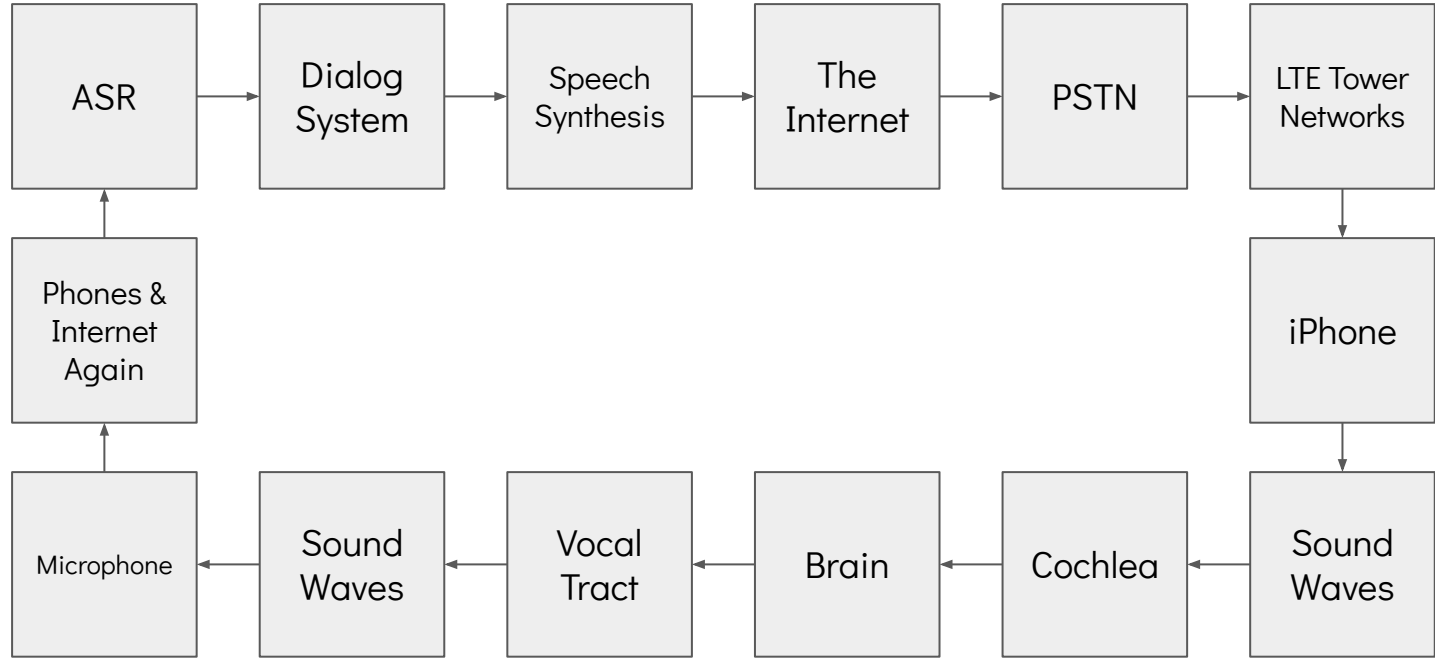
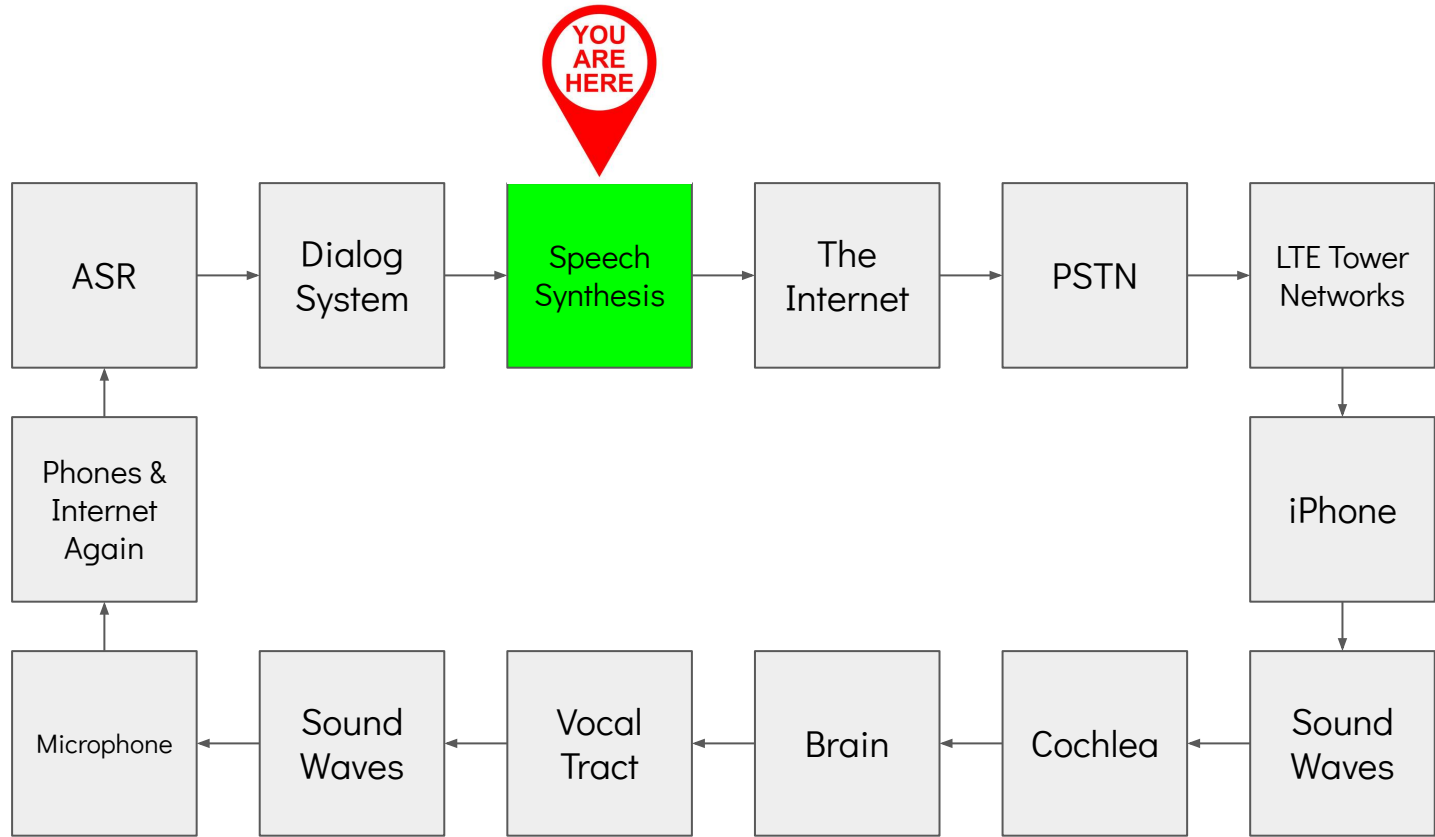


IAP: TEXT-TO-SPEECH (TTS)

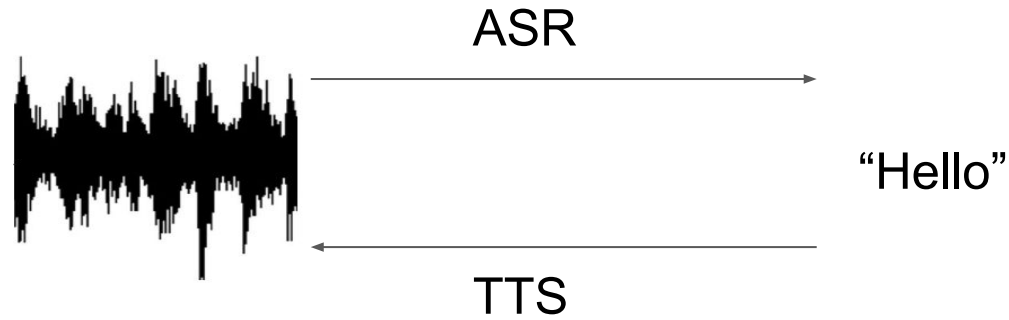
Jan 25, 2023

Recall from Lecture 1



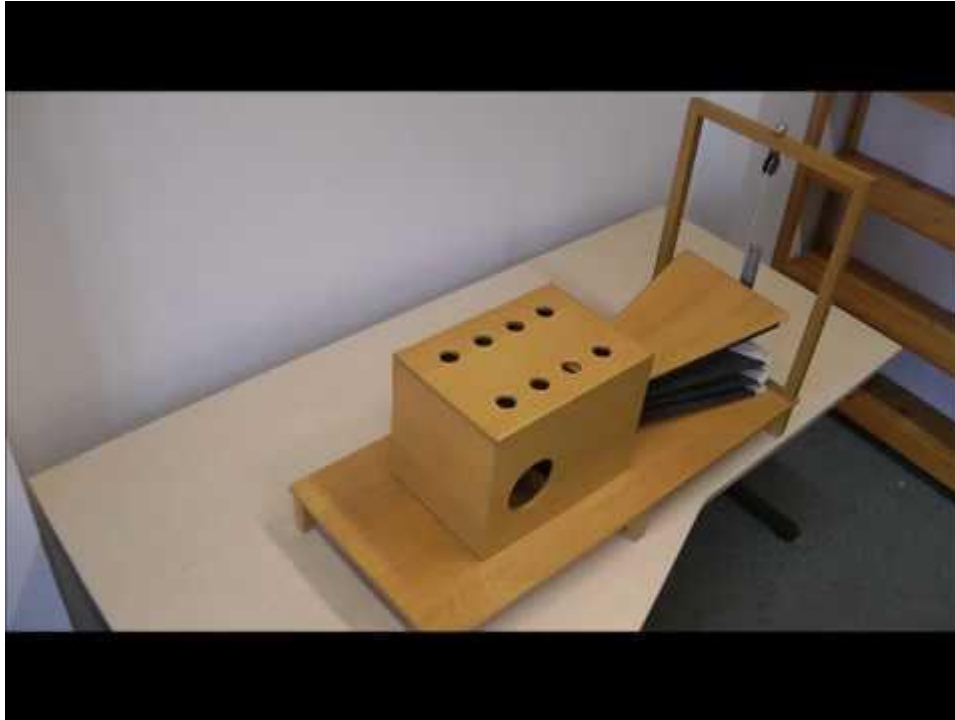


TTS: Text-to-Speech



TTS history

- 1769 Wolfgang von Kempelen's speaking machine



TTS history

- 1939 VODER



TTS history

- 1961 IBM 7094



TTS history

- 1980s DECtalk



TTS history

- 1980s DECtalk



TTS history

- 2022 NaturalSpeech (neural network)

“as effectually to rebuke and abash the profane spirit of the more insolent and daring of the criminals.”



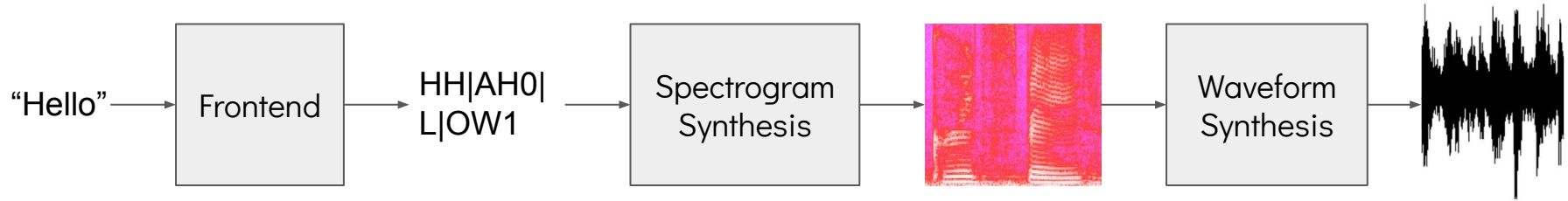
“it is not possible to state with scientific certainty that a particular small group of fibers come from a certain piece of clothing.”



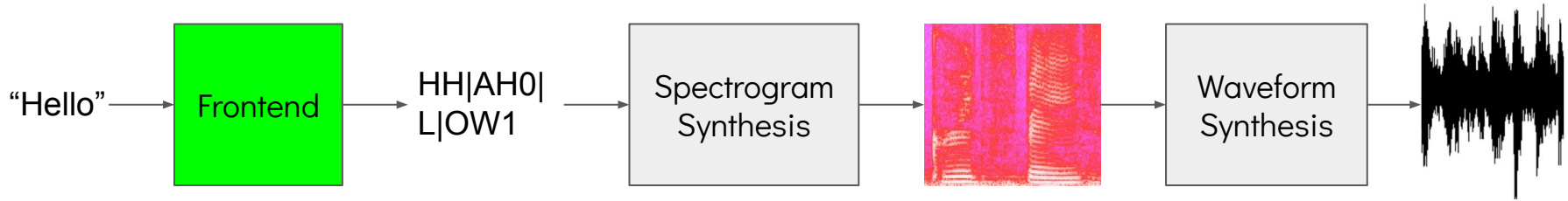
Why TTS

- Speech is the natural way humans communicate, with transmission rate of about 120 - 180 words per minute, without need of special equipments
- Speech contains other information other than words, eg. emotions, sarcasm, personality. Makes computer feel “alive” and natural for humans to interact with
- Eg. siri, alexa, etc

Text-to-Phonemes-to-Spectrogram-to-Speech



Part I: Text-to-Phonemes



Part I: Text-to-Phonemes

- Text, eg. “Hello Mr. Scodary, you won \$10000!”
- Normalize -> “hello mister scodary, you won ten thousand dollars”
- Phonemize -> “HH|AH0|L|OW1| |M|IH1|S|T|ER0| |S|K|OW0|D|EH0|R|IY0| |,|
|Y|UW1| |W|AH1|N| |T|EH1|N| |TH|AW1|Z|AH0|N|D| |D|AA1|L|ER0|Z”

Why Phonemes?

- Easy to train model
- Some use characters instead
- Some use a mixture of characters and phonemes (eg randomly convert 30% words to phonemes)

Text normalization

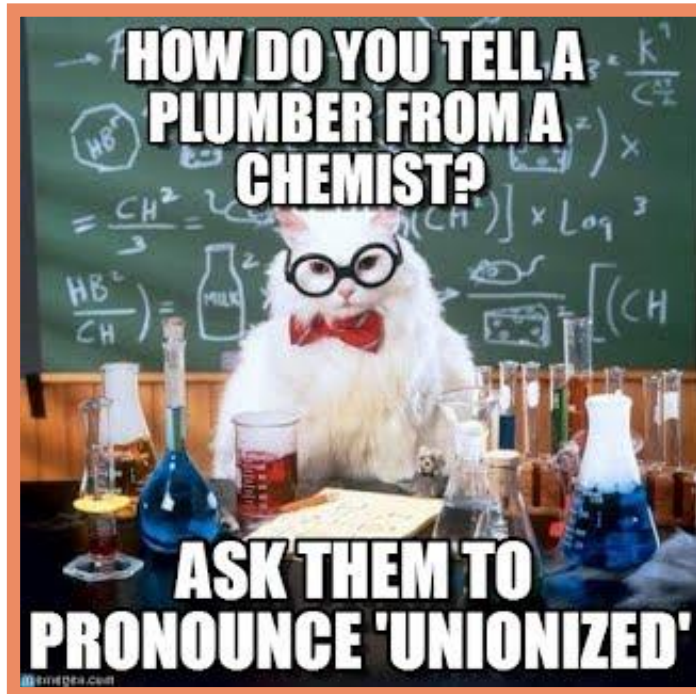
- Mostly rule-based, using regex
- Emails and dot com, eg “lokman@gmail.com” -> “lokman at gmail dot com”
- Numbers, eg “1,234,567” -> “one million, two hundred and thirty four thousand, five hundred and sixty seven”
- Dollars, eg “\$4,000” -> “four thousand dollars”
- Years, eg “2021” -> “twenty twenty one”

Text-to-Phoneme

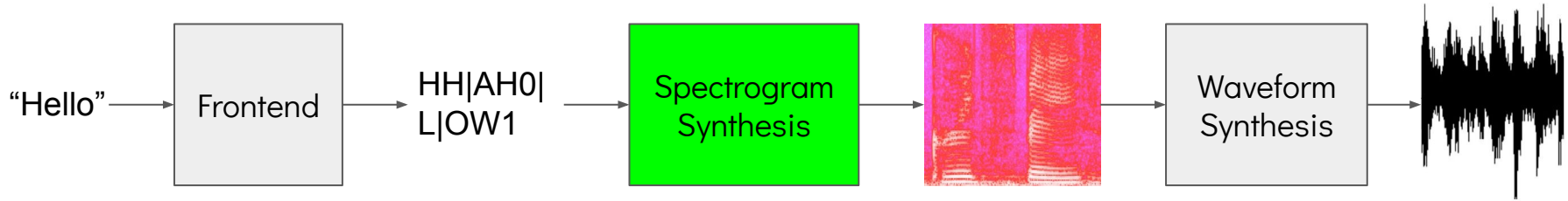
- aka grapheme-to-phoneme, or g2p
- Eg. “ten thousand” -> |T|EH1|N| |TH|AW1|Z|AH0|N|D|
- Dictionary lookup
- For out-of-vocab words, run a simple seq-to-seq model
- We can correct mispronunciations here without retraining model

Challenge: Homographs

- For homographs, eg “we will **record** this call” vs. “let me look up your **record**”: use part-of-speech tagger, assign different phonemes for **verbs** vs **nouns**



Part II: Phonemes-to-Spectrogram



Why Spectrograms?

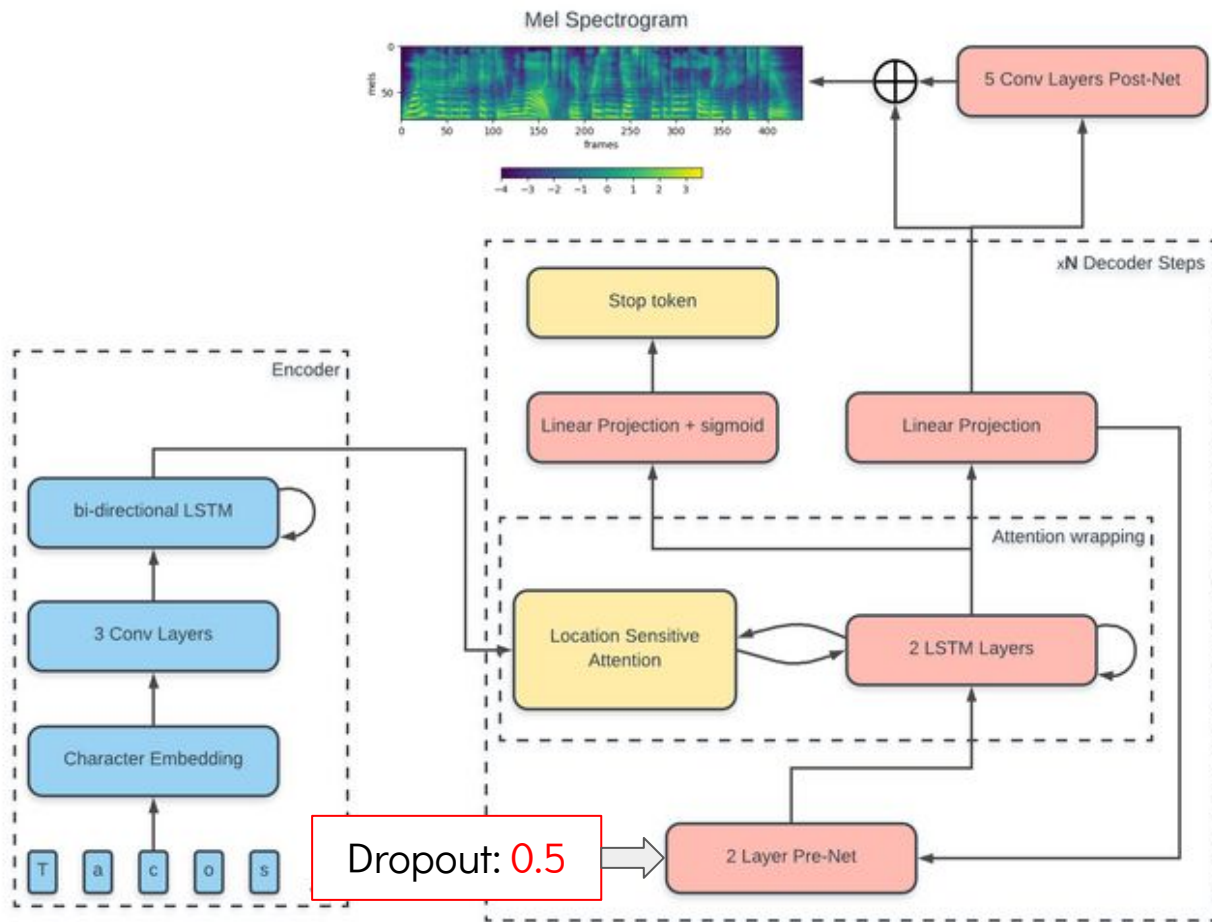
- Easy to train model
- Mel-spectrograms:
wav -> Fourier transform -> take magnitude -> focus on human speech frequency bands (see ASR lecture)
- Mel-spectrograms is a good representation of human speech
- Useful for duration modelling (number of mel-frames \propto time duration)

Tacotron2

- Attention-based
- Auto-regressive
- MOS 4.53
(human: 4.58)
- 26M params

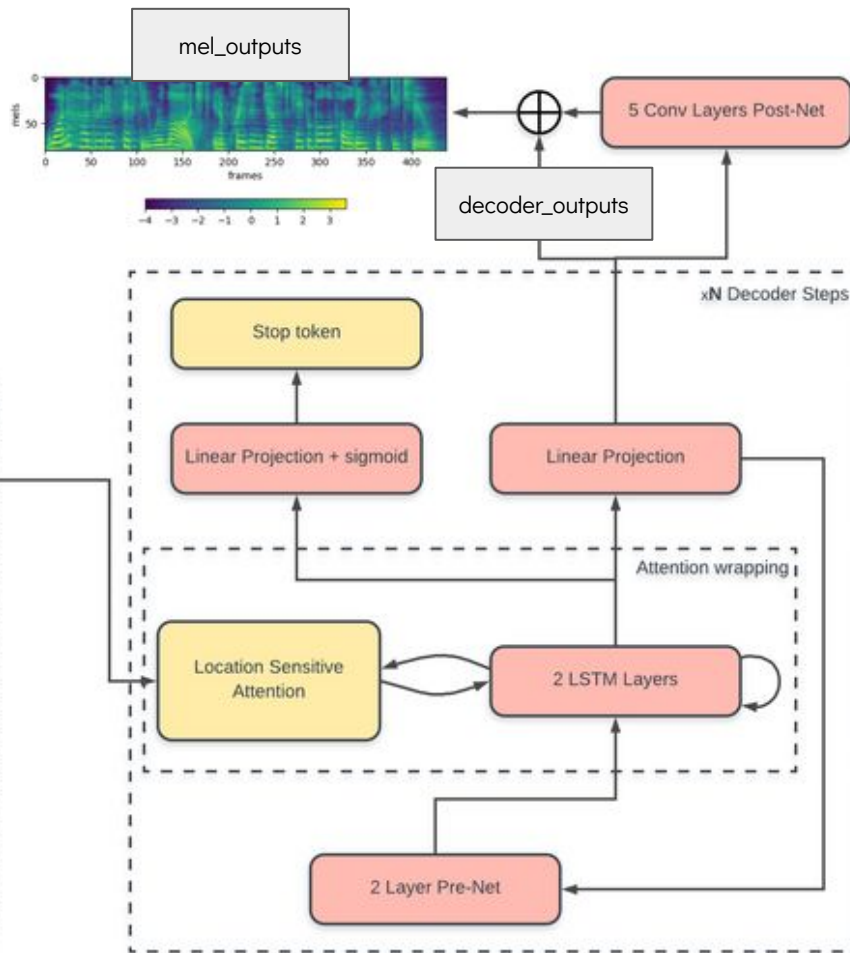
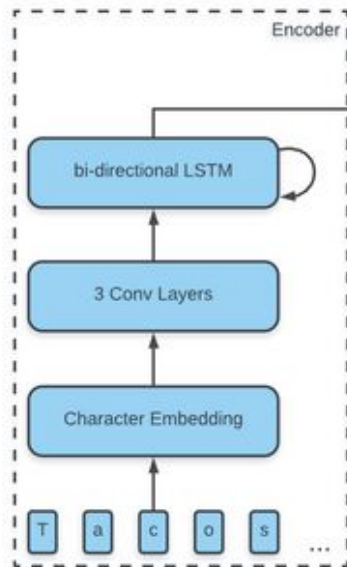
[Shen et. al. \(2018\)](#)

*These authors really like tacos.



Tacotron2 Loss

$$L = L_{pre} + L_{post} + L_{stop}$$

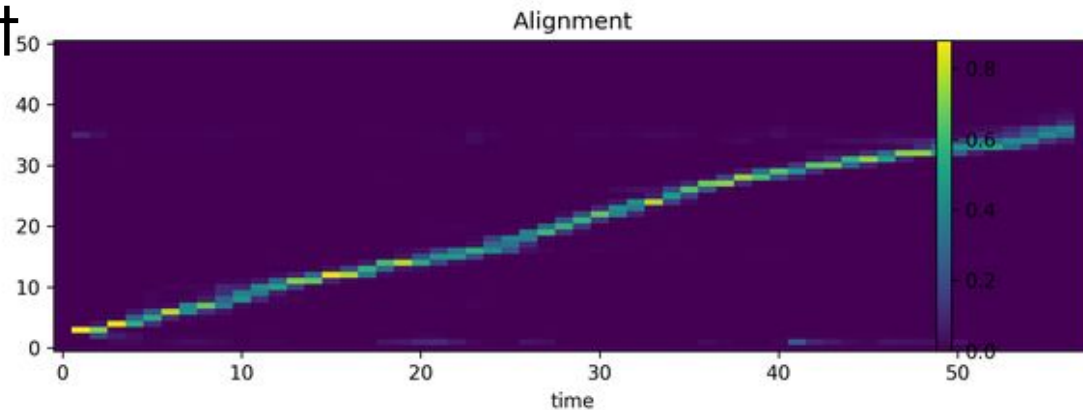


[Shen et. al. \(2018\)](#)

*These authors really like tacos.

Attention Alignment

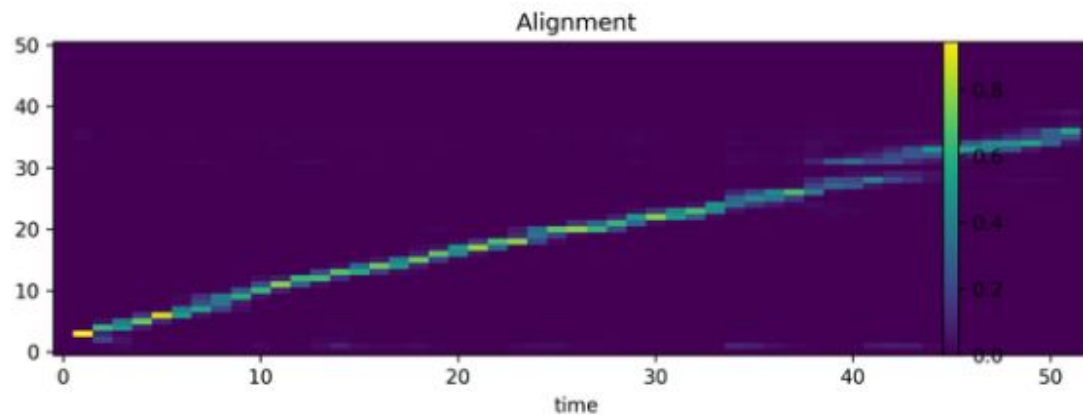
Good:



Could be better:



My name is Grace thank you for calling

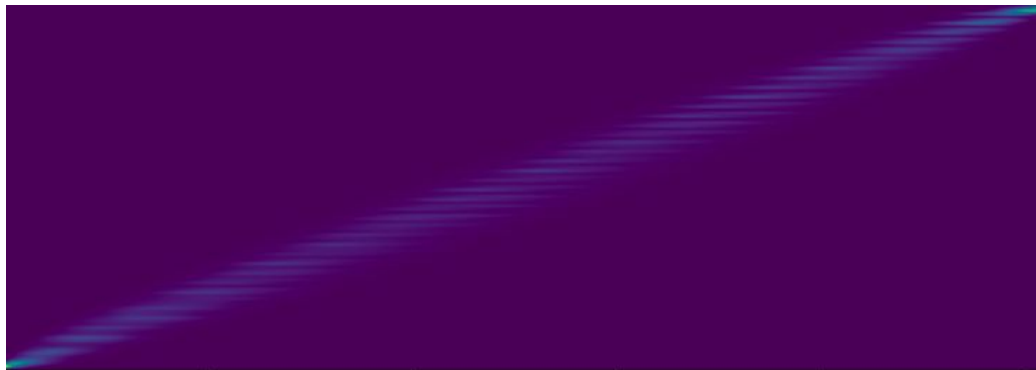


Good alignment=good generalization
Useful for debugging

Thanks for calling bright smile dental

Improving Attention Alignment?

- Add beta-binomial prior to attention in early stages of training
- Add Alignment sharpness to loss



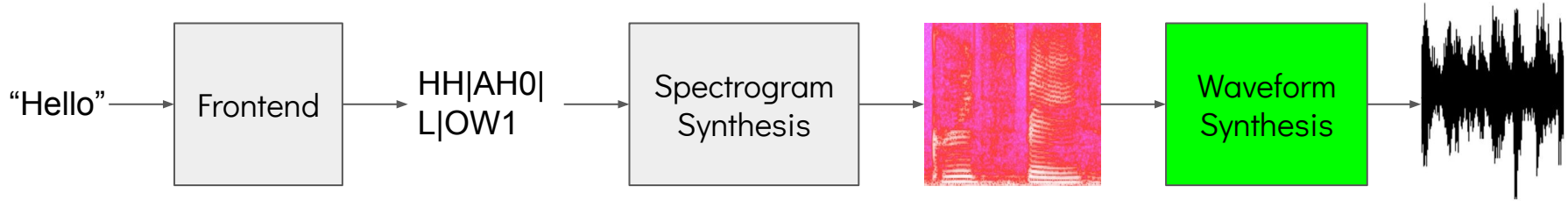
$$L_{ForwardSum} = L_{CTC}(\mathcal{A})$$

- Focus Rate

rate F to measure how an attention head is close to diagonal: $F = \frac{1}{S} \sum_{s=1}^S \max_{1 \leq t \leq T} a_{s,t}$, where S and T are the lengths of the ground-truth spectrograms and phonemes, $a_{s,t}$ donates

[One TTS Alignment To Rule Them All](#)
[RAD-TTS FastSpeech](#)

Part III: Spectrogram-to-Waveform



Griffin-Lim

- Algorithm that reconstruct time domain signal, by filling in missing phases, from given magnitude

Start with random initial phase: `angles`

```
spectrogram.magnitude, spectrogram.angles = magnitude, angles
for i in range(num_iters):
    inverse = iSTFT(spectrogram)
    spectrogram = STFT(inverse)
    spectrogram.magnitude = magnitude
final_inverse = iSTFT(spectrogram)
```

- No parameters
- Quality not very good

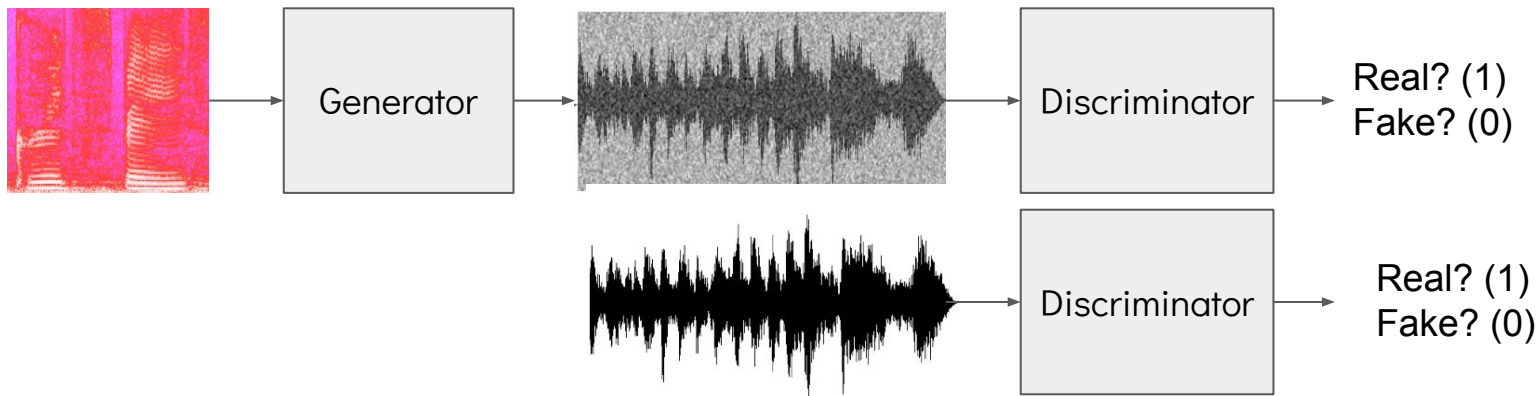
Neural waveform synthesis

- WaveNet (high quality but very slow)
- Parallel WaveNet (hard to train)
- WaveGlow (Flow+WaveNet, fast and good.)
- HiFiGAN (even faster with no loss in quality)

See [224S lectures](#) for WaveNet/WaveGlow

HiFiGAN

- Generative Adversarial Network



$$\mathcal{L}_{Adv}(D; G) = \mathbb{E}_{(x,s)} \left[(D(x) - 1)^2 + (D(G(s)))^2 \right]$$

$$\mathcal{L}_{Adv}(G; D) = \mathbb{E}_s \left[(D(G(s)) - 1)^2 \right]$$

HiFiGAN: Generator

- Lightweight (~14M params)
- Non-autoregressive

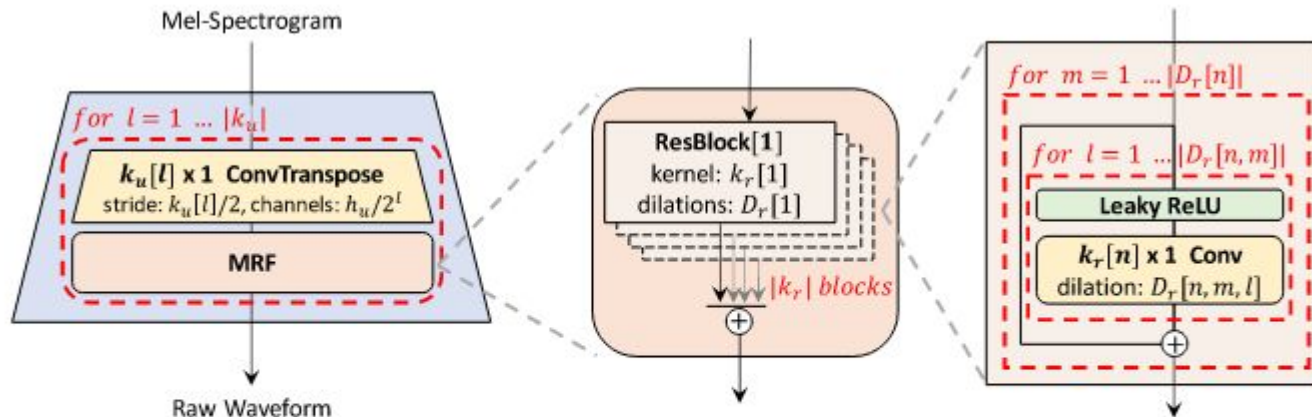


Figure 1: The generator upsamples mel-spectrograms up to $|k_u|$ times to match the temporal resolution of raw waveforms. A MRF module adds features from $|k_r|$ residual blocks of different kernel sizes and dilation rates. Lastly, the n -th residual block with kernel size $k_r[n]$ and dilation rates $D_r[n]$ in a MRF module is depicted.

HiFiGAN: Discriminators

- MultiPeriodDiscriminators (MPD) and MultiScaleDiscriminators (MSD) (71M parameters)

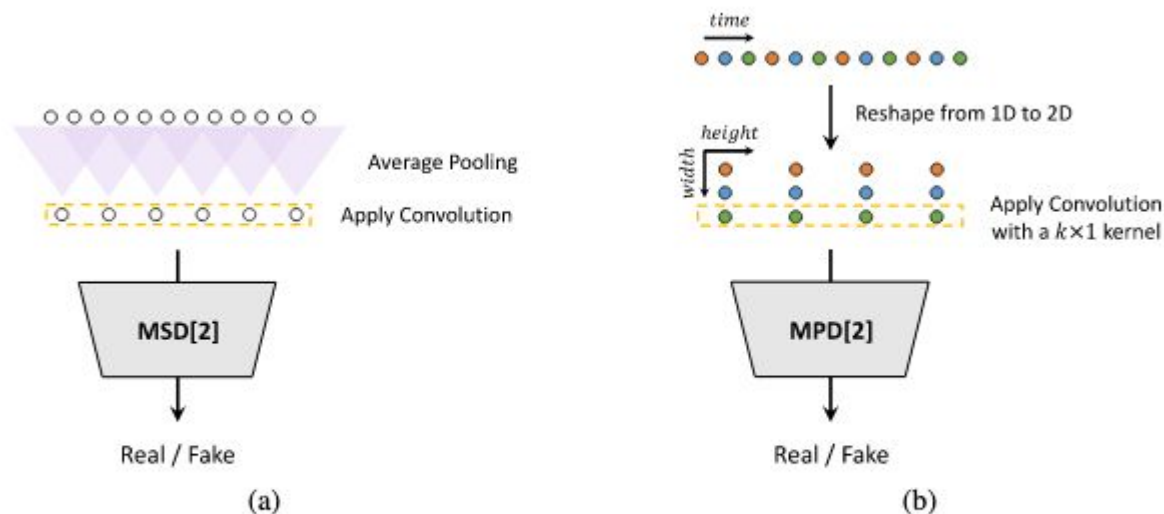


Figure 2: (a) The second sub-discriminator of MSD. (b) The second sub-discriminator of MPD with period 3.

HiFiGAN: additional loss terms

- Mel-spectrogram loss:

$$\mathcal{L}_{Mel}(G) = \mathbb{E}_{(x,s)} \left[\|\phi(x) - \phi(G(s))\|_1 \right]$$

(generator should output waveforms that converts to same mel as ground-truth waveform)

- Feature-matching loss:

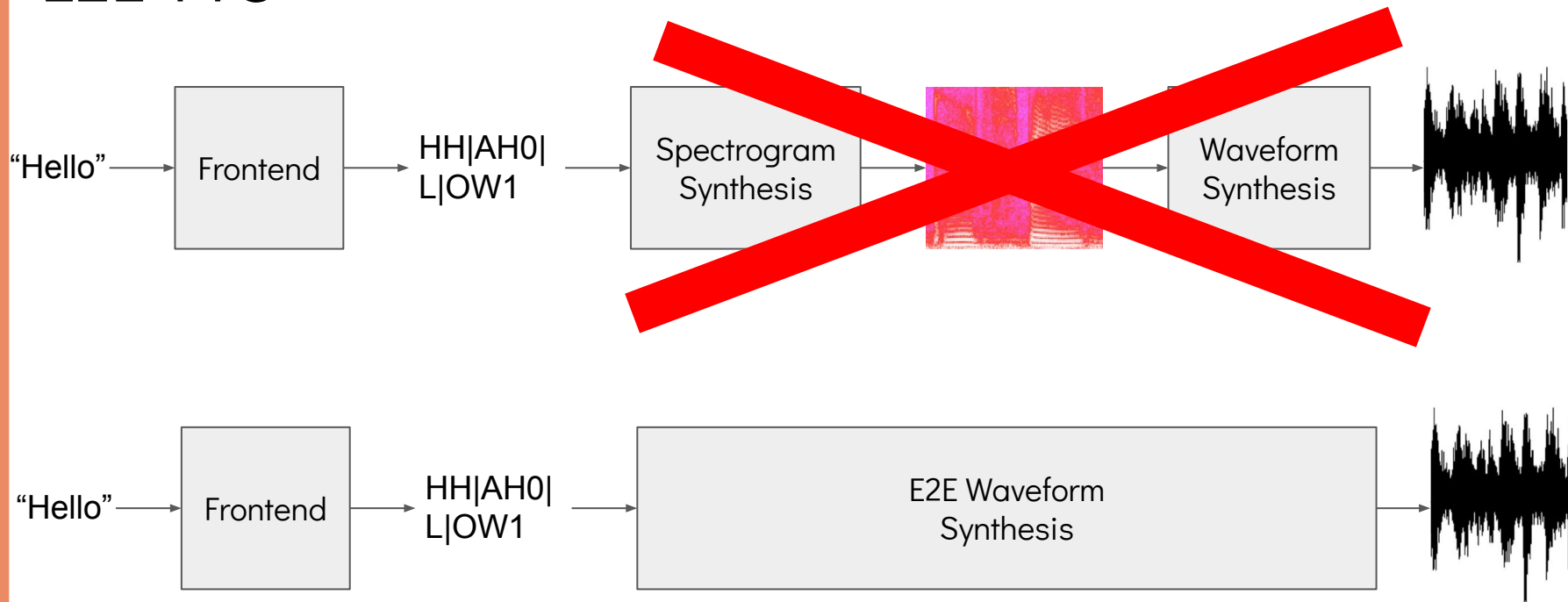
$$\mathcal{L}_{FM}(G; D) = \mathbb{E}_{(x,s)} \left[\sum_{i=1}^T \frac{1}{N_i} \|D^i(x) - D^i(G(s))\|_1 \right]$$

(generated waveforms and ground-truth waveforms should have the same features inside discriminators)

- Final loss:

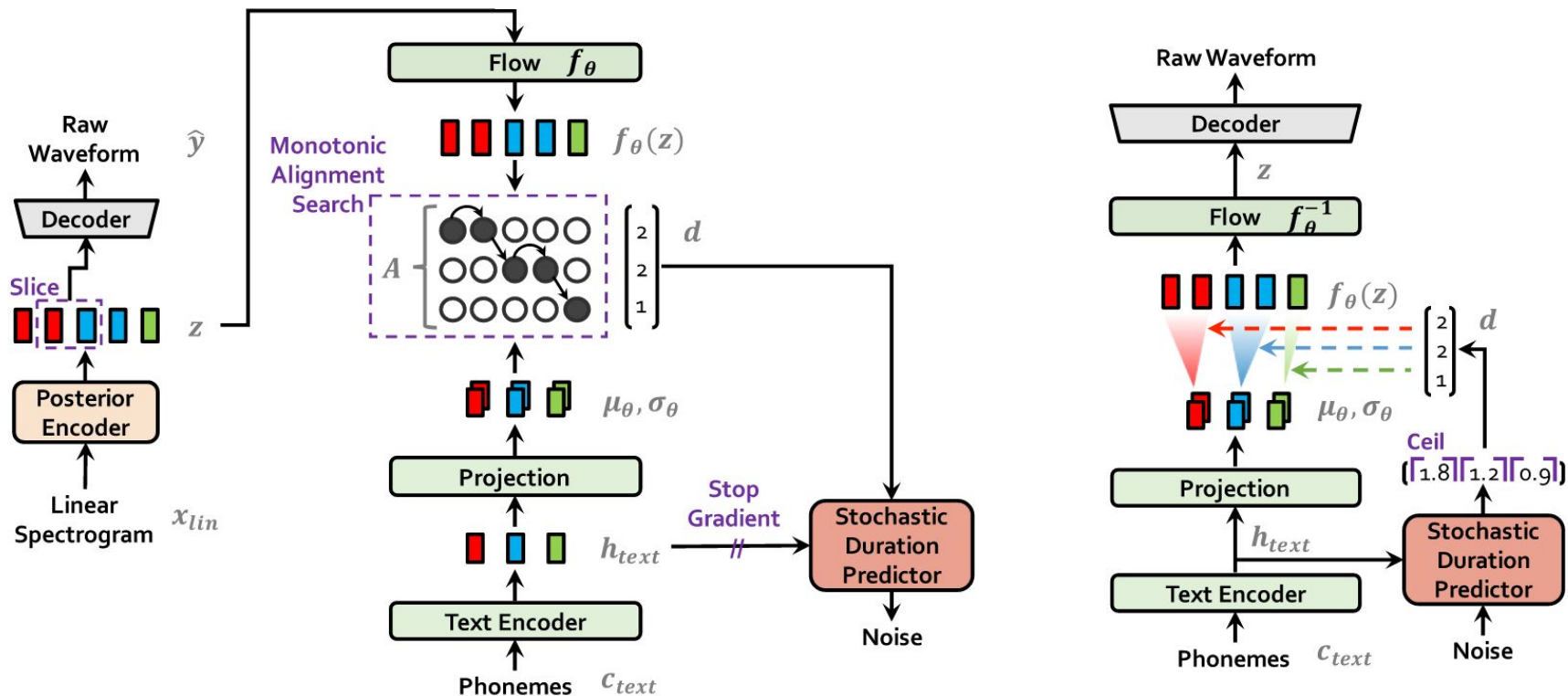
$$\mathcal{L}_G = \sum_{k=1}^K \left[\mathcal{L}_{Adv}(G; D_k) + \lambda_{fm} \mathcal{L}_{FM}(G; D_k) \right] + \lambda_{mel} \mathcal{L}_{Mel}(G)$$
$$\mathcal{L}_D = \sum_{k=1}^K \mathcal{L}_{Adv}(D_k; G)$$

E2E TTS



E2E TTS: VITS

$$L_{vae} = L_{recon} + L_{kl} + L_{dur} + L_{adv}(G) + L_{fm}(G)$$



Kim et al
<https://arxiv.org/abs/2106.06103>

(a) Training procedure

(b) Inference procedure

E2E TTS: VITS

High MOS (mean-opinion-score)

No need to train alignment

No need to train vocoder separately

Controllable duration

Non-autoregressive, thus very fast compared to eg. Tacotron 2

E2E TTS: NaturalSpeech

Variation on VITS, with:

- Phoneme pretraining
- Differentiable durator
- Memory bank in wave decoder
- MOS matches human speaker on LJSpeech

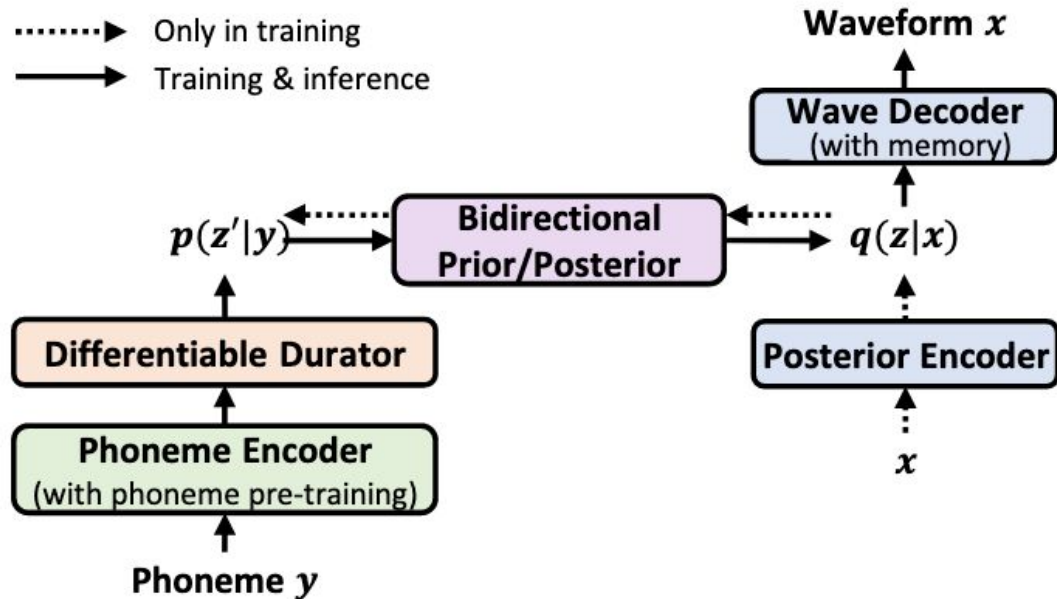
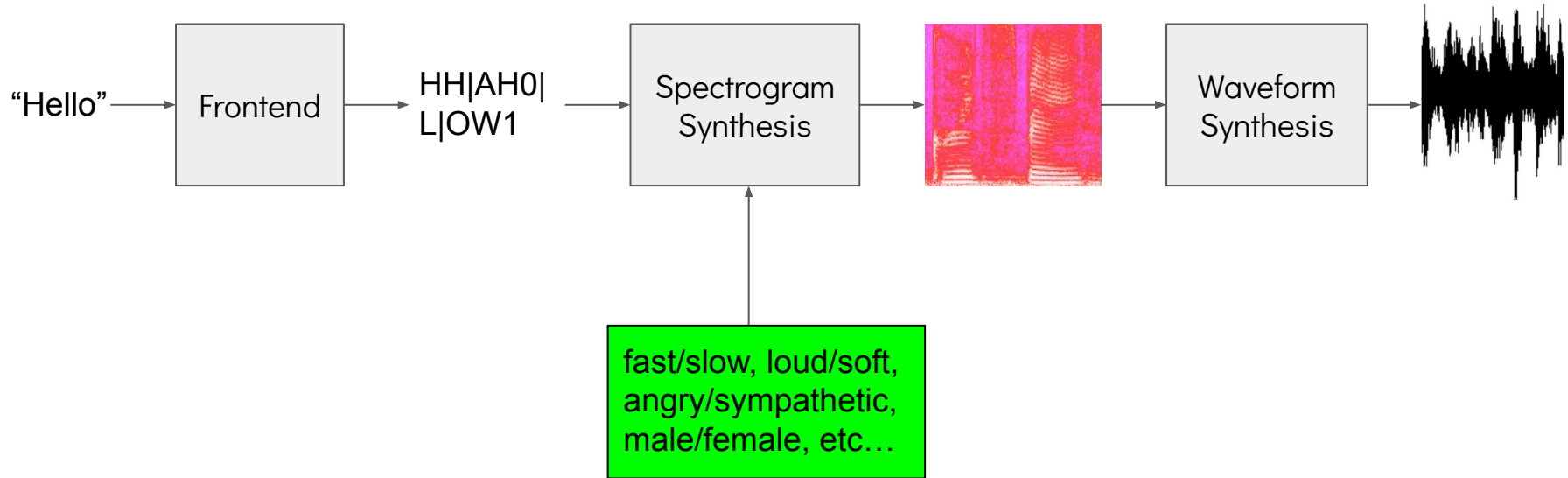


Figure 1: System overview of NaturalSpeech.

Part IV: Control



Why control?

- TTS is a one-to-many mapping
- In dialog systems, can tailor to specific users for specific situations
- Eg. old people may prefer slower, louder speech
- Eg. medical bots/customer service bots should be sympathetic
- Eg. Different accents for different geographic users

Case study: FastSpeech2

- Use Variance Adaptor to predict energy/pitch from hidden states

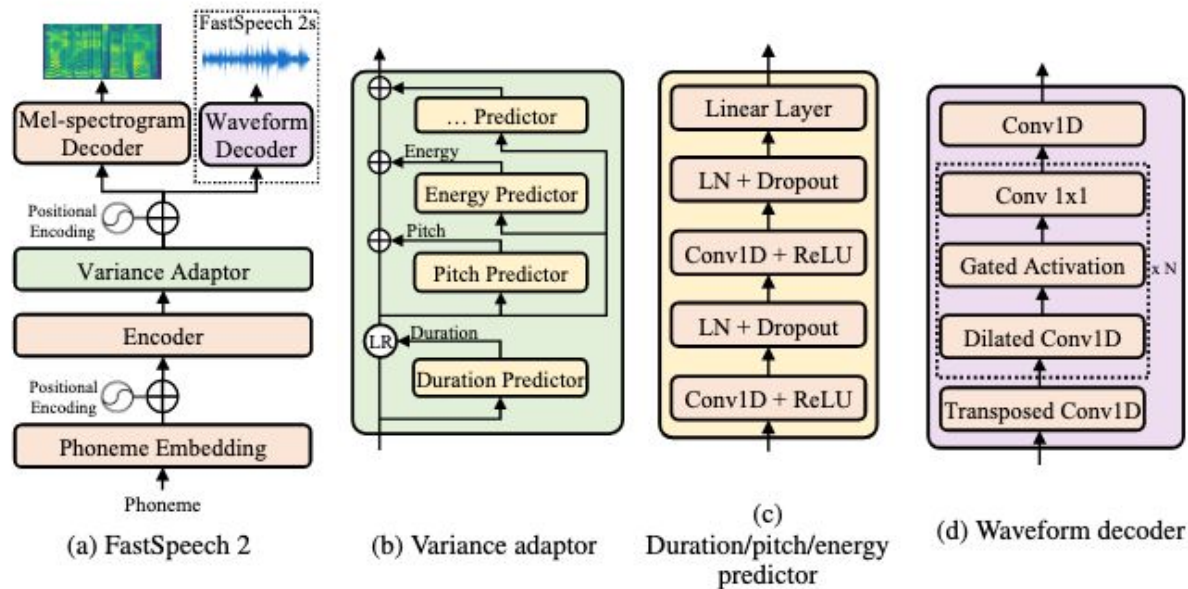
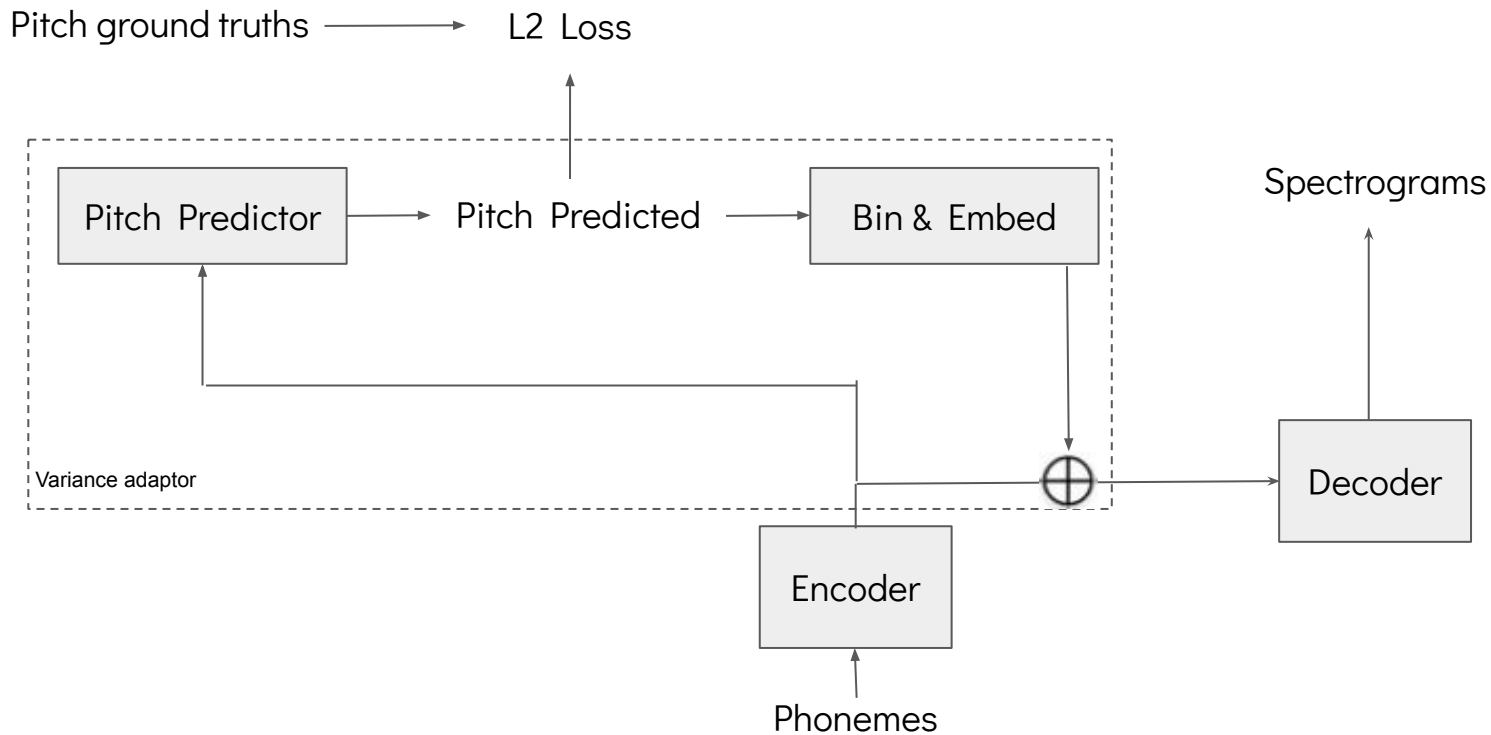


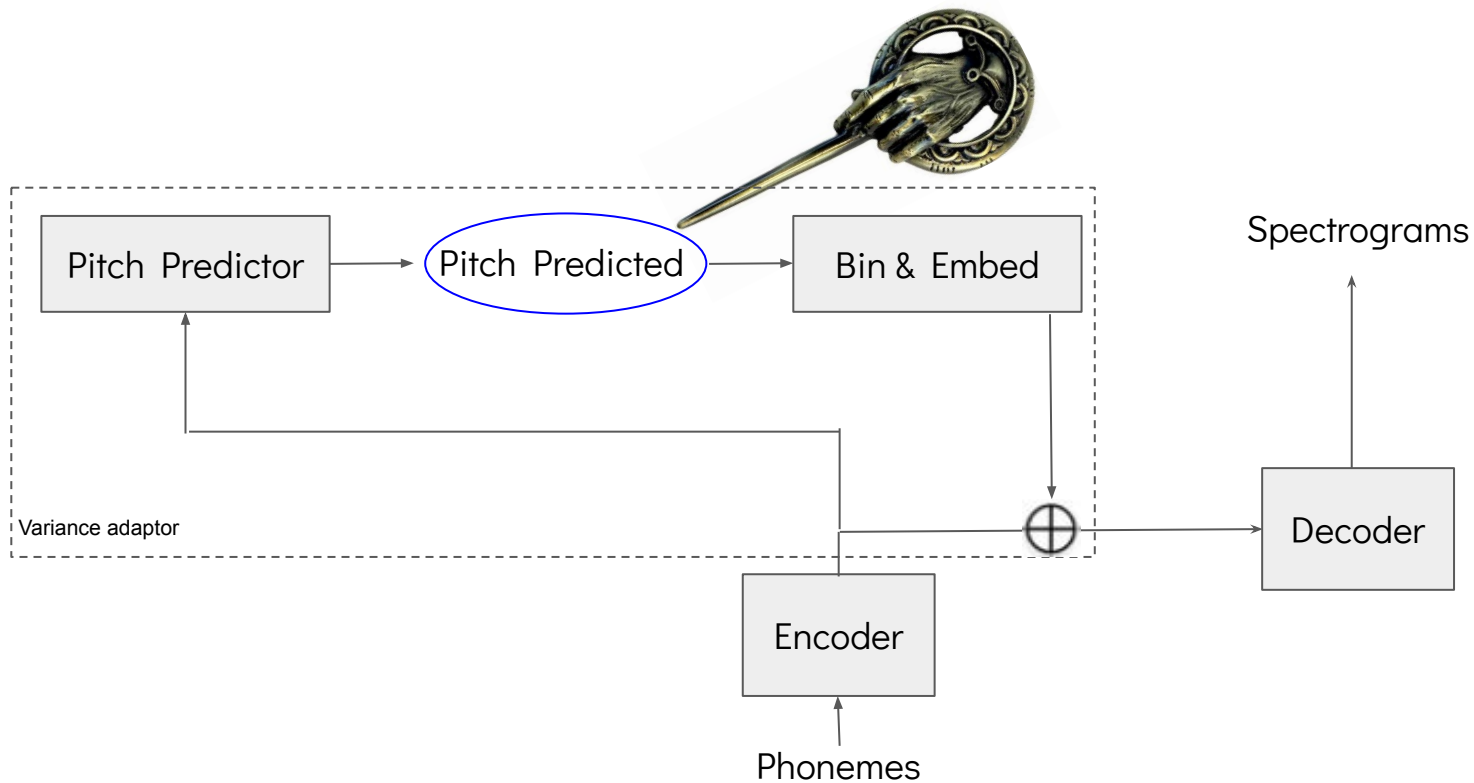
Figure 1: The overall architecture for FastSpeech 2 and 2s. LR in subfigure (b) denotes the length regulator proposed in FastSpeech. LN in subfigure (c) denotes layer normalization.

<https://arxiv.org/pdf/2006.04558.pdf>

Variance Adaptor in training



Variance Adaptor in inference



Zero-shot speaker adaptation

Train a TTS with input (text, ref speaker wav) to mimic voice of ref speaker

Speaker ref: 

“thank you for calling optum my name is grace how may i help you today”

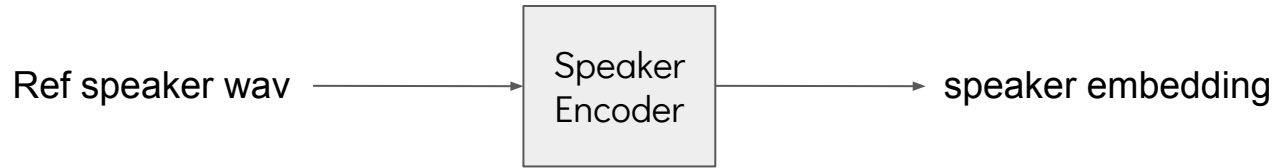
Output: 

Zero-shot speaker adaptation

Train a TTS with input (text, ref speaker wav) to mimic voice of ref speaker

Preliminary: train speaker encoder, on dataset of (wav, speaker_id) pairs

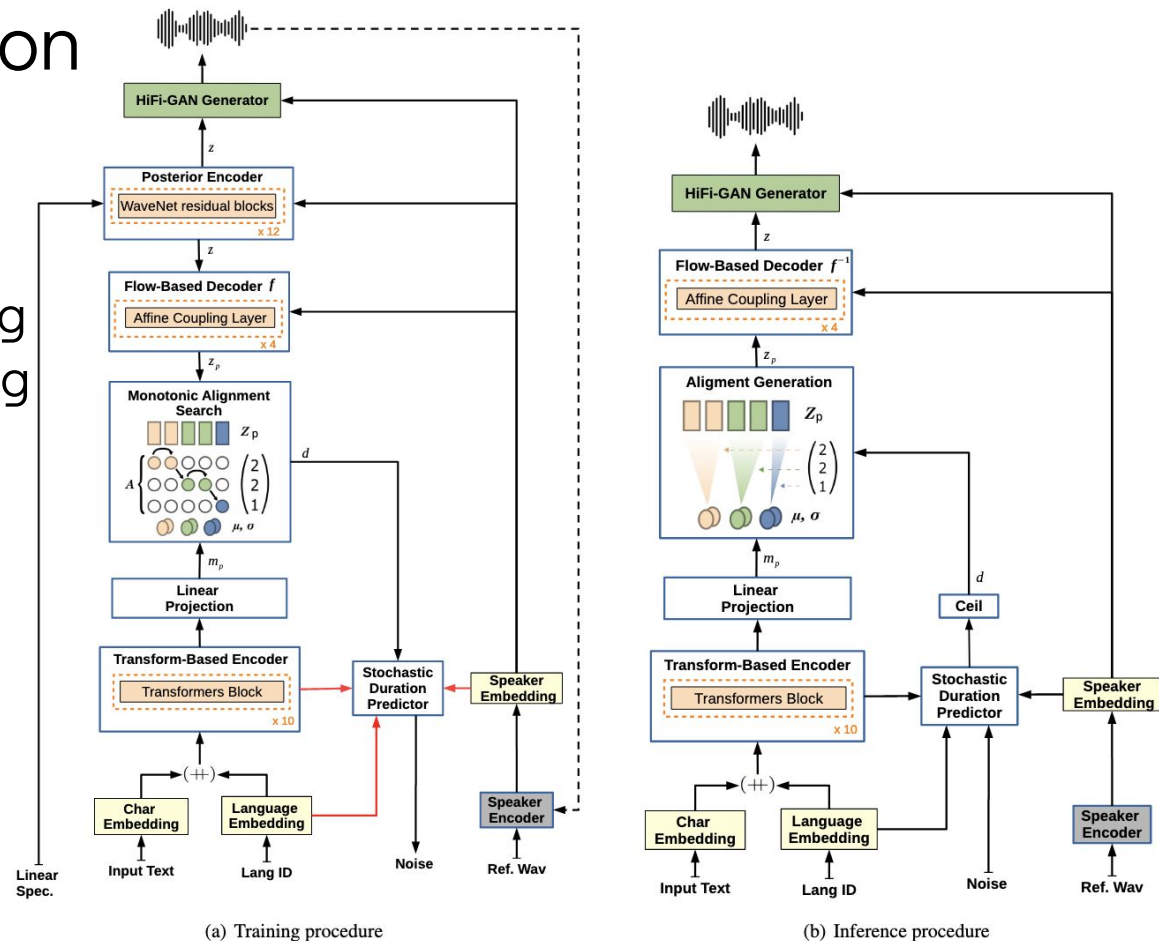
Idea: Encode voice style but not the texts



s. t. wavs with same speaker id -> “near” embeddings,
wavs with different speaker ids -> “far” embeddings

Speaker adaptation

YourTTS
Variation of VITS
w/ speaker embedding
& language embedding



Casanova et al.
<https://arxiv.org/pdf/2112.02418.pdf>

TTS ethics

Modern TTS is fully capable of fooling humans

Only uses ref speaker's voice with permission

Disclose your dialog system is a bot

“hello this is lokman please send ten thousand dollars to me”

Output:




An Exercise to the Reader for Next Time

- How would you normalize the following sentence:
“Louis XI owes president Xi \$1,911.11, in the year 1911”?
- Go to [Microsoft's TTS](#), generate a speech wav for the following text:
“The time has come. Execute order 66”
Please make it sound as sinister as possible
- Compared to training general ASR, does single speaker TTS require more or less data? Is TTS model size larger or smaller than ASR?

Exercises from Last Time

- With HMM model in 27p, what would be the most probable state when your spouse sent 😄 😞 😄 ?
 $P(\text{SFS}, \text{HHH}) = P(\text{S}|\text{H}) * P(\text{H}|\emptyset) * P(\text{F}|\text{H}) * P(\text{H}|\text{H}) * P(\text{S}|\text{H}) * P(\text{H}|\text{H}) = 0.9 * 0.8 * 0.1 * 0.7 * 0.9 * 0.7 = 0.031752$
 $P(\text{SFS}, \text{HBH}) = P(\text{S}|\text{H}) * P(\text{H}|\emptyset) * P(\text{F}|\text{B}) * P(\text{B}|\text{H}) * P(\text{S}|\text{H}) * P(\text{H}|\text{B}) = 0.9 * 0.8 * 0.8 * 0.3 * 0.9 * 0.4 = 0.062208$
 $P(\text{SFS}, \text{HBB}) = P(\text{S}|\text{H}) * P(\text{H}|\emptyset) * P(\text{F}|\text{B}) * P(\text{B}|\text{H}) * P(\text{S}|\text{B}) * P(\text{B}|\text{B}) = 0.9 * 0.8 * 0.8 * 0.3 * 0.2 * 0.6 = 0.020736$
 $P(\text{SFS}, \text{BBB}) = P(\text{S}|\text{B}) * P(\text{B}|\emptyset) * P(\text{F}|\text{B}) * P(\text{B}|\text{B}) * P(\text{S}|\text{B}) * P(\text{B}|\text{B}) = 0.2 * 0.2 * 0.8 * 0.6 * 0.2 * 0.6 = 0.002304$
 $P(\text{SFS}, \text{HHB}) = P(\text{S}|\text{H}) * P(\text{H}|\emptyset) * P(\text{F}|\text{H}) * P(\text{H}|\text{H}) * P(\text{S}|\text{B}) * P(\text{B}|\text{H}) = 0.9 * 0.8 * 0.1 * 0.7 * 0.2 * 0.3 = 0.003024$
 $P(\text{SFS}, \text{BHH}) = P(\text{S}|\text{B}) * P(\text{B}|\emptyset) * P(\text{F}|\text{H}) * P(\text{H}|\text{B}) * P(\text{S}|\text{H}) * P(\text{H}|\text{H}) = 0.2 * 0.2 * 0.9 * 0.4 * 0.9 * 0.7 = 0.001008$
 $P(\text{SFS}, \text{BHB}) = P(\text{S}|\text{B}) * P(\text{B}|\emptyset) * P(\text{F}|\text{H}) * P(\text{H}|\text{B}) * P(\text{S}|\text{B}) * P(\text{B}|\text{H}) = 0.2 * 0.2 * 0.9 * 0.4 * 0.2 * 0.3 = 0.000096$
 $P(\text{SFS}, \text{BBH}) = P(\text{S}|\text{B}) * P(\text{B}|\emptyset) * P(\text{F}|\text{B}) * P(\text{B}|\text{B}) * P(\text{S}|\text{H}) * P(\text{H}|\text{B}) = 0.2 * 0.2 * 0.8 * 0.6 * 0.9 * 0.4 = 0.006912$
*S:Smile, F:Frown, H:Happy, B:Bad
- When you have 1 hours of labeled speech data, how do you want to train speech recognition?
 - fine-tuning from pre-trained model(wav2vec 2.0, Whisper, ...)
 - web-crawling for more data, hire people for data generation & labeling
 - train smaller model
- To make noise robust speech recognition, what can you do with modern e2e speech recognition?
 - Apply SpecAugment, Data Augmentation, Real noisy training data
 - Train noise canceling model and apply before ASR

Solutions to TTS Exercises

- How would you normalize the following sentence:
“Louis XI owes president Xi \$1,911.11, in the year 1911”?
“Louis the eleventh owes president Xi one thousand nine hundred eleven dollars and eleven cents, in the year nineteen eleven”?
- Go to [Microsoft's TTS](#), generate a speech wav for the following text:
“The time has come. Execute order 66”
Please make it sound as sinister as possible 
- Compared to training general ASR, does single speaker TTS require more or less data? Is TTS model size larger or smaller than ASR?
General ASR needs to recognize many accents and many different speakers whereas single speaker TTS only needs to produce one voice. TTS requires much less data (~10hours) vs ASR (>1k hours)
TTS model size is also much smaller (~10M params vs 100M params)