NaturalSpeech: End-to-End Text-to-Speech Synthesis With Human-Level Quality

Xu Tan, Senior Member, IEEE, Jiawei Chen, Haohe Liu, Jian Cong, Chen Zhang, Yanqing Liu, Xi Wang, Yichong Leng, Yuanhao Yi, Lei He, Sheng Zhao, Senior Member, IEEE, Tao Qin, Senior Member, IEEE, Frank Soong, Fellow, IEEE, and Tie-Yan Liu, Fellow, IEEE

Abstract—Text-to-speech (TTS) has made rapid progress in both academia and industry in recent years. Some questions naturally arise that whether a TTS system can achieve human-level quality, how to define/judge that quality, and how to achieve it. In this paper, we answer these questions by first defining the human-level quality based on the statistical significance of subjective measure and introducing appropriate guidelines to judge it, and then developing a TTS system called NaturalSpeech that achieves human-level quality on benchmark datasets. Specifically, we leverage a variational autoencoder (VAE) for end-to-end text-to-waveform generation, with several key modules to enhance the capacity of the prior from text and reduce the complexity of the posterior from speech, including phoneme pre-training, differentiable duration modeling, bidirectional prior/posterior modeling, and a memory mechanism in VAE. Experimental evaluations on the popular LJSpeech dataset show that our proposed NaturalSpeech achieves a 0.01 CMOS (comparative mean opinion score) to human recordings at the sentence level, with Wilcoxon signed rank test at p-level $p > 0.05$, which demonstrates no statistically significant difference from human recordings for the first time.

Index Terms—Text-to-speech, speech synthesis, human-level quality, variational auto-encoder, end-to-end.

I. INTRODUCTION

Text to speech (TTS) aims at synthesizing intelligible and natural speech from text [1], and has made rapid progress in recent years due to the development of deep learning. Neural network based TTS has evolved from CNN/RNN-based models [2], [3], [4], [5], [6], [7], [8] to Transformer-based models [9], [10], [11], from autoregressive models [2], [3], [9] to other generative models (VAE, GAN, Flow, diffusion) [12], [13], [14], [15], from cascaded acoustic models/vocoders [2], [3], [4], [10], [16], [17] to fully end-to-end models [15], [18], [19].

Building TTS systems with human-level quality has always been the dream of practitioners in speech synthesis. While current TTS systems achieve high voice quality, they still have a quality gap compared with human recordings. To pursue this goal, several questions need to be answered: 1) how to define human-level quality in text-to-speech synthesis? 2) how to judge whether a TTS system has achieved human-level quality or not? 3) how to build a TTS system to achieve human-level quality? In this paper, we conduct a comprehensive study of these problems in TTS. We first give a formal definition of human-level quality in TTS based on a statistical and measurable way (see Definition 1). Then we introduce some guidelines to judge whether a TTS system has achieved human-level quality with a hypothesis test. Using this judging method, we found several previous TTS systems have not achieved it (see Table 1).

In this paper, we further develop a fully end-to-end text-to-waveform generation system called NaturalSpeech to bridge the quality gap to recordings and achieve human-level quality. Specifically, inspired by image/video/waveform generation [15], [20], [21], we leverage variational autoencoder (VAE) [22] to compress the high-dimensional speech ($x$) into continuous frame-level representations (denoted as posterior $q(z|x)$), which are used to reconstruct the waveform (denoted as $p(x|z)$). The corresponding prior (denoted as $p(z|y)$) is obtained from the text sequence $y$. Considering the posterior from speech is more complicated than the prior from text, we design several modules (see Fig. 1) to match the posterior and prior as close to each other as possible, to enable text-to-speech synthesis through $p(z|y) \rightarrow p(x|z)$:

- We leverage large-scale pre-training on the phoneme encoder to extract better representations from phoneme sequence [23] (Section III-B).
- We leverage a fully differentiable durator that consists of a duration predictor and an upsampling layer to improve the duration modeling [24] (Section III-C).

1Since duration is very important in TTS, especially in non-autoregressive TTS, we name the module related to duration modeling as durator, including but not limited to the functionalities of duration prediction and hidden expansion. It is common to come up with a new term to revolutionize the concept in the speech community, such as vocoder, cepstrum.
Table I

| System                  | MOS       | Wilcoxon p-value | CMOS     | Wilcoxon p-value |
|-------------------------|-----------|------------------|----------|------------------|
| Human Recordings        | 4.52 ± 0.11 | -                | 0        | -                |
| FastSpeech 2 [18] + HiFiGAN [17] | 4.32 ± 0.10 | 1.0e-05           | -0.39    | 5.1e-20          |
| Glow-TTS [13] + HiFiGAN [17] | 4.33 ± 0.10 | 1.3e-06           | -0.23    | 8.7e-17          |
| Grad-TTS [14] + HiFiGAN [17] | 4.37 ± 0.10 | 0.0127            | -0.23    | 1.2e-11          |
| VITS [15]               | 4.49 ± 0.10 | 0.2429            | -0.19    | 2.9e-04          |

Note that the Wilcoxon p-value in MOS is conducted using Wilcoxon rank sum test [35], instead of the Wilcoxon signed rank test in CMOS, due to no paired comparison in MOS evaluation. For Grad-TTS, we use 1000 steps for inference.

Fig. 1. System overview of NaturalSpeech.

- We design a bidirectional prior/posterior module based on Flow models [25], [26], [27] to further enhance the prior \( p(z|y) \) and reduce the complexity of posterior \( q(z|x) \) (Section III-D).
- We propose a memory-based VAE to reduce the complexity of the posterior needed to reconstruct waveform (Section III-E).

Compared to previous TTS systems, NaturalSpeech has several advantages: 1) Reduce training-inference mismatch. In previous cascaded acoustic model/vocoder pipeline [13], [14], [18] and explicit duration prediction [13], [15], [18], both mel-spectrogram and duration suffer from training-inference mismatch since ground-truth values are used in training the vocoder and mel-spectrogram decoder while predicted values are used in inference. Our fully end-to-end text-to-waveform generation and differentiable durator can avoid the training-inference mismatch. 2) Alleviate one-to-many mapping problem. One text sequence can correspond to multiple speech utterances with different variation information (e.g., F0, duration, speed, pause, prosody, etc). Previous works only using variance adaptor [11], [18] to predict F0/duration cannot well handle the one-to-many mapping problem. Our memory-based VAE and bidirectional prior/posterior can reduce the complexity of the posterior and enhance the prior, which helps relieve the one-to-many mapping problem. 3) Improve representation capacity. Previous models are not powerful enough to extract good representations from phoneme sequence [13], [14], [15] and learn complicated data distribution in speech [18]. Our large-scale phoneme pre-training, powerful generative models (e.g., Flow and VAE), and memory mechanism in VAE can learn better text representations and speech data distributions.

We conduct experimental evaluations on the widely adopted single-speaker LJSpeech dataset [28] and multi-speaker VCTK dataset [29] to measure the voice quality of our NaturalSpeech system. Based on the proposed judgment guidelines, NaturalSpeech achieves similar quality with human recordings in terms of MOS (mean opinion score) and CMOS (comparable MOS). Specifically, the speech generated by NaturalSpeech achieves \(-0.01 \) CMOS compared to recordings on the LJSpeech dataset, with p-level \( p \gg 0.05 \) under the Wilcoxon signed rank test, which demonstrates that NaturalSpeech can generate speech with no statistically significant difference from recordings.

II. DEFINITION AND JUDGMENT OF HUMAN-LEVEL QUALITY IN TTS

In this section, we introduce the formal definition of human-level quality in text-to-speech synthesis, and describe how to judge whether a TTS system achieves human-level quality or not.

A. Definition of Human-Level Quality

We define human-level quality in a statistical and measurable way.

Definition 1: If there is no statistically significant difference between the quality scores of the speech generated by a TTS system and the quality scores of the corresponding human recordings on a test set, then this TTS system achieves human-level quality on this test set.

Note that by claiming a TTS system achieves human-level quality on a test set, we do not mean that a TTS system can surpass or replace humans, but the quality of this TTS system is statistically indistinguishable from human recordings on this test set.

B. Judgment of Human-Level Quality

1) Judgment Guideline: While there are some objective metrics to measure the quality gap between the generated speech and human recordings, such as PESQ [30], STOI [31], SI-SDR [32], they are not reliable to measure the perception quality in TTS [33], [34]. Therefore, we use subjective evaluation to measure the voice quality. Previous works usually use mean opinion score (MOS) with 5 points (from 1 to 5) to compare the generated speech with recordings. However, MOS is not sensitive enough to the difference in voice quality since the judge simply rates the quality of each sentence alone from the...
two systems with no paired comparison. Thus, we choose a
comparative mean opinion score (CMOS) with 7 points (from
−3 to 3) as the evaluation metric, where each judge measures
the voice quality by comparing samples from two systems head
by head. We further conduct Wilcoxon signed rank test [35] to
measure whether the two systems are significantly different or
not in terms of CMOS evaluation.

Therefore, we list the judgment guidelines of human-level
quality as follows: 1) Each utterance from the TTS system and
human recordings should be listened to and compared side-by-
side by more than 20 judges, who should be native language
speakers, and at least 50 test utterances from each system should
be used in the judgment following the common practice in TTS
evaluations [4], [10], [13], [14]. 2) The speech generated by
the TTS system has no statistically significant difference from
human recordings, if and only if the average CMOS is close to 0
and the p-level of Wilcoxon signed rank test satisfies p > 0.05.

2) Judgment of Previous TTS Systems: Based on these guide-
lines, we test whether current TTS systems can achieve human-
level quality or not on the LJSpeech dataset. The systems we
study include: 1) FastSpeech 2 [18] + HiFiGAN [17], 2) Glow-
TTS [13] + HiFiGAN [17], 3) Grad-TTS [14] + HiFiGAN [17],
4) VITS [15]. We reproduce the results of all these systems
on our own, which can match or even beat the quality in their
original papers (note that the HiFiGAN vocoder is fine-tuned
on the predicted mel-spectrograms for better synthesis quality).
We use 50 test utterances, each with 20 judges for MOS and
CMOS evaluation. As shown in Table I, although the current
TTS systems can achieve close MOS to recordings, they have
a large CMOS gap to recordings, with Wilcoxon signed rank
test at p-level p ≪ 0.05, which shows statistically significant
difference from human recordings.

To understand where and how the quality gap to recordings
comes from, we conduct a systematic study on the current TTS
systems, which can help us to find the problems, and is equally
important (if not more) than solving the problems. Specifically,
we choose a state-of-the-art TTS system using FastSpeech
2 [18] as the acoustic model and HiFiGAN [17] as the vocoder,
which consists of four components: phoneme encoder, variance
adaptor, mel-spectrogram decoder, and vocoder. We design a
series of comparison experiments to measure the quality gap (in
terms of CMOS) of each component to its corresponding upper
bound. We conduct analyses from this order (from the closest to
waveform to the farthest): vocoder, mel-spectrogram decoder,
variance adaptor, and phoneme encoder.

- **Vocoder:** We study the quality drop on the vocoder by
  comparing the two settings: 1) waveform generated by
  vocoder with ground-truth mel-spectrograms as input; 2)
ground-truth waveform (human recordings). The CMOS is
shown in Table II. It can be seen that when taking
ground-truth mel-spectrograms as input, the waveform
generated by the vocoder has some but not a huge gap
to human recordings. However, we need to pay attention to
the training-inference mismatch in the vocoder: in training,
the vocoder takes ground-truth mel-spectrograms as input,
while in inference, it takes predicted mel-spectrograms as input.

  - **Mel-spectrogram Decoder:** We study the quality drop on
    the mel-spectrogram decoder by comparing the two set-
    tings: 1) mel-spectrograms generated by mel-spectrogram
decoder with ground-truth F0 and duration as input; 2)
ground-truth mel-spectrograms (extracted from human
recordings). We use the vocoder to convert the
mel-spectrograms in the two settings into waveform for evalua-
tion. As shown in Table II, the predicted mel-spectrograms
have a 0.15 CMOS drop compared to the ground-truth
mel-spectrograms.

  - **Variance Adaptor:** We study the quality drop on the
    variance adaptor by comparing the predicted F0/duration
    with the ground-truth F0/duration. We need the mel-
spectrogram decoder and vocoder to generate the wave-
form for evaluation in the two settings. As shown in
Table II, the predicted F0/duration has a 0.14 CMOS drop
compared to the ground-truth F0/duration.

  - **Phoneme Encoder:** Since it is not straightforward to con-
    struct the upper bound of the phoneme encoder, we analyze
the approximate quality drop through backward verifica-
tion, by improving the phoneme encoder for better voice
quality. We conduct large-scale phoneme pre-training on
the phoneme encoder and fine-tune it with the FastSpeech 2
training pipeline, and achieve a 0.12 CMOS gain, as shown
in Table II, which demonstrates the phoneme encoder has
improvement space.

According to the above experimental studies, we analyze
several reasons causing the quality drop in each component: 1)
Training-inference mismatch. Ground-truth mel-spectrogram,
F0, and duration are used in training, while predicted values
are used in inference, which causes a mismatch in the input
of the vocoder and mel-spectrogram decoder. Fully end-to-end
text-to-waveform optimization is helpful to eliminate this mis-
match. 2) One-to-many mapping problem. Text-to-speech map-
ing is one-to-many, where a text sequence can correspond to

2Ideally, we should also use ground-truth phoneme hidden sequence as input. However, a ground-truth hidden sequence cannot be obtained. Thus, this
comparison setting is just an approximation.
multiple speech utterances with different variation information (e.g., F0, duration, speed, pause, prosody, etc). Current systems usually use a variance adaptor to predict variance information (e.g., F0, duration) to alleviate this problem, which is not enough to handle this problem [15], [19], [24]. We should rethink previous methods on variance information and come up with some thorough and elegant solutions. 3) Lack of representation capacity. Current models are not powerful enough to extract good representations from phoneme sequences and learn complicated data distribution in speech. More advanced methods such as large-scale pre-training and powerful generative models are critical to enhancing the learning capacity.

III. DESCRIPTION OF NATURALSPEECH SYSTEM

To bridge the quality gap to human recordings, we develop NaturalSpeech, a fully end-to-end text-to-waveform generation model. We first describe the design principle of our system (Section III-A), and then introduce each module of this system (Sections III-B–III-E) and training/inference pipeline (Section III-F), and finally explain why our system can bridge the quality gap to human recordings (Section III-G).

A. Design Principle

Inspired by image/video generation [21], [36], [37], [38], [39] that uses VQ-VAE [20], [40], [41] to compress high-dimensional image into low-dimensional representations to ease the generation [42], we leverage VAE [22] to compress high-dimensional speech \( x \) into frame-level representations \( z \) (i.e., \( z \) is sampled from posterior distribution \( q(z|x) \)), which are used to reconstruct the waveform (denoted as \( p(z|x) \)). In the general formulation of VAE, the prior \( p(z) \) is chosen to be standard isotropic multivariate Gaussian. To enable conditional waveform generation from input text in TTS, we predict \( z \) from phoneme sequence \( y \), i.e., \( z \) is sampled from predicted prior distribution \( p(z|y) \). We jointly optimize the VAE and the prior prediction with gradients propagating to both \( q(z|x) \) and \( p(z|y) \). Derived from the evidence lower bound [22], the loss function consists of a waveform reconstruction loss \(- \log p(x|z)\) and a Kullback-Leibler divergence loss between the posterior \( q(z|x) \) and the prior \( p(z|y) \), i.e., \( KL[q(z|x)||p(z|y)]\).

Considering the posterior from speech is more complicated than the prior from text, to match them as close as possible to enable text-to-waveform generation, we design several modules to simplify the posterior and to enhance the prior, as shown in Fig. 1. First, to learn a good representation of phoneme sequence for better prior prediction, we pre-train a phoneme encoder on a large-scale text corpus using masked language modeling on phoneme sequence (Section III-B). Second, since the posterior is at the frame level while the phoneme prior is at the phoneme level, we need to expand the phoneme prior according to its duration to bridge the length difference. We leverage a differentiable durator to improve duration modeling (Section III-C). Third, we design a bidirectional prior/posterior module to enhance the prior or simplify the posterior (Section III-D). Fourth, we propose a memory-based VAE that leverages a memory bank through Q-K-V attention [43] to reduce the complexity of posterior needed to reconstruct the waveform (Section III-E).

B. Phoneme Encoder

The phoneme encoder \( \theta_{\text{phi}} \) takes a phoneme sequence \( y \) as input and outputs a phoneme hidden sequence. To enhance the representation capability of the phoneme encoder, we conduct large-scale phoneme pre-training. Previous works [44] conduct pre-training in character/word level and apply the pre-trained model to phoneme encoder, which will cause inconsistency, and the works [45] directly using phoneme pre-training will suffer from limited capacity due to too small size of phoneme vocabulary. To avoid these issues, we leverage mixed-phoneme pre-training [23], which uses both phoneme and sup-phoneme (adjacent phonemes merged together) as the input of the model, as shown in Fig. 2(c). The benefit of using both phoneme and sup-phoneme has been well demonstrated in [23]. When using masked language modeling [46], we randomly mask some sup-phoneme tokens and their corresponding phoneme tokens and predict the masked phoneme and sup-phoneme at the same time. After mixed phoneme pre-training, we use the pre-trained model to initialize the phoneme encoder of our TTS system.

C. Differentiable Durator

The differentiable durator \( \theta_{\text{dur}} \) takes a phoneme hidden sequence as input, and outputs a sequence of prior distribution at the frame level, as shown in Fig. 2(a). We denote the prior distribution as \( p(z'|y; \theta_{\text{phi}}, \theta_{\text{dur}}) = p(z'|y; \theta_{\text{pri}}) \), where \( \theta_{\text{pri}} = [\theta_{\text{phi}}, \theta_{\text{dur}}] \). The differentiable durator \( \theta_{\text{dur}} \) consists of several modules: 1) a duration predictor \( \theta_{\text{dp}} \) that builds upon the phoneme encoder to predict the duration for each phoneme, 2) a learnable upsampling layer \( \theta_{\text{lu}} \) that leverages the predicted duration to learn a projection matrix to extend the phoneme hidden sequence from phoneme level to frame level in a differentiable way [24], and 3) two additional linear layers on the expanded hidden sequence to calculate the mean and variance of the prior distribution \( p(z'|y; \theta_{\text{lu}}) \). Compared to simply repeating each phoneme hidden sequence with the predicted duration in a hard way, the learnable upsampling layer enables more flexible duration adjustment for each phoneme. Also, the learnable upsampling layer makes the phoneme to frame expansion differentiable, and thus can be jointly optimized with other modules in the TTS system. We introduce the detailed formulation of differentiable durator as follows.

1) Duration Predictor: The input to the duration predictor \( \theta_{\text{dp}} \) is phoneme hidden sequence \( H_{n \times h} \) where \( n, h \) is the phoneme sequence length and hidden dimension size, and the output is the estimated phoneme duration \( d_{n \times 1} \). The duration predictor \( \theta_{\text{dp}} \) consists of 3 layers of one-dimensional convolution, with ReLU activation, layer normalization, and dropout between each layer.

2) Learnable Upsampling Layer: The learnable upsampling layer \( \theta_{\text{lu}} \) takes phoneme duration \( d_{n \times 1} \) as input and upsamples phoneme hidden sequence \( H_{m \times h} \) to frame-level sequence \( O_{m \times h} \) where \( m \) is the number of frames [24]. First, we calculate
the duration start and end matrices $S_{m \times n}$ and $E_{m \times n}$ by

$$S_{i,j} = i - \sum_{k=1}^{j-1} d_k, \quad E_{i,j} = \sum_{k=1}^{j} d_k - i,$$

(1)

where $S_{i,j}$ indexes the $(i,j)$th element in the matrix. We calculate the primary attention matrix $W_{m \times n \times q}$ and auxiliary context matrix $C_{m \times n \times p}$ following [24]:

$$W = \text{Softmax}(\text{MLP}([S, E, \text{Expand} \left(\text{Conv1D} \left(\text{Proj} \left(\text{H} \right)\right)\right)])),$$

(2)

$$C = \text{MLP}([S, E, \text{Expand} \left(\text{Conv1D} \left(\text{Proj} \left(\text{H} \right)\right)\right)]),$$

(3)

where $\text{Proj} \cdot \cdot \cdot$ represents one linear layer with input and output dimensions of $h$. $\text{Conv1D} \cdot \cdot \cdot$ is one-dimensional convolution operation with layer normalization and Swish activation [47]. The input and output dimensions of $\text{Conv1D} \cdot \cdot \cdot$ are $h$ and $8$. $\text{Expand} \cdot \cdot \cdot$ means adding an extra dimension by repeating the input matrix by $m$ times. $\cdot$ stands for matrix concatenation along the hidden dimension, and gets a hidden dimension of $10 = 1 + 1 + 8$. $\text{MLP} \cdot \cdot \cdot$ is a two-layer full-connected network with Swish activations. The numbers underneath MLP denote the input and output hidden dimensions. We set $p = 2$ and $q = 4$. The Softmax operation is performed on the phoneme sequence time dimension. We calculate the frame-level hidden sequence output $O_{m \times h}$ with the following equation:

$$O = \text{Proj} \left(\text{W} \cdot \text{H} \right) + \text{Proj} \left(\text{Einsum} \left(\text{W}, \text{C}\right)\right),$$

(4)

where $\text{Einsum} \cdot \cdot \cdot$ represents the einsum operation (‘qmnp, mnpq -> qmpn’). $\text{W}, \text{C}$). We first permute $\text{W}$ from $m \times n \times q$ to $q \times m \times n$ for computation, and after we get $\text{WH}$ with shape $q \times m \times h$ and $\text{Einsum} \left(\text{W}, \text{C}\right)$ with shape $q \times m \times p$, we reshape them to $m \times qh$ and $m \times qp$ respectively for final projection to dimension $m \times h$.

3) Linear Layers for Mean and Variance: We further map $O_{m \times h}$ with a mean and variance linear layer to get the frame-level prior distribution parameter $\mu(y; \theta_{\text{pri}})$ and $\sigma(y; \theta_{\text{pri}})$, and get the prior distribution $p(z' | y; \theta_{\text{pri}}) = \mathcal{N}(z'; \mu(y; \theta_{\text{pri}}), \sigma(y; \theta_{\text{pri}}))$.

We optimize the duration prediction, learnable upsampling layer, and mean/variance linear layers together with the TTS model in a fully differentiable way, which can reduce the training-inference mismatch in previous duration prediction (ground-truth duration is used in training while predicted duration is used in inference) [13, 15, 18] and better use duration in a soft and flexible way instead of a hard expansion, hence the side-effect of inaccurate duration prediction is mitigated.

D. Bidirectional Prior/Posterior

As shown in Fig. 2(b), we design a bidirectional prior/posterior module to enhance the capacity of the prior $p(z' | y; \theta_{\text{pri}})$ or to reduce the complexity of the posterior $q(z | x; \phi)$ where $\phi$ is the posterior encoder, since there is an information gap between the posterior obtained from speech sequence and the prior obtained from phoneme sequence. We choose a Flow model [25, 26, 27, 48] as the bidirectional prior/posterior module (denoted as $\theta_{\text{bpp}}$) since it is easy to optimize and has a nice property of invertibility.

1) Reduce Posterior With Backward Mapping $f^{-1}$: The bidirectional prior/posterior module can reduce the complexity of posterior from $q(z | x; \phi) \rightarrow q(z' | x; \phi, \theta_{\text{bpp}}$) through the backward mapping $f^{-1}(z; \theta_{\text{bpp}})$, i.e., for $z \sim q(z | x; \phi), z' = f^{-1}(z; \theta_{\text{bpp}}) \sim q(z' | x; \phi, \theta_{\text{bpp}})$. The objective is to match the simplified posterior $q(z' | x; \phi, \theta_{\text{bpp}})$ to the prior $p(z' | y; \theta_{\text{pri}})$ by...
using the KL divergence loss as follows:

\[
\mathcal{L}_{\text{bwd}}(\phi, \theta_{\text{bpp}}, \theta_{\text{pri}}) = KL[q(z'|x; \phi, \theta_{\text{bpp}}) || p(z'|y; \theta_{\text{pri}})]
\]

\[
= \int q(z'|x; \phi) \cdot \log \frac{q(z'|x; \phi, \theta_{\text{bpp}})}{p(z'|y; \theta_{\text{pri}})} dz'
\]

\[
= \int q(z|x; \phi) \cdot \det \frac{\partial f^{-1}(z; \theta_{\text{bpp}})}{\partial z} \cdot \log \frac{q(z|x; \phi)}{p(f^{-1}(z; \theta_{\text{bpp}})|y; \theta_{\text{pri}})} \cdot \det \frac{\partial f^{-1}(z; \theta_{\text{bpp}})}{\partial z} dz
\]

\[
= \int q(z|x; \phi) \cdot \log \frac{q(z|x; \phi)}{p(f^{-1}(z; \theta_{\text{bpp}})|y; \theta_{\text{pri}})} \cdot \det \frac{\partial f^{-1}(z; \theta_{\text{bpp}})}{\partial z} dz,
\]

where the third equality (the second line) in (5) is obtained via the change of variables: \(dz' = \det \frac{\partial f^{-1}(z; \theta_{\text{bpp}})}{\partial z} dz\), and \(q(z'|x; \phi, \theta_{\text{bpp}}) = q(z|x; \phi) \cdot \det \frac{\partial f^{-1}(z; \theta_{\text{bpp}})}{\partial z}\) according to the inverse function theorem.

2) **Enhance Prior With Forward Mapping** \(f\): The bidirectional prior/posterior module can enhance the capacity of prior from \(p(z'|y; \theta_{\text{pri}})\) to \(p(z'|y; \theta_{\text{pri}}, \theta_{\text{bpp}})\) through the forward mapping \(f(z'; \theta_{\text{bpp}})\), i.e., \(z' \sim p(z'|y; \theta_{\text{pri}})\), \(z = f(z'; \theta_{\text{bpp}}) \sim p(z|y; \theta_{\text{pri}}, \theta_{\text{bpp}})\). The objective is to match the enhanced prior \(p(z'; y; \theta_{\text{pri}}, \theta_{\text{bpp}})\) to the posterior \(q(z|x; \phi)\) using the KL divergence loss as follows:

\[
\mathcal{L}_{\text{fwd}}(\phi, \theta_{\text{bpp}}, \theta_{\text{pri}}) = KL[p(z'|y; \theta_{\text{pri}}, \theta_{\text{bpp}})|| q(z'|x; \phi)]
\]

\[
= \int p(z|y; \theta_{\text{pri}}, \theta_{\text{bpp}}) \cdot \log \frac{p(z|y; \theta_{\text{pri}}, \theta_{\text{bpp}})}{q(z|x; \phi)} \cdot dq(z|x; \phi)
\]

\[
= \int p(z'|y; \theta_{\text{pri}}) \cdot \log \frac{q(z'|x; \phi)}{p(f(z'; \theta_{\text{bpp}})|x; \phi)} \cdot \det \frac{\partial f(z'; \theta_{\text{bpp}})}{\partial z'} dz'
\]

\[
= \mathbb{E}_{z' \sim p(z'|y; \theta_{\text{pri}})} \log p(z'|y; \theta_{\text{pri}}) - \mathbb{E}_{z' \sim p(z'|y; \theta_{\text{pri}})} \log q(f(z'; \theta_{\text{bpp}})|x; \phi)
\]

\[
\det \frac{\partial f(z'; \theta_{\text{bpp}})}{\partial z'}(z'),
\]

where the third equality (the second line) in (6) is obtained via the change of variables: \(dz' = \det \frac{\partial f(z'; \theta_{\text{bpp}})}{\partial z'} dz\), and \(p(z|y; \theta_{\text{pri}}, \theta_{\text{bpp}}) = p(z'|y; \theta_{\text{pri}}) \cdot \det \frac{\partial f^{-1}(z; \theta_{\text{bpp}})}{\partial z} = p(z'|y; \theta_{\text{pri}}) \cdot \det \frac{\partial f^{-1}(z; \theta_{\text{bpp}})}{\partial z'}\) according to the inverse function theorem, similar to that in (5).

By using backward and forward loss functions, both directions of the Flow model are considered in training, which can reduce the training-inference mismatch in the previous Flow models that train in the backward direction but infer in the forward direction.

3) **Alternative Formulation of Forward/Backward Mapping**: Alternatively, we also provide another formulation of the bidirectional prior/posterior by directly using KL loss to match two distributions. For the backward loss, we directly match the posterior \(q(z|x; \phi)\) to the prior \(p(z|y; \theta_{\text{pri}})\)

\[
\mathcal{L}_{\text{bwd}}(\phi, \theta_{\text{bpp}}, \theta_{\text{pri}}) = KL[q(z|x; \phi)|| p(z|y; \theta_{\text{pri}})]
\]

\[
= \mathbb{E}_{z \sim q(z|x; \phi)} \log q(z|x; \phi) - \mathbb{E}_{z \sim q(z|x; \phi)} \log p(z|y; \theta_{\text{pri}})
\]

\[
= \mathbb{E}_{z \sim q(z|x; \phi)} \log q(z|x; \phi) - \mathbb{E}_{z \sim q(z|x; \phi)} \log p(f^{-1}(z; \theta_{\text{bpp}})|y; \theta_{\text{pri}}))
\]

\[
\det \frac{\partial f^{-1}(z; \theta_{\text{bpp}})}{\partial z},
\]

where \(f^{-1}(z; \theta_{\text{bpp}}) = z'\), and \(p(z|y; \theta_{\text{pri}}) = p(f^{-1}(z; \theta_{\text{bpp}})|y; \theta_{\text{pri}}))\)

\[
\det \frac{\partial f(z'; \theta_{\text{bpp}})}{\partial z'}(z'),
\]

where \(f(z'; \theta_{\text{bpp}}) = z\), and \(q(z'|x; \phi) = q(f(z'; \theta_{\text{bpp}})|x; \phi)\)

\[
\det \frac{\partial f(z'; \theta_{\text{bpp}})}{\partial z'}(z'),
\]

where the variable change \(z' \sim q(z'|x; \phi)\) according to the change of variable rule.

E. **VAE With Memory**

The posterior \(q(z|x; \phi)\) in the original VAE model is used to reconstruct the speech waveform and thus is more complicated than the prior from the phoneme sequence. To further relieve the burden of prior prediction, we simplify the posterior by designing a memory-based VAE model. The high-level idea of this design is that instead of directly using \(z \sim q(z|x; \phi)\) for waveform reconstruction, we just use \(z\) as a query to attend to a memory bank and use the attention result for waveform reconstruction, as shown in Fig. 2(d). In this way, the posterior \(z\) is only used to determine the attention weights in the memory bank and thus is largely simplified. The waveform reconstruction loss based on memory VAE can be formulated as

\[
\mathcal{L}_{\text{rec}}(\phi, \theta_{\text{dec}}) = -\mathbb{E}_{z \sim q(z|x; \phi)} \log p(x) | \text{Attention}\ (z, M, M), \theta_{\text{dec}}
\]

\[
\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QW_Q(KW_K)^T}{\sqrt{h}}\right)VW_V \omega,
\]

where \(\theta_{\text{dec}}\) denotes the waveform decoder, which covers not only the original waveform decoder but also the model parameters related to the memory mechanism, including the memory bank \(M\) and the attention parameters \(W_Q, W_K, W_V, \omega\), where \(M \in \mathbb{R}^{L \times h}\) and \(W_v \in \mathbb{R}^{h \times h}, L\) is the size of the memory bank and \(h\) is the hidden dimension.
the waveform loss in \( L_{\text{rec}} \) and \( L_{\text{e2e}} \) as negative log-likelihood loss for simplicity. Actually following [17], \( L_{\text{rec}} \) consists of GAN loss, feature mapping loss, and mel-spectrogram loss, while \( L_{\text{e2e}} \) consists of only GAN loss. We do not use soft-DTW in \( L_{\text{e2e}} \) since we found GAN loss can still perform well with mismatched lengths. We introduce the details of the waveform loss in Section III-F3.

2) Soft Dynamic Time Warping in KL Loss: Since the frame-level prior distribution \( p(z_i^j|\mathbf{y}; \theta_{\text{pri}}) \) usually has different lengths from the ground-truth speech frames, the standard KL loss cannot be applied. Therefore, we use a soft dynamic time warping (Soft-DTW) of KL loss for \( L_{\text{bwd}} \) and \( L_{\text{fwd}} \) to circumvent this mismatch.

The Soft-DTW version of the KL loss for \( L_{\text{bwd}} \) can be obtained by recursive calculation

\[
\begin{align*}
\gamma_{i,j} \text{ is the KL divergence loss between the simplified posterior } q(z_i^j|x; \phi, \theta_{\text{bwd}}) \text{ from frame } i \text{ to frame } j \text{ with the best alignment.} \quad &KL[q(z_i^j|x; \phi, \theta_{\text{bwd}})||p(z_i^j|\mathbf{y}; \theta_{\text{pri}})] \text{ is defined in (5), } \\
\min_y \text{ a soft-min operator, which is defined as } &\min_{(a_1, \ldots, a_n)} (-\gamma \log \Sigma e^{-\frac{a_i}{\gamma}}) \text{ and } \gamma = 0.01. \quad \text{warp is a warp penalty for not choosing the diagonal path and is set as 0.07.} \quad \text{p(z_i^j|\mathbf{y}; \theta_{\text{pri}}) is the } j^{th} \text{ frame of the simplified posterior, and } p(z_i^j|\mathbf{y}; \theta_{\text{pri}}) \text{ is the } j^{th} \text{ frame of the prior.}
\end{align*}
\]

The Soft-DTW version of KL loss for \( L_{\text{fwd}} \) is similar to that of \( L_{\text{bwd}} \), which can be defined as

\[
\begin{align*}
\gamma_{i,j} \text{ is the KL divergence loss between the enhanced prior } &p(z_i^j|x; \theta_{\text{fwd}}) \text{ from frame } i \text{ to frame } j \text{ with the best alignment.} \quad &KL[p(z_i^j|x; \theta_{\text{fwd}})||q(z_i^j|x; \phi)] \text{ is defined in (6), } \\
\min_y \text{ a soft-min operator, which is defined as } &\min_{(a_1, \ldots, a_n)} (-\gamma \log \Sigma e^{-\frac{a_i}{\gamma}}) \text{ and } \gamma = 0.01. \quad \text{warp is a warp penalty for not choosing the diagonal path and is set as 0.07.} \quad \text{p(z_i^j|\mathbf{y}; \theta_{\text{pri}}) is the } j^{th} \text{ frame of the enhanced prior, and } q(z_i^j|x; \phi) \text{ is the } j^{th} \text{ frame of the posterior.}
\end{align*}
\]

3) Waveform Decoder Loss: Instead of using negative log-likelihood loss in waveform reconstruction and prediction in (9) and (10), we use GAN loss, feature mapping loss, and mel-spectrogram loss as used in [17].

GAN Loss: The GAN loss follows LS-GAN [50], which is defined as follows. The generator is trained with loss function \( L_G \) while the discriminator is train with loss function \( L_D \)

\[
\begin{align*}
L_G &= E_z[1 - D(G(z))^2], \quad L_D = E_{x,z}[1 - D(x)^2] + D(G(z))^2,
\end{align*}
\]
where is the ground-truth waveform and is the input of the waveform decoder. We follow [15] for the design of discriminators.

Feature Mapping Loss: The feature mapping loss consists of the L1 distance between real samples and fake samples in terms of the intermediate feature in each layer of the discriminator, which can be formulated as

\[ E_{(x,z)} \left[ \sum_{l} \frac{1}{N_l} \| D_l^f(x) - D_l^f(G(z)) \|_1 \right], \quad (15) \]

where is the layer index in discriminator, \( D_l^f(\cdot) \) and \( N_l \) are the features and the number of features in the \( l \)th layer of the discriminator, respectively.

Mel-Spectrogram Loss: The mel-spectrogram loss is the L1 distance between the mel-spectrogram of the ground-truth waveform and that of the generated waveform, which can be defined as

\[ E_{(x,z)} = \| S(x) - S(G(z)) \|_1, \quad (16) \]

where \( S(\cdot) \) is the function that converts the waveform into the corresponding mel-spectrogram.

G. Advantages of NaturalSpeech

We explain how the designs in our NaturalSpeech system can close the quality gap to recordings.

- **Reduce training-inference mismatch**: We directly generate waveform from text and leverage a differentiable durator to ensure a fully end-to-end optimization, which can reduce the training-inference mismatch in the cascaded acoustic model/vocoder [13], [14], [18], [24] and explicit duration prediction [13], [15], [18]. Note that although VAE and Flow can have training-inference mismatch inherently (waveform is reconstructed from the posterior in training while predicted from the prior in inference for VAE, and Flow is trained in the backward direction and inferred in the forward direction), we design the backward/forward loss in (5) and (6) and the end-to-end loss in (10) to alleviate this problem.

- **Alleviate one-to-many mapping problem**: Compared to previous methods using reference encoder [11], [51], [52], [53] or F0/energy extraction [18] for variation information modeling, our posterior encoder \( \phi \) in VAE acts like a reference encoder that can extract all the necessary variance information in posterior distribution \( q(z|x; \phi) \). We do not predict F0 explicitly since it can be learned implicitly in the posterior encoder and the memory bank of VAE. To ensure the prior and posterior can match with each other, on the one hand, we simplify the posterior with memory VAE and backward mapping in the bidirectional prior/posterior module, and on the other hand, we enhance the prior with phoneme pre-training, differentiable durator, and forward mapping in the bidirectional prior/posterior module. Thus, we can alleviate the one-to-mapping problem to a large extent.

- **Increase representation capacity**: We leverage large-scale phoneme pre-training to extract better representation from the phoneme sequence and leverage the advanced generative models (Flow, VAE, GAN) to capture the speech data distributions better, and also design a memory mechanism in VAE, which can enhance the representation capacity of the TTS models for better voice quality.

We further list the difference between our NaturalSpeech and previous TTS systems as follows: 1) Compared to previous autoregressive TTS models such as Tacotron 1/2 [3], [4], WaveNet [2], TransformerTTS [9], and Wave-Tacotron [54], our NaturalSpeech is non-autoregressive in nature with a fast inference speed. 2) Compared to the previous systems with cascaded acoustic model and vocoder, such as Tacotron 1/2 [3], [4], FastSpeech 1/2 [10], [18], ParallelTacotron 2 [24], Glow-TTS [13], and Grad-TTS [14], we are fully end-to-end with no cascaded errors. 3) Compared to previous systems with various reference encoders and F0/duration prediction, such as FastSpeech 2 [18], AdaSpeech [52], and DelightfulTTS [11], we unify all the variance information with a posterior encoder and model the duration in a fully differentiable way. 4) Compared to previous fully end-to-end TTS systems such as EATS [19], FastSpeech 2s [18], and VITS [15], we bridge the quality gap to recordings with advanced model designs to closely match the prior and posterior in the VAE framework.

IV. EXPERIMENTS AND RESULTS

A. Experimental Settings

1) Datasets: We first evaluate our proposed NaturalSpeech on the LJSpeech dataset [28], which is widely used for benchmarking TTS. LJSpeech is a single-speaker English corpus and consists of 13,100 audio and text transcripts, with a total length of nearly 24 hours at a sampling rate of 22.05 kHz. We randomly split the dataset into a training set with 12,500 samples, a validation set with 100 samples, and a test set with 500 samples. We will release the training/validation/test split. We further evaluate NaturalSpeech on the multi-speaker VCTK dataset [29], which contains 109 speakers and 44 hours speech data. We follow [15] for the training/validation/test split of VCTK dataset. For phoneme pre-training on phoneme encoder, we collect a large-scale text corpus with 200 million sentences from the news-crawl dataset [55]. Note that we do not use any extra paired text and speech data except for the LJSpeech dataset. We conduct several preprocessing on the speech and text sequences: 1) We convert the text/character sequence into phoneme sequence [56] using a grapheme-to-phoneme tool [57]. 2) We use linear spectrograms as the input of the posterior encoder [15], instead of the original waveform sequence for simplicity. The linear spectrograms are obtained by short-time Fourier transform (STFT) with FFT size, window size, and hop size of 1,024, 1,024, and 256, respectively. 3) For the mel-spectrogram loss on the waveform decoder, we obtain the mel-spectrograms by applying 80-dimension mel-filterbanks on the linear spectrograms of the speech waveform.

2) Model Configurations: Our phoneme encoder is a stack of 6 Feed-Forward Transformer (FFT) blocks [10], where each block consists of a multi-head attention layer and a 1D convolution feed-forward layer, with hidden size of 192. In the
differentiable durator, the duration predictor consists of 3-layer convolution. We use 4 consecutive affine coupling layers [58] in our bidirectional prior/posterior module following [15]. We discard the scaling operation in the affine transform to stabilize the bidirectional training. The shifting in the affine transform is estimated by a 4-layer WaveNet [2] with a dilation rate of 1. The posterior encoder is based on a 16-layer WaveNet with a kernel size of 5 and a dilation rate of 1. The waveform decoder consists of 4 residual convolution blocks following [17], where each block has 3 layers of 1D convolution. We perform transpose convolution for upsampling at every convolution block at a rate of \([8,8,2,2]\). The details of the hyperparameters of NaturalSpeech are listed in the Supplementary Materials. The number of model parameters for \(\theta\) is 28.7 M, and for the discriminators is 46.7 M. Note that only \(\theta_{\text{pho}}, \theta_{\text{dur}}, \theta_{\text{bpp}}, \) and \(\theta_{\text{dec}}\) with 28.7 M model parameters are used in inference.

3) Training Details: We pre-train our phoneme encoder on 200 M phoneme sequences, which are converted from text with grapheme-to-phoneme conversion. The size of the phoneme dictionary is 182. We learn the sup-phoneme using Byte-Pair Encoding (BPE) [59] with a sup-phoneme dictionary size of 30,088. We conduct the pre-training on 8 NVIDIA A100 GPUs with 80G memory (we only use A100 for phoneme pre-training, and use V100 for the remaining training of NaturalSpeech), with a total batch size of 1,024 sentences for 120k training steps. The mask ratio for sup-phoneme is 15%.

We initialize the phoneme encoder with pre-trained weights and train our NaturalSpeech system on 8 NVIDIA V100 GPUs with 32 G memory, with a dynamic batch size of 8,000 speech frames (under hop size of 256) per GPU, and a total 15k training epochs. We use AdamW optimizer [60] with \(\beta_1 = 0.8, \beta_2 = 0.99\). The initial learning rate is \(2 \times 10^{-4}\), with a learning rate decay factor \(\gamma = 0.999875\) in each epoch, i.e., the learning rate is multiplied by \(\gamma\) in every epoch. We find it helpful to stabilize the training of our system and achieve better results through a warmup stage with 1k epochs at the beginning of the training, and a tuning stage with 2k epochs at the end of the training. In the tuning stage, we only use \(L_{\text{ce}}\) to tune the model. We freeze the parameters of the posterior encoder, waveform decoder, phoneme encoder, and bidirectional prior/posterior, and only update the durator for full end-to-end duration optimization.

### B. Comparison With Human Recordings

We first compare the speech generated by NaturalSpeech with human recordings in terms of both MOS and CMOS evaluation on the LJSpeech dataset. As described in Section II, we use 50 test utterances, each with 20 judges for evaluation. As shown in Tables III and IV, our system achieves similar quality scores with human recordings in both MOS and CMOS. Importantly, our system achieves \(-0.01\) CMOS compared to recordings, with a Wilcoxon p-value \(p \geq 0.05\), which demonstrates the speech generated by our system has no statistically significant difference from human recordings.\(^3\) \(^4\) Thus, our NaturalSpeech achieves human-level quality according to the definition and judgment in Section II.

Besides the single-speaker LJSpeech dataset, we further train NaturalSpeech on the multi-speaker VCTK dataset, and conduct CMOS evaluation on the VCTK test set. As shown in Table V, NaturalSpeech achieves 0.04 CMOS scores compared to human recordings, with a Wilcoxon p-value \(p \geq 0.05\), demonstrating that NaturalSpeech can also achieve human-level quality on a multi-speaker dataset.

### C. Comparison With Previous TTS Systems

We compare our NaturalSpeech with previous TTS systems including: 1) FastSpeech 2 [18] + HiFiGAN [17], 2) Glow-TTS [13] + HiFiGAN [17], 3) Grad-TTS [14] + HiFiGAN [17], and 4) VITS [15] on the LJSpeech dataset. We reproduce the results of all these systems on our own, which can match or even beat the quality in their original papers (note that the HiFiGAN vocoder is fine-tuned on the predicted mel-spectrograms for better synthesis quality). Both the MOS and CMOS results are shown in Table VI. It can be seen that our NaturalSpeech achieves better voice quality than these systems in terms of both MOS and CMOS.

### D. Ablation Studies and Method Analyses

1) Ablation Studies: We further conduct ablation studies to verify the effectiveness of each module in our system, as shown in Table VII. We describe the ablation studies as follows: 1)

\(^3\)Audio samples can be found in https://speechresearch.github.io/naturalspeech/

\(^4\)Note that some human recordings in LJSpeech dataset may contain strange rhythm ups and downs that affect the rating score. To ensure the human recordings used for evaluation are of good quality, we let judges exclude the recordings with strange rhythms from evaluation. Otherwise, our NaturalSpeech will achieve better CMOS than human recordings. In a CMOS test without excluding bad recordings, NaturalSpeech achieves \(+0.09\) CMOS better than recordings.
By removing phoneme pre-training, we do not initialize the phoneme encoder from pre-trained weights but just random initialization, which brings $-0.09$ CMOS drop, demonstrating the effectiveness of phoneme pre-training. 2) By removing the differentiable durator, we do not use the learnable upsampling layer and end-to-end duration optimization but just use duration predictor for hard expansion. In this way, we use tonal alignment search [13] to provide the duration label to train the duration predictor through the whole training process. Removing the differentiable durator causes a $-0.12$ CMOS drop, demonstrating the importance of end-to-end optimization in duration modeling. 3) By removing the bidirectional prior/posterior module, we only use $L_{\text{swd}}$ in training and do not use $L_{\text{swd}}$. It brings a $-0.09$ CMOS drop, showing the gain by leveraging bidirectional training to bridge the gap between posterior and prior. 4) By removing the memory mechanism in VAE, we use the original VAE for waveform reconstruction, which causes a $-0.06$ CMOS drop, showing the effectiveness of memory in VAE to simplify the posterior.

2) Inference Latency: We compare the inference speed of our NaturalSpeech with previous TTS systems. We measure the latency by using an NVIDIA V100 GPU with a batch size of 1 sentence and averaging the latency over the sentences in the LJSpeech test set. The results are shown in Table VIII. The model components $\theta_{\text{plp}}, \theta_{\text{dur}}, \theta_{\text{bpp}},$ and $\theta_{\text{dec}}$ in NaturalSpeech are used in inference, with 28.7 M model parameters. Our NaturalSpeech achieves faster or comparable inference speed when compared with the previous systems, and achieves better voice quality.

V. CONCLUSION AND DISCUSSIONS

In this paper, we conduct a systematic study of the problems related to human-level quality in TTS. We first give a formal definition of human-level quality, describe the guidelines to judge it, and further build a TTS system called NaturalSpeech to achieve human-level quality. Specifically, after analyzing the quality gap on several competitive TTS systems, we develop a fully end-to-end text-to-waveform generation system, with several designs to close the gap to human recordings, including phoneme pre-training, differentiable durator, bidirectional prior/posterior module, and memory mechanism in VAE. Evaluations on the popular single-speaker LJSpeech dataset and multi-speaker VCTK dataset demonstrate that our NaturalSpeech achieves human-level quality with CMOS evaluations, with no statistically significant difference from human recordings for the first time.

Note that by claiming our NaturalSpeech system achieves human-level quality on the LJSpeech/VCTK dataset, we do not mean that we can surpass or replace human, but the quality of NaturalSpeech is statistically indistinguishable from human recordings on these datasets. Meanwhile, although our evaluations are conducted on the LJSpeech/VCTK dataset, we believe the technologies in NaturalSpeech can be applied to other languages, speakers, and styles to improve the general synthesis quality. We will further try to achieve human-level quality in more challenging datasets or scenarios, such as expressive voices, long-form audiobook voices, and singing voices that have more dynamic, diverse, and contextual prosody in our future work.

REFERENCES

[1] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, “A survey on neural speech synthesis,” 2021, arXiv:2106.15561.
[2] A. van den Oord et al., “WaveNet: A generative model for raw audio,” 2016, arXiv:1609.03499.
[3] J. Shen et al., “Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2018, pp. 4779–4783.
[4] Y. Wang et al., “Tacotron: Towards end-to-end speech synthesis,” in Proc. Ann. Conf. Int. Speech Commun. Assoc., 2017, pp. 4006–4010.
[5] S. O Ankl et al., “Deep voice: Real-time neural text-to-speech,” in Proc. Int. Conf. Mach. Learn., PMLR, 2017, pp. 195–204.
[6] A. Gibiansky et al., “Deep voice 2: Multi-speaker neural text-to-speech,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, pp. 2966–2974.
[7] W. Ping et al., “Deep voice 3: 2000-speaker neural text-to-speech,” in Proc. Int. Conf. Learn. Representations, 2018, pp. 214–217.
[8] H. Tachibana, K. Uenoyma, and S. Aihara, “Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2018, pp. 4784–4788.
[9] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, “Neural speech synthesis with transformer network,” in Proc. AAAI Conf. Artif. Intell., 2019, pp. 6706–6713.
[10] Y. Ren et al., “FastSpeech: Fast, robust and controllable text to speech,” in Proc. 33rd Int. Conf. Neural Inf. Process. Syst., 2019, Art. no. 285.
[11] Y. Liu et al., “DelightfulTTS: The Microsoft speech synthesis system for Blizzard challenge 2021,” 2021, arXiv:2110.12612.
[12] R. Prenger, R. Valle, and B. Catanzaro, “WaveGlow: A flow-based generative network for speech synthesis,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2019, pp. 3617–3621.
[13] J. Kim, S. Kim, J. Kong, and S. Yoon, “Glowl-TTS: A generative flow for text-to-speech via monotonic alignment search,” in Proc. Adv. Neural Inf. Process. Syst., 2020, Art. no. 676.

[14] L. Popov, I. Wark, V. Golovchans, T. Sadekova, and K. Kudinov, “Grad-TTS: A diffusion probabilistic model for text-to-speech,” in Proc. Int. Conf. Mach. Learn., PMLR, 2021, pp. 8599–8608.

[15] J. Kim, J. Kong, and J. Son, “Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech,” in Proc. Int. Conf. Mach. Learn., PMLR, 2021, pp. 5530–5540.

[16] N. Kalchbrenner et al., “Efficient neural audio synthesis,” in Proc. Int. Conf. Mach. Learn., PMLR, 2018, pp. 2410–2419.

[17] J. Kong, J. Kim, and J. Bae, “HIFI-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” in Proc. Adv. Neural Inf. Process. Syst., 2020, no. 1428.

[18] Y. Ren et al., “FastSpeech 2: Fast and high-quality end-to-end text to speech,” in Proc. Int. Conf. Learn. Representations, 2021.

[19] J. Donahue, S. Dieleman, M. Bifflowski, E. Olsen, and K. Simonyan, “End-to-end adversarial text-to-speech,” in Proc. Int. Conf. Learn. Representations, 2021.

[20] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, pp. 6309–6318.

[21] A. Ramesh et al., “Parallel Tacotron 2: A non-autoregressive neural TTS model with differentiable duration modeling,” 2021, arXiv:2103.14574.

[22] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2013, arXiv:1312.6114.

[23] G. Zhang et al., “Mixed-phoneme BERT: Improving BERT with mixed phone and sup-phoneme representations for text to speech,” 2022, arXiv:2203.17290.

[24] I. Elia et al., “Parallel Tacotron 2: A non-autoregressive neural TTS model with differentiable duration modeling,” 2021, arXiv:2103.14574.

[25] L. Dinh, D. Krueger, and Y. Bengio, “NICE: Non-linear independent components estimation,” 2014, arXiv:1410.8516.

[26] P. Durk et al., “Improved variational inference with inverse autoregressive flow,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 4743–4751.

[27] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1 × 1 convolutions,” in Proc. 32nd Int. Conf. Neural Inf. Process. Syst., 2018, pp. 10236–10245.

[28] K. Ito, “The LJ speech dataset,” 2017. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[29] C. Veaux et al., “Superseded-CSR VCTK corpus: English multi-speaker corpus for CSR voice cloning toolkit,” 2016. [Online]. Available: http://datashare.is.ed.ac.uk/handle/10283/2651

[30] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Egberts, “Perceptual evaluation of speech quality (PESQ)—A new method for speech quality assessment of telephone networks and codecs,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2001, pp. 749–752.

[31] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “An algorithm for intelligibility prediction of time–frequency weighted noisy speech,” IEEE Trans. Audio Speech Lang. Process., vol. 19, no. 7, pp. 2125–2136, Sep. 2011.

[32] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “SDR-halved or well done?,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2019, pp. 626–630.

[33] C.-C. Lo et al., “MOSNet: Deep learning-based objective assessment for voice conversion,” in Proc. Annu. Conf. Int. Speech Commun. Assoc., 2019, pp. 1541–1545.

[34] Y. Leng, X. Tan, S. Zhao, F. Soong, X.-Y. Li, and T. Qin, “MBNet: MOS prediction for synthesized speech with mean-bias network,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2021, pp. 391–395.

[35] F. Wilcoxon, “Individual comparisons by ranking methods,” in Breakthroughs in Statistics, Berlin, Germany: Springer, 1992, pp. 196–202.

[36] M. Ding et al., “CogView: Mastering text-to-image generation via transformers,” 2021, arXiv:2105.13290.

[37] C. Wu et al., “NUWA: Visual synthesis pre-training for neural visual world creation,” 2021, arXiv:2111.12417.

[38] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, “VideoGPT: Video generation using VQ-VAE and transformers,” 2021, arXiv:2104.10157.

[39] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev, “Latent video transformer,” 2020, arXiv:2006.10704.

[40] A. Razavi, A. van den Oord, and O. Vinyals, “Generating diverse high-fidelity images with VQ-VAE-2,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 14866–14876.

[41] P. Esser, R. Rombach, and B. Ommer, “Taming transformers for high-resolution image synthesis,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 12873–12883.

Xu Tan (Senior Member, IEEE) is a principal research scientist and research manager with Microsoft Research Asia, Beijing, China. He works on deep learning and natural language/speech/music processing. He has published more than 100 papers on top AI conferences and served as the area chair of some conferences (e.g., NeurIPS and AAAI). He has developed machine translation systems that achieved human parity in Chinese-English translation, in 2018 and won several champions in the WMT machine translation competition, in 2019, and designed several popular language/speech/music systems, such as MASS, FastSpeech/NaturalSpeech, and Muzic, and transferred many research works to Microsoft products.

Jiawei Chen is currently working toward the master’s degree with the South China University of Technology, Guangzhou, China. He is currently an intern with Microsoft Azure Speech, Beijing, China. His main research interests include deep learning and speech/music processing.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Haohe Liu is currently working toward the PhD degree with the University of Surrey, Guildford, U.K., and was an intern with Microsoft Azure Speech, Beijing, China. His main research interests include topics related to speech, music, and general audio.

Jian Cong received the master’s degree from Northwestern Polytechnical University, Xi’an, China, and was an intern with Microsoft Azure Speech, Beijing, China. His main research interests include machine learning and text-to-speech synthesis.

Chen Zhang received the master’s degree from Zhejiang University, Hangzhou, China, and is currently a software engineer with Microsoft Azure Speech, Beijing, China. Her main research interest is text-to-speech synthesis.

Yanqing Liu is currently a principal software engineer with Microsoft Azure Speech, Beijing, China. His main research interest is text-to-speech synthesis.

Xi Wang received the master’s degree from the Beijing University of Posts and Telecommunications, Beijing, China. She is currently a principal speech scientist with Microsoft Azure Speech, Beijing, China. Her main research interest is text-to-speech synthesis.

Yichong Leng is currently working toward the PhD degree with the University of Science and Technology of China, Hefei, China, and is an intern with Microsoft Research Asia, Beijing, China. His main research interests include natural language processing, speech, and audio processing.

Yuanhao Yi received the master’s degree from the University of Science and Technology of China, Hefei, China. He is currently a software engineer with Microsoft Azure Speech, Beijing, China. His main research interest is text-to-speech synthesis.

Lei He received the bachelor’s and PhD degrees from Tsinghua University, Beijing, China. He is a principal applied science manager with Microsoft Azure Speech, Beijing, China. His main research interests include artificial intelligence, human language processing, speech synthesis, speech recognition, and pronunciation assessment.

Sheng Zhao (Senior Member, IEEE) received the master’s degree from Tsinghua University, Beijing, China. He is a partner group engineer manager with Microsoft Azure Speech, Beijing. He takes charge of the speech synthesis group in Microsoft Azure Cognitive Services. His main research interests include artificial intelligence, human language processing, speech synthesis, voice conversion, and speech recognition.

Tao Qin (Senior Member, IEEE) received the bachelor’s and PhD degrees from Tsinghua University, Beijing, China. He is a senior principal researcher and research manager with Microsoft Research Asia. He has authored/co-authored more than 100 papers in refereed conferences. His research interests include machine learning (with a focus on deep learning and reinforcement learning), artificial intelligence (with applications to language and speech processing), game theory and multiagent systems (with applications to cloud computing, online and mobile advertising, e-commerce), information retrieval, and computational advertising. He is a senior member of ACM.

Frank Soong (Fellow, IEEE) received the BS degree from National Taiwan University, Taipei, Taiwan, the MS degree from the University of Rhode Island, South Kingstown, Rhode Island, and the PhD degree from Stanford University, Stanford, California, all in electrical engineering. He works on fundamental research on speech and its practical applications, including automatic speech recognition, speaker verification, and text-to-speech synthesis. His professional research career spans four decades, first with Bell Labs in the USA, then with ATR Interpreting Telephony Lab in Japan, before joining Microsoft Research Asia, in 2004. He was a member of the Speech and Language Technical Committee, IEEE Signal Processing Society, and other IEEE functions, including an associate editor for the IEEE Speech and Audio Transactions and chairing IEEE Workshop.

Tie-Yan Liu (Fellow, IEEE) received the bachelor’s and PhD degrees from Tsinghua University, Beijing, China. He is a distinguished scientist and assistant managing director of Microsoft Research Asia, leading the machine learning research area. He is very well known for his pioneer work on learning to rank and computational advertising. He is an adjunct/honorary professor with Carnegie Mellon University (CMU), Pittsburgh, Pennsylvania, University of Nottingham, Nottingham, U.K., and several other universities in China. He has authored/co-authored more than 200 papers in refereed conferences and journals, e.g., SIGIR, WWW, ICML, KDD, NeurIPS, IJCAI, AAAI, ACL, IEEE Transactions on Pattern Analysis and Machine Intelligence, and so on, with more than 30,000 citations. His recent research interests include deep learning, natural language and speech processing, AI for science, etc. He is a fellow of ACM.