Neural Speech Synthesis with Transformer Network

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu

University of Electronic Science and Technology of China

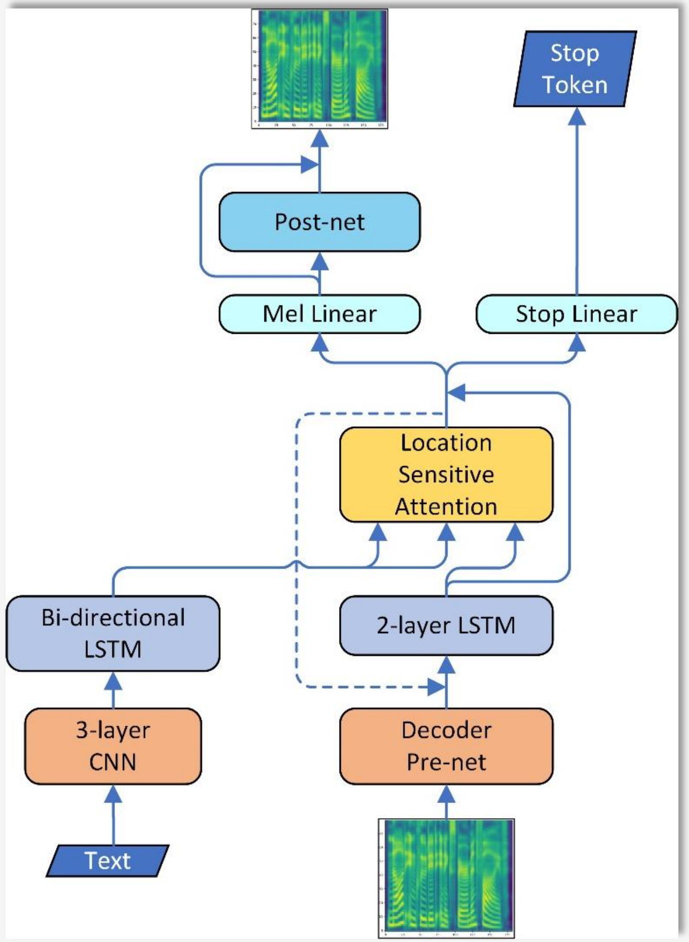
Microsoft Research Asia

Microsoft STC Asia

Tacotron2

A neural network architecture for speech synthesis directly from text

- **3-layer CNN**: extracts a longer-term text context.
- **Bi-directional LSTM**: encoder.
- **Location sensitive attention**: connects encoder and decoder.
- **Decoder pre-net**: a 2-layer fully connected network.
- **2-layer LSTM**: decoder.
- **Mel linear**: a fully-connected layer, generates mel spectrogram frames.
- **Stop linear**: a fully-connected layer, predicts the stop token for each frame.
- **Post-net**: a 5-layer CNN with residual connections, refines the mel spectrogram.



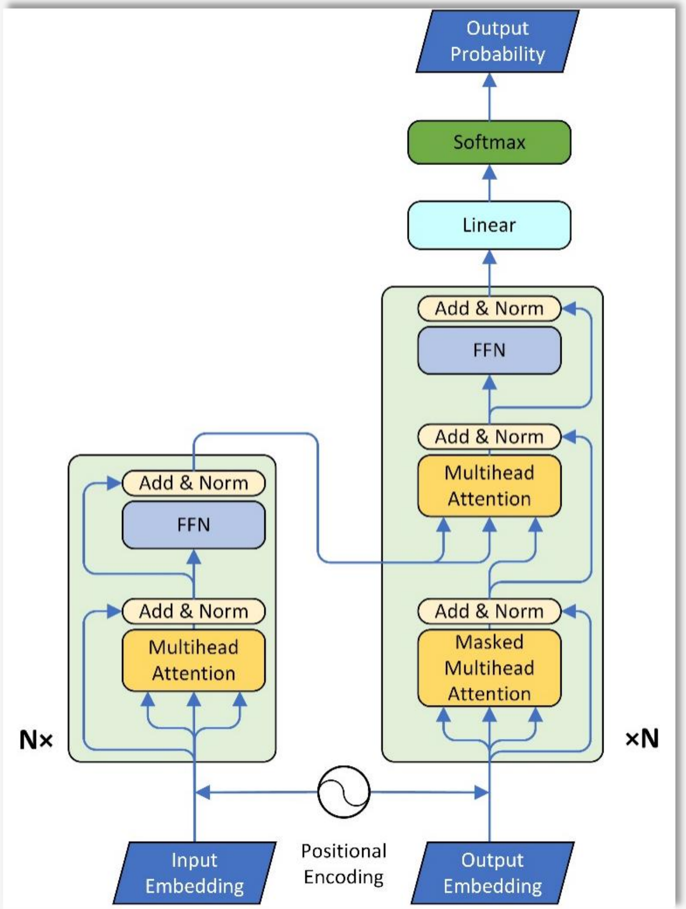
Transformer

A sequence to sequence network based solely on attention mechanisms

- **Encoder:** 6 blocks.
- **Decoder**: 6 blocks.
- **Positional embeddings**: add positional information (PE) to input embeddings

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$
$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

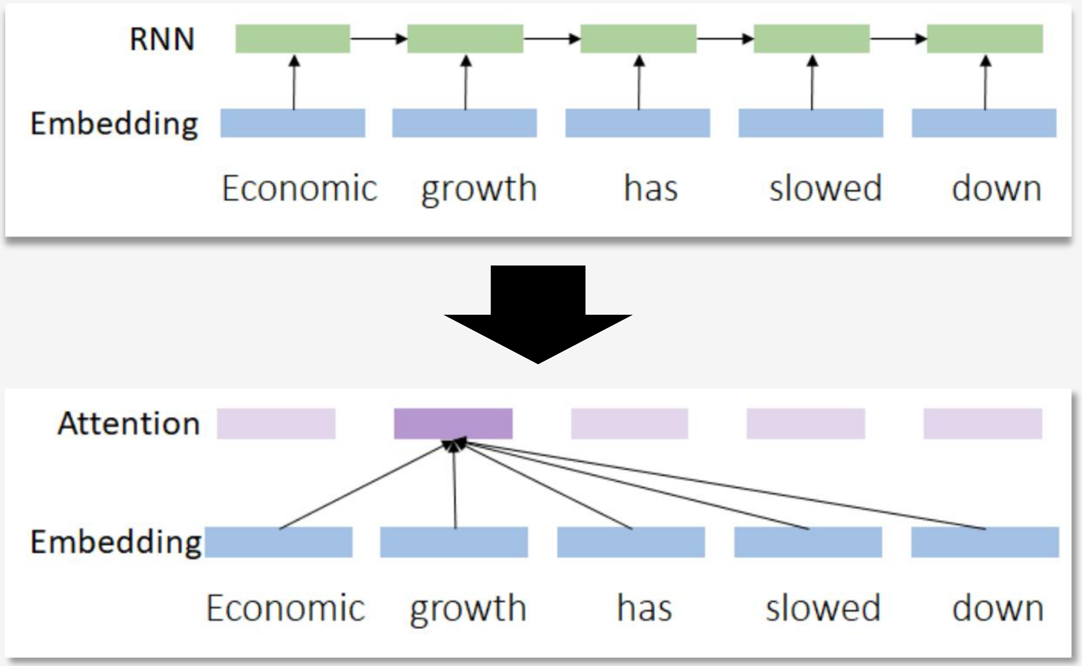
- **(Masked) Multi-head attention**:
  - Splits each Q, K and V into 8 heads
  - Calculates attention contexts respectively
  - Concatenates 8 context vectors
- **FFN**: feed forward network, 2 fully connected layers.
- **Add & Norm**: residual connection and layer normalization.



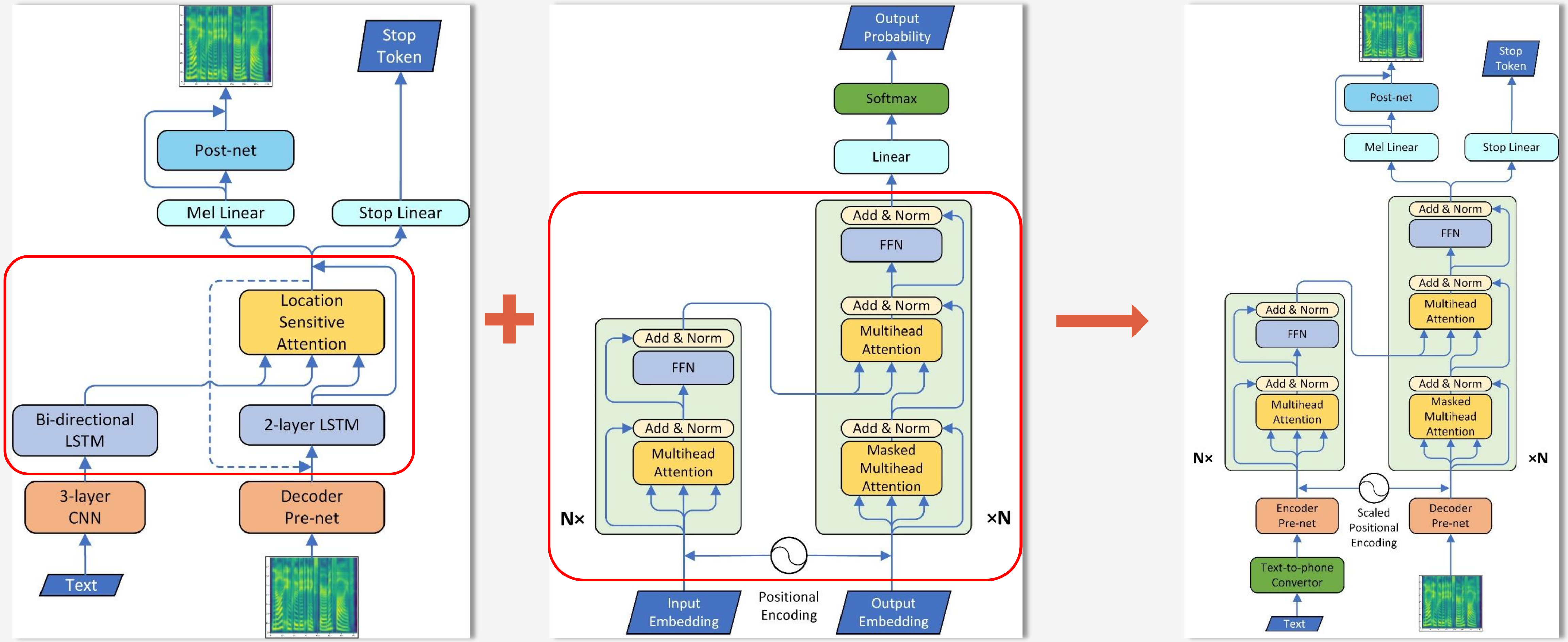
Neural TTS with Transformer

Why apply Transformer in TTS

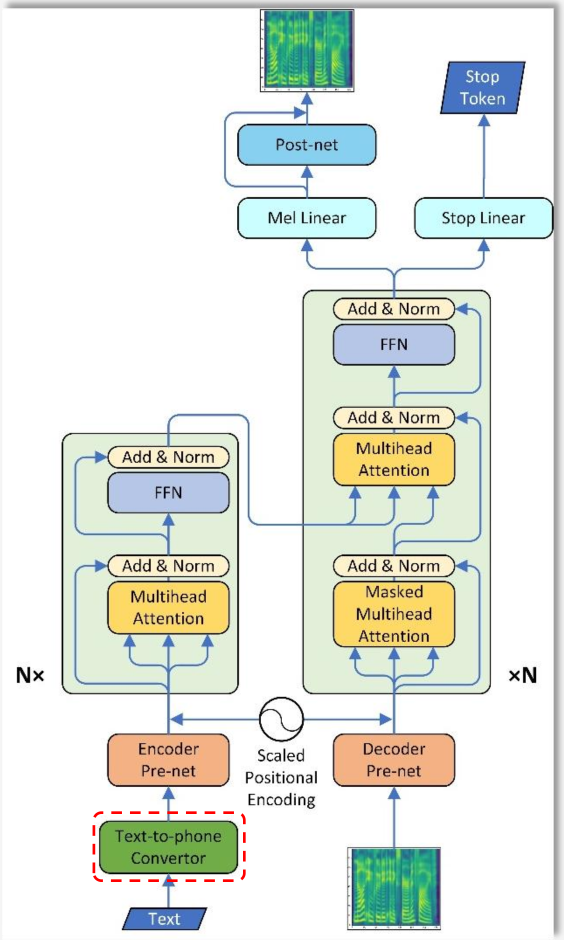
- **Parallel training**
  Frames of an input sequence can be provided in parallel.
- **Long range dependencies**
  Self attention injects global context of the whole sequence into each input frame.



Neural TTS with Transformer

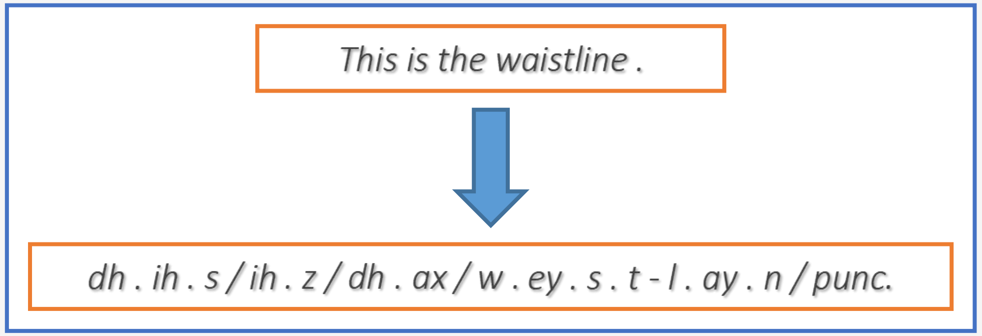


Neural TTS with Transformer

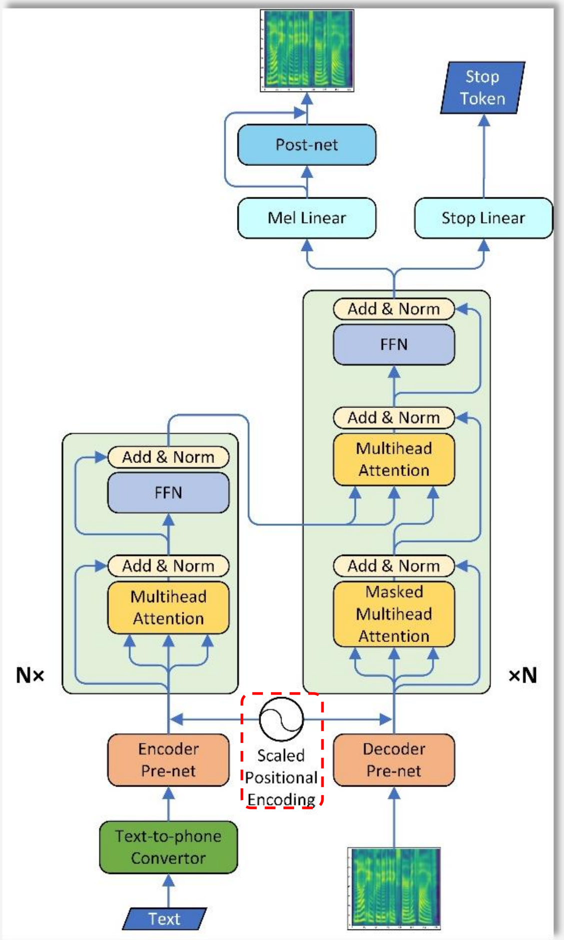


Text-to-Phoneme Converter

- Difficult to learn all the regularities without sufficient training data
- Some exceptions have too few occurrences for neural networks to learn
- Convert text into phonemes by rule:



Neural TTS with Transformer



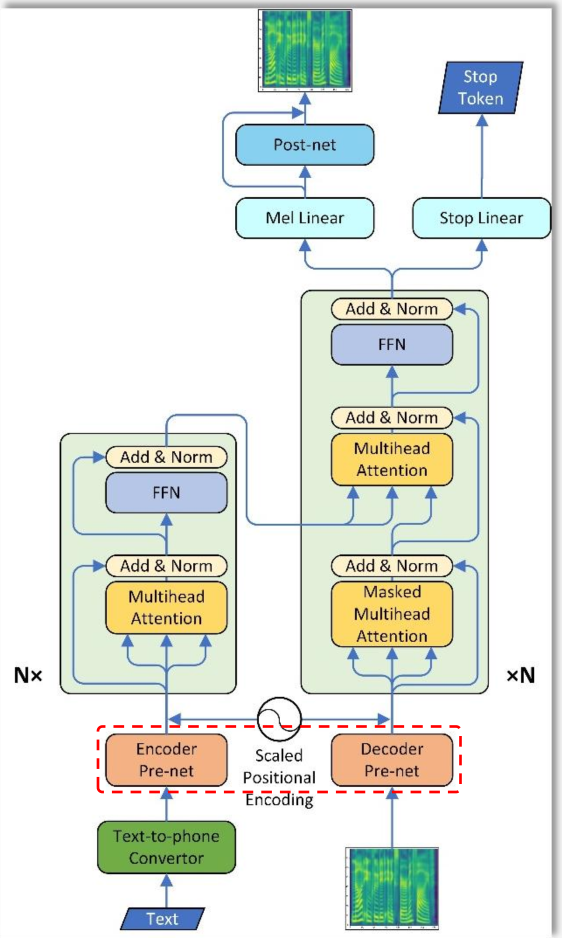
Scaled Positional Encoding

- Transformer adds information about the relative or absolute position by adding a positional embedding (PE) to input embeddings
- In TTS scenario, texts and mel spectrograms may have different scales
  - Scale-fixed positional embeddings may impose heavy constraints on both the encoder and decoder pre-nets
- Add a trainable weight to positional embeddings

$$x_i = prenet(phoneme_i) + \alpha \cdot PE(i)$$

  - Positional embeddings can adaptively fit the scales of both encoder and decoder pre-nets' output

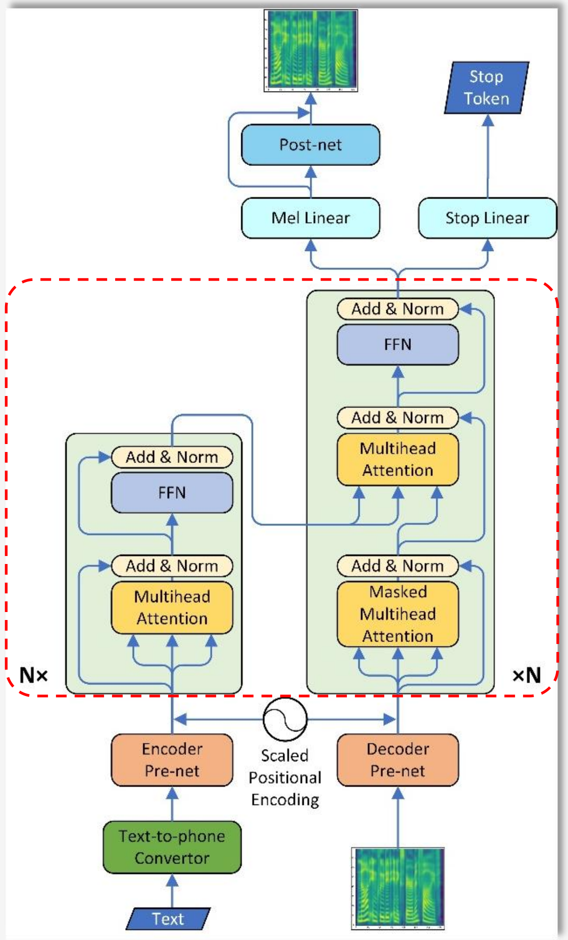
Neural TTS with Transformer



Pre-nets of Encoder and decoder

- Similar structure and function as in Tacotron2
- An additional linear projection is appended
  - Positional embeddings are in $[-1, 1]$
  - After *relu*, the outputs of pre-nets are in $[0, +\infty)$
  - Re-center to have a 0-centered range

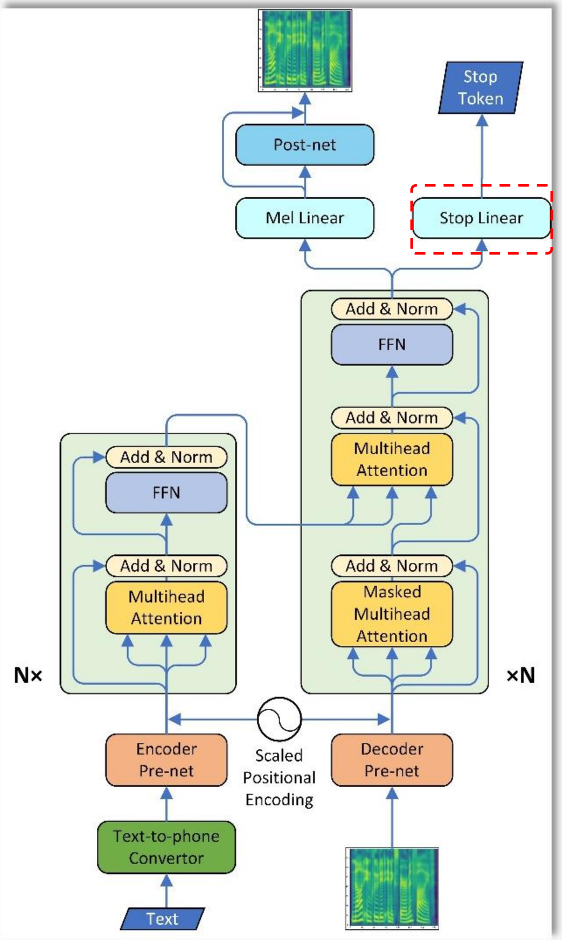
Neural TTS with Transformer



Encoder and decoder

- Model the frame relationship in multiple different aspects.
- Inject the global context of the whole sequence into each frames.
- Enable parallel computing to improve training speed.

Neural TTS with Transformer



Stop linear

- Predicts whether should inference stop at current frame.
- Unstoppable inference may occur
  - During training, each sequence has only one "stop" while hundreds of “continue”.
  - Imbalance positive/negative samples result in biased stop linear.
  - Solution: impose a weight (5.0 ~ 8.0) on the "stop" token when calculating binary cross entropy loss during training.

Experiment

Training Setup

- 4 Nvidia Tesla P100
- Internal US English female dataset
  - 25-hour professional speech
  - 17584 text-wave pairs
- Dynamic batch size
  - Various sample number
  - Fixed mel spectrogram frame number

Text-to-Phoneme Conversion and Pre-process

- Phoneme type:
  - Normal phonemes
  - Word boundaries
  - Syllable boundaries
  - Punctuations
- Process pipeline:
  - Sentence separation
  - Text normalization
  - Word segmentation
  - Obtaining pronunciation

WaveNet Settings

- Sample rate: 16000
- Frame rate (frames per second): 80
- 2 QRNN layers
- 20 dilated layers
- Residual and dilation channel size: 256

Experiment

Training Time Comparison

	Tacotron2	Transformer
Single step (batch size=~16)	~ 1.7s	~ 0.4s
Total time	~4.5 days	~3 days

Inference Time Comparison

	Tacotron2	Transformer
Synthesize 1s spectrogram	~ 0.13s	~ 0.36s

Evaluation

Evaluation setup

- 38 fixed examples with various lengths
- Each audio is listened to by at least 20 testers (8 testers Shen et al. (2017))
- Each tester listens less than 30 audios

Baseline model

- Tacotron2
  - Use phone sequence as inputs
  - Other structure are same as Google's version

Results

System	MOS	CMOS
Tacotron2	4.39 ± 0.05	0
Our Model	4.39 ± 0.05	**0.048**
Ground Truth	4.44 ± 0.05	-

CMOS: comparison mean option score. Testers listen to two audios each time and evaluates how the latter feels comparing to the former using a score in [−3, 3] with intervals of 1

Evaluation

Generated sample comparison

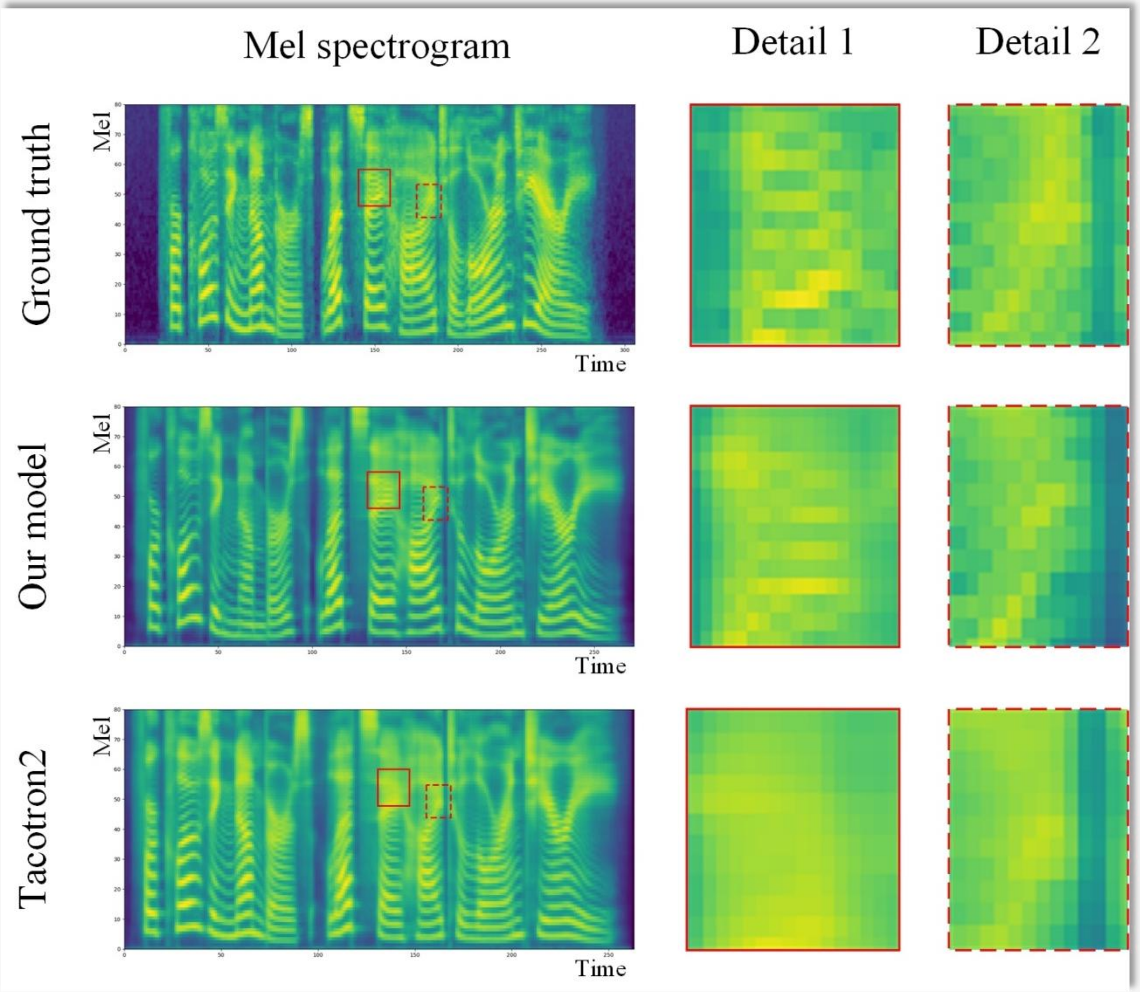
Our model			
Tacotron2			

More samples at https://neuraltts.github.io/transformertts/

Evaluation

Mel spectrogram details

- Our model does better in reproducing high frequency region



Ablation Studies

Re-centering Pre-net’s Output

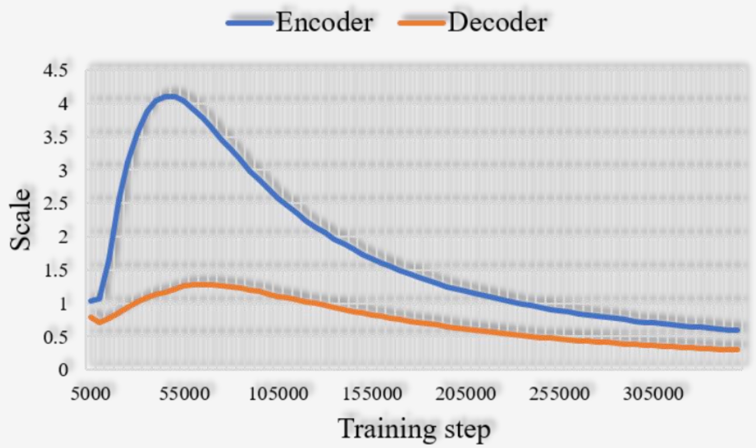
- Re-project pre-nets’ outputs for consistent center with positional embeddings
- Center-consistent PE performs slightly better

Re-projection	MOS
No	4.32 ± 0.05
Yes	$\mathbf{4.36} \pm 0.05$
Ground Truth	4.43 ± 0.05

Ablation Studies

Different Positional Encoding Methods

- Final positional embedding scales of encoder and decoder are different
- Trainable scale performs slightly better.
- Reason:
  - Constraint on encoder and decoder pre-nets are relaxes
  - Positional information are more adaptive for different embedding spaces

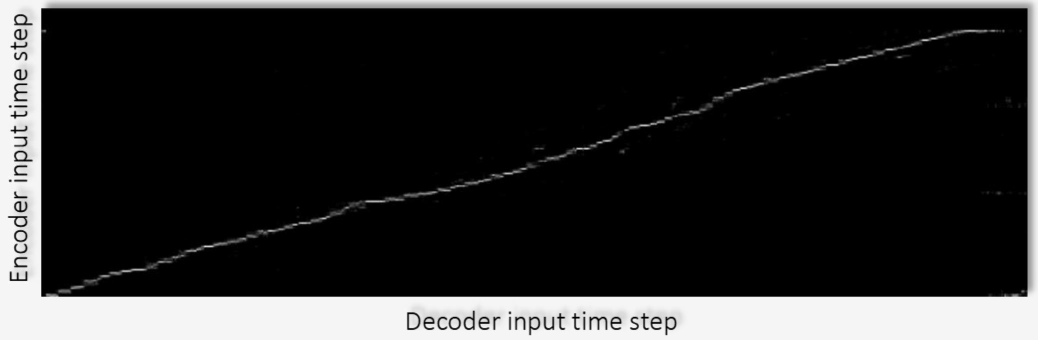


PE Type	MOS
Original	4.37 ± 0.05
Scaled	$\mathbf{4.40} \pm 0.05$
Ground Truth	4.41 ± 0.04

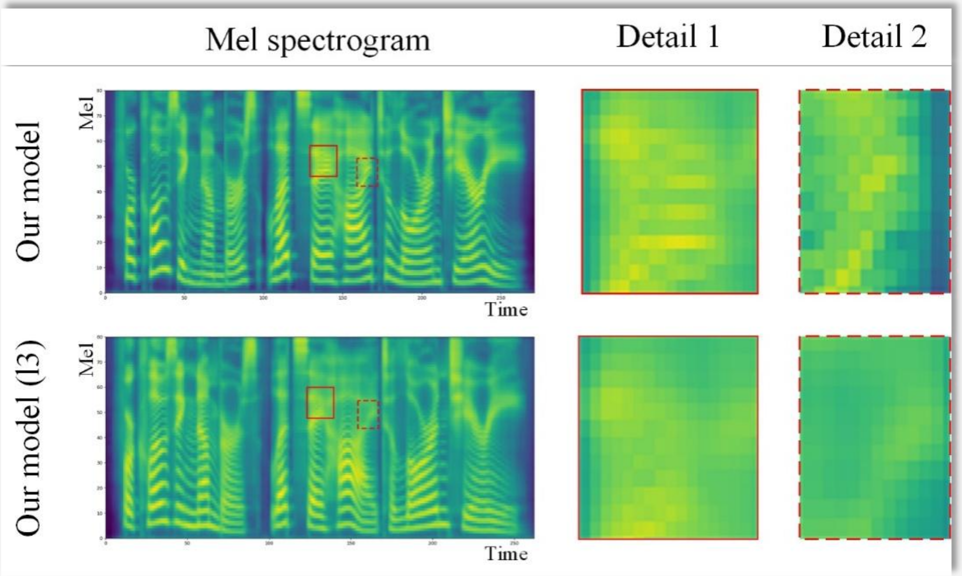
Ablation Studies

**Different Hyper-Parameter:
layer number impact**

- For encoder-decoder attention, only alignments from certain heads of the beginning 2 layers' are interpretable diagonal lines



- More layers can still refine the synthesized mel spectrogram and improve audio quality



Ablation Studies

Different Hyper-Parameter:
head number impact

- Reducing head numbers harms performance

Head Number	MOS
4-head	4.39 ± 0.05
8-head	$\mathbf{4.44} \pm 0.05$
Ground Truth	4.47 ± 0.05

- Comparison of time consuming (in second) per training step of different layer and head numbers

	3-layer	6-layer
4-head	-	0.44
8-head	0.29	0.50

(Tested on 4 GPUs with dynamic batch size)

Thank you!