CLTTS: A Cooperative Learning Strategy for Non-Autoregressive Text-to-Speech

Wei Zhao

Zhejiang University

December 7, 2023

Abstract

Non-autoregressive text-to-speech (TTS) has recently received a lot of attention due to its reliability and fast reasoning. Despite its outstanding achievement, non-autoregressive speech synthesis still faces some critical challenges. A major issue is that non-autoregressive methods necessitate an external toolkit to align the speech with the transcript, thus substantially complicating the process of building the model. Besides, non-autoregressive methods suffer from the one-to-many mapping issue, where the same transcript may correspond to speech in numerous styles. This problem may harm the expressiveness of the generated speech because the model tends to provide output with an average style. To address the above issues, this paper proposes a cooperative learning strategy for non-autoregressive speech synthesis. Specifically, the suggested method employs both an autoregressive and a non-autoregressive TTS model during the training procedure. The autoregressive model is trained as a partner at each iteration, providing essential alignment information and also the prosody embedding of the speech to the non-autoregressive model. After receiving the above useful knowledge, the non-autoregressive model can be further trained without relying on external alignment tools. Meanwhile, the prosody embedding from the autoregressive model and the pitch information from the raw audio can be utilised together to alleviate the one-to-many mapping problem. Experimental results demonstrate that our approach can produce comparable speech to the most popular FastSpeech 2 model while drastically reducing the complexity of constructing a non-autoregressive TTS model.
Graphical Abstract

CLTTS: A Cooperative Learning Strategy for Non-Autoregressive Text-to-Speech

Wei Zhao, Zuyi Wang, Li Xu
Highlights

CLTTS: A Cooperative Learning Strategy for Non-Autoregressive Text-to-Speech

Wei Zhao, Zuyi Wang, Li Xu

- We propose a cooperative learning method in which the non-autoregressive TTS model is trained along with the autoregressive TTS model to get the duration data it needs for training. With the suggested method, our non-autoregressive TTS model can be trained from scratch without relying on an external alignment toolkit.

- To supplement the non-autoregressive modeling with more prosodic knowledge, we also introduce a self-supervised autoregressive TTS model into the cooperative learning schema. The prosody embedding acquired via self-supervised learning may be an effective complement to conventional artificial features of speech, such as pitch and energy. With the help of both the prosody embedding and pitch information, the non-autoregressive TTS model is capable of generating highly expressive speech.

- We conduct extensive experiments on the CSMSC and LJSpeech datasets. Experimental results demonstrate that our model can perform comparable to the most popular FastSpeech 2 with a significantly simpler construction process.
CLTTS: A Cooperative Learning Strategy for Non-Autoregressive Text-to-Speech

Wei Zhao\textsuperscript{a,b}, Zuyi Wang\textsuperscript{a,b}, Li Xu\textsuperscript{a,b,}\* \\
\textsuperscript{a}College of Electrical Engineering, Zhejiang University, Hangzhou, 310027, China  
\textsuperscript{b}Institute of Robotics, Zhejiang University, Yuyao, 315400, China

Abstract

Non-autoregressive text-to-speech (TTS) has recently received a lot of attention due to its reliability and fast reasoning. Despite its outstanding achievement, non-autoregressive speech synthesis still faces some critical challenges. A major issue is that non-autoregressive methods necessitate an external toolkit to align the speech with the transcript, thus substantially complicating the process of building the model. Besides, non-autoregressive methods suffer from the one-to-many mapping issue, where the same transcript may correspond to speech in numerous styles. This problem may harm the expressiveness of the generated speech because the model tends to provide output with an average style. To address the above issues, this paper proposes a cooperative learning strategy for non-autoregressive speech synthesis. Specifically, the suggested method employs both an autoregressive and a non-autoregressive TTS model during the training procedure. The autoregressive model is trained as a partner at each iteration, providing essential alignment information and also the prosody embedding of the speech to the non-autoregressive model. After receiving the above useful knowledge, the non-autoregressive model can be further trained without relying on external alignment tools. Meanwhile, the prosody embedding from the autoregressive model and the pitch information from the raw audio can be utilised together to alleviate the one-to-many mapping problem. Experimental results demonstrate that our approach can produce comparable speech to the most popular FastSpeech 2 model while drastically reducing the complexity of constructing a non-autoregressive TTS model.

\*Corresponding author: xupower@zju.edu.cn
1. Introduction

Speech synthesis has advanced quickly in recent years, thanks to the boost of deep learning. In particular, the end-to-end TTS with the sequence-to-sequence neural network, such as Tacotron[1], Tacotron 2[2], DCTTS[3], DeepVoice 3[4], and Transformer TTS[5], have all achieved remarkable success. These models can be categorised as autoregressive models that recursively produce the spectrum frame by frame. However, the autoregressive speech synthesis method has two inherent drawbacks. One is that the recursive generating schema leads to a slow inference speed. The other is that the neural attention mechanism used to align the transcript with the speech spectrum is insufficiently robust, which could result in pronunciation errors.

To address the above problems, researchers have proposed some non-autoregressive neural TTS models. The non-autoregressive model generates the speech spectrum entirely in parallel, thus being significantly faster than the autoregressive method. Additionally, the non-autoregressive model often employs external toolkits to obtain the pronunciation durations of input phonemes and then aligns the transcript with the spectrum according to these durations, guaranteeing robustness for industrial deployment. Ren suggests the famous non-autoregressive FastSpeech[6], which performs similarly to the prior autoregressive TTS models. However, FastSpeech needs to acquire input phoneme durations from a pre-trained autoregressive TTS model. Moreover, knowledge distillation is conducted using the output of the pre-trained autoregressive model to mitigate the one-to-many mapping problem. In conclusion, the reliance on external alignment tools and the multi-modal issue have been the two major challenges for non-autoregressive speech synthesis.

To free the model from dependence on external alignment tools, Zeng et al.[7] introduce AlignTTS to align the text with the spectrum via a mixed-density network. Ma et al.[8] present a trainable position encoding to build a specialized alignment model that can be trained together with the non-autoregressive TTS model. Miao et al.[9] propose using the index mapping vector to learn the alignment information in an end-to-end training procedure. Elias et al.[10] present a differential duration model that can be directly
integrated into the non-autoregressive TTS model. However, these methods usually demand human intervention to perform tedious multi-stage training or necessitate an advanced degree of mathematical expertise. JDI-T[11] introduces a joint training method in which the non-autoregressive and autoregressive models share the same text encoder, and the non-autoregressive model obtains the duration from the autoregressive decoder. Despite JDI-T being straightforward, the joint training framework is unstable and may lead to poor performance.

Researchers also present some studies to address the multi-modal problem in non-autoregressive TTS. FastSpeech 2[12] suggests incorporating both a pitch predictor and an energy predictor into the TTS decoder to improve the expressiveness of the predicted spectrum. Similarly, FastPitch[13] introduces a higher-granularity pitch predictor and also achieves promising results. There are further investigations that use advanced neural network structures, such as generating adversarial networks (GAN)[14, 15, 16] and diffusion models[17, 18, 19], to produce more expressive speech. Nevertheless, GAN-based models are hard to train, and diffusion models require a time-consuming sampling process. It is worth noting that prior studies usually focus on only one of the major challenges in non-autoregressive speech synthesis. Few efforts are made to address the two concerns mentioned above at the same time.

In this paper, we suggest a simple but effective way for non-autoregressive models to avoid using external alignment tools and improve the expressiveness of speech at the same time. Specifically, we propose a cooperative learning strategy in which the non-autoregressive TTS model is trained alongside the autoregressive TTS model to get the duration information of input phonemes. During the training procedure, the autoregressive and non-autoregressive TTS models are updated with separate optimizers to ensure stable gradient backpropagation. To supplement the non-autoregressive modeling with more prosodic knowledge, we also introduce a self-supervised autoregressive TTS model into the cooperative learning schema. The prosody embedding acquired via self-supervised learning may be an effective complement to conventional artificial features of speech, such as pitch and energy. Taking durations and prosody embedding from the cooperative partner as training labels, our non-autoregressive TTS model employs two corresponding predictors to estimate the above knowledge according to the input. Besides, a pitch predictor is also incorporated into the non-autoregressive model. With the help of all the predicted information, our non-autoregressive TTS
model no longer requires external alignment tools and is simultaneously capable of generating highly expressive speech. Experimental results demonstrate that our model can perform comparable to the most popular FastSpeech 2 with a significantly simpler construction process.

2. Related work

2.1. Autoregressive speech synthesis

The autoregressive speech synthesis uses neural attention-based sequence-to-sequence networks[20, 21, 22] to directly transform the text into the spectrum, as illustrated in Figure 1. In particular, the model recursively predicts the spectrum frame $y_t$ at the current time step according to the encoder output $(x_1, x_2, \ldots, x_T')$ and previously generated spectrum frames $(y_1, y_2, \ldots, y_{t-1})$. It can be observed from Equation 1 that the autoregressive TTS model can provide an accurate estimation of the joint probability density $p_{AR}(Y \mid X; \theta)$ of the spectrum via the chain rule. $\theta$ denotes the trainable parameters of the neural network.

$$p_{AR}(Y \mid X; \theta) = \prod_{t=1}^{T} p(y_t \mid y_{1:t-1}, x_{1:T'}; \theta)$$

Figure 1: Architecture of an autoregressive TTS model.
Due to their straightforward pipeline and excellent performance, autoregressive TTS models have been quite popular for a long time. To be more specific, Figure 2 gives an illustration of the well-known Tacotron 2 model, which mainly consists of three parts: an encoder, a decoder, and a post-net. The encoder, including an input embedding layer, three convolution layers, and a bidirectional LSTM, obtains semantic representations from the input text for the decoder. The decoder first uses two linear layers to project the reference spectrum into a hidden embedding space. Then, two stacked attention LSTMs receive the above hidden states and provide a context vector via the neural attention mechanism. Finally, another two stacked decoder LSTMs are utilised to predict the spectrum according to this context vector. The postnet, which contains five convolution layers, is used to refine the produced spectrum from a global perspective.

2.2. Non-autoregressive speech synthesis

Unlike autoregressive speech synthesis, the non-autoregressive TTS model produces all the frames of a spectrum in parallel, as shown in Figure 3. The
frame at a given step is independent of the frames at other steps. Therefore, the feed-forward non-autoregressive model is not capable of accurately estimating the joint probability distribution of the spectrum through the chain rule, which leads to the one-to-many mapping problem. Specifically, the non-autoregressive TTS model first predicts the duration $t$ of each input phoneme according to the encoder outputs $(x_1, x_2, \ldots, x'_T)$ and then produces the spectrum according to the expanded encoder outputs, as shown in Equation 2.

$$p_{NA}(Y \mid X; \theta) = p_L(T \mid x_1:T'; \theta) \cdot \prod_{t=1}^{T} p(y_t \mid x_1:T'; \theta)$$  (2)

Benefiting from the parallel generative schema, non-autoregressive TTS models have a significantly faster inference speed than autoregressive ones. Furthermore, non-autoregressive models are more robust for eliminating pronunciation errors since they use durations rather than neural attention to align the text with the spectrum. Figure 4 shows the well-known non-autoregressive TTS baseline, FastSpeech, which is composed of an encoder, a length regulator, and a decoder. Either the encoder or decoder is a stack of $N$ identical feed-forward transformer blocks, where each block contains a self-attention module and a convolution module. Position encodings are added before the encoder and decoder, respectively, to inject temporal information.
for the model. The non-autoregressive TTS model employs a length regulator to bridge the length gap between input and output. In particular, the non-autoregressive model usually resorts to external tools to get the detailed duration of each input phoneme and expand the length of encoder outputs to be consistent with the speech spectrum. During the training, a duration predictor is also trained to predict the duration of each phoneme according to the encoder outputs. Therefore, the non-autoregressive TTS models are capable of predicting durations in inference.

Figure 4: The detailed structure of FastSpeech.

3. The proposed method

Non-autoregressive speech synthesis usually faces the thorny problem that an external alignment toolkit is required to obtain durations for each input token. When developing a TTS model, the employment of such toolkits leads to substantial labour costs. Besides, the non-autoregressive TTS model also requires prosody information, such as pitch and energy, to improve the expressiveness of the speech. Considering these artificial features reflect the speech variance in a specific aspect, there may exist other more comprehensive descriptions for the prosody. In this paper, we propose a cooperative
learning strategy for the non-autoregressive TTS model to simplify its construction pipeline and enhance the expressiveness of the output speech at the same time. Our model is able to be trained from scratch automatically without relying on external tools.

3.1. Basic cooperative learning strategy

Figure 5: Basic cooperative learning strategy for the non-autoregressive TTS.

Figure 5 demonstrates a basic cooperative learning strategy for non-autoregressive speech synthesis. Our method employs both the autoregressive TTS model and the non-autoregressive TTS model in the same training process. At each iteration, the current batch of input data is first sent to the autoregressive model A to implement forward and backward propagation. Then the optimizer A is used to update the parameters of model A. Meanwhile, the neural attention alignment produced by the autoregressive model is converted to the durations of input phonemes, as shown in Equation 3, where $d_i$ represents the duration of $i$-th input phoneme, $S$ represents the length of spectrum, $s$ denotes the index of a single spectrum frame and $t$ denotes the index of an input phoneme. Afterward, the input data and the obtained durations are sent to the non-autoregressive model B for training. A separate optimizer B is further employed to update the parameters of model B. Algorithm 1 summarises the detailed process of the basic cooperative
learning strategy for non-autoregressive speech synthesis.

\[ d_i = \sum_{s=1}^{S} \left[ \arg\max_{t} a_{s,t} = i \right] \]  

(3)

Algorithm 1 The basic cooperative learning strategy

```python
for input in Dataloader do
    output_A, attention = Model_A(input)
    loss_A = LossFunction_A(output_A, target_A)
    loss_A.backward()
    Optimizer_A.update()
    duration = AttentionToDuration(attention.detach())
    output_B = Model_B(input, duration)
    loss_B = LossFunction_B(output_B, target_B)
    loss_B.backward()
    Optimizer_B.update()
end for
```

In particular, we take Tacotron 2 as the autoregressive model in our basic cooperative learning method since Tacotron 2 is capable of learning a robust alignment between text and spectrum due to its location-sensitive attention. The only difference we introduce is that all the LSTM networks in the original Tacotron 2 model are replaced with the GRU network to save GPU memory and speed up the training. Besides, the FastSpeech model is employed as the non-autoregressive model to build a baseline for the suggested method.

3.2. Improved cooperative learning strategy

The basic cooperative learning strategy can help the non-autoregressive TTS model obtain pronunciation durations automatically without relying on an external alignment toolkit. However, the other challenging problem, i.e., the one-to-many mapping problem, has not been taken into account. To that end, we introduce an improved cooperative learning approach to further enhance the performance of non-autoregressive speech synthesis. Compared to the prior basic cooperative learning method, two improvements are introduced in the current approach, as shown in Figure 6. First, we introduce an autoencoder-like autoregressive model instead of the regular autoregressive model. The autoencoder TTS model employs an additional spectrum
encoder except for the text encoder, which extracts a highly compact representation from the spectrum. These hidden states are concatenated together with the output from the text encoder, and the decoder receives the concatenated hidden representations to construct the spectrum. Considering that the text encoder provides semantic content, the output of the spectrum encoder would mainly offer prosody information.

Second, we present a non-autoregressive TTS model that can predict prosody according to the input text. As some studies [23] have shown, fundamental frequency, i.e., pitch, can generally denote the speech rhythm. Thus, introducing the pitch embedding into a non-autoregressive model may improve its performance effectively. Moreover, the prosody embedding learned via the self-supervised autoregressive model is additionally introduced to the non-autoregressive TTS model as a supplement.

![Figure 6: Improved cooperative learning strategy for the non-autoregressive TTS.](image)

3.2.1. Self-supervised autoregressive TTS

Figure 7 illustrates the autoencoder-like autoregressive model, which has a similar architecture as the TTS model in [24]. This model has almost the same architecture as Tacotron 2, except that an extra spectrum encoder is introduced to obtain the prosody embedding. This prosody embedding is then concatenated with the outputs of the text encoder, and the decoder finally produces the spectrum with the neural attention mechanism based on the concatenated representations.
As shown in Figure 8, the spectrum encoder receives the input with a shape of $l \times d$ and downsamples the spectrum via a 6-layer convolutional network. Each convolution layer is composed of $3 \times 3$ filters with $2 \times 2$ stride, Relu activation, and Batch normalization. In order to guarantee that the output is compact enough to contain merely prosody information, padding is not introduced during the convolution. The number of convolution filters is 32, 32, 64, 64, 128, and 128, respectively. After the downsampling of convolution layers, the output can be unrolled as hidden states with the shape of $\lfloor l \rfloor \times 128 \lfloor d \rfloor$. Subsequently, a 128-unit GRU network is used to process these hidden states, and the output at the final time step is specified as the prosody embedding learned from the spectrum.

3.2.2. Non-autoregressive TTS with prosody predictor

To improve the expressiveness of the speech, we introduce a prosody predictor into the non-autoregressive model, as demonstrated in Figure 9. Our non-autoregressive TTS model has a similar structure as the prior FastSpeech, except that a prosody predictor is added on top of the text encoder. The prosody predictor produces the prosody information according to the output of the text encoder and injects the predicted information into the
original encoder output. Then, the TTS model generates the spectrum in the same manner as FastSpeech.

Specifically, Figure 10 illustrates the detailed structure of the prosody predictor, which includes two prediction branches. The first prediction branch contains two 1-D convolution layers, followed by Relu activation, layer normalisation, and dropout, respectively. A linear unit is subsequently utilised to produce the final pitch estimation. It is worth mentioning that we don’t use the original pitch extracted from the audio as the prediction target. Instead, we average the original pitch according to the durations of the input phonemes and take the averaged pitch as the training label, as in [13]. The second prediction branch is to predict the prosody embedding learned by the autoregressive TTS model. While receiving the encoder outputs, a GRU network is applied to summarise the semantic information, and the hidden state at the last time step is taken as the estimation for the prosody embedding. Another two linear units with Relu and Tanh activation, respectively, are employed to improve the accuracy of the prediction. The mean square error (MSE) loss is used to guide the training of the prosody predictor.

We inject the pitch and prosody information produced by the above prediction branches into the encoder outputs, as shown in Equation 4, where $h \in \mathbb{R}^{n \times d}$ denotes the encoder outputs, $f \in \mathbb{R}^{n}$ denotes the pitch estimation, and $p \in \mathbb{R}^{e}$ denotes the prosody embedding. The 1-D convolution is used to
Figure 9: The proposed non-autoregressive TTS model with prosody predictor

Project the pitch estimation $f$ to the same space as encoder outputs $h$, and the pitch estimation is then added to the encoder outputs. Since the prosody embedding $p$ is a time-independent state, we broadcast and concatenate it to the encoder outputs. Finally, a linear unit is used to recover the shape of concatenated hidden states from $n \times (d + e)$ to $n \times d$.

$$\hat{h} = \text{Concat}( (h + \text{Conv1d}(f)), p)$$

4. Experiments

4.1. Experimental settings

We compare our presented method with the state-of-the-art FastSpeech 2 via both subjective and objective evaluations. Considering Lim et al. propose a jointly trained duration-informed Transformer (JDI-T)[11], which similarly converts the neural attention alignment from the autoregressive decoder into durations required by the non-autoregressive decoder, we also introduce this model into our experiments. In addition, we also conduct ablation studies
to validate each modification suggested in the proposed cooperative learning strategy. Table 1 summarises the detailed configuration of each non-autoregressive model in the experiments. We ensure that all experimental models have a similar number of parameters. Besides, these models are built on the basis of the ESPnet-TTS toolkit[25], which offers reproducible implementations of popular speech synthesis models.

We adopt the LJSpeech[26] and CSMSC[27] datasets to evaluate experimental methods. The LJSpeech dataset contains approximately 13,000 English audio clips recorded by a female speaker. However, its recording quality is not good enough. Each audio file in LJSpeech is a single-channel 16-bit PCM WAV with a sample rate of 22050 Hz. The CSMSC dataset contains 10,000 Chinese standard Mandarin audios also recorded by a professional female speaker. Each audio in CSMSC is a single-channel, 16-bit PCM WAV with a sample rate of 48000 Hz. We set the sampling rate of both datasets to 22050 Hz in the experiments. Each dataset is divided into three subsets: 250 samples for the test set, 250 samples for the validation set, and all the rest for the training set. The Mel spectrum is calculated through a short-time Fourier transformation (STFT) using a 50 ms frame size, 12.5 ms frame hop, a Hann window function, and 80 Mel scale filter banks, followed by a log transformation. Meanwhile, the G2PC[28] and G2P[29] open-source frontend tools are employed to convert input text to phonemes, respectively.
<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>FastSpeech 2</th>
<th>JDI-T</th>
<th>CLTTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phoneme Embedding Dimension</td>
<td>256</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Encoder Prenet Layers</td>
<td>N/A</td>
<td>3</td>
<td>N/A</td>
</tr>
<tr>
<td>Encoder Prenet Conv1d Kernel</td>
<td>N/A</td>
<td>512</td>
<td>N/A</td>
</tr>
<tr>
<td>Encoder Prenet Conv1d Filter Size</td>
<td>N/A</td>
<td>5</td>
<td>N/A</td>
</tr>
<tr>
<td>Encoder Layers</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Encoder Hidden</td>
<td>256</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Encoder Conv1d Kernel</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Encoder Conv1d Filter Size</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Encoder Attention Heads</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Decoder Layers</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Decoder Hidden</td>
<td>256</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Decoder Conv1d Kernel</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Decoder Conv1d Filter Size</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Decoder Attention Heads</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Encoder/Decoder Dropout</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Duration/Pitch/Energy Predictor Conv1d Kernel</td>
<td>256</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Duration/Pitch/Energy Predictor Conv1d Filter Size</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Duration/Pitch/Energy Predictor Dropout</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Prosody Embedding Dimension</td>
<td>N/A</td>
<td>N/A</td>
<td>128</td>
</tr>
<tr>
<td>Prosody Embedding Predictor GRU Hidden</td>
<td>N/A</td>
<td>N/A</td>
<td>256</td>
</tr>
</tbody>
</table>
The Adam optimizer\cite{kingma2014adam} is used for the training of all models with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. The warm-up learning rate strategy suggested in \cite{he2016deep} is also applied. The learning rate is set to 0.001, and the warm-up step is set to 4000. We train the experimental models for 100K steps with a batch size of 32 on 2 NVIDIA RTX 2080Ti GPUs. The WaveGlow\cite{vahdat2019waveglow} vocoder released by NVIDIA is employed to reconstruct audio samples from the generated spectrum. Since this model is trained on LJSpeech, we fine-tuned it on CSMSC for Mandarin TTS experiments.

4.2. Results

We use Mel cepstrum distortion (MCD) and mean opinion score (MOS) to evaluate the performance of experimental models. The MCD represents the Euclidean distance between spectra from the ground truth audio sample and the synthesised audio sample. However, lower MCD values do not guarantee subjectively better performance. Thus, the MOS evaluation, where each audio sample is scored by listeners on a five-point scale, is taken as a more effective indicator of model performance. Table 2 demonstrates the experimental results of different models on the LJSpeech dataset. It can be observed that models without introducing any prosodic information, i.e., JDI-T and CLTTS, have much lower MCD values. Particularly, the CLTTS model may achieve the lowest MCD value. Nevertheless, models without prosodic information lead to poor performance on MOS tests.

For subjective evaluation, our proposed model with the basic cooperative learning strategy outperforms the JDI-T model significantly, with a gap of 0.44. When the enhanced cooperative learning method is used, the MOS of our CLTTS can be further improved. In particular, our model achieves a very comparable performance to FastSpeech 2, with only a gap of 0.04 after the pitch information is introduced. It is notable that when the prosody embedding is further introduced, the MOS value of CLTTS decreases to some extent. We conjecture that the limited recording quality of LJSpeech hinders the cooperative autoregressive model from learning an effective prosody embedding. Therefore, we further conduct experiments on the CSMSC dataset, which contains high-quality and expressive audio samples.

Table 3 shows that the MCD evaluation on CSMSC presents consistent results with the tests on LJSpeech. For the MOS evaluation, our non-autoregressive TTS model with the basic cooperative learning method still precedes the JDI-T model with a gap of 0.13. Besides, the CLTTS model with pitch prediction is able to achieve the same performance as FastSpeech.
2. Once the prediction of prosody embedding is introduced into the CLTTS model, its performance can be further enhanced and even achieve higher MOS values than FastSpeech 2. This result suggests that the cooperative autoregressive TTS model can learn available prosody embedding on the CSMSC dataset, which may be an effective complement to the pitch information. In the meantime, our approach is easier to implement, and it does not rely on any external alignment tools.

Table 2: Results on the LJSpeech dataset (with 95% confidence intervals).

<table>
<thead>
<tr>
<th>Method</th>
<th>MOS</th>
<th>MCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth</td>
<td>4.49±0.05</td>
<td>N/A</td>
</tr>
<tr>
<td>FastSpeech 2</td>
<td>4.21±0.06</td>
<td>5.81±0.06</td>
</tr>
<tr>
<td>JDI-T</td>
<td>2.95±0.07</td>
<td>5.67±0.06</td>
</tr>
<tr>
<td>CLTTS</td>
<td>3.39±0.07</td>
<td>5.60±0.06</td>
</tr>
<tr>
<td>CLTTS (+F0)</td>
<td>4.17±0.06</td>
<td>5.78±0.06</td>
</tr>
<tr>
<td>CLTTS (+F0, PE)</td>
<td>4.09±0.06</td>
<td>5.76±0.06</td>
</tr>
</tbody>
</table>

Table 3: Results on the CSMSC dataset (with 95% confidence intervals).

<table>
<thead>
<tr>
<th>Method</th>
<th>MOS</th>
<th>MCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth</td>
<td>4.46±0.05</td>
<td>N/A</td>
</tr>
<tr>
<td>FastSpeech 2</td>
<td>4.08±0.07</td>
<td>5.54±0.06</td>
</tr>
<tr>
<td>JDI-T</td>
<td>3.75±0.06</td>
<td>5.20±0.06</td>
</tr>
<tr>
<td>CLTTS</td>
<td>3.88±0.06</td>
<td>5.19±0.06</td>
</tr>
<tr>
<td>CLTTS (+F0)</td>
<td>4.09±0.06</td>
<td>5.27±0.06</td>
</tr>
<tr>
<td>CLTTS (+F0, PE)</td>
<td>4.11±0.06</td>
<td>5.26±0.06</td>
</tr>
</tbody>
</table>

4.3. Case Studies

We conduct in-depth case studies on the synthesised spectrum from experimental models to investigate our method further. The spectra produced by the plain CLTTS and the JDI-T models are first compared, as shown in Figures 11 and 12. We can observe that the spectrum from our CLTTS model may present more details than the JDI-T model. Although both CLTTS and
JDI-T show promising results when trained on the CSMSC dataset, the spectrum from the CLTTS model is more distinct in the high-frequency domain.

Figure 11: The spectrum produced by JDI-T (A) and CLTTS (B) on CSMSC

Figure 12: The spectrum produced by JDI-T (A) and CLTTS (B) on LJSpeech

We further compare the spectrum from the CLTTS before and after the injection of pitch information, respectively. As shown in Figures 13 and 14, predicting pitch helps the CLTTS model produce a more expressive spectrum, which can be clearly observed when the models are trained on the LJSpeech dataset. Although both CLTTS and CLTTS (F0) perform quite well for the CSMSC dataset, CLTTS (F0) still presents more expressive details, as shown in the red rectangles in Figure 13.
Figure 13: The spectrum produced by CLTTS (A) and CLTTS (+F0) (B) on CSMSC

Figure 14: The spectrum produced by CLTTS (A) and CLTTS (+F0) (B) on LJSpeech
Finally, in order to explore how the prosody embedding learned by the cooperative autoregressive TTS model enhances the performance of non-autoregressive models, the case study on the pitch curve of speech from CLTTS (F0) and CLTTS (F0, PE) is compared. As can be seen in Figure 15, the pitch variation of the speech from the CLTTS (F0, PE) model is closer to the ground truth than the CLTTS (F0) model, indicating that the prosody embedding can help the non-autoregressive TTS model learn a more similar rhythm to the target speech.

5. Conclusion

This paper suggests a cooperative learning strategy to address the two most challenging problems in non-autoregressive speech synthesis. With the proposed method, our non-autoregressive TTS model can obtain the durations of the input polyphones from its cooperative autoregressive TTS model, thus eliminating the dependency on an external alignment toolkit. At the same time, the presented non-autoregressive TTS model can achieve comparable performance to FastSpeech 2 by introducing pitch and prosody embedding prediction into the model. In addition, the proposed cooperative
learning strategy can also be integrated with other types of generative networks, such as generative adversarial networks and diffusion networks, which remain for our future study.

References


