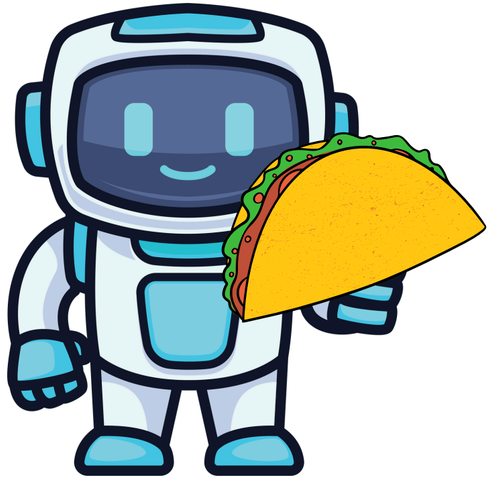
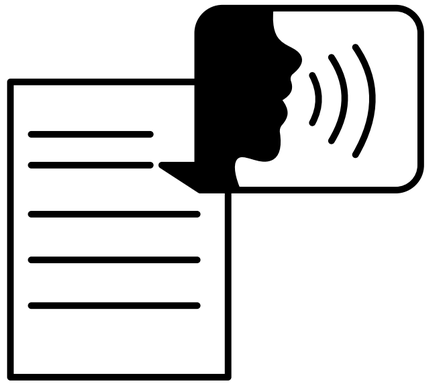
TACOTRON

TOWARDS END-TO-END
SPEECH SYNTHESIS





Released: March 2017

INTRODUCTION

Tacotron is an **end-to-end text-to-speech (TTS)** model that synthesizes speech directly from text using a **sequence-to-sequence framework with attention.**

Unlike traditional TTS pipelines, Tacotron **eliminates** the need for **domain-specific modules** like linguistic feature extraction, duration prediction, and vocoders.

DOMAIN AND TASK

- **Google** developed Tacotron to enhance the naturalness and intelligibility of AI-generated speech, particularly for Google Assistant

- **Tacotron** aimed to produce human-like speech.

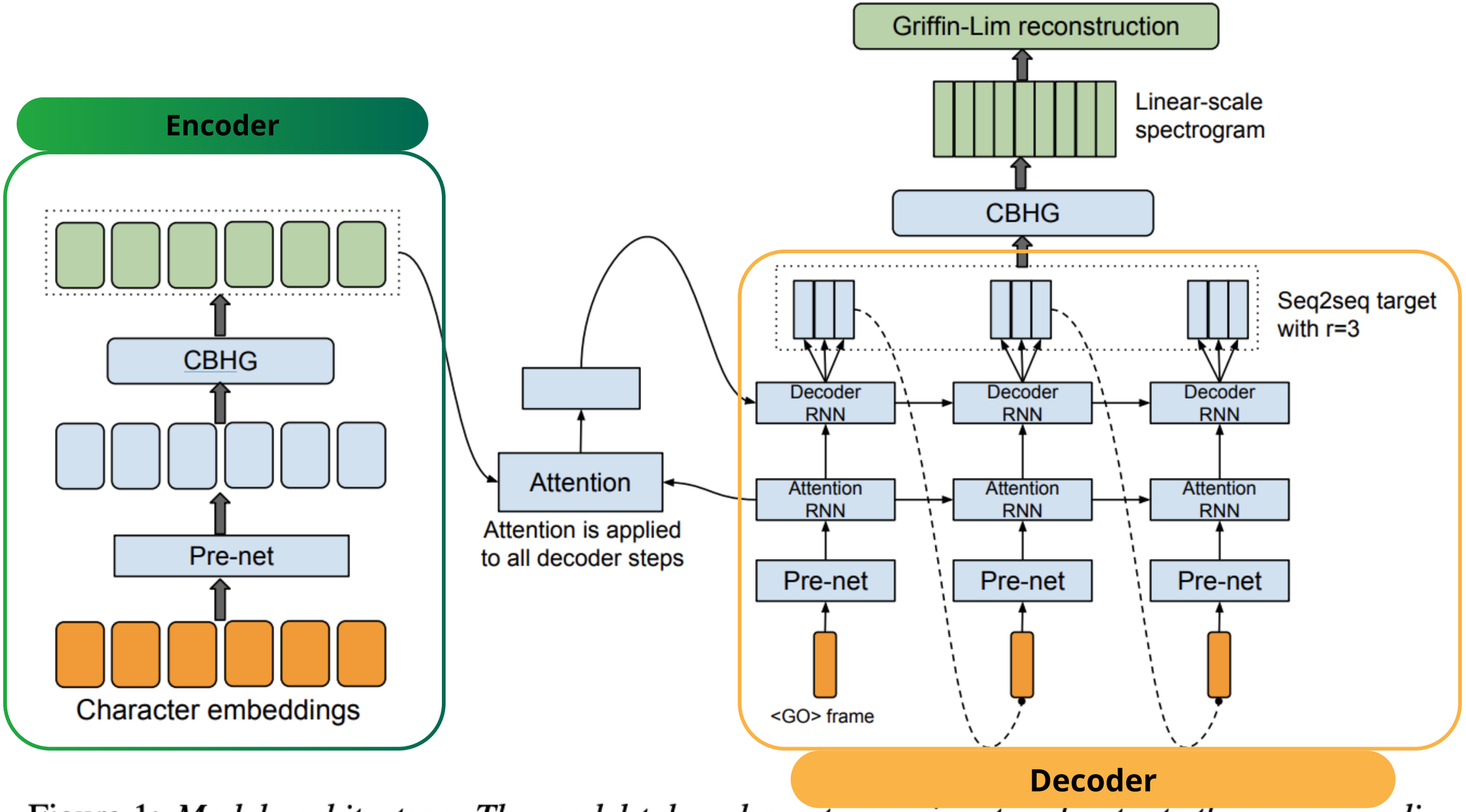


Figure 1: *Model architecture. The model takes characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesize speech.*

ENCODER

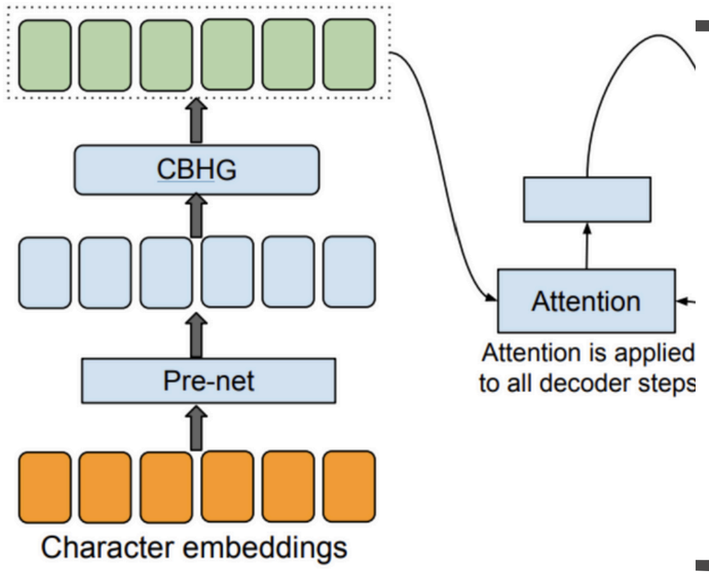
processes the input character sequence by converting it into **one-hot** encodings and then **mapping it to trainable 256D embeddings.**

→ **Pre-net** with **2 fully connected layers** (256 → 128D) using ReLU activation and **Dropout (0.5)** for regularization, **reducing dimensions** and stabilizing input for alignment learning.

→ The embeddings are then refined by the **CBHG Module**, which extracts **sequential patterns and hierarchical features.**

The **output** is a high-level encoded representation of size [Batch, Seq Len, 256], **sent to the attention** mechanism for alignment with the **decoder**.

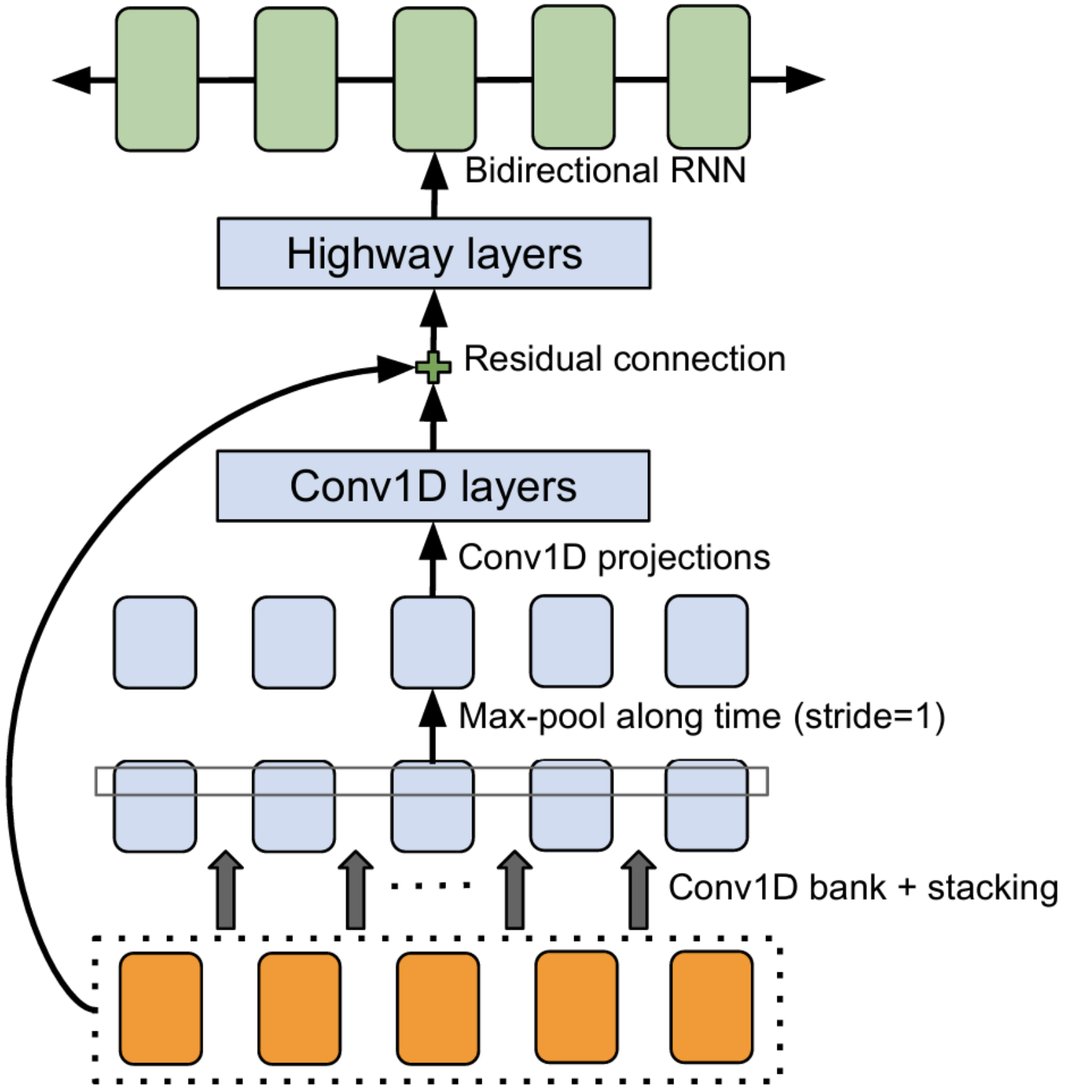
Advantages: **reduced overfitting**, improved **generalization** to unseen text, and enhanced pronunciation **accuracy**.



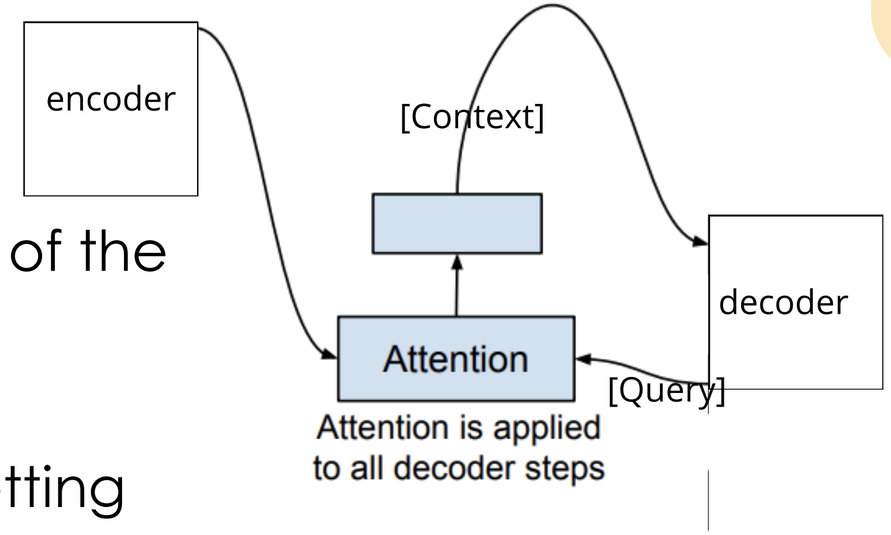
CBHG

CBHG (Convolutional Bank + Highway + GRU)
extracts hierarchical and sequential features.
CBHG in **Encoder** improving attention alignment.

- **Conv1D Bank:** captures local patterns at multiple timescales using **filters of varying widths**.
- **Max-Pooling:** reduces noise and enhances important temporal features.
- **Highway Networks:** adaptive gates, filtering features.
- **Bidirectional GRU** processes the sequence in both **forward and backward** directions, for **global context** is retained.
- **Residual connections:** prevent gradient vanishing and **preserve information** from earlier layers



BAHDANAU'S ATTENTION



- Allows the model to focus on specific parts of the text while generating audio frames.

- The attention mechanism responsible for letting the decoder knows **which parts of the encoder output to focus** on **when predicting the next mel-scale Spectrogram Frame**.

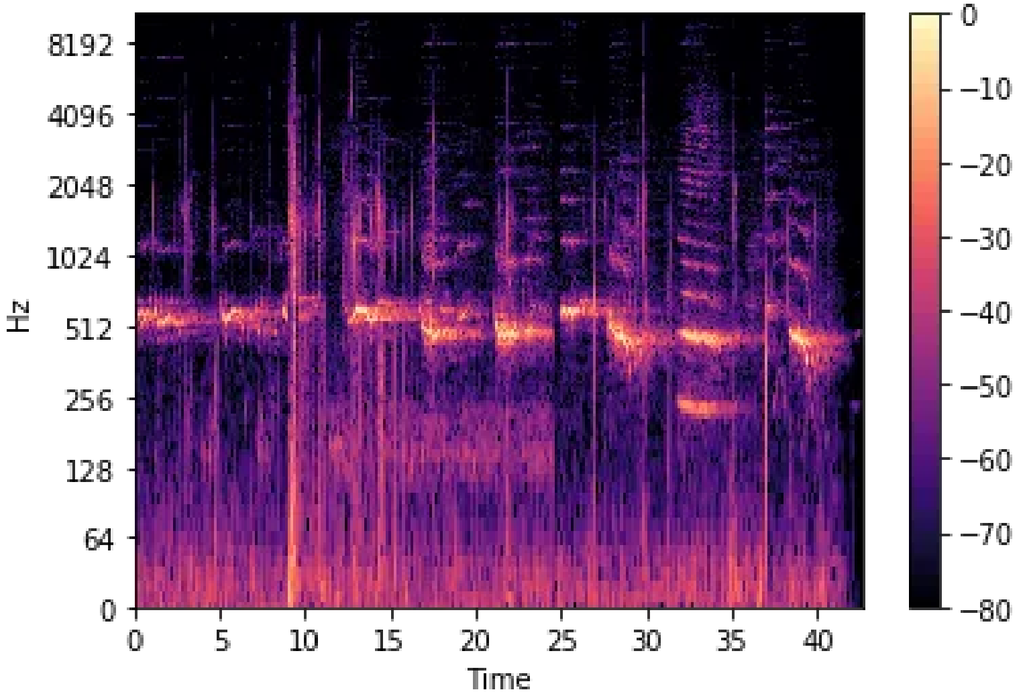
Main Difference

Bahdanau's attention is sequential, processing step-by-step, while **Self-Attention** is fully parallelizable, allowing faster computations and allowing context from both past and future.

WHAT?

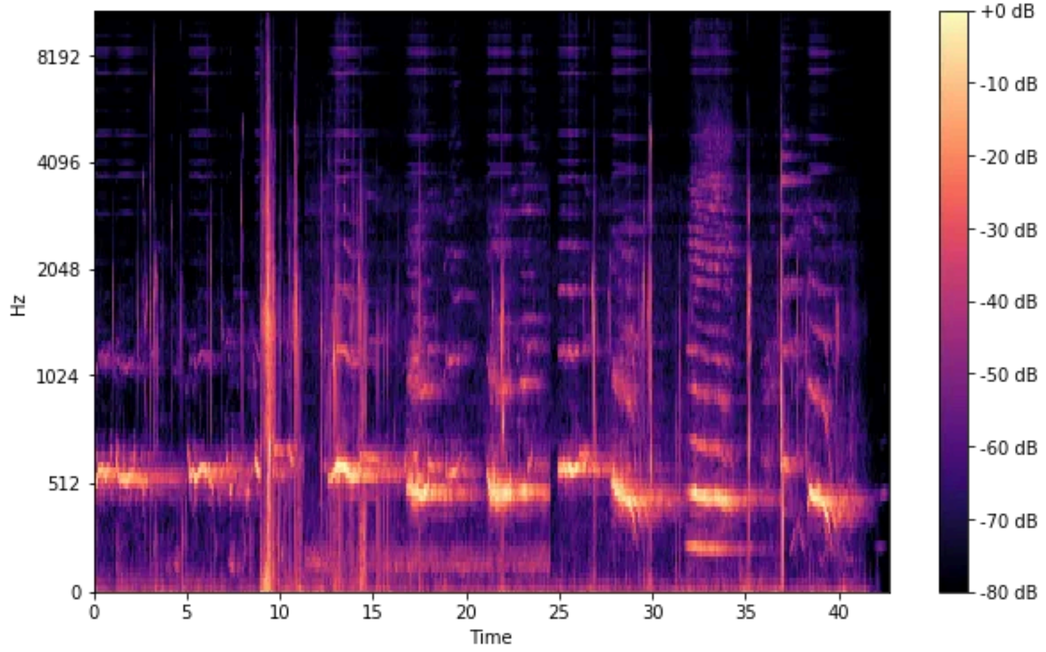
Spectrogram

visual representation of the spectrum of frequencies of a signal as it varies with time



The Mel Scale

is the result of some non-linear transformation of the frequency scale. Mel Scale is constructed such that sounds of equal distance from each other on the Mel Scale, also “sound” to humans as they are equal in distance from one another.



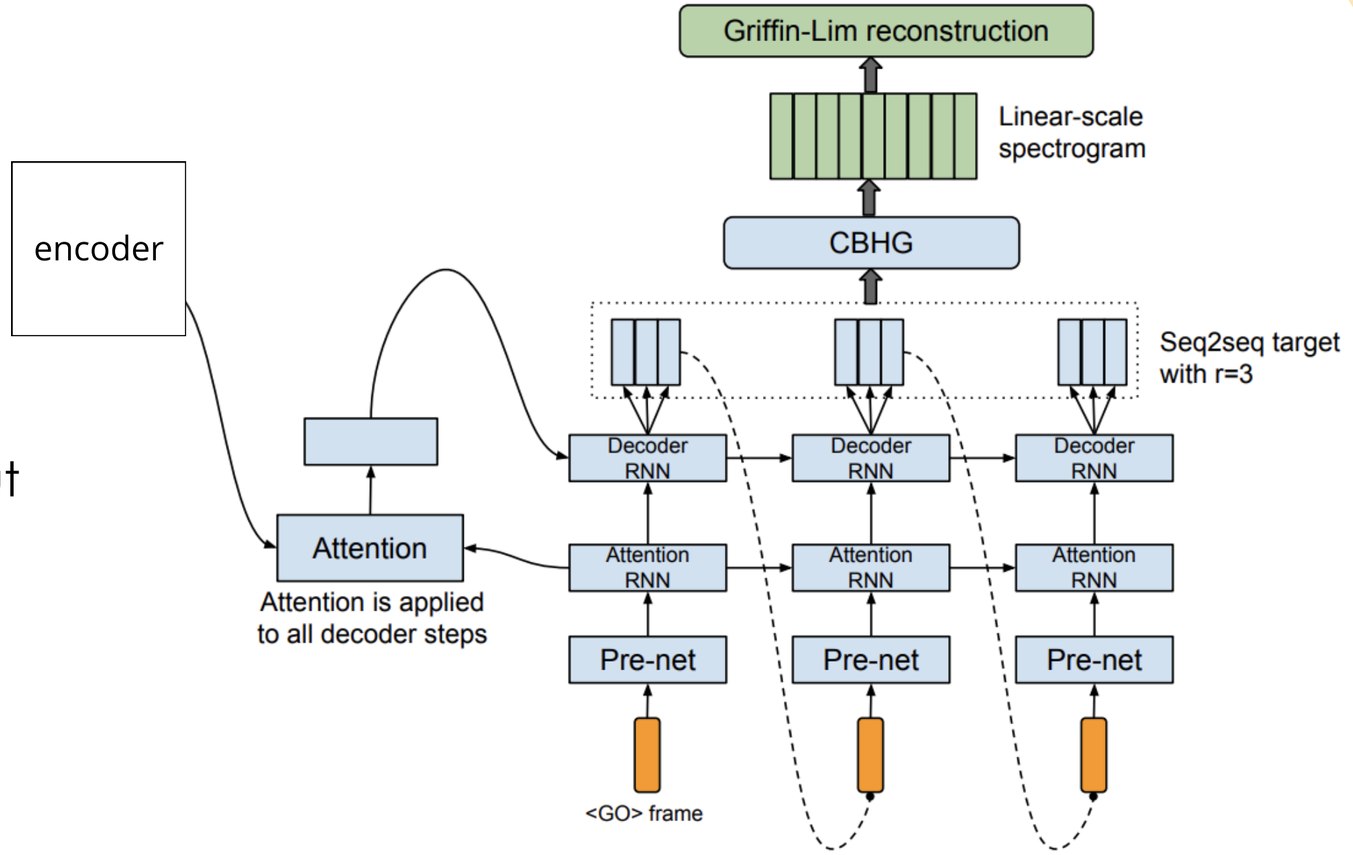
We know now what is a Spectrogram, and also what is the Mel Scale, so the Mel Spectrogram, is, rather surprisingly, a Spectrogram with the Mel Scale as its y axis.

DECODER

The decoder predicts **mel-spectrogram frames**

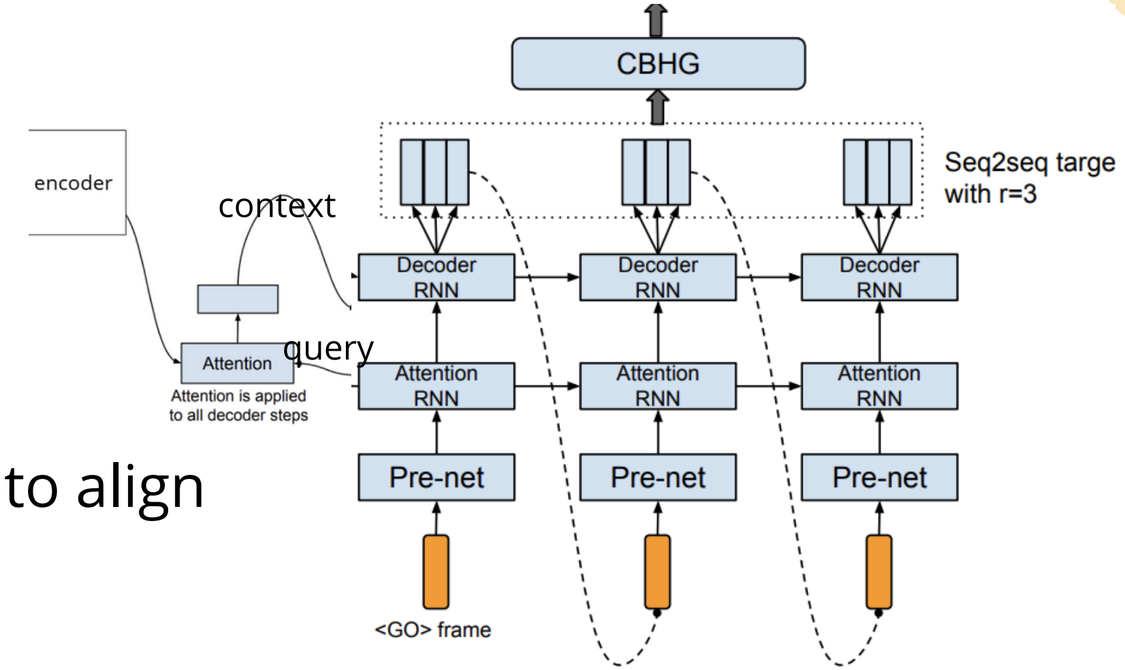
It takes the **context vector** (output from the attention module)

and uses it to generate spectrogram frames.



DECODER

- **Decoder Pre-net –** Feed Forward Net, Processes previous output, improves convergence
- **Attention RNN –** Creates query vector to align with encoder outputs
- **Decoder RNN –** Combines context and attention output for final step



<GO> Frame Initialization:
all-zero frame

ATTENTION & DECODER

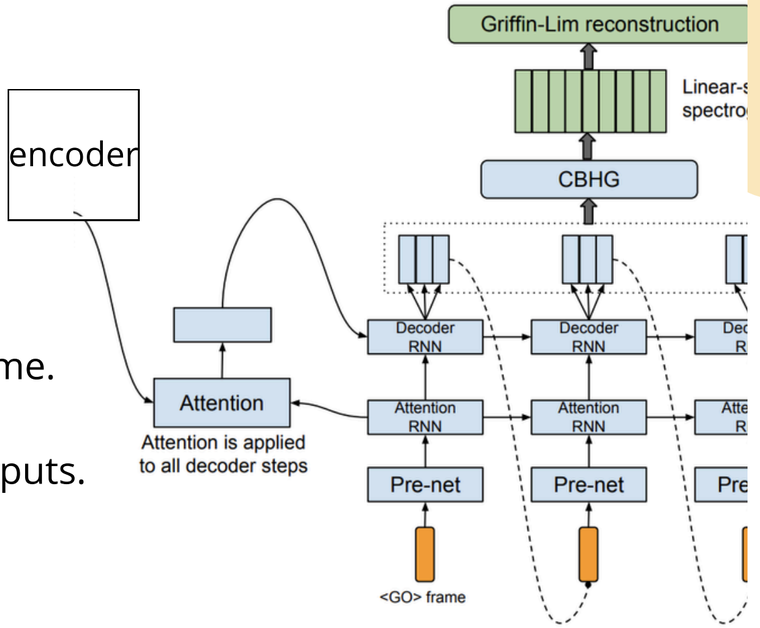
Input to Attention:
From encoder - a sequence of feature vectors.
Focus on Relevant Text:
At every decoder step, the attention mechanism determines which parts of the encoder's output are most relevant for generating the current spectrogram frame.

It computes a context vector, which is a weighted average of the encoder outputs. The weights are determined by the attention scores.

Output of Attention:
The **context vector** is passed to the **decoder RNN**, helping it generate the next mel-spectrogram frame in alignment with the input text.

At every decoder step, the **attention RNN (in decoder)** generates a **query vector**, which tells the attention mechanism what the decoder currently needs from the encoder outputs.
"What part of the input text should I focus on right now?"

The attention mechanism computes a context vector and sent it **back to the decoder**, helping it produce the next spectrogram frame.



POST-PROCESSING NET→ WAVEFORM SYNTHESIS

Primary Task: Convert the mel-spectrogram (produced by the decoder) into a **linear-scale spectrogram**, which is required for **waveform synthesis**

Architecture Used:

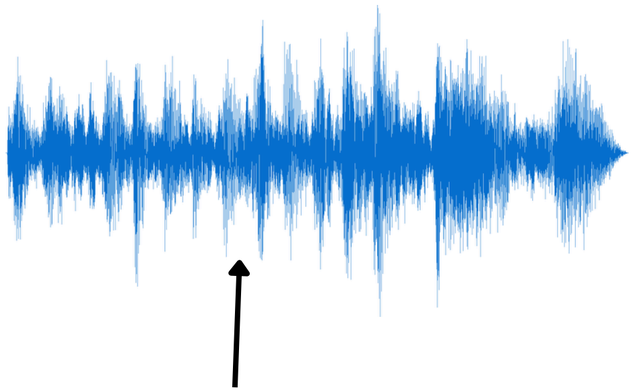
CBHG module, similar to the one in the encoder, but adapted to handle spectrogram data.

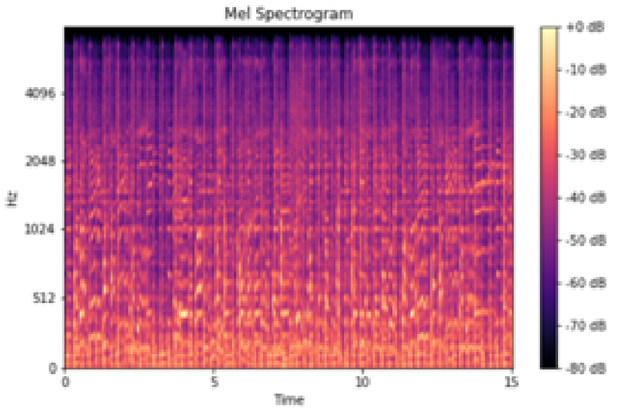
- Final Output Shape: [Batch Size, Time Steps, **Linear Spectrogram** Dim (1025)].

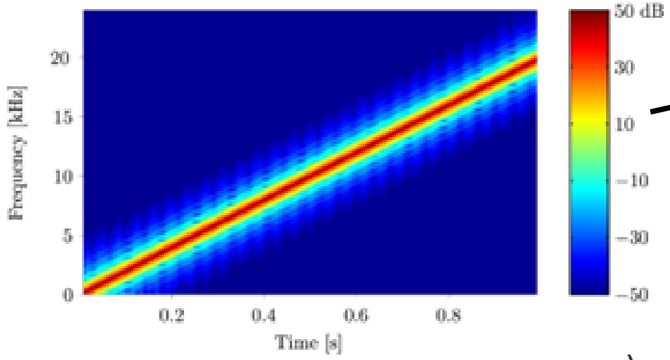
Griffin-Lim for Waveform Synthesis

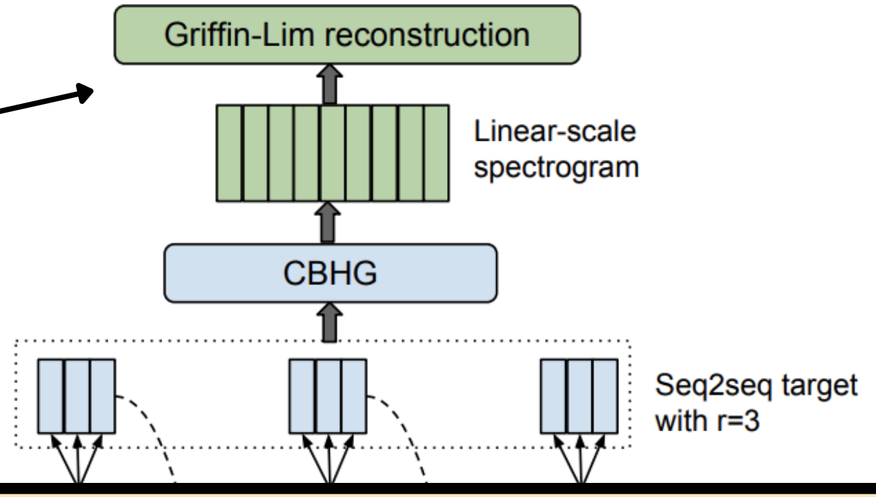
The **linear-scale spectrogram** produced by the post-processing net is **converted into a raw audio waveform**

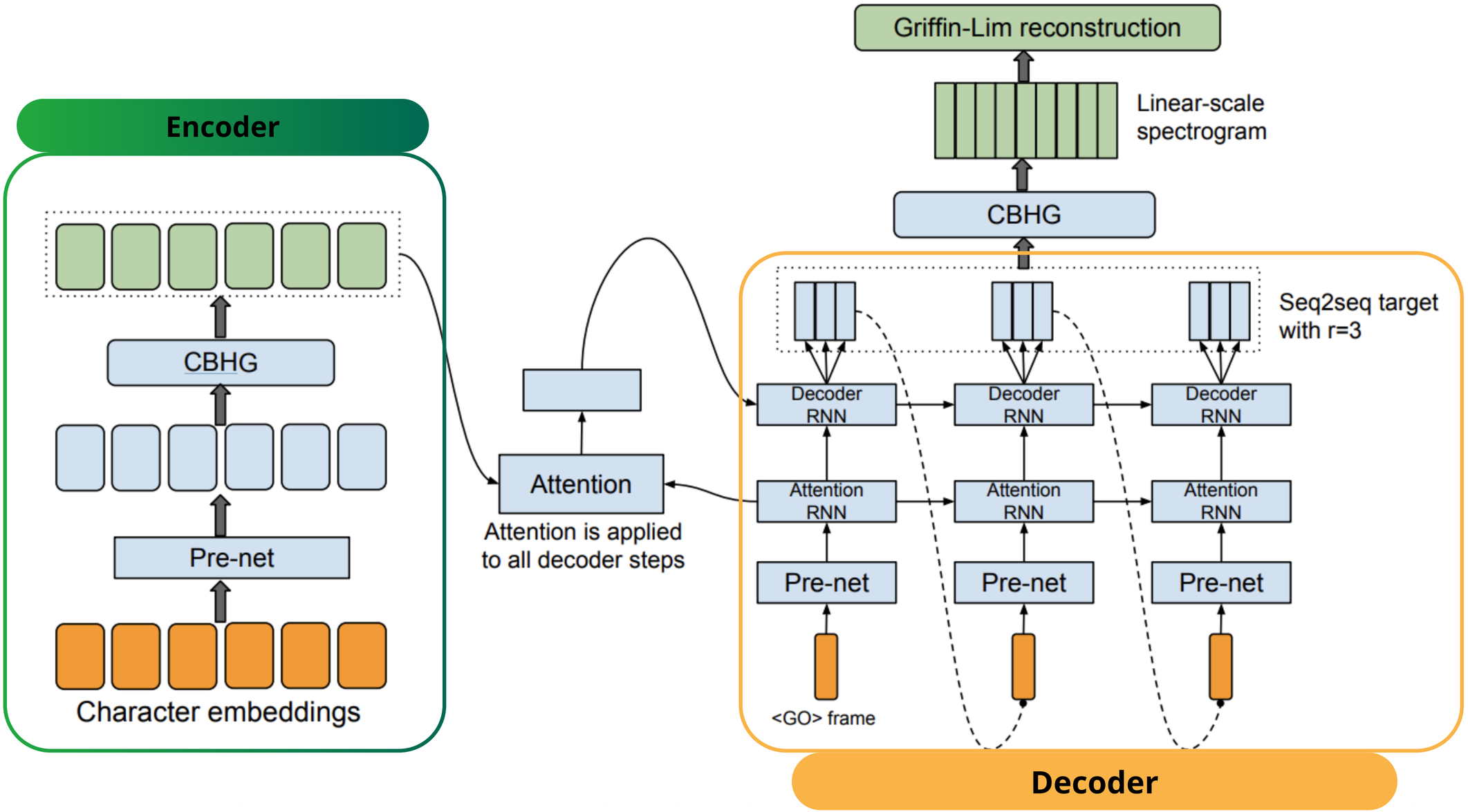
Why Used: It is simple, efficient, and sufficient for Tacotron's needs, avoiding the complexity of neural vocoders.










Griffin-Lim reconstruction
Linear-scale spectrogram
CBHG
Seq2seq target with r=3
Encoder
CBHG
Pre-net
Character embeddings
Attention
Attention is applied to all decoder steps
Decoder RNN
Attention RNN
Pre-net
<GO> frame
Decoder

TRAINING

- Train the model to **predict mel-spectrograms accurately from text**.
- Optimize both the decoder output (mel-spectrogram) and the post-processing net output (linear spectrogram).

Training Configuration

Dataset: 24.6 hours of North American English speech data by a professional female speaker.
Text normalization: dr -> doctor , f.e->for example , 3->three , $ -> dollar

Model Hyperparameters

- **Reduction Factor (r):** 2 **-** reduces decoder timesteps by predicting two frames per step
- **Batch Size:** 32

- **Loss Function:**
  L1 Loss (**M**ean **A**bsolute **E**rror) for both:
  - Seq2seq **decoder**: Predicting **mel-spectrogram**.
  - **Post-processing** net: Predicting **linear-scale spectrogram**

  Equal weight is given to both losses.

- **Optimizer : Adam** with **learning rate** decay:
  Start at 0.001, and decay to:
  0.0005 after 500K steps
  0.0003 after 1M steps
  0.0001 after 2M steps

TRAINING

- Tacotron was trained **end-to-end on text-audio pairs**, learning directly from character inputs to spectrogram outputs.
- **Teacher forcing** was used, where the decoder received ground-truth spectrogram frames instead of its own predictions during training.

- The model optimized **L1 loss** (Mean Absolute Error), comparing **predicted and real mel-spectrograms and linear-scale spectrograms.**
- The decoder generates frames **sequentially**, with each step depending on the previous step's context.

- Training ran for 2 million steps with a batch size of 32, covering **approximately 7,220 epochs.**

- During inference, only the last predicted frame of each step is fed into the next step to maintain smooth speech synthesis.

TRAINING

Challenges in Training

Handling Zero-Padding During Training

Problem:

- Tacotron uses **padded sequences** for batches, but **masking out** the padded frames can cause issues during inference, such as **repeated sounds** or poor stop token prediction.

Solution:

- Include **padded frames in training** by reconstructing the zero-padded frames, ensuring the model learns when to stop emitting outputs.

A loss mask is a technique used in deep learning to ignore specific parts of the data when calculating the loss during training. This is particularly important in sequence models, like Tacotron, when working with variable-length inputs or outputs.

TRAINING

Training loop

Prepare Data:
Convert text to sequences of characters.
Convert audio to mel-spectrograms and linear spectrograms.

Forward Pass:
Input text → Encoder → Decoder → Mel-Spectrogram.
Mel-Spectrogram → Post-Processing Net → Linear Spectrogram.

Compute Loss:
Use the ground-truth mel-spectrogram and linear spectrogram to calculate the total loss.

Backward Pass:
Compute gradients and update weights using the optimizer.

Adjust Learning Rate:
Update the learning rate based on the current step.

Repeat for All Batches.

MEAN OPINION SCORE (MOS)

Table 2: 5-scale mean opinion score evaluation.

	mean opinion score
Tacotron	3.82 ± 0.085
Parametric	3.69 ± 0.109
Concatenative	4.09 ± 0.119

Experiment:

- 100 unseen phrases, rated on a 5-point Likert scale.
- 8 ratings per phrase, using headphones.

- **MOS**: 3.82, **outperforming parametric baselines**. (despite Griffin-Lim artifacts)

- **High-quality natural speech** from characters.

EXPERIMENTS

Attention alignment on a test phrase

Vanilla Seq2Seq:

Requires scheduled sampling for alignment.

Issues:

Poor alignment: Attention gets stuck for multiple frames.

Synthesized speech is unintelligible with unnatural durations.

CBHG Encoder:

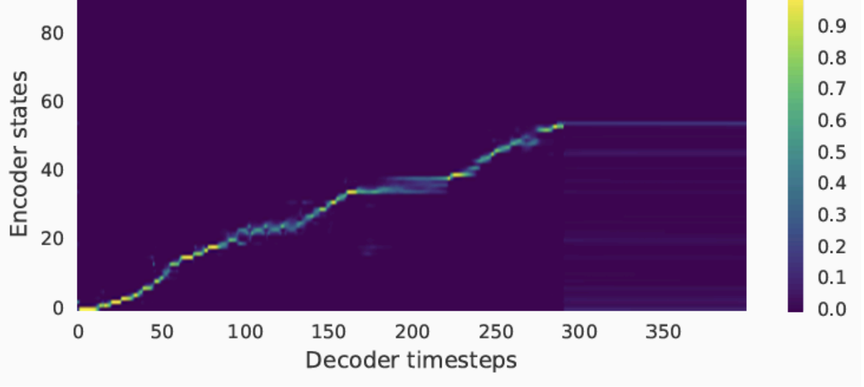
Compared to a 2-layer GRU encoder:

CBHG Advantages:

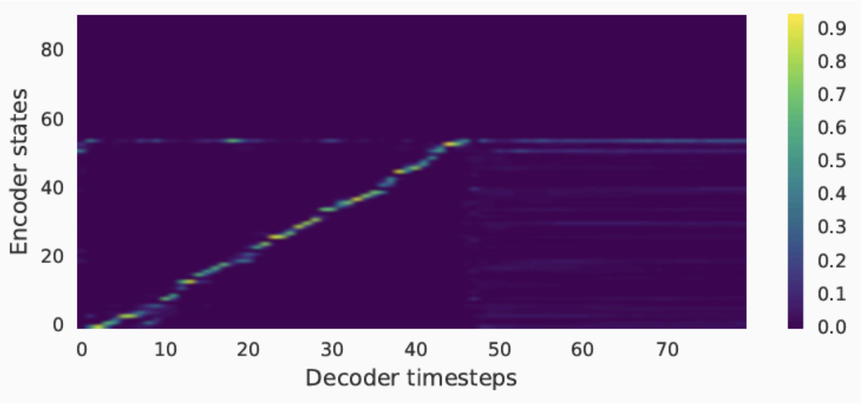
Cleaner, smoother alignments.

Better generalization for long/complex phrases.

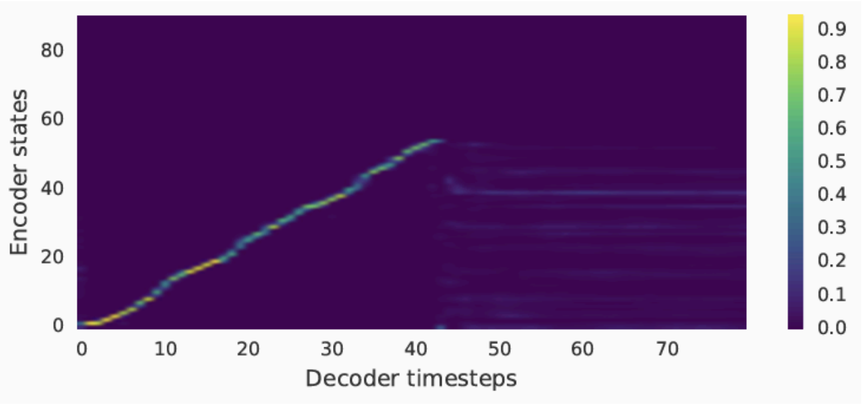
Fewer mispronunciations.



(a) Vanilla seq2seq + scheduled sampling



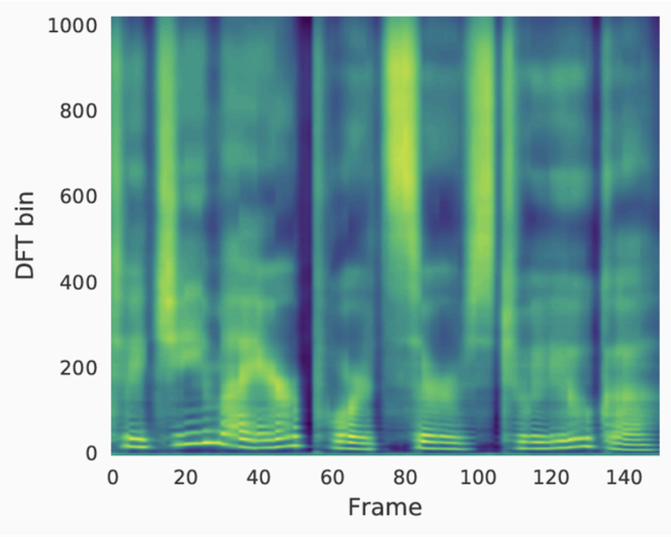
(b) GRU encoder



(c) Tacotron (proposed)

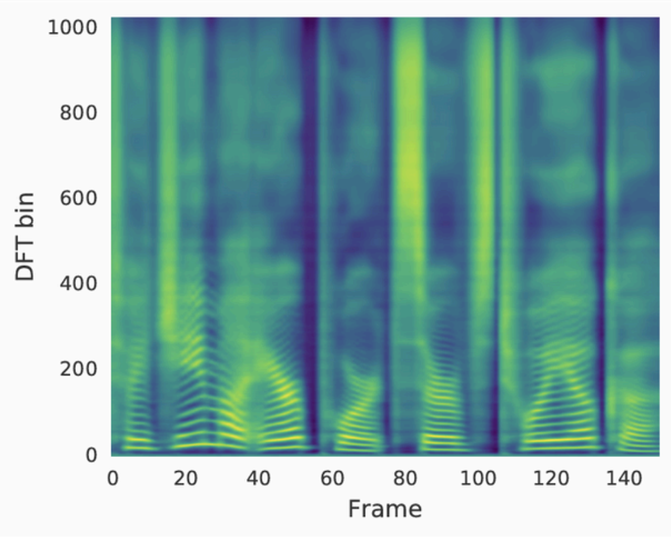
POST-PROCESSING NET ANALYSIS

Without Post-Processing Net:
Linear-scale spectrogram is directly predicted.
Poor harmonic resolution and high-frequency formant structure.



(a) Without post-processing net

With Post-Processing Net:
Improved harmonic resolution and formant structure.
Smoother, more natural-sounding audio.



(b) With post-processing net