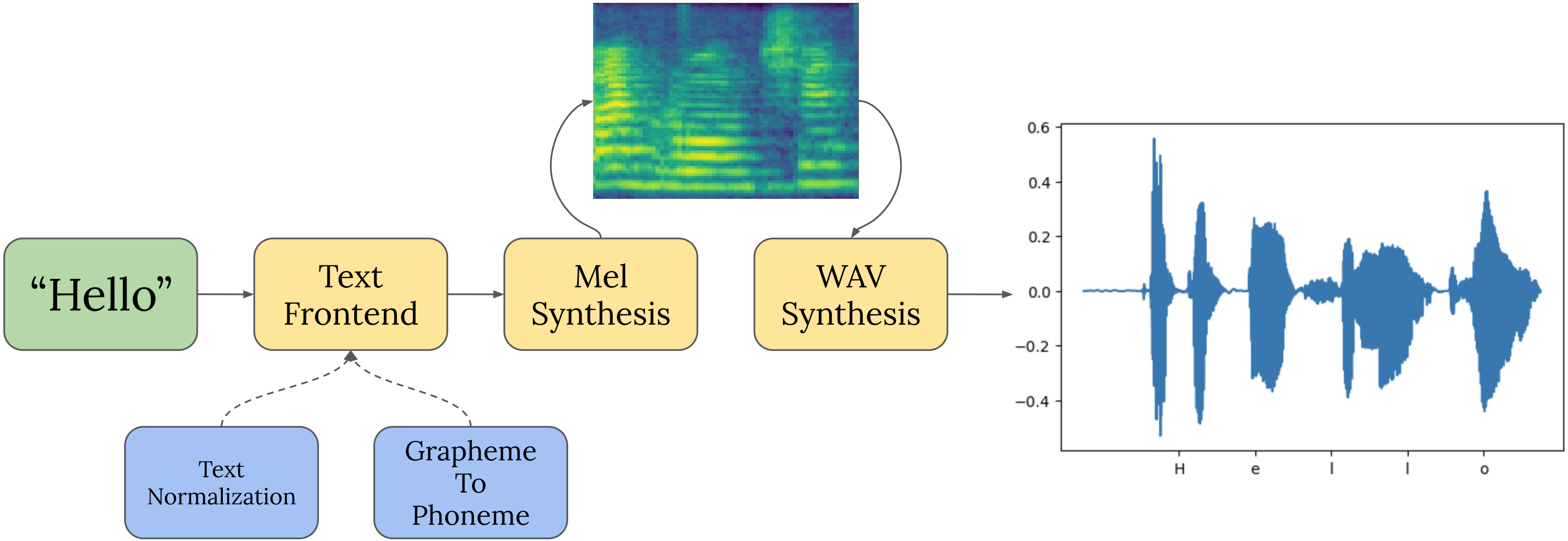
Text-to-Speech

Acoustic Models

What is a TTS?


"Hello"
Text Frontend
Mel Synthesis
WAV Synthesis
Text Normalization
Grapheme To Phoneme
0.6
0.4
0.2
0.0
−0.2
−0.4
H
e
l
l
o

How to Measure Quality?

- There’s no correct answer
- Subjective perception
- A lot of types of mistakes

How to Measure Quality?

- Overall impression
- Intelligibility
- Similarity
- Naturalness
- Pleasantness
- Intonation and pauses
- Emotions
- Listening effort

Mean Opinion Score (MOS)

- Ask people to rate audio
- Typically in the range from 1 to 5
- Average scores

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

$$MOS = \frac{\sum_{n=1}^{N} R_n}{N}$$

Mean Opinion Score (MOS)

- Crowdsourcing
- Confidence intervals

$$CI = \bar{x} \pm z\frac{s}{\sqrt{n}}$$

Mean Opinion Score (MOS)

- Large variance due to human factor
- Less variance more money
- Hard to catch “hard” mistakes

Side-by-Side

- Compare pair of instances and choose more “preferable”
- Answer is binary
- Possibly evaluate small improvements
- Sensitive to “hard” mistakes

What About Public Datasets?

- LJSpeech
- LibriTTS
- CommonVoice
- OpenTTS (Russian)

Wait! Why are we introducing MelSpectrogram?

- Train two component separately
    - Faster and easier
- MelSpectrogram is smoother than waveform
- Easier to train using a MSE

Grapheme VS Phoneme

- A phoneme is simply a sound
- A grapheme is the smallest fundamental unit in written language
- There isn't bijection because depend on context

Phoneme	Example	Translation
AA	odd	AA D
AE	at	AE T
AH	hut	HH AH T
AO	ought	AO T
AW	cow	K AW

Acoustic Models

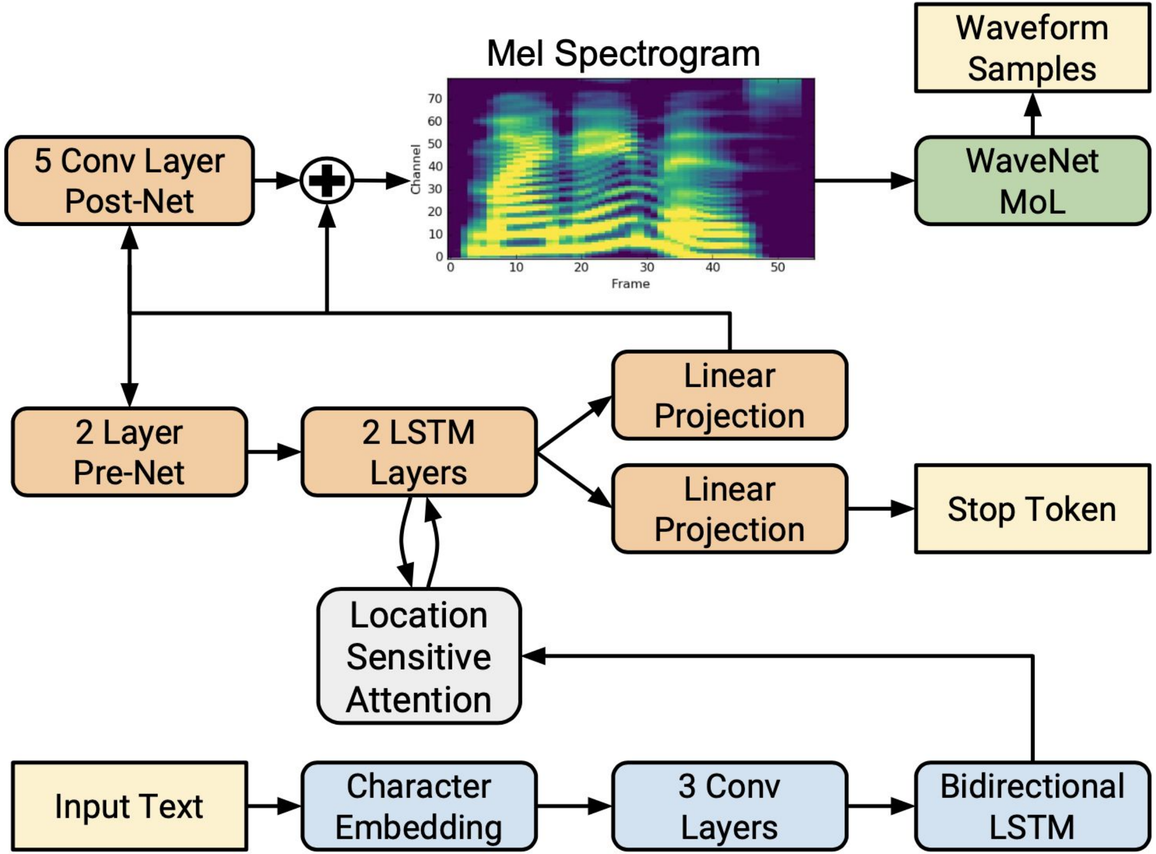
- Tacotron
    - Guided Attention
- GST & SE
- FastSpeech

Tacotron 2

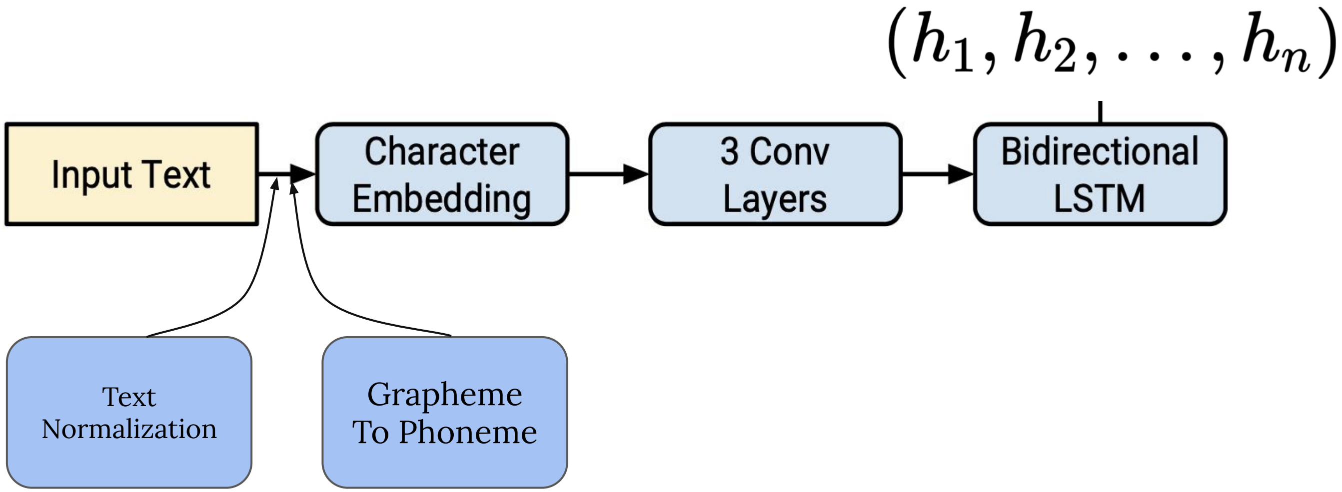
- Text Encoder
- Attention
- MelSpectrogram Decoder
- Pre&Post Nets
- MSE & CE criterions

Tacotron 2

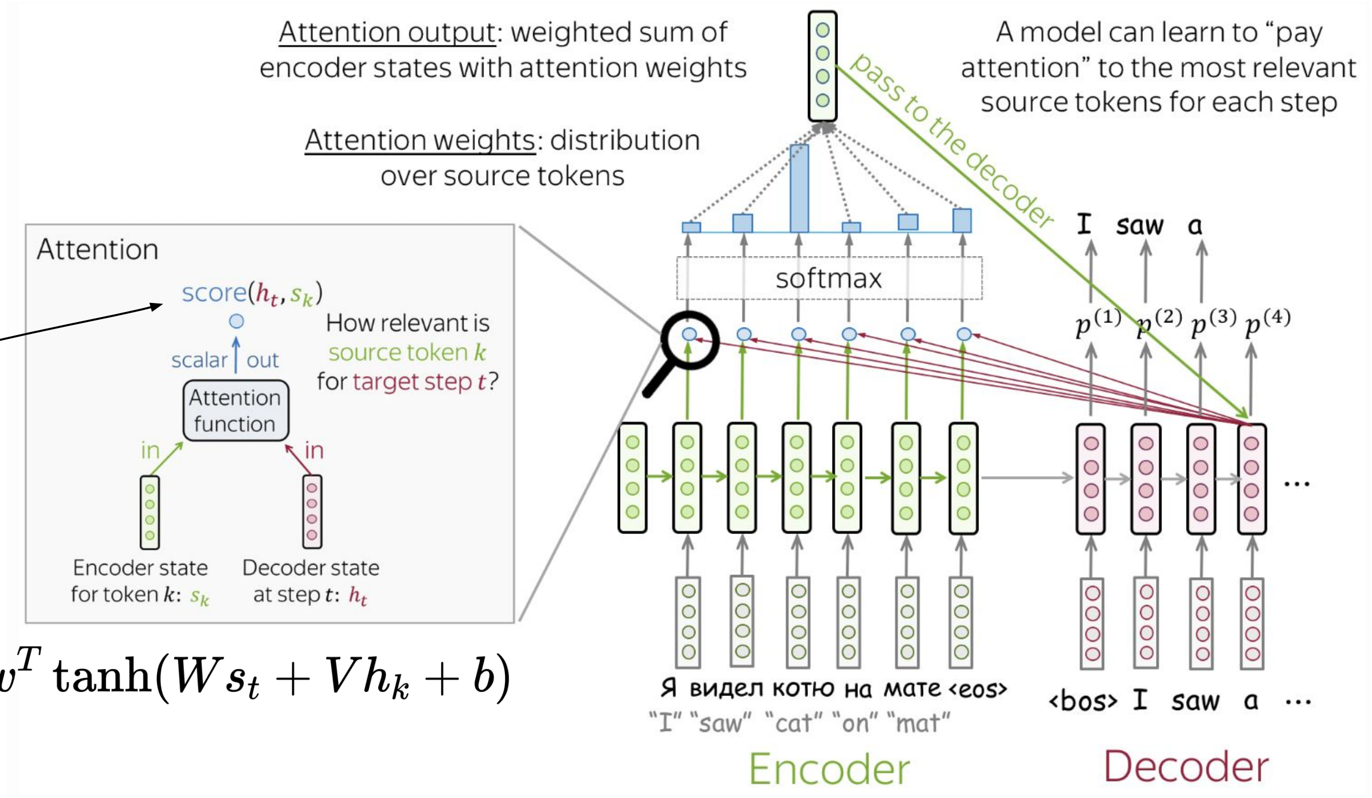
$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{pre}} + \mathcal{L}_{\text{post}} + \text{StopToken} \\ \mathcal{L}_{\text{pre}} &= \text{MSE}(x, \hat{x}_{\text{pre}}) \\ \mathcal{L}_{\text{pre}} &= \text{MSE}(x, \hat{x}_{\text{post}}) \\ \text{StopToken} &= \text{CE}(h, \mathbb{I}[h = \text{stop}])\end{aligned}$$



Text Encoder



Attention



$$\text{score}_{t,k} = w^T \tanh(W s_t + V h_k + b)$$

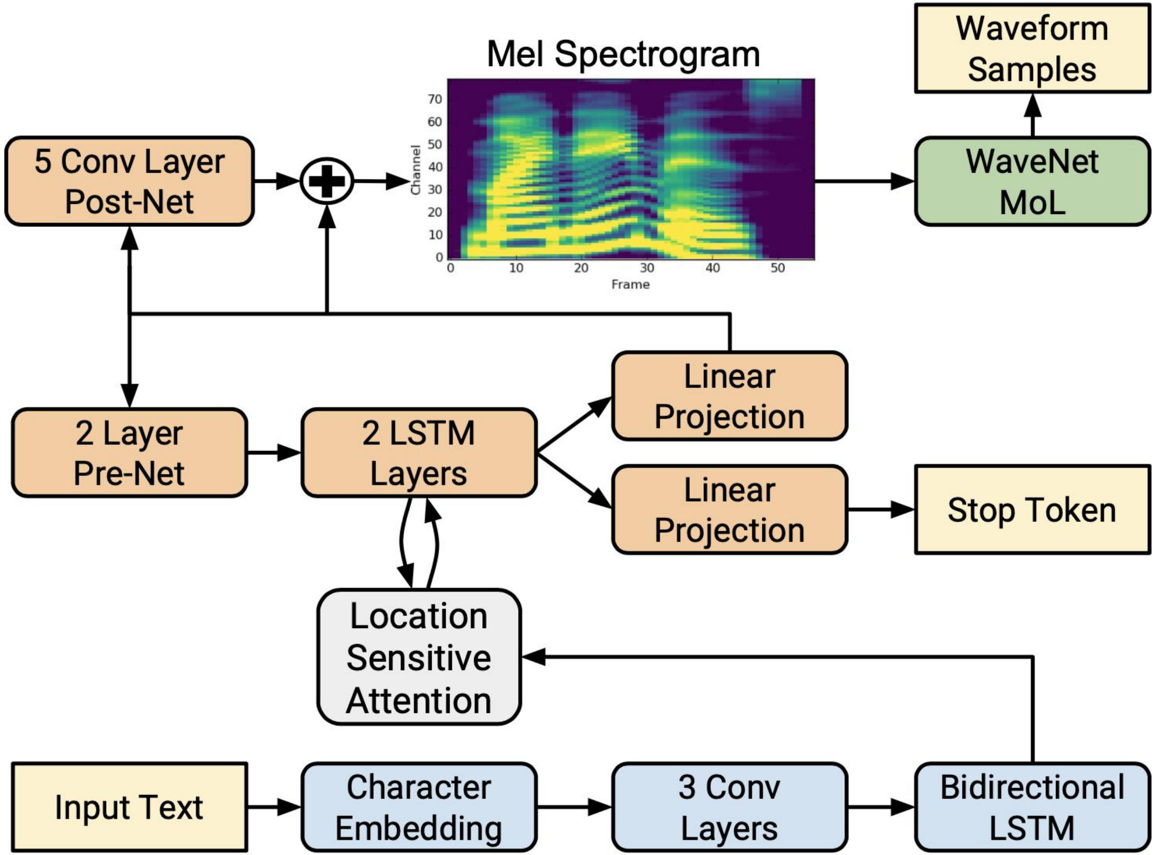
https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Tacotron 2

$$\alpha_i = \mathrm{Attend}(s_{i-1}, \alpha_{i-1}, h)$$
$$g_i = \sum_{j=1}^{L} \alpha_{i,j} h_j$$
$$y_i \sim \mathrm{Generate}(s_{i-1}, g_i)$$

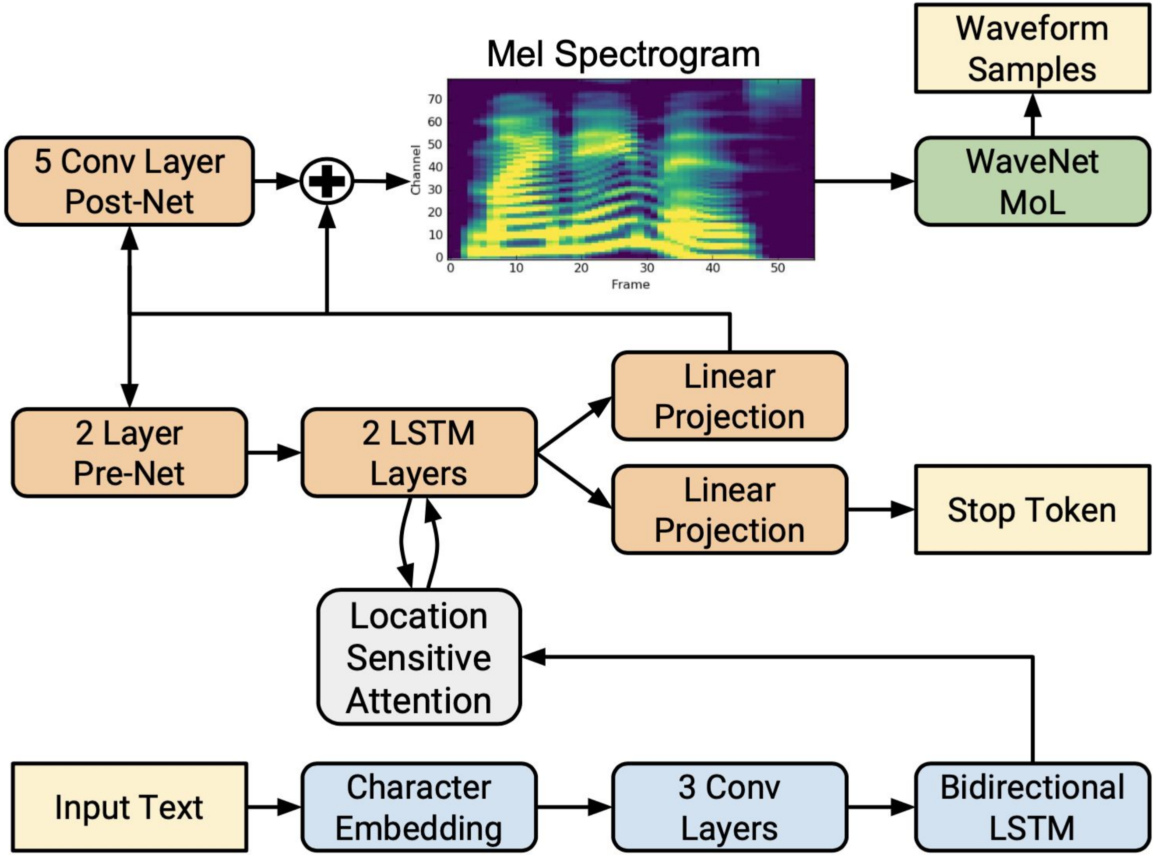
$$e_{i,j} = \mathrm{Score}(s_{i-1}, \alpha_{i-1,j}, h_j)$$
$$\alpha_{i,j} = \exp(e_{i,j}) / \sum_{j=1}^{L} \exp(e_{i,j})$$

$$f_i = F * \alpha_{i-1}$$
$$e_{i,j} = w^\top \tanh(W s_{i-1} + V h_j + U f_{i,j} + b)$$



Tacotron 2

- Neighboring predicted mels are correlated
- Turn on dropout on inference in PreNet

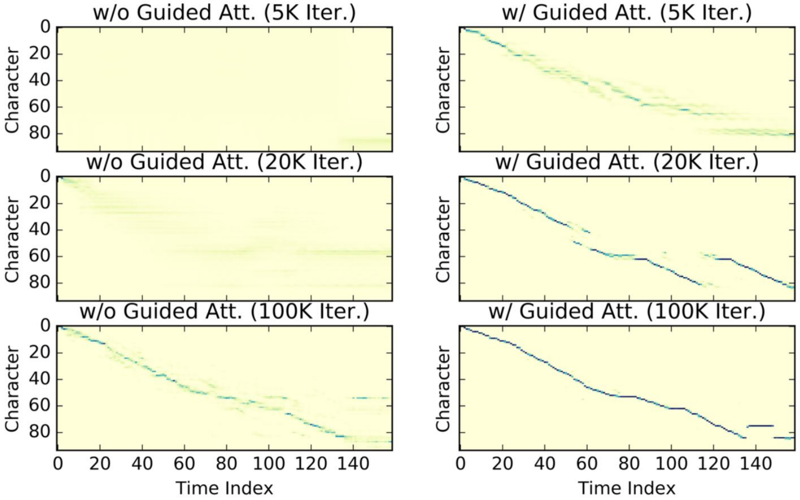


Guided Attention

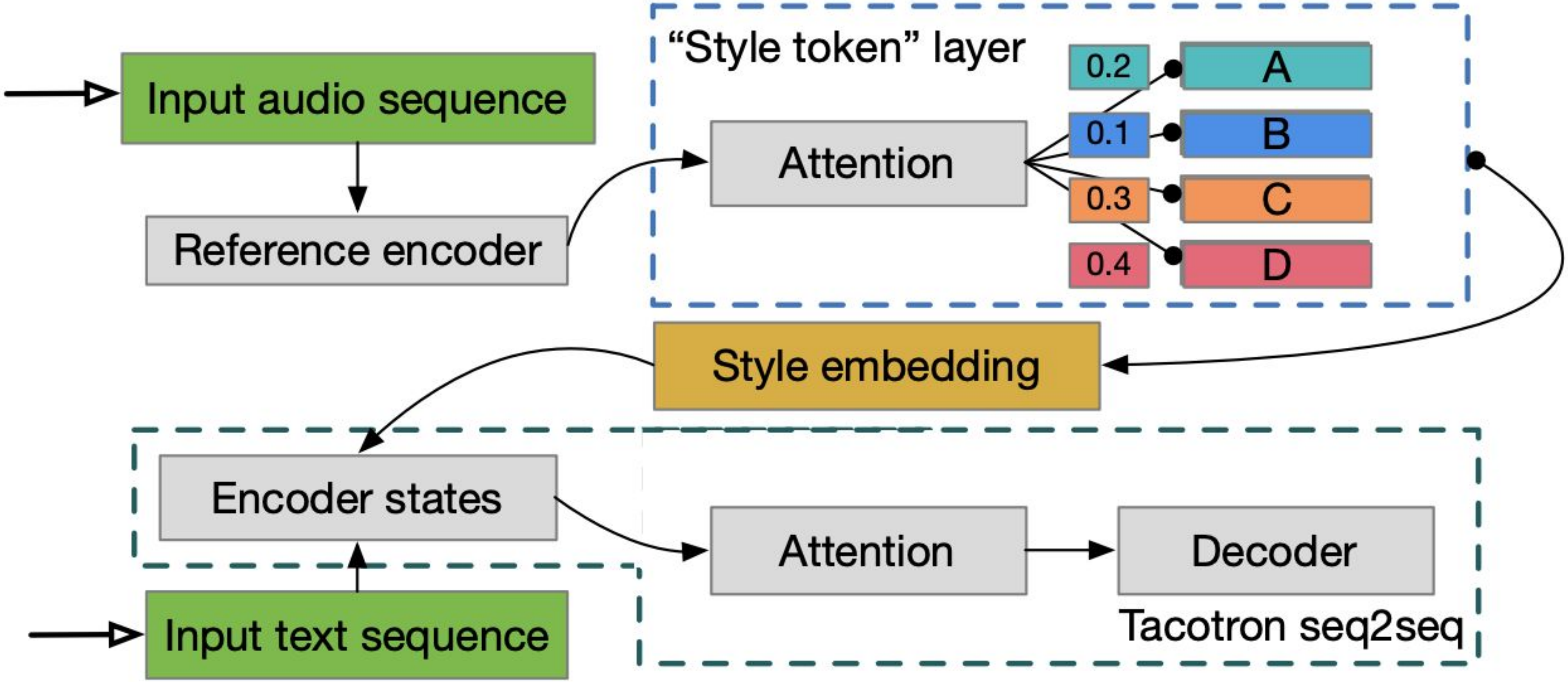
- Hard to learn attention from scratch
- Text position n progresses linearly to time t:

$$n \sim at, \text{ where } a \sim N/T$$

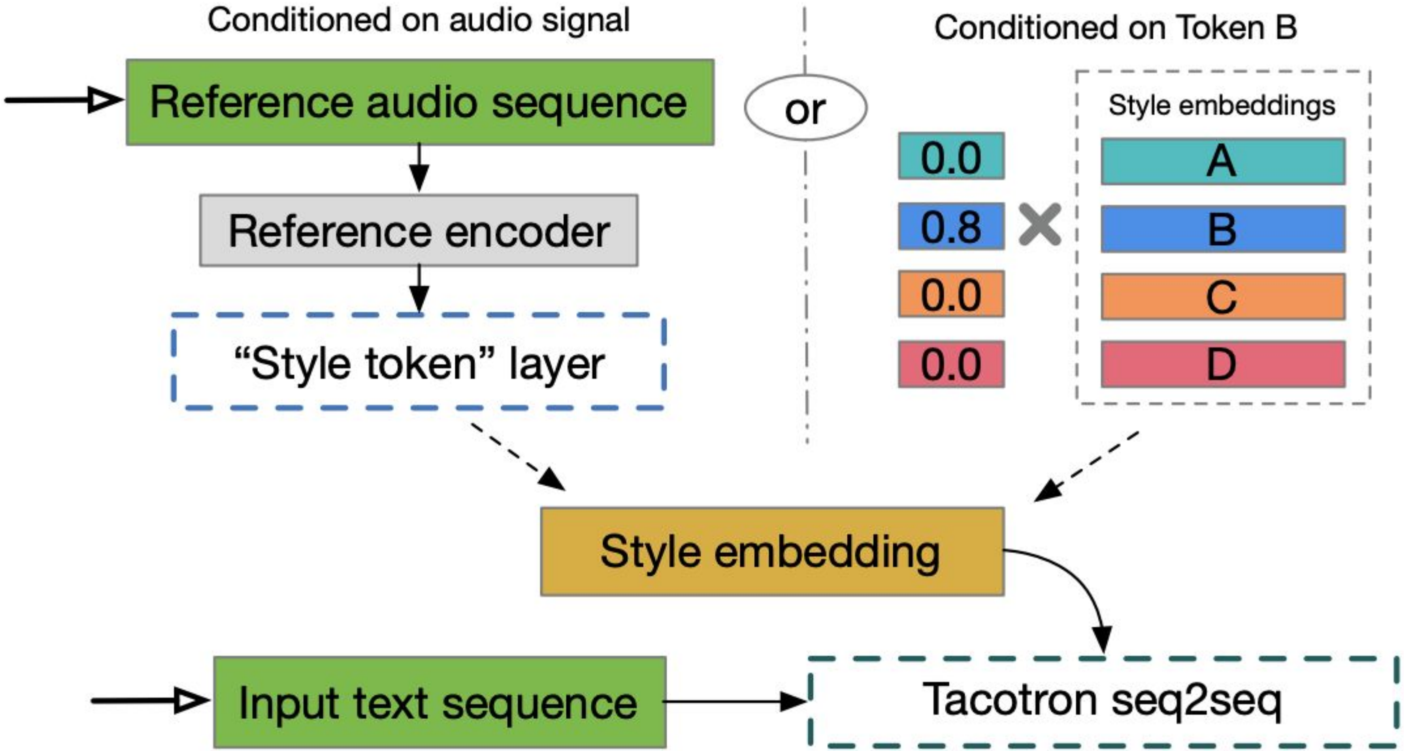
- Prompts attention matrix A to be "diagonal"
- $\mathcal{L}_{\text{att}}(A) = \mathbb{E}_{nt}[A_{nt}W_{nt}], \text{ where } W_{nt} = 1 - \exp\{-(n/N - t/T)^2/2g^2\}$



Global Style Token (GST)

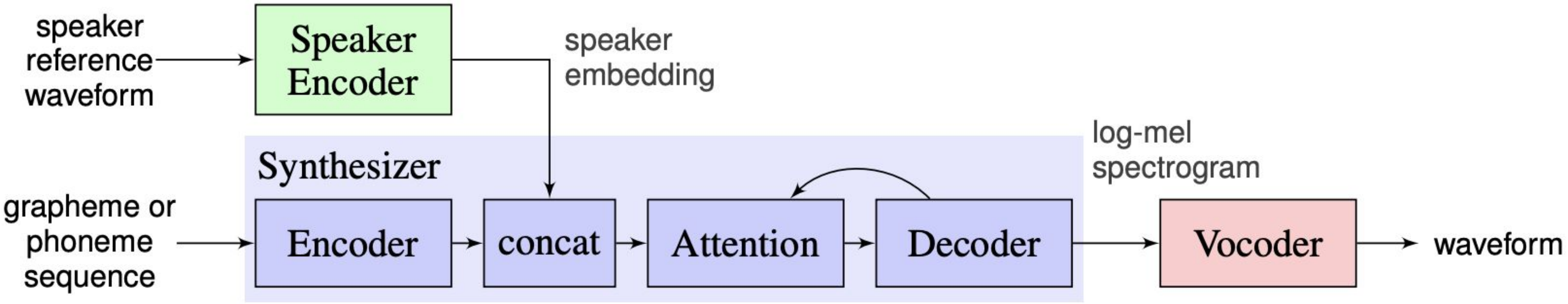


Global Style Token (GST)



Multispeaker TTS

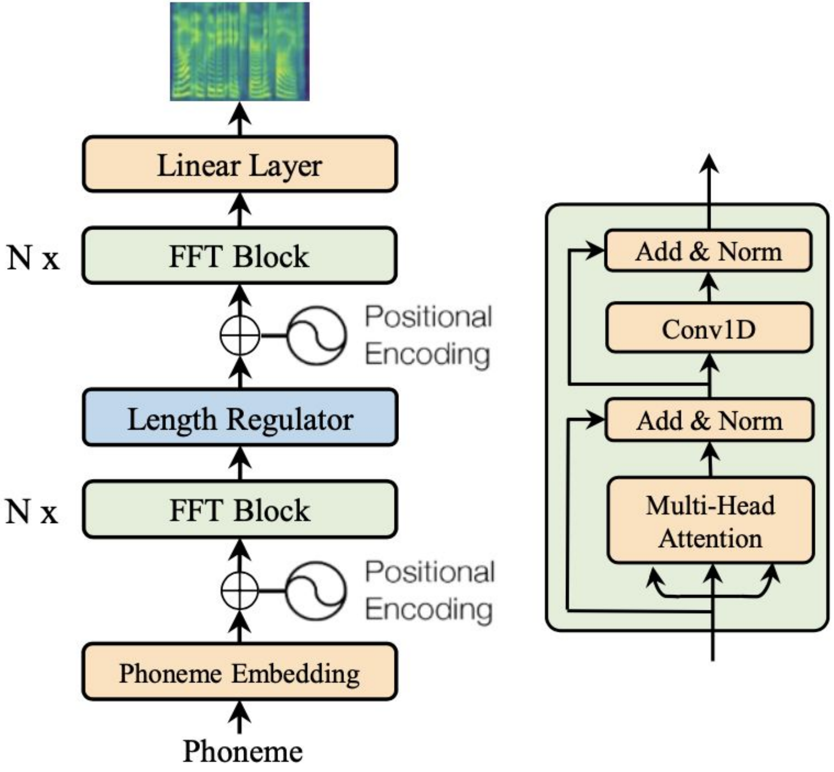
- Speaker encoder (SE) trained on a speaker verification task
- Text-independent task
- Thousands of speakers



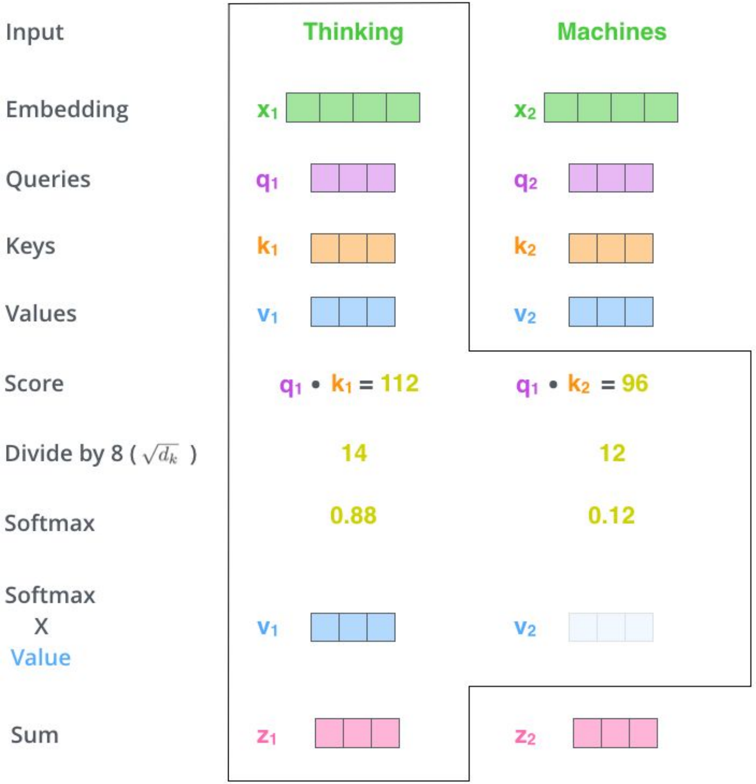
FastSpeech

- Parallel Mel Generation
- Feed-forward Transformer blocks
  - 1D Conv instead Linear
- Length Regulator
- Duration predictor

FastSpeech



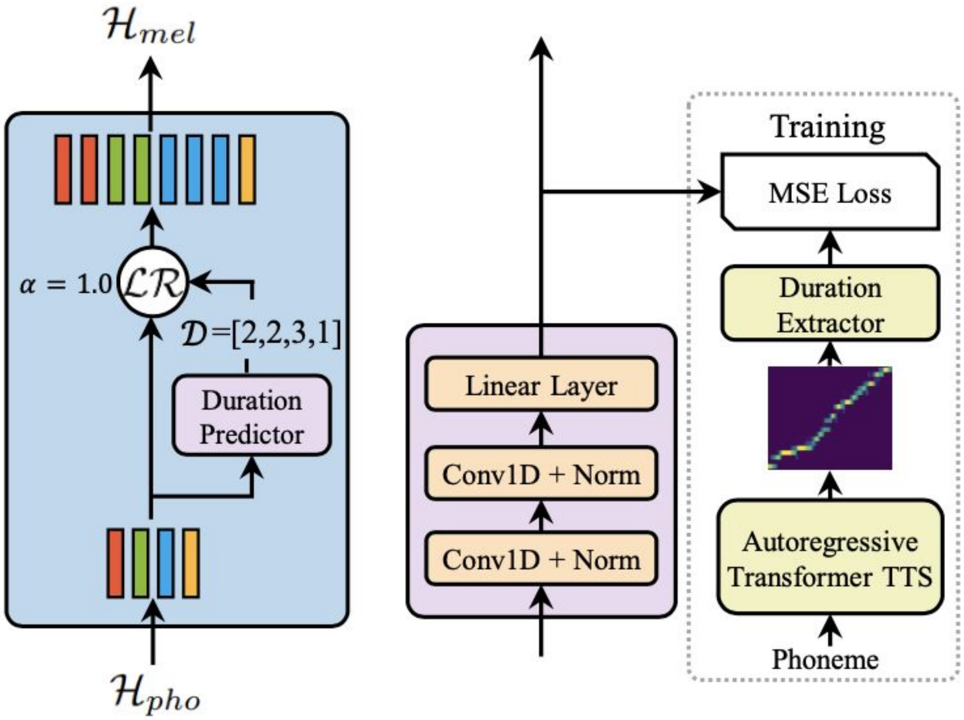
Linear Layer
N x
FFT Block
Positional Encoding
Length Regulator
N x
FFT Block
Positional Encoding
Phoneme Embedding
Phoneme
Add & Norm
Conv1D
Add & Norm
Multi-Head Attention



Input
Thinking
Machines
Embedding
x1
x2
Queries
q1
q2
Keys
k1
k2
Values
v1
v2
Score
q1 • k1 = 112
q1 • k2 = 96
Divide by 8 ($\sqrt{d_k}$)
14
12
Softmax
0.88
0.12
Softmax
X
Value
v1
v2
Sum
z1
z2

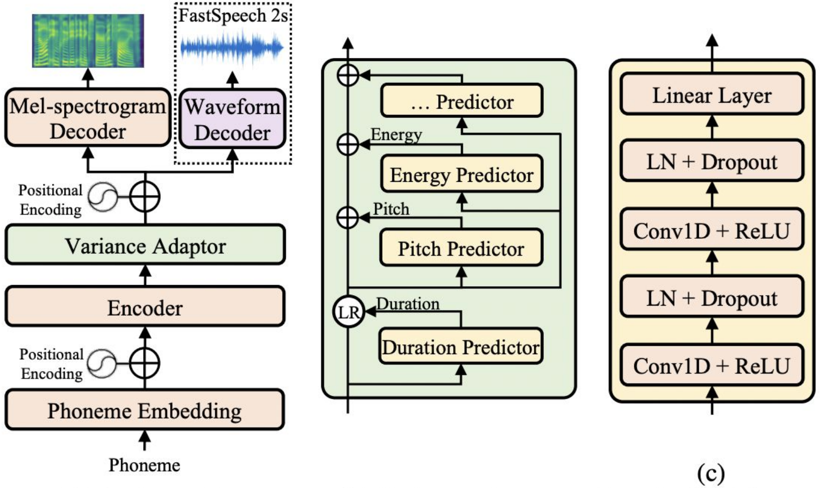
FastSpeech

- Take phoneme duration from pretrained model
- α linearly control duration



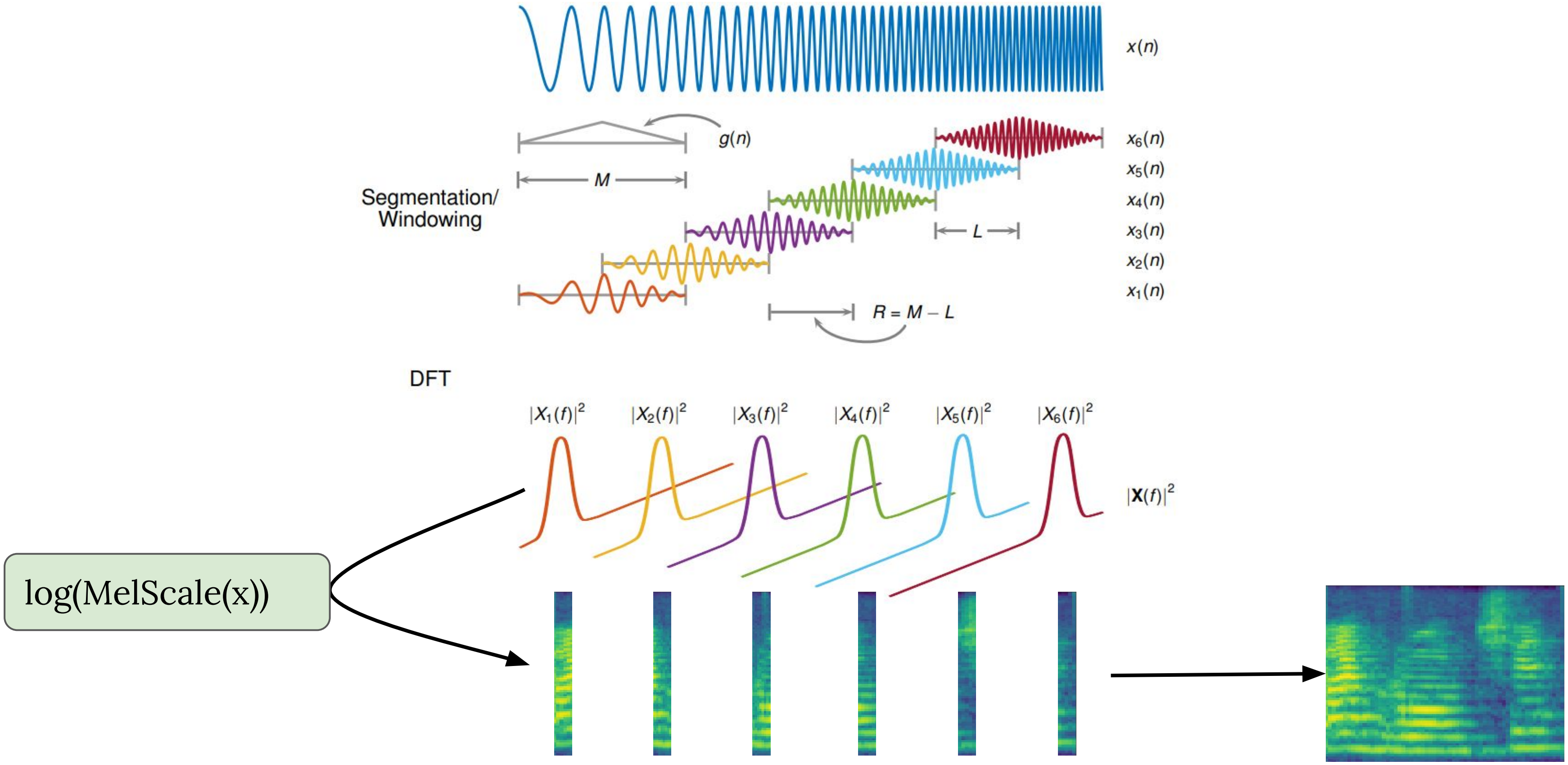
FastSpeech 2

- **Phoneme duration**, which represents how long the speech voice sounds
- **Pitch**, which is a key feature to convey emotions and greatly affects the speech prosody
- **Energy**, which indicates framelevel magnitude of mel-spectrograms



(a) FastSpeech 2 (b) Variance adaptor (c) Duration/pitch/energy predictor

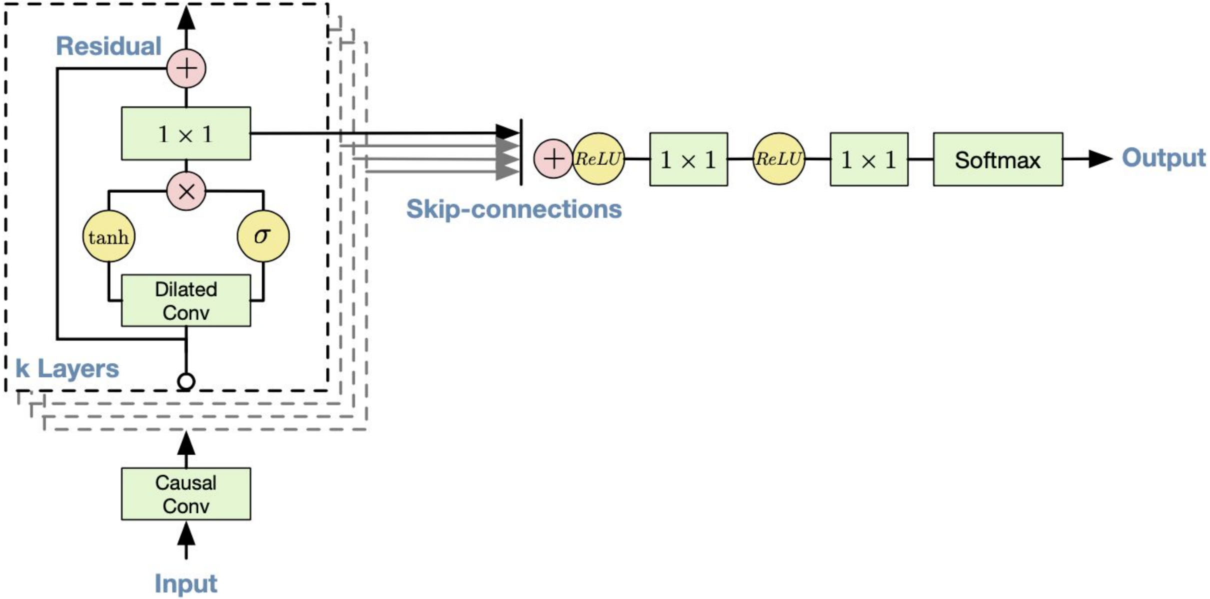
What is MelSpectrogram?



WaveNet

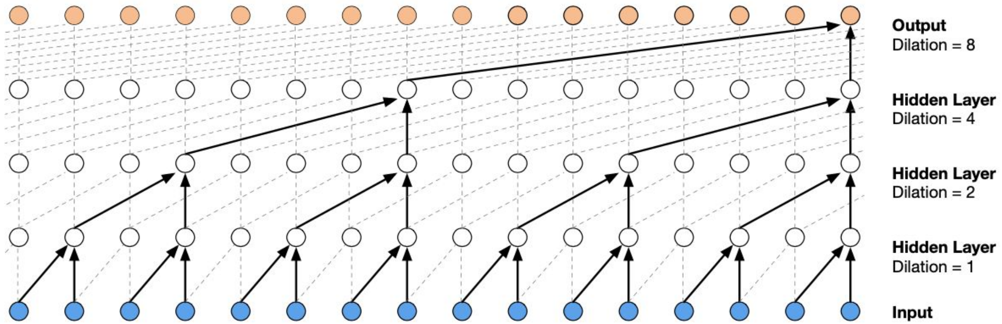
$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

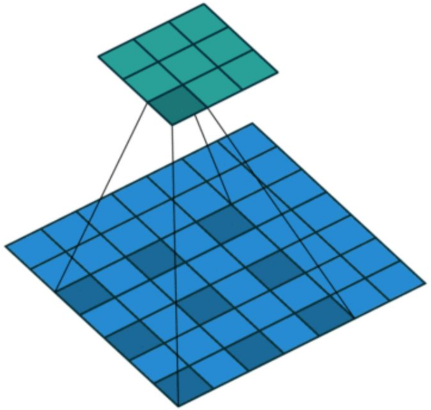
$$p(x_t | x_i, \ldots, x_{t-1}) \sim \mathrm{Cat}(\pi_\theta)$$



Dilated Convolution

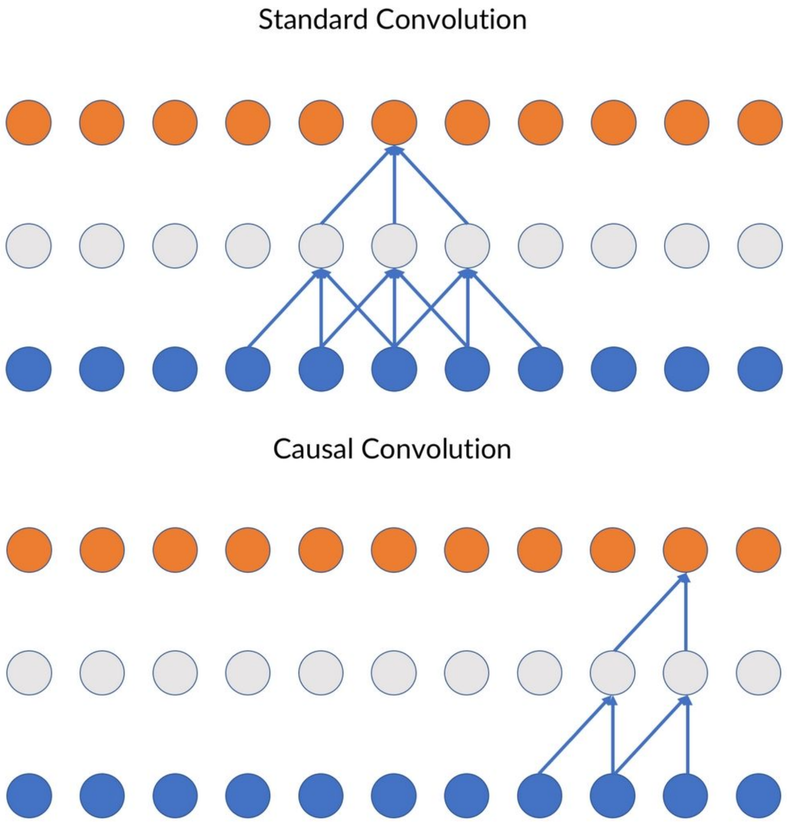
- Increase receptive field
- Allow modeling long time dependencies





Causal Convolution

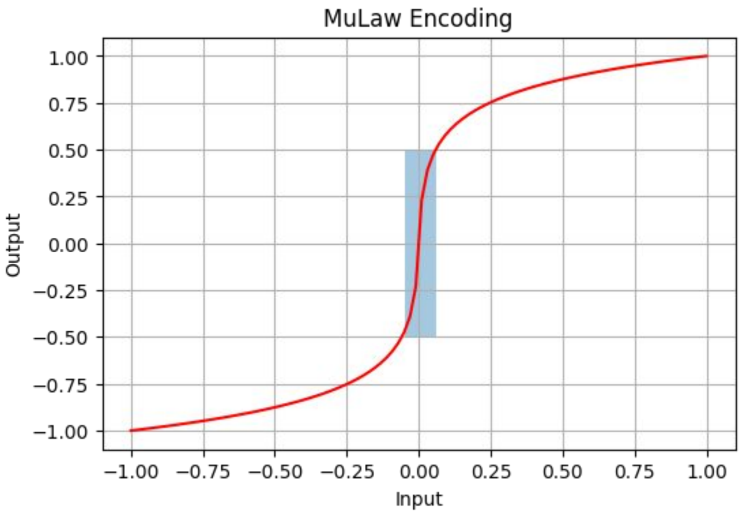
- $p(x_{t+1} \mid x_1, \ldots, x_t)$
- Don't use padding in Conv
- Use a separate Pad



Mu Law Encoding

- $$f(x_t) = \text{sign}(x_t)\frac{\ln(1+\mu|x_t|)}{\ln(1+\mu)}$$

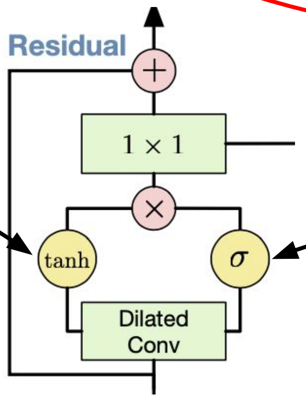
- 16-bit WAV contain 2^16 values
- Softmax will die :(
- Human hearing on a **logarithmic** scale
- **Low-amplitude** sounds in **high** resolution
- **High-amplitude** sounds in **low** resolution

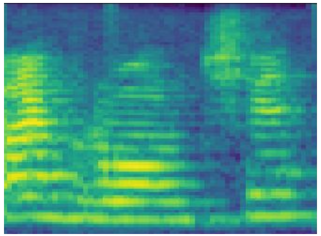


http://www.young-engineering.com/docs/YoungEngineering_ALaw_and_MuLaw_Companding.pdf

(Condition) Gated Mechanism

- $\mathbf{z} = \tanh(W_{f,k} * \mathbf{x}) \odot \sigma(W_{g,k} * \mathbf{x})$
- $\mathbf{z} = \tanh(W_{f,k} * \mathbf{x} + V_{f,k} * \mathbf{y}) \odot \sigma(W_{g,k} * \mathbf{x} + V_{g,k} * \mathbf{y})$





But how do we align the WAV and Mel?

But how do we align the WAV and Mel?

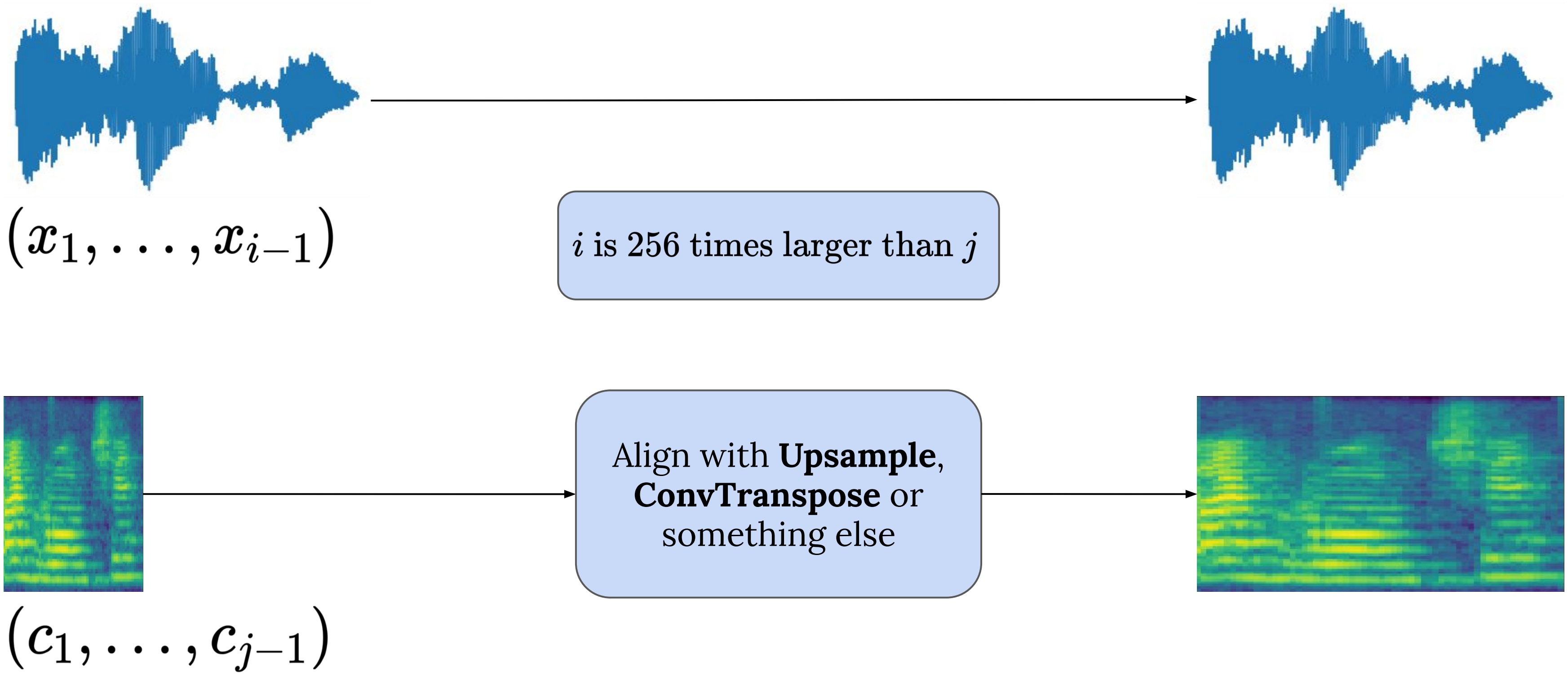
$$p(x_i | x_1, \ldots, x_{i-1})$$

add a condition using **MelSpec**

$$p(x_i | x_1, \ldots, x_{i-1}; c_1, \ldots, c_{j-1})$$

Samples and MelSpec are **not time-aligned**!

But how do we align the WAV and Mel?



Upsample is our everything!

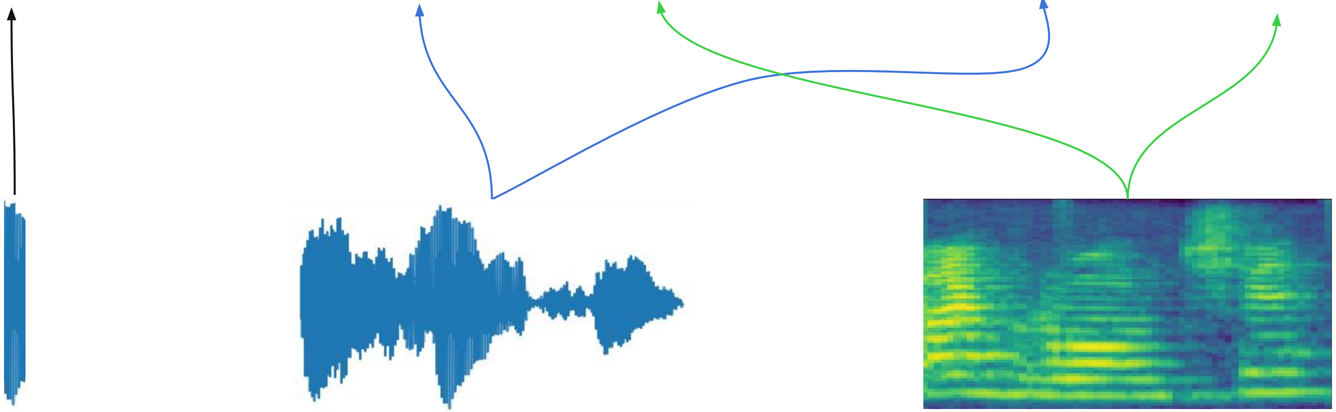

```
torch.nn.Upsample(hop_size, mode='linear')
```

```
Upsample(size=256, mode=linear)
```

```
torch.nn.ConvTranspose1d(..., stride=hop_size)
```

But how do we align the WAV and Mel?

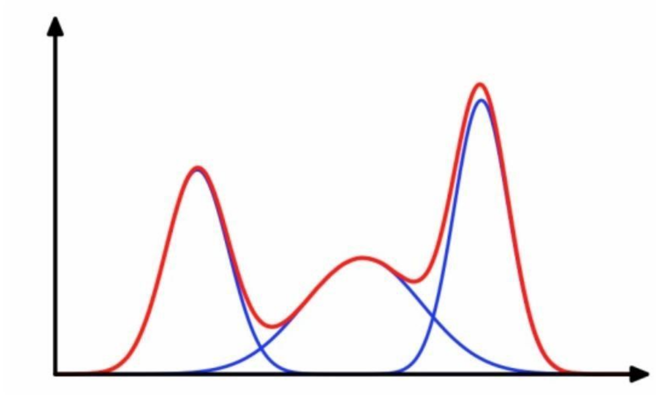
$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x} + V_{f,k} * \mathbf{y}) \odot \sigma(W_{g,k} * \mathbf{x} + V_{g,k} * \mathbf{y})$$



x_i $(x_1, \ldots, x_{i-1})$ $(c_1, \ldots, c_{i-1})$

What about a loss function?

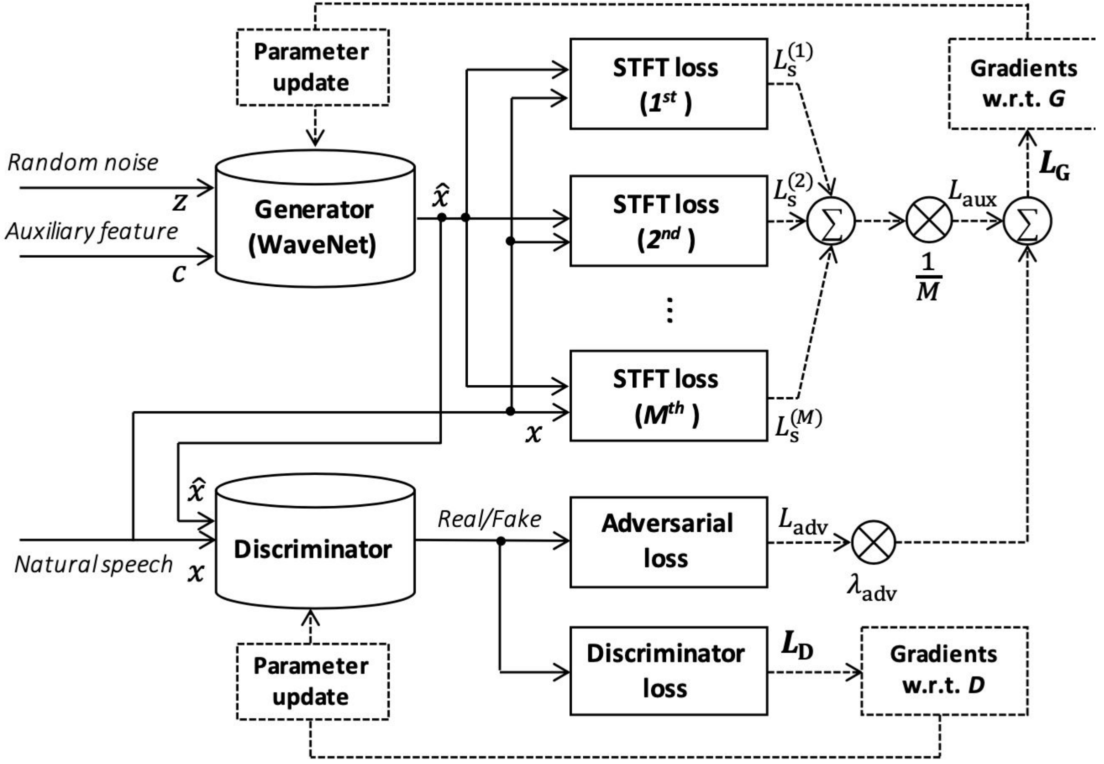
- Categorical distribution
- Normal distribution
- Logistic distribution
- Mixture of Normals or Logistics
- Use torch.distributions :)



What if… we just learn to map Mels to WAVs?

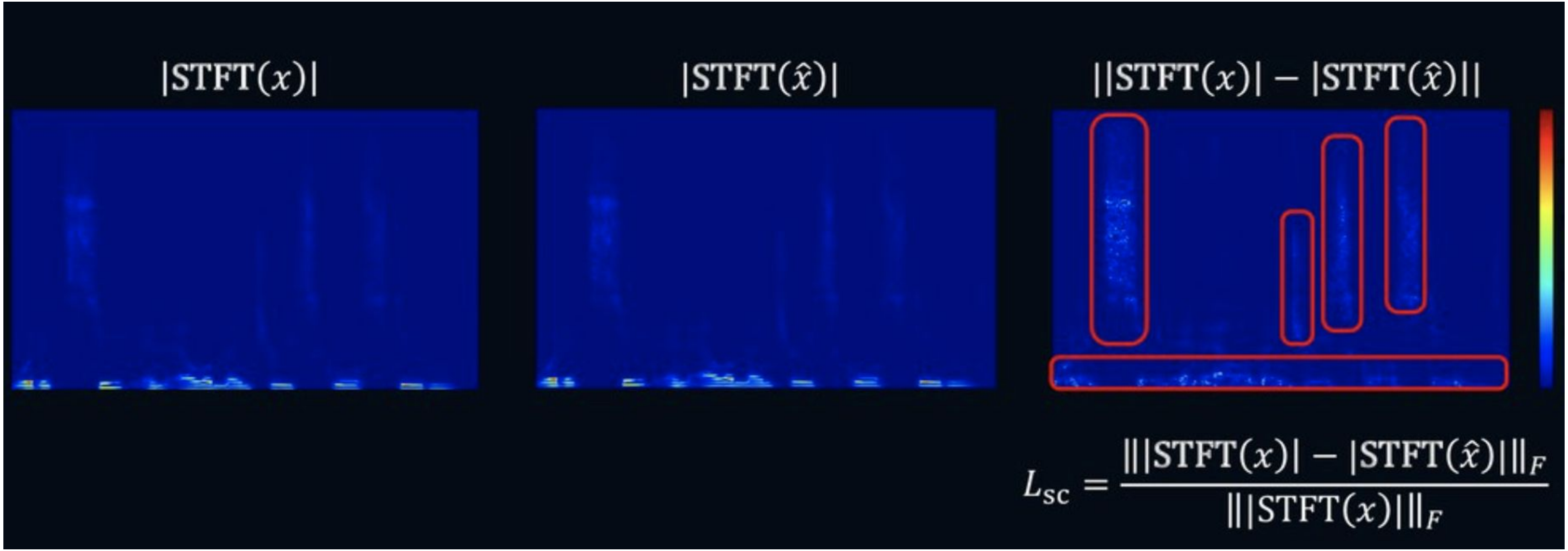
Parallel WaveGAN

- Use WaveNet based Generator
- Additionally use **multi-STFT loss**
- **LSGAN**



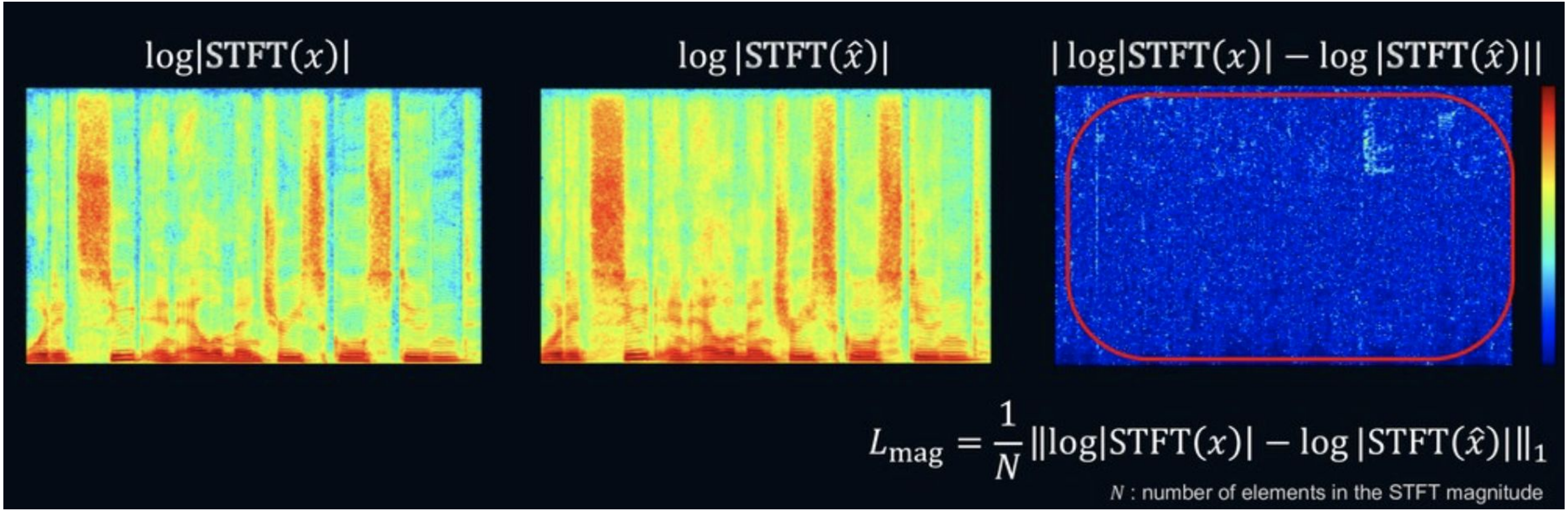
STFT Loss

Spectral Convergence part



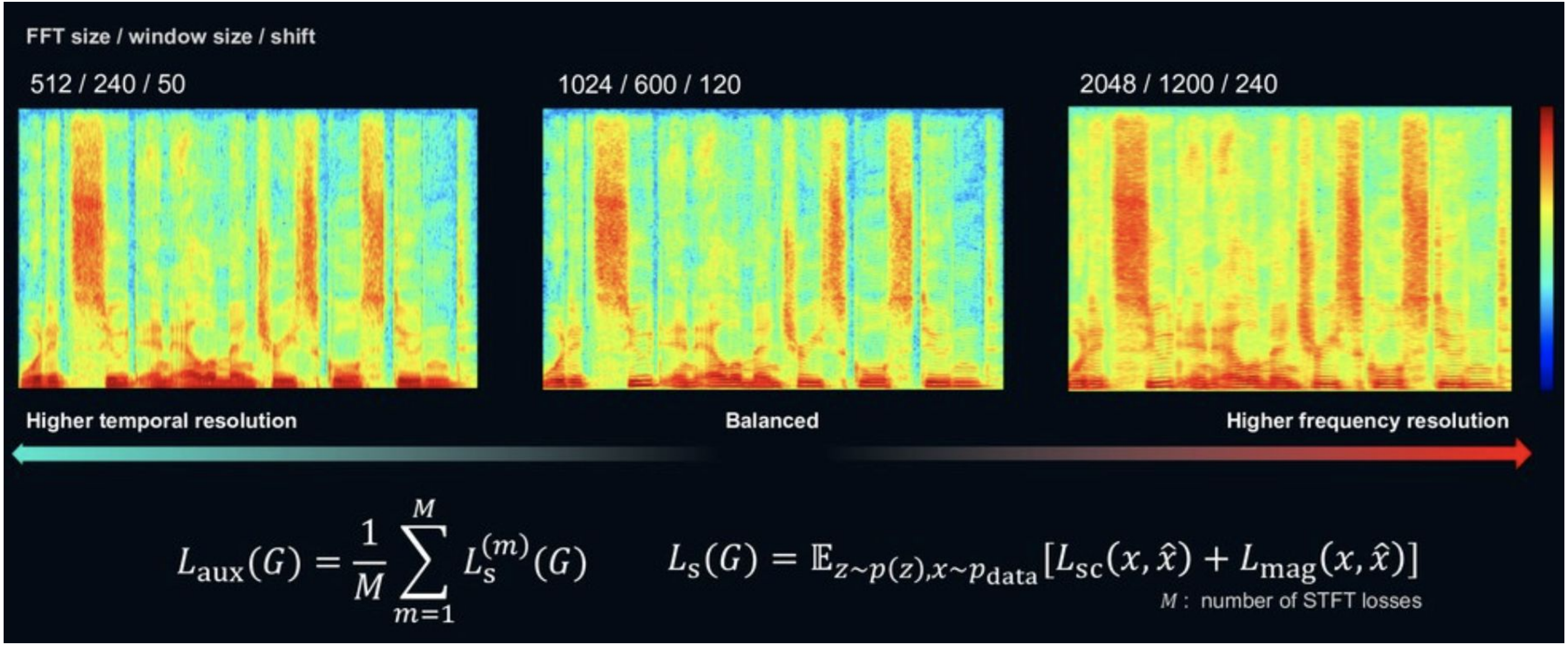
STFT Loss

Log scale STFT magnitude part



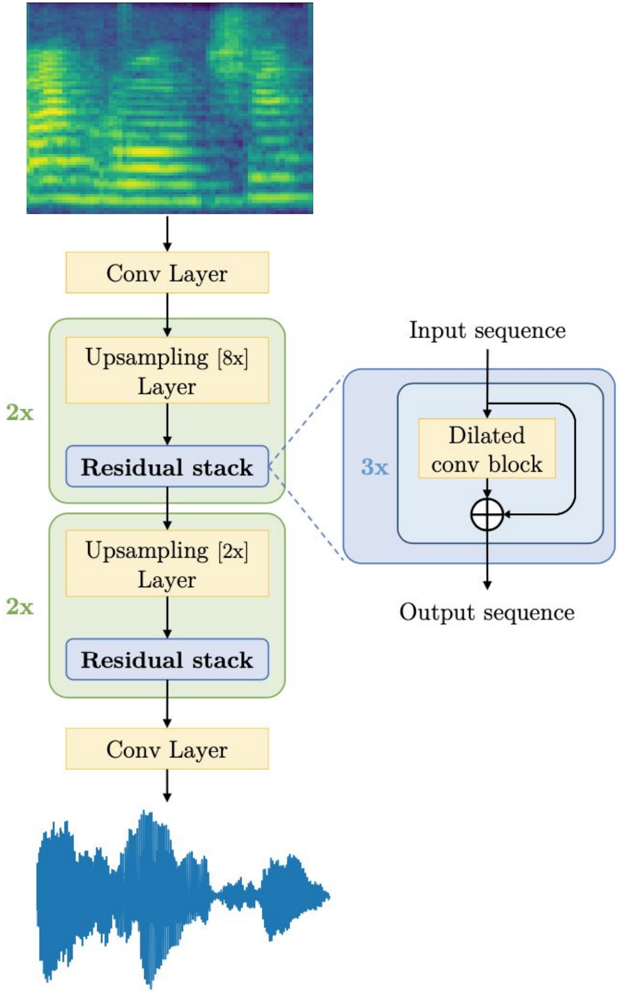
STFT Loss

Multi Resolution



MelGAN

- Non-autoregressive
- Incredibly fast and don't require any kind of distillation



MelGAN

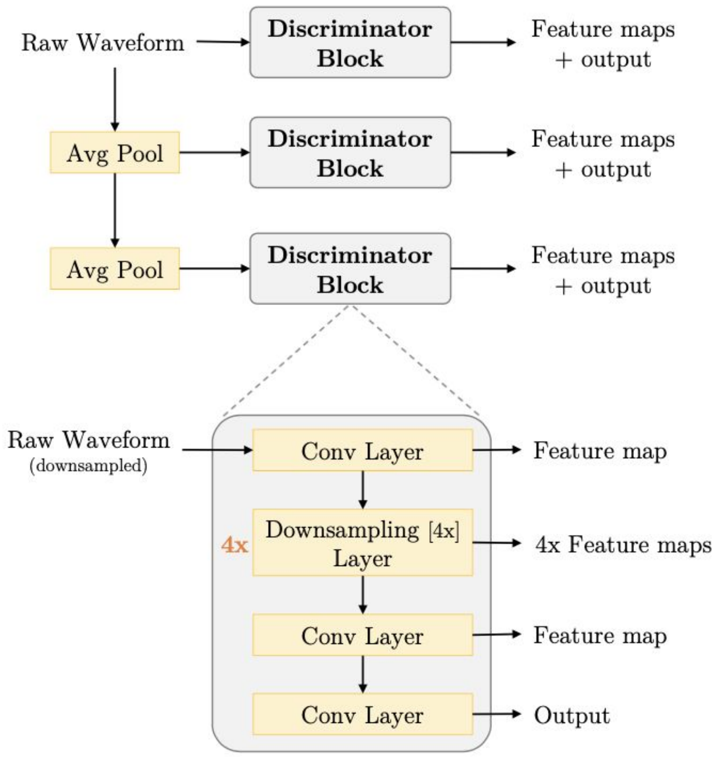
- Multiscale discriminator
- Feature Matching

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

- Hinge Loss

$$\min_{D_k} \mathbb{E}_x[\min(0, 1 - D_k(x))] + \mathbb{E}_{s,z}[\min(0, 1 + D_k(G(s,z)))], \forall k = 1,2,3$$

$$\min_G \mathbb{E}_{s,z}\left[\sum_{k=1,2,3} -D_k(G(s,z))\right]$$



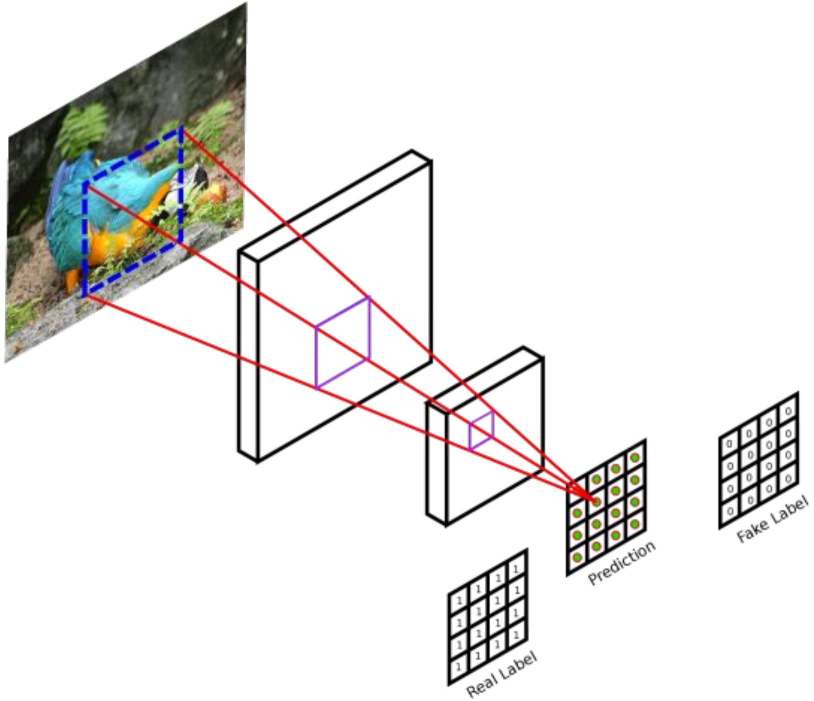
Weight Normalization

- Low-cost calculations
- Don't store additional weight
- Don't have train/test domain gap in statistics

$$\mathbf{w} = \frac{g}{\|\mathbf{v}\|}\mathbf{v}$$

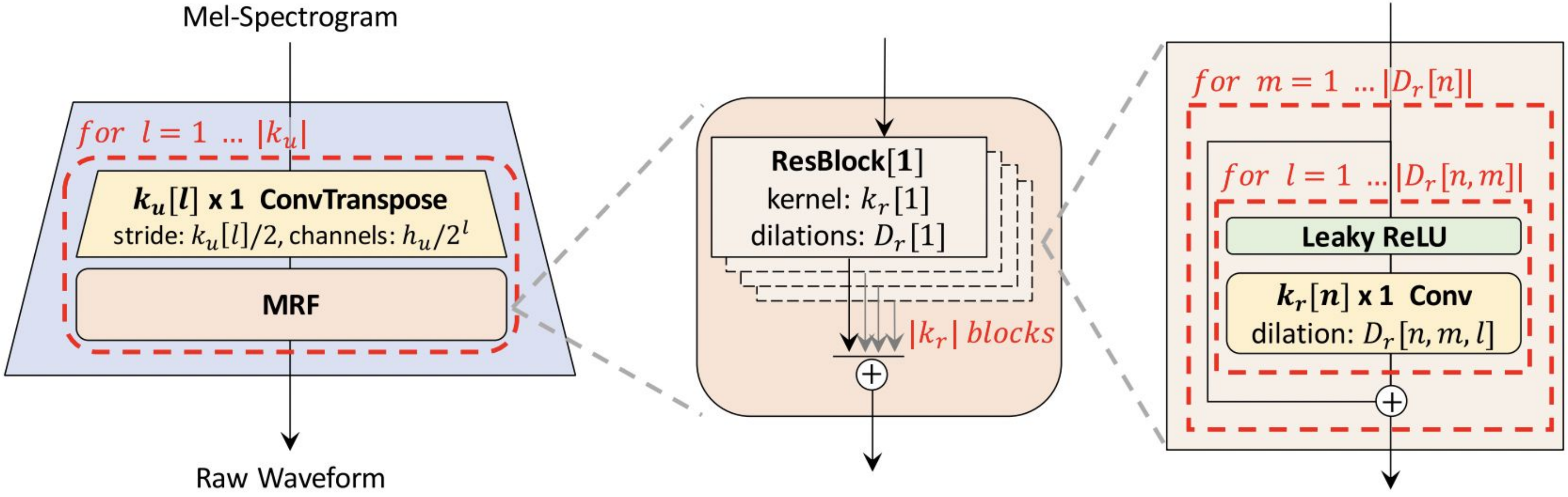
Markovian Discriminator

- Don’t classify entire audio sequences
- Classify random overlapped chunks



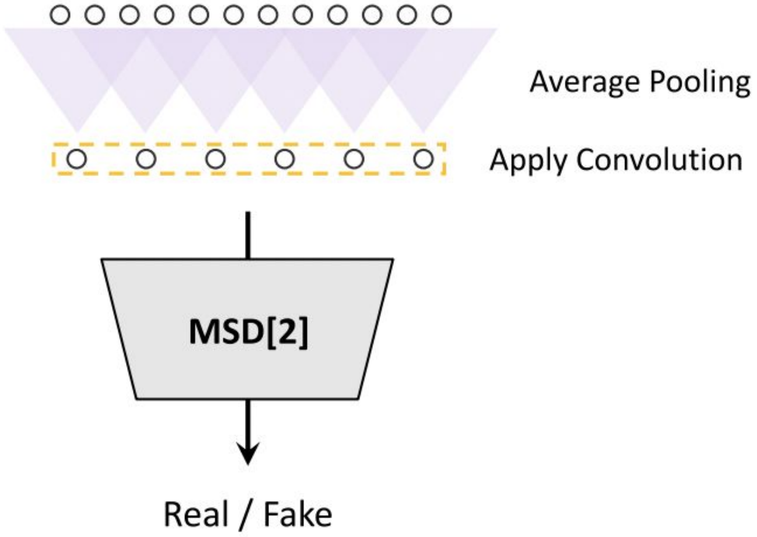
HiFi GAN

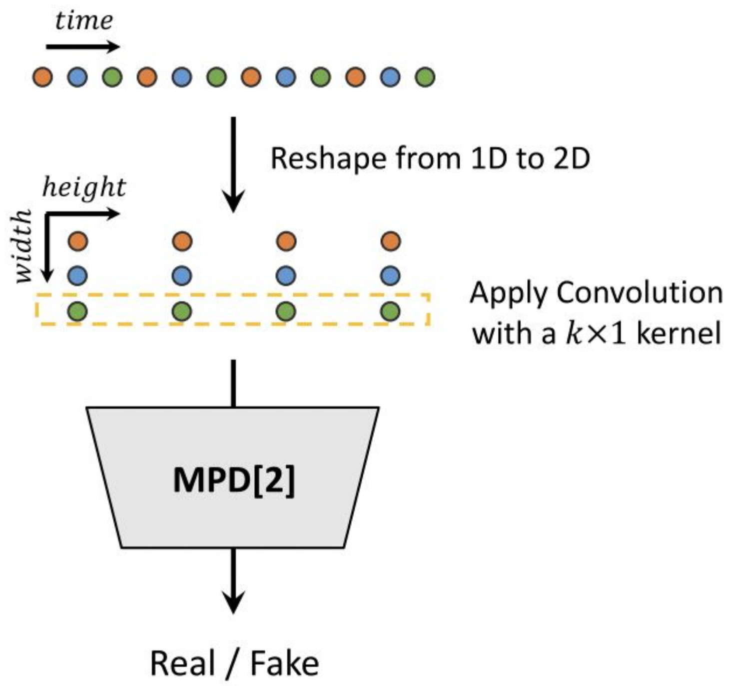
Generator



h_u	k_u	k_r	D_r
512	[16, 16, 4, 4]	[3, 7, 11]	[[1, 1], [3, 1], [5, 1]] ×3

HiFi GAN
Discriminator





HiFi GAN
Criterions

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

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

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