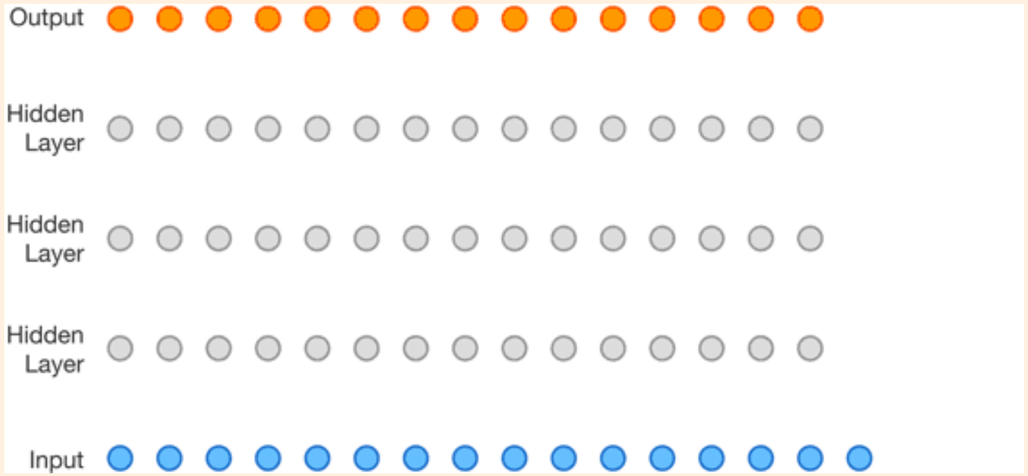
Neural TTS Overview

TTS Overview

Towards Neural TTS: WaveNet

- Introduced by DeepMind in 2016
- Replaces the Acoustic Model and Vocoder
- Still needs to be connected to an independent, pre-trained Linguistic Model/Text Encoder
  —> Not End-to-End
- Set a new standard for speech synthesis
- However, it was slow due to its auto-regressive, sample-level generation



Two-stage Models: Tacotron

- Introduced by Google in 2017
- Sequence-to-sequence framework trained to generate Mel-spectrograms from input text
- Text encoder and Acoustic model are trained end-to-end
- However, the vocoder is still not trained as part of the framework:
  - Uses Griffin-Lim, a Fourier transform based algorithm to reconstruct an audio waveform from a spectrogram
- Concatenative systems still outperform it

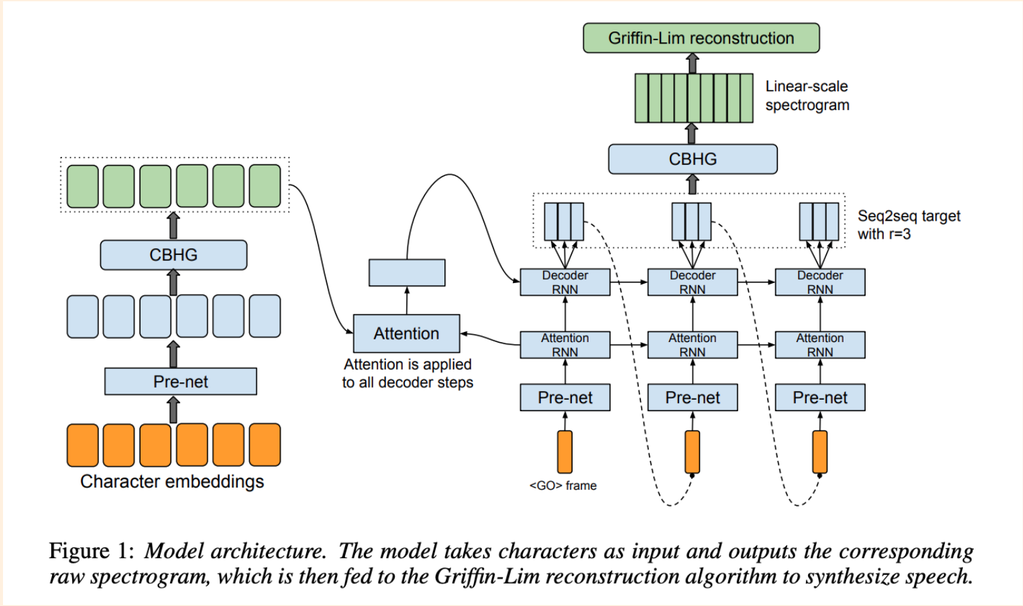


Figure 1: *Model architecture. The model takes characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesize speech.*

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

Two-stage Models: Tacotron 2

- Uses WaveNet as the vocoder
- Achieved very natural results that surpass concatenative methods
- Still not completely end-to-end as the WaveNet vocoder is pre-trained separately

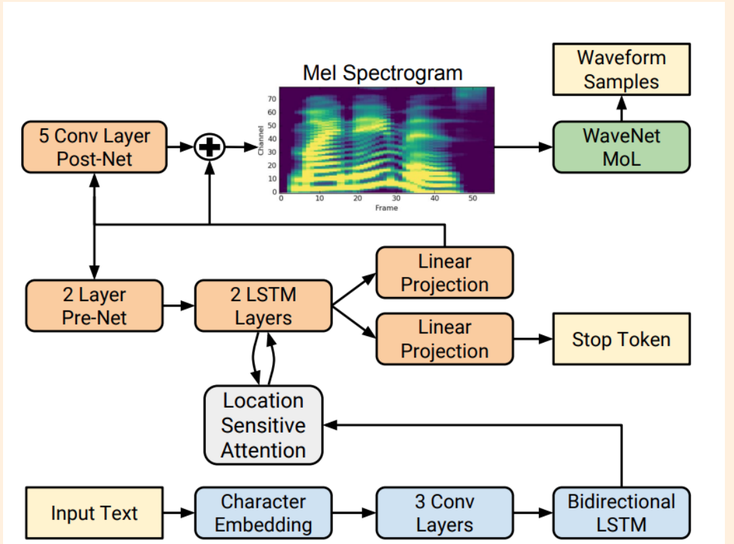


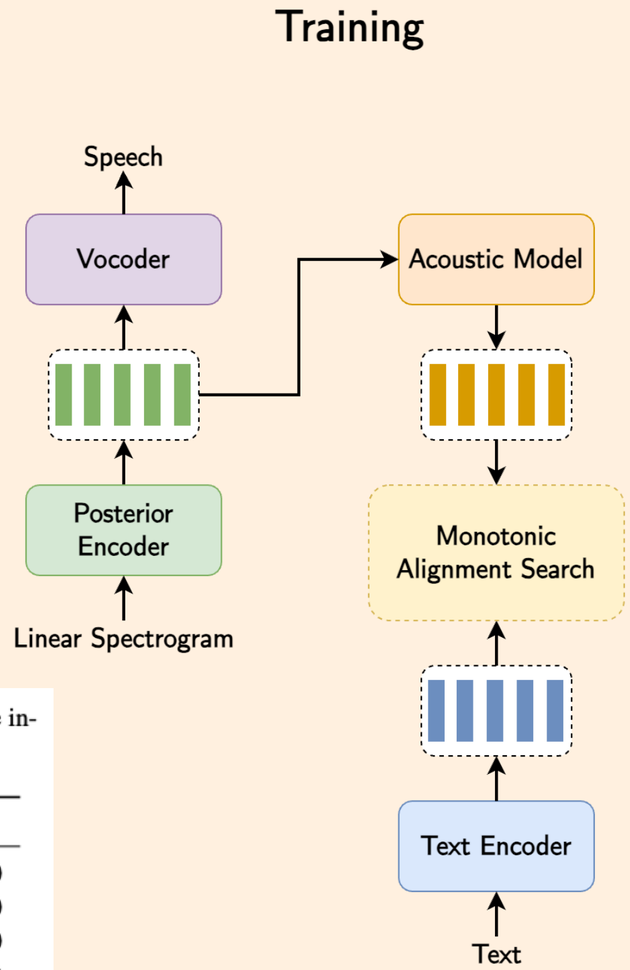
Fig. 1. Block diagram of the Tacotron 2 system architecture.

System	MOS
Parametric	3.492 ± 0.096
Tacotron (Griffin-Lim)	4.001 ± 0.087
Concatenative	4.166 ± 0.091
WaveNet (Linguistic)	4.341 ± 0.051
Ground truth	4.582 ± 0.053
Tacotron 2 (this paper)	$\mathbf{4.526 \pm 0.066}$

Table 1. Mean Opinion Score (MOS) evaluations with 95% confidence intervals computed from the t-distribution for various systems.

End-to-end Models: VITS

- Trained end-to-end to generate a speech waveform from input text
- During training, a speech-to-speech task is used to encode speech into an intermediate representation, and learn to decode it back to waveform using the vocoder
- There is also a text-to-embedding task to map from task to the same intermediate representation
- Both are trained jointly
- Surpasses two-stage models in terms of naturalness



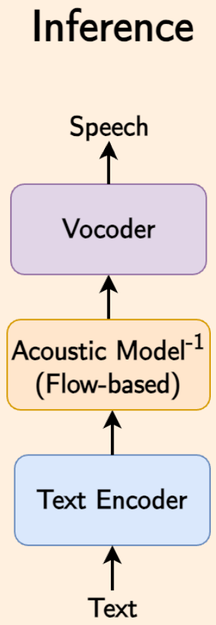


Table 1. Comparison of evaluated MOS with 95% confidence intervals on the LJ Speech dataset.

Model	MOS (CI)
Ground Truth	4.46 (±0.06)
Tacotron 2 + HiFi-GAN	3.77 (±0.08)
Tacotron 2 + HiFi-GAN (Fine-tuned)	4.25 (±0.07)
Glow-TTS + HiFi-GAN	4.14 (±0.07)
Glow-TTS + HiFi-GAN (Fine-tuned)	4.32 (±0.07)
VITS (DDP)	4.39 (±0.06)
VITS	**4.43 (±0.06)**

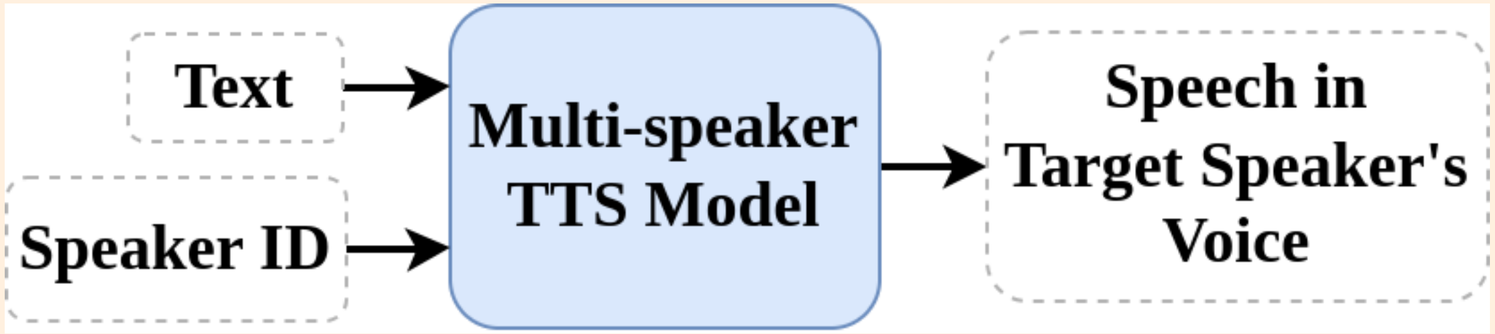
Two-stage vs End-to-end

- Both two-stage and end-to-end models have pros and cons

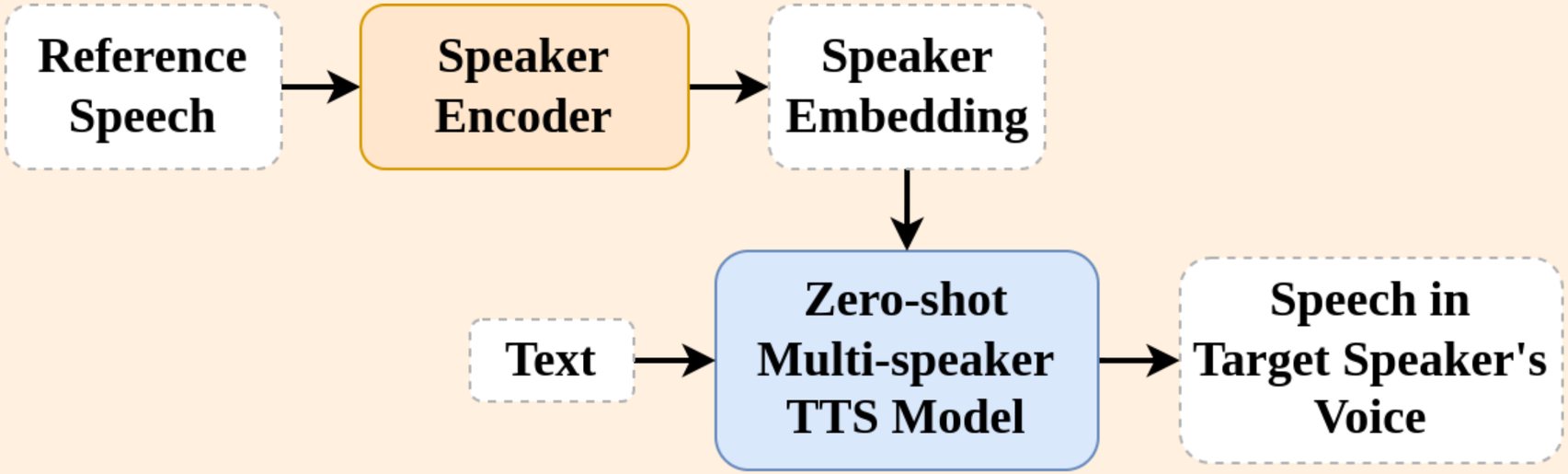
- **Two-stage:**
  - Pros: intermediate representation is interpretable, blocks can be replaced in a modular way, vocoder can be trained on untranscribed speech data
  - Cons: can suffer from error propagation, handcrafted intermediate representations have limitations (e.g. spectral features)

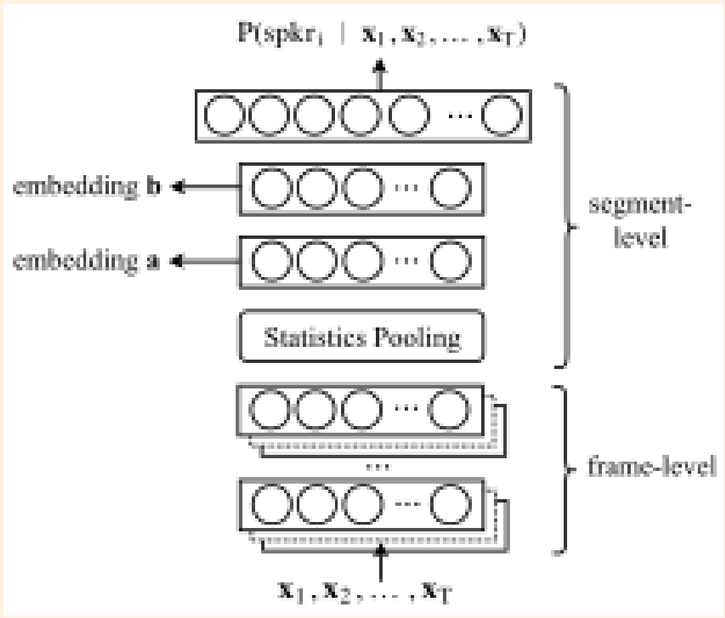
- **End-to-end:**
  - Pros: simplified training pipeline, components are tuned jointly leading to less error propagation, achieve higher naturalness
  - Cons: reduced flexibility, intermediate representation is less interpretable, can suffer from oversmoothing and mispronunciation

Multi-speaker TTS



Zero-shot Multi-speaker TTS





Speaker embedding extraction

Neural embeddings as Intermediate Representations

- Recently, two-stage models that use features from Speech Foundation Models (aka Self-Supervised Learning models) as intermediate representations
- They enjoy the flexibility of two-stage models, while not suffering from limitations due to handcrafted intermediate representations
- Speech foundation model embeddings have very interesting properties which enable new use cases and possibilities

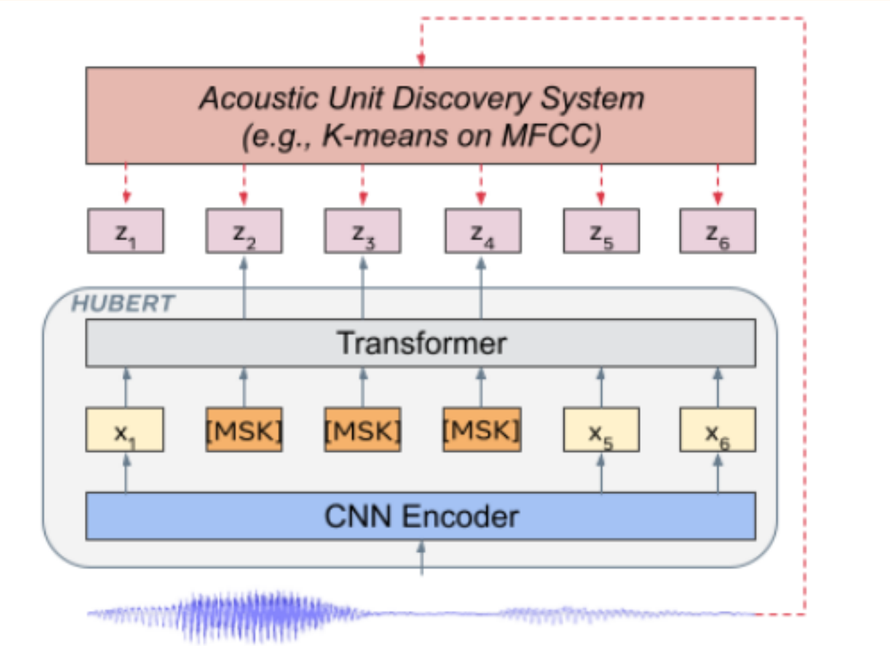
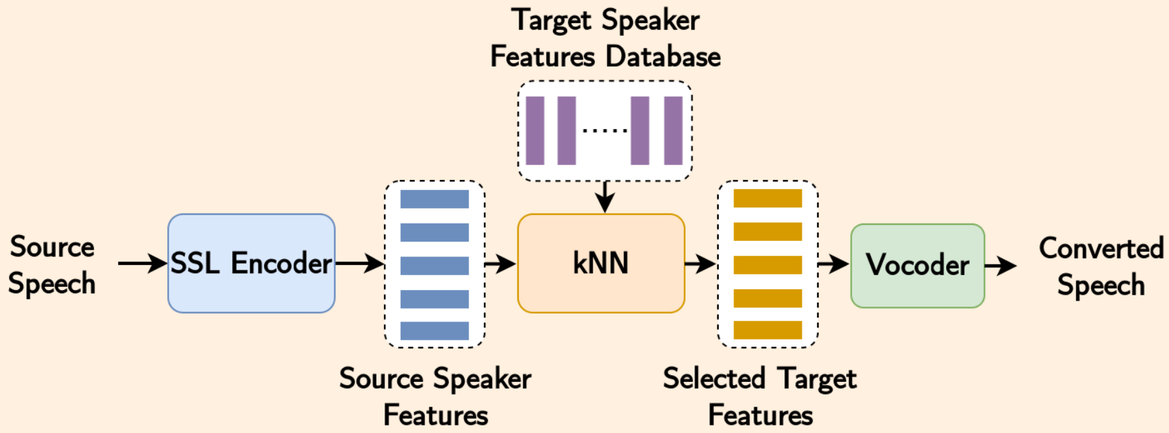


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

Example: kNN-VC

- SSL models such as Wav2Vec2, HuBERT, WavLM, encode speech into a sequence of frames, each corresponding to a window of 25ms of speech, with a 20ms hop between windows.
- Property: frames are linearly close if they contain similar phonetic, linguistic information, even if they are spoken by very different voices
- This enables very simple Voice Conversion: kNN-VC



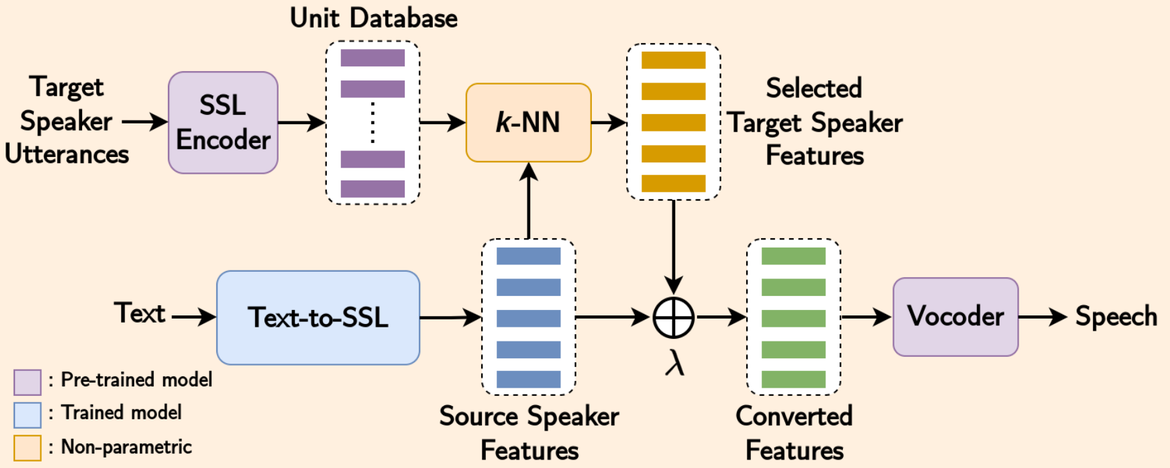
Source (VITS)

Reference speaker

Converted output

Multi-speaker Zero-shot kNN-TTS

- Can extend the same idea to TTS
- Two-stage framework which uses SSL embeddings as intermediate representations
- Using SSL features + kNN enables simple zero-shot multi-speaker TTS with no need for speaker embeddings
- Can also linearly interpolate between features to achieve Voice Morphing



$$y_{\text{converted}} = \lambda\, y_{\text{selected}} + (1-\lambda)\, y_{\text{source}}$$

λ = 0

λ = 0.25

λ = 0.5

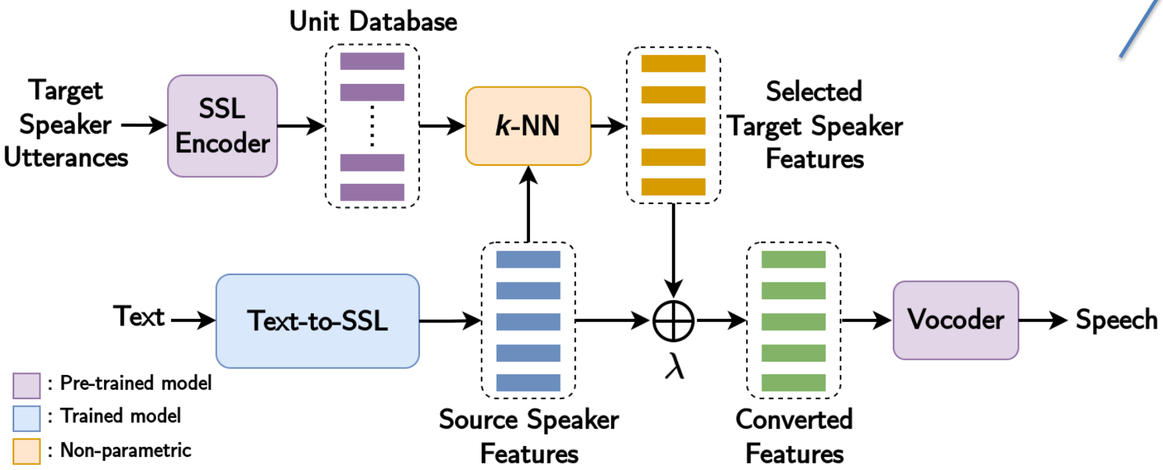
λ = 0.75

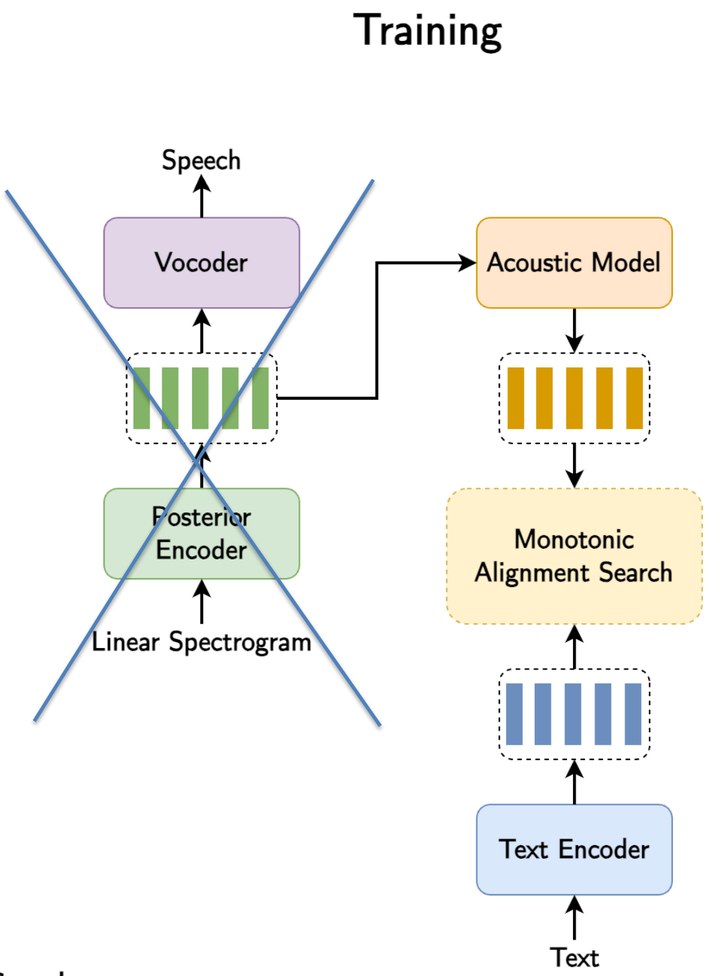
λ = 1

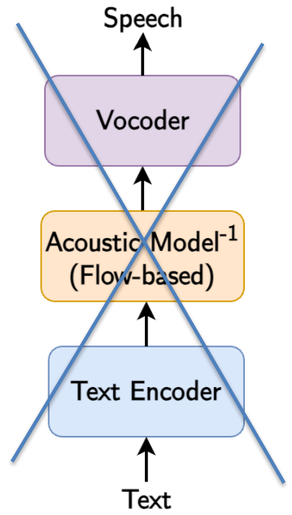
Ground truth

Multi-speaker Zero-shot kNN-TTS: Modular version of VITS

- Intermediate representation does not needs to be trained. It can be SSL model, e.g., WavLM.
- Vocoder does not needs to be trained, e.g., kNN-VC vocoder which takes WavLM representation as input.
- Only text-to-SSL embedding needs to be trained. It can be done with a single speakers data.







Future

- Neural TTS models have reached a naturalness level comparable to human speech
- However, it is still challenging to generate highly expressive speech (e.g. sports commentator)
- Defining and generating the subtle nuances of human expressiveness is difficult, making it challenging to guide models in a practical and versatile manner
- Many approaches are being explored, such as using text prompts to specify the desired speech style or tone
- Seamless context-switching between different languages is another use case that is still challenging

Example: ChatGPT Advanced Voice