrt{1-\beta_{i+1}})\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_{i+1}, i+1)$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i' + \sqrt{\beta_{i+1}}\,\mathbf{z}$
  Predictor
  for $j = 1$ **to** M **do**
  Corrector
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_i, i) + \sqrt{2\epsilon_i}\,\mathbf{z}$
return $\mathbf{x}_0$

Deep Generative Models—Probability Flow ODE

- A corresponding deterministic process to SDE: ODE (Ordinary Differential Equation) [Song, 2020]

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t,$$



$$\mathbf{x}_i = \mathbf{x}_{i+1} + \frac{1}{2}(\sigma_{i+1}^2 - \sigma_i^2)\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_{i+1}, \sigma_{i+1}), \quad i = 0, 1, \cdots, N-1.$$

$$\mathbf{x}_i = (2 - \sqrt{1 - \beta_{i+1}})\mathbf{x}_{i+1} + \frac{1}{2}\beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_{i+1}, i+1), \quad i = 0, 1, \cdots, N-1.$$

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
- Diffusion
  - Diff-TTS, Grad-TTS, DiffGAN-TTS, PriorGrad

	Acoustic Model	Input→Output	AR/NAR	Modeling	Structure
RNN	Tacotron [382]	Ch→LinS	AR	Seq2Seq	Hybrid/RNN
	Tacotron 2 [303]	Ch→MelS	AR	Seq2Seq	RNN
	DurIAN [418]	Ph→MelS	AR	Seq2Seq	RNN
	Non-Att Tacotron [304]	Ph→MelS	AR	/	Hybrid/CNN/RNN
	MelNet [367]	Ch→MelS	AR	/	RNN
CNN	DeepVoice [8]	Ch/Ph→MelS	AR	/	CNN
	DeepVoice 2 [87]	Ch/Ph→MelS	AR	/	CNN
	DeepVoice 3 [270]	Ch/Ph→MelS	AR	Seq2Seq	CNN
	ParaNet [268]	Ph→MelS	NAR	Seq2Seq	CNN
	DCTTS [332]	Ch→MelS	AR	Seq2Seq	CNN
	SpeedySpeech [361]	Ph→MelS	NAR	/	CNN
	TalkNet 1/2 [19, 18]	Ch→MelS	NAR	/	CNN
Transformer	TransformerTTS [192]	Ph→MelS	AR	Seq2Seq	Self-Att
	MultiSpeech [39]	Ph→MelS	AR	Seq2Seq	Self-Att
	FastSpeech 1/2 [290, 292]	Ph→MelS	NAR	Seq2Seq	Self-Att
	AlignTTS [429]	Ch/Ph→MelS	NAR	Seq2Seq	Self-Att
	JDI-T [197]	Ph→MelS	NAR	Seq2Seq	Self-Att
	FastPitch [181]	Ph→MelS	NAR	Seq2Seq	Self-Att
	AdaSpeech 1/2/3 [40, 403, 404]	Ph→MelS	NAR	Seq2Seq	Self-Att
	DenoiSpeech [434]	Ph→MelS	NAR	Seq2Seq	Self-Att
	DeviceTTS [126]	Ph→MelS	NAR	/	Hybrid/DNN/RNN
	LightSpeech [220]	Ph→MelS	NAR	/	Hybrid/Self-Att/CNN
Flow	Flow-TTS [234]	Ch/Ph→MelS	NAR*	Flow	Hybrid/CNN/RNN
	Glow-TTS [159]	Ph→MelS	NAR	Flow	Hybrid/Self-Att/CNN
	Flowtron [366]	Ph→MelS	AR	Flow	Hybrid/RNN
	EfficientTTS [235]	Ch→MelS	NAR	Flow	Hybrid/CNN
VAE	GMVAE-Tacotron [119]	Ph→MelS	AR	VAE	Hybrid/RNN
	VAE-TTS [443]	Ph→MelS	AR	VAE	Hybrid/RNN
	BVAE-TTS [187]	Ph→MelS	NAR	VAE	CNN
	Para. Tacotron 1/2 [74, 75]	Ph→MelS	NAR	VAE	Hybrid/Self-Att/CNN
GAN	GAN exposure [99]	Ph→MelS	AR	GAN	Hybrid/RNN
	TTS-Stylization [224]	Ch→MelS	AR	GAN	Hybrid/RNN
	Multi-SpectroGAN [186]	Ph→MelS	NAR	GAN	Hybrid/Self-Att/CNN
Diffusion	Diff-TTS [141]	Ph→MelS	NAR*	Diffusion	Hybrid/CNN
	Grad-TTS [276]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN
	PriorGrad [185]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN

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

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

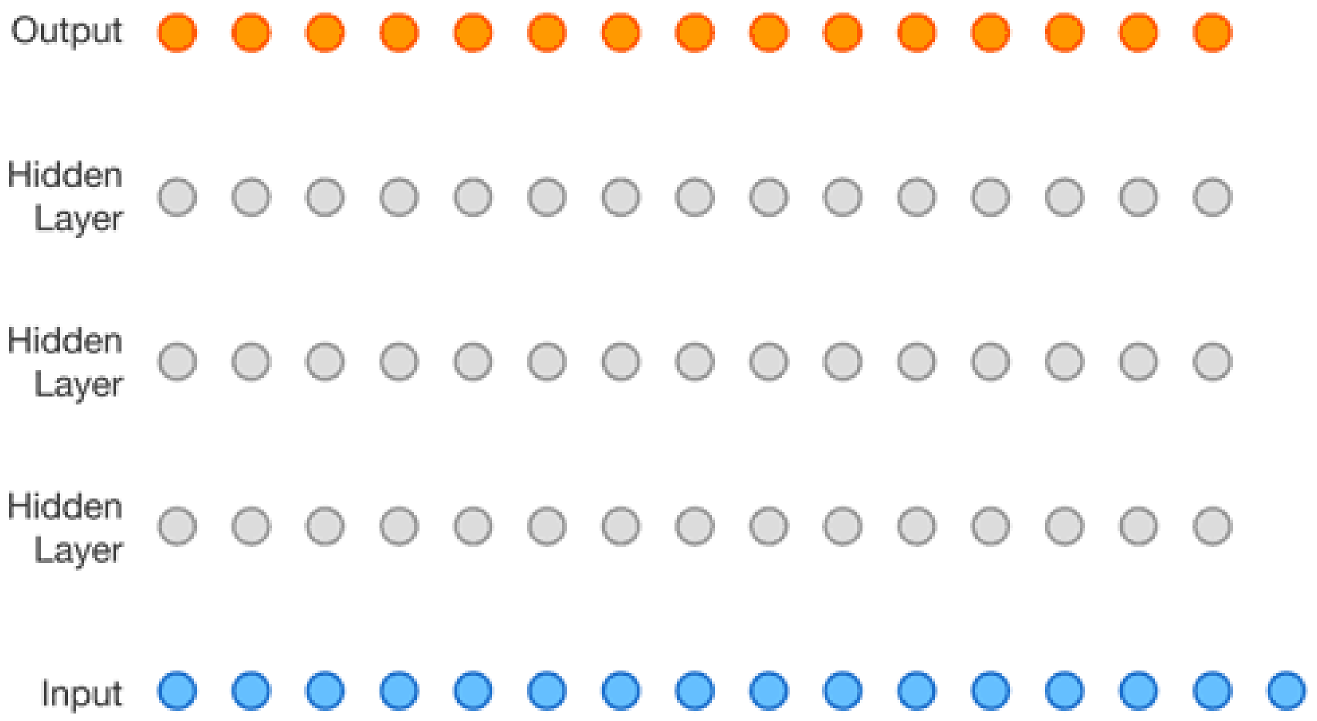
Deep Generative Models—Examples in End-to-End TTS

- Autoregressive models
  - Char2Wav
- Flow
  - ClariNet, Wave-Tacotron
- GAN
  - FastSpeech 2s, EATS
- Diffusion
  - WaveGrad 2
- VAE+Flow+GAN
  - VITS, NaturalSpeech

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

Autoregressive Model for TTS

- WaveNet: autoregressive model with dilated causal convolution

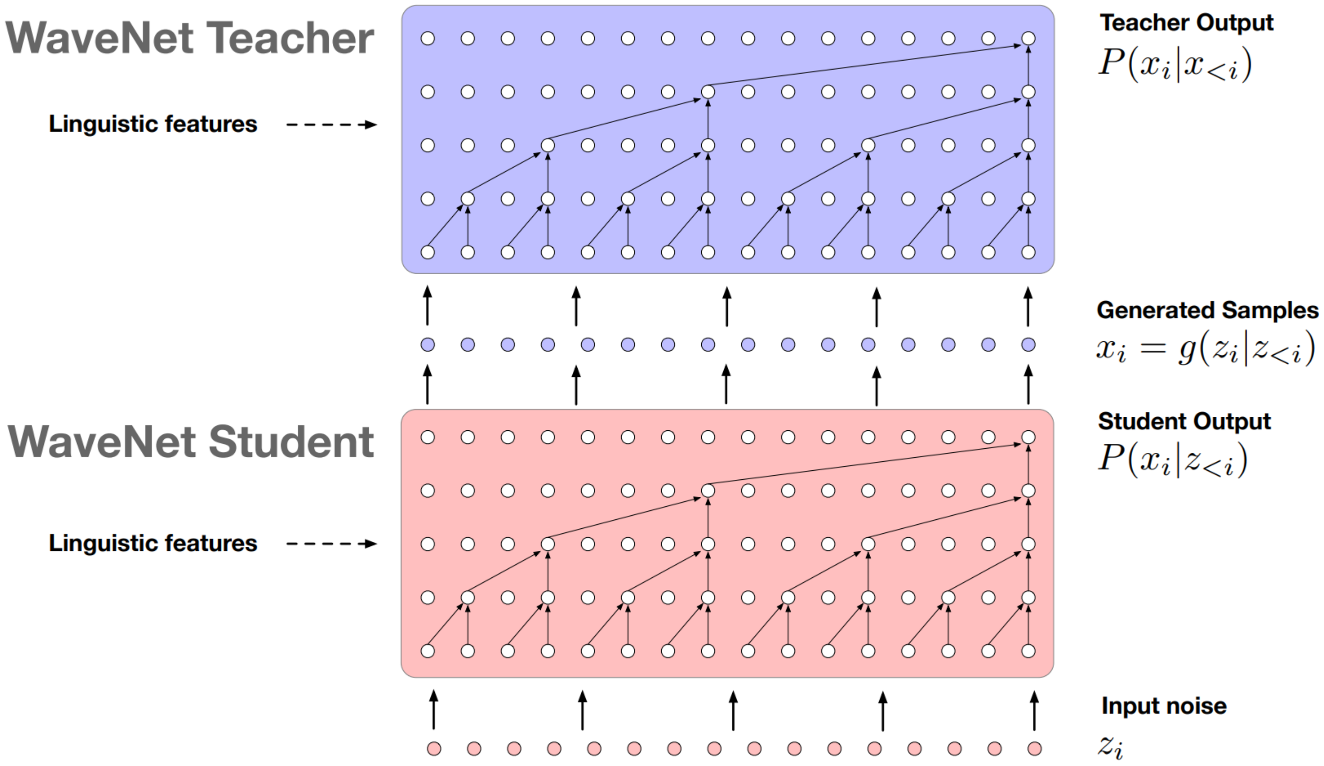


- Other works
  - Acoustic model: Tacotron 1/2, DeepVoice 3, TransformerTTS
  - Vocoder: SampleRNN, WaveRNN

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



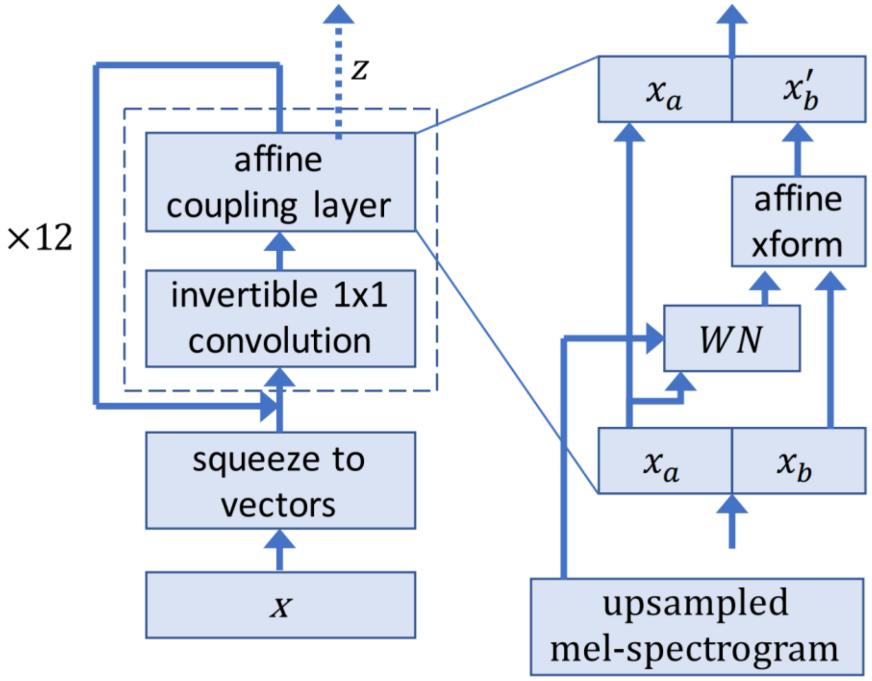
Flow for TTS

- WaveGlow (Bipartite)
  - Flow based transformation

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

  - Affine Coupling Layer

$$x_a, x_b = split(x)$$
$$(\log s, t) = WN(x_a, mel\text{-}spectrogram)$$
$$x_b\prime = s \odot x_b + t$$
$$f_{coupling}^{-1}(x) = concat(x_a, x_b\prime)$$



- Other works
  - FloWaveNet, WaveFlow

Flow for TTS

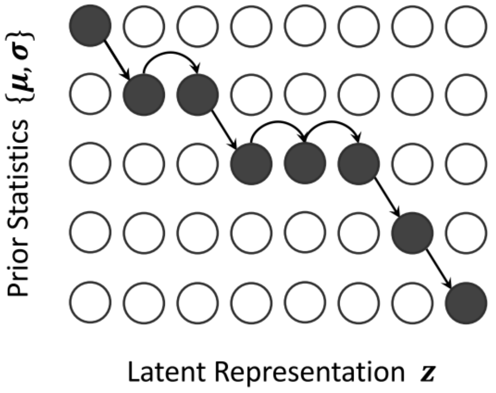
- Glow-TTS (Bipartite) for acoustic model
  - Log likelihood

$$\log P_X(x|c) = \log P_Z(z|c) + \log \left| \det \frac{\partial f_{dec}^{-1}(x)}{\partial x} \right|$$

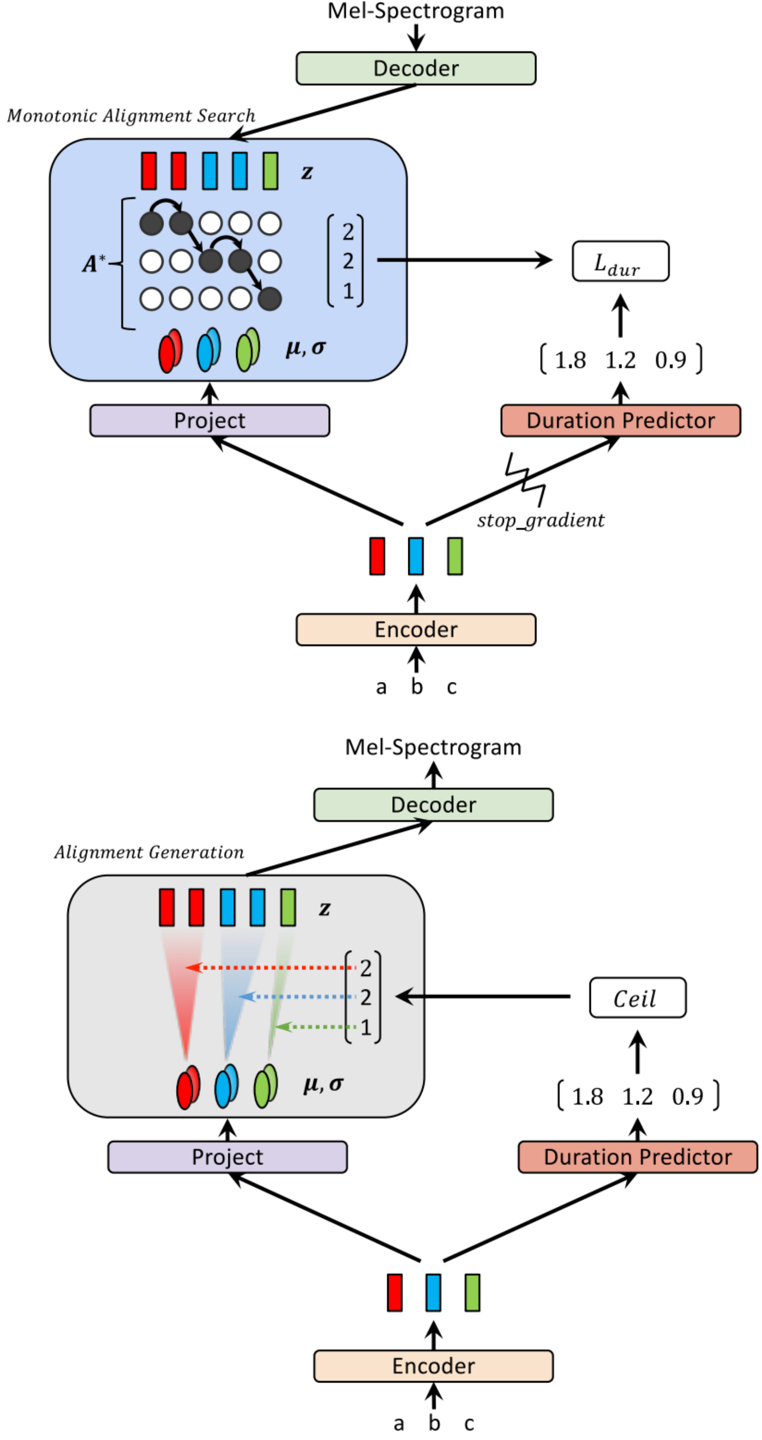
  - Prior is learnt from phoneme text

$$\log P_Z(z|c;\theta,A) = \sum_{j=1}^{T_{mel}} \log \mathcal{N}(z_j; \mu_{A(j)}, \sigma_{A(j)})$$

  - Alignment A is obtained by monotonic alignment search



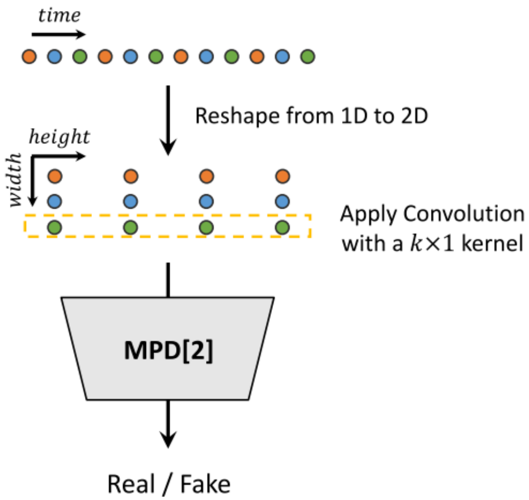
- Other works
  - FlowTTS, Flowtron

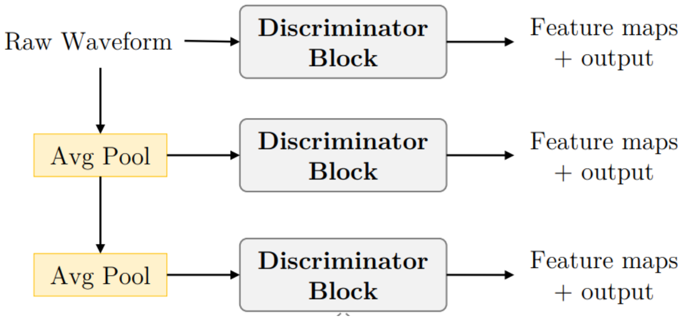


GAN for TTS

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

GAN	Generator	Discriminator	Loss
WaveGAN [68]	DCGAN [287]	/	WGAN-GP [97]
GAN-TTS [23]	/	Random Window D	Hinge-Loss GAN [198]
MelGAN [178]	/	Multi-Scale D	LS-GAN [231] Feature Matching Loss [182]
Par.WaveGAN [402]	WaveNet [254]	/	LS-GAN, Multi-STFT Loss
HiFi-GAN [174]	Multi-Receptive Field Fusion	Multi-Period D, Multi-Scale D	LS-GAN, STFT Loss, Feature Matching Loss
VocGAN [408]	Multi-Scale G	Hierarchical D	LS-GAN, Multi-STFT Loss, Feature Matching Loss
GED [96]	/	Random Window D	Hinge-Loss GAN, Repulsive loss

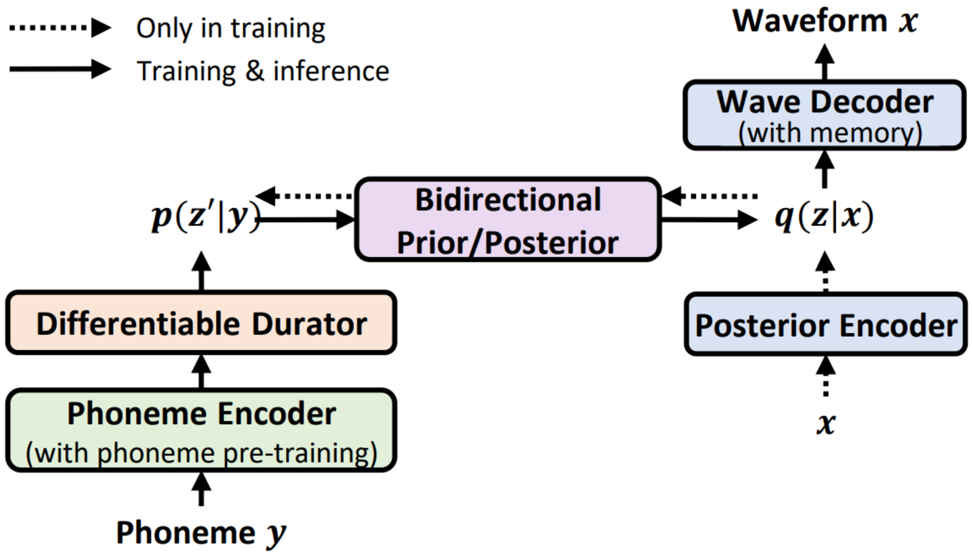




- Other works
    - Para. WaveGAN, BigVGAN
    - FastSpeech 2s, EATS

VAE + Flow + GAN for TTS

- NaturalSpeech for fully end-to-end TTS
  - Reconstruction: z~q(z|x), x~p(x|z)
  - Prior prediction: z~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



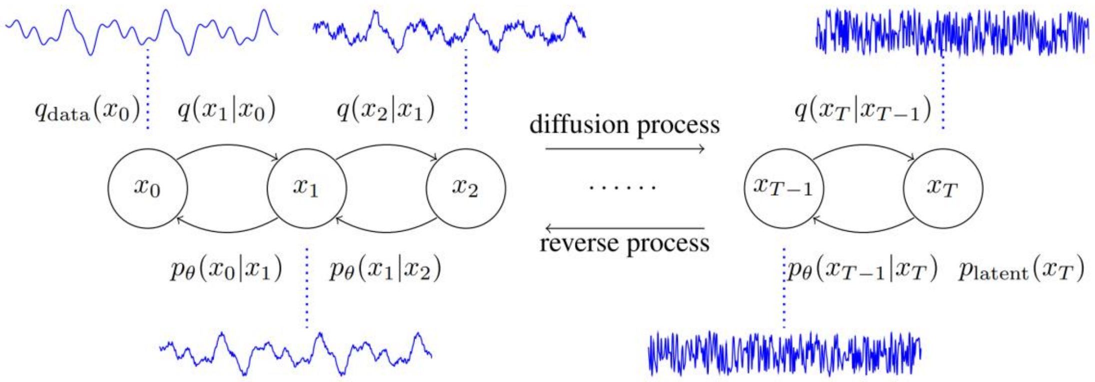
Human Recordings	NaturalSpeech	Wilcoxon p-value
4.58 ± 0.13	4.56 ± 0.13	0.7145

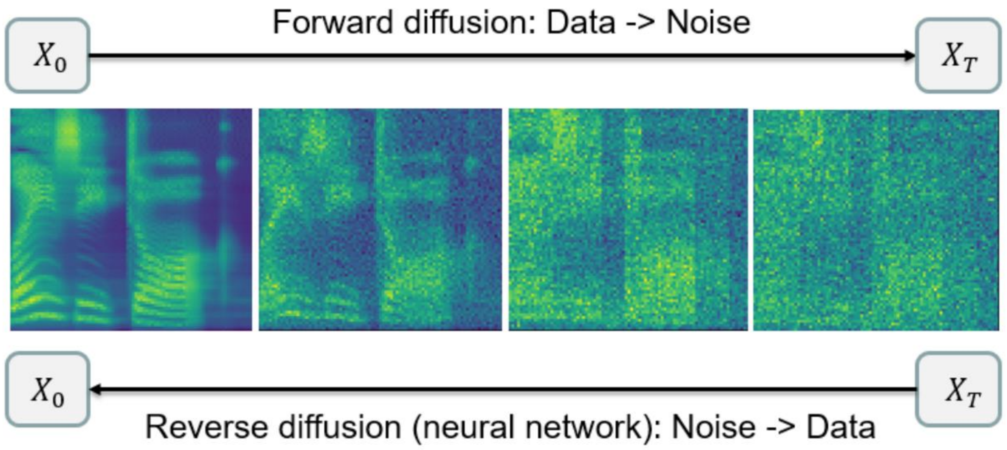
Human Recordings	NaturalSpeech	Wilcoxon p-value
0	-0.01	0.6902

System	MOS	CMOS
FastSpeech 2 [18] + HiFiGAN [17]	4.32 ± 0.15	-0.33
Glow-TTS [13] + HiFiGAN [17]	4.34 ± 0.13	-0.26
Grad-TTS [14] + HiFiGAN [17]	4.37 ± 0.13	-0.24
VITS [15]	4.43 ± 0.13	-0.20
NaturalSpeech	4.56 ± 0.13	0

Diffusion for TTS

- Vocoder: DiffWave, WaveGrad
- Acoustic model: Diff-TTS, Grad-TTS



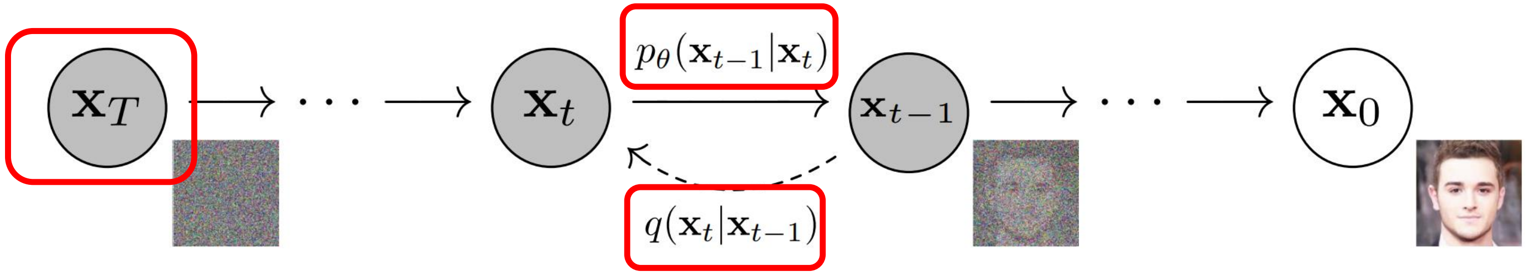


Diffusion—Speedup

- Sampling steps, latency

System	RTF
FastSpeech 2 [18] + HiFiGAN [17]	0.011
Glow-TTS [13] + HiFiGAN [17]	0.021
Grad-TTS [14] (1000) + HiFiGAN [17]	4.120
Grad-TTS [14] (10) + HiFiGAN [17]	0.082
VITS [15]	0.014
NaturalSpeech	0.013

Diffusion—Speedup



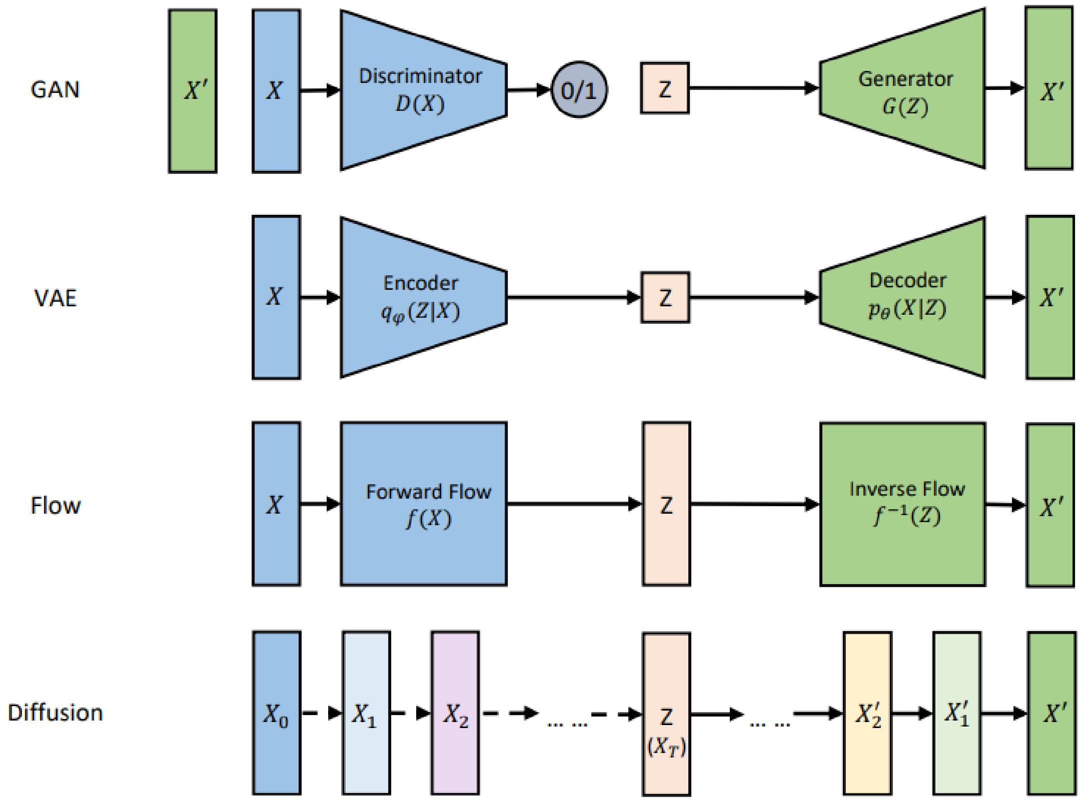
- **Prior distribution**: standard Gaussian → non-standard, e.g., PriorGrad, SpecGrad, Grad-TTS, DDGM
- **Forward process**: fixed → learnable, e.g., Variational diffusion models
- **Diffusion + X**
  - Diffusion + GAN: e.g., DiffusionGAN
  - Diffusion + VAE: e.g., Latent Diffusion
  - Diffusion + KD: e.g., Progressive Distillation
- **Diffusion assumption**: Markovian → non-Markovian: e.g., DDIM
- **Reverse process** (noise levels, schedule, or variance): fixed → learnable, e.g., BDDM, Improved DDPM
- **SDE/ODE solver**: e.g., Euler-Maruyama, Runge-Kutta, adaptive-size SDE, PNDM, DPM-Solver, DPM-Solver++

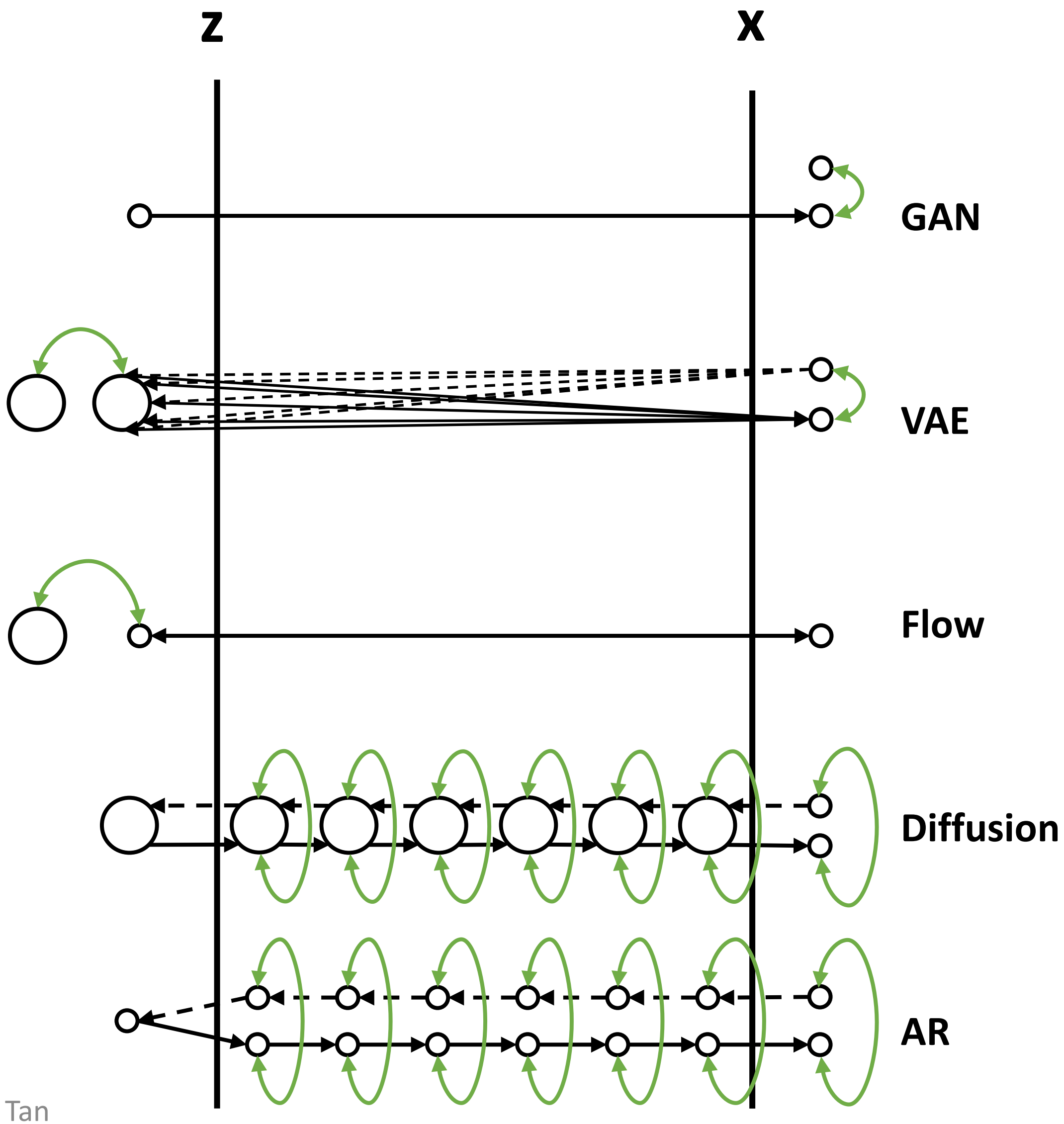
Outline

- 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

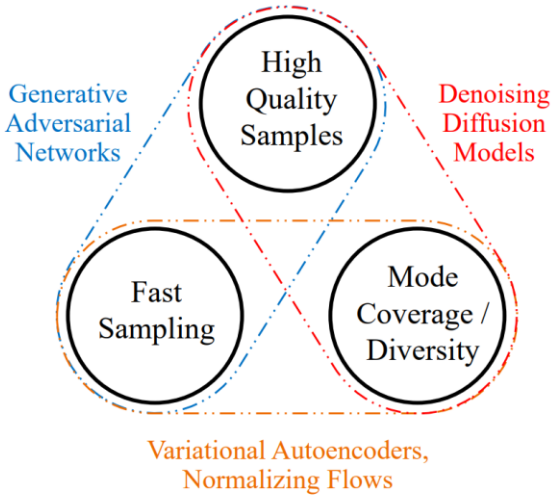




Deep Generative Models—Comparisons

- Pros and cons

Generative Models	AR	Flow	VAE	Diffusion	SMLD	SDE	ODE	GAN
High-Quality	Y	N	N	Y	Y	Y	Y	Y
Fast Sampling	N	Y*	Y	N	N	N	N	Y
Mode Diversity	Y	Y	Y	Y	Y	Y	Y	N
Likelihood Estimation	Y	Y	Y*	Y*	N	N	Y	N
Latent Manipulation	N	Y	Y	Y*	Y*	Y*	Y*	Y*
Error Propagation	Y	N*	N	Y	Y	Y	Y	N
Stable Training	Y	Y	N*	Y	Y	Y	Y	N



[Xiao, 2021]

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

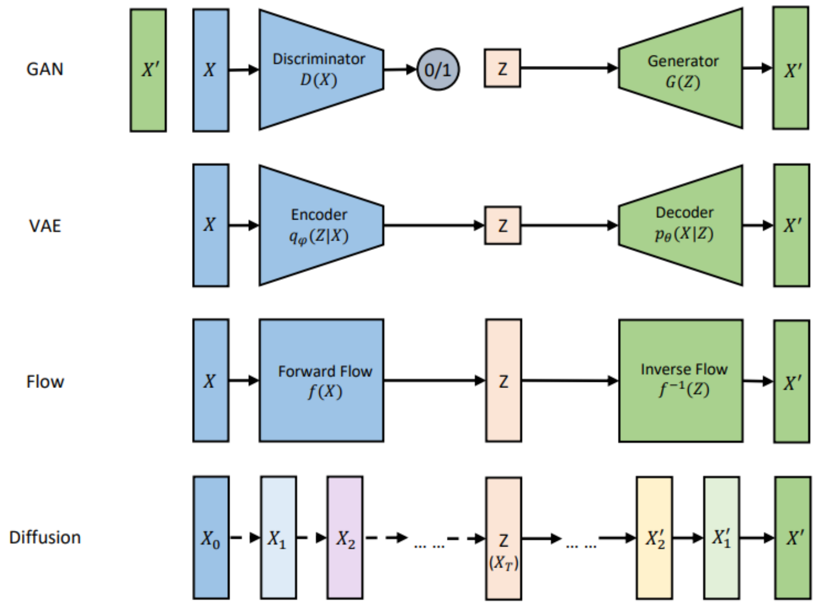
Summary

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

duration, pitch, sound volume, prosody, speaker, style, emotion, etc

Text → 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?

Generative Models	AR	Flow	VAE	Diffusion	SMLD	SDE	ODE	GAN
High-Quality	Y	N	N	Y	Y	Y	Y	Y
Fast Sampling	N	Y*	Y	N	N	N	N	Y
Mode Diversity	Y	Y	Y	Y	Y	Y	Y	N
Likelihood Estimation	Y	Y	Y*	Y*	N	N	Y	N
Latent Manipulation	N	Y	Y	Y*	Y*	Y*	Y*	Y*
Error Propagation	Y	N*	N	Y	Y	Y	Y	N
Stable Training	Y	Y	N*	Y	Y	Y	Y	N

  - 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?

Generative Models	AR	Flow	VAE	Diffusion	SMLD	SDE	ODE	GAN
High-Quality	Y	N	N	Y	Y	Y	Y	Y
Fast Sampling	N	Y*	Y	N	N	N	N	Y
Mode Diversity	Y	Y	Y	Y	Y	Y	Y	N
Likelihood Estimation	Y	Y	Y*	Y*	N	N	Y	N
Latent Manipulation	N	Y	Y	Y*	Y*	Y*	Y*	Y*
Error Propagation	Y	N*	N	Y	Y	Y	Y	N
Stable Training	Y	Y	N*	Y	Y	Y	Y	N

  - 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?

The Landscape of Deep Generative Learning

Reference

See the references in:

A Survey on Neural Speech Synthesis

https://arxiv.org/pdf/2106.15561.pdf

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/

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

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/