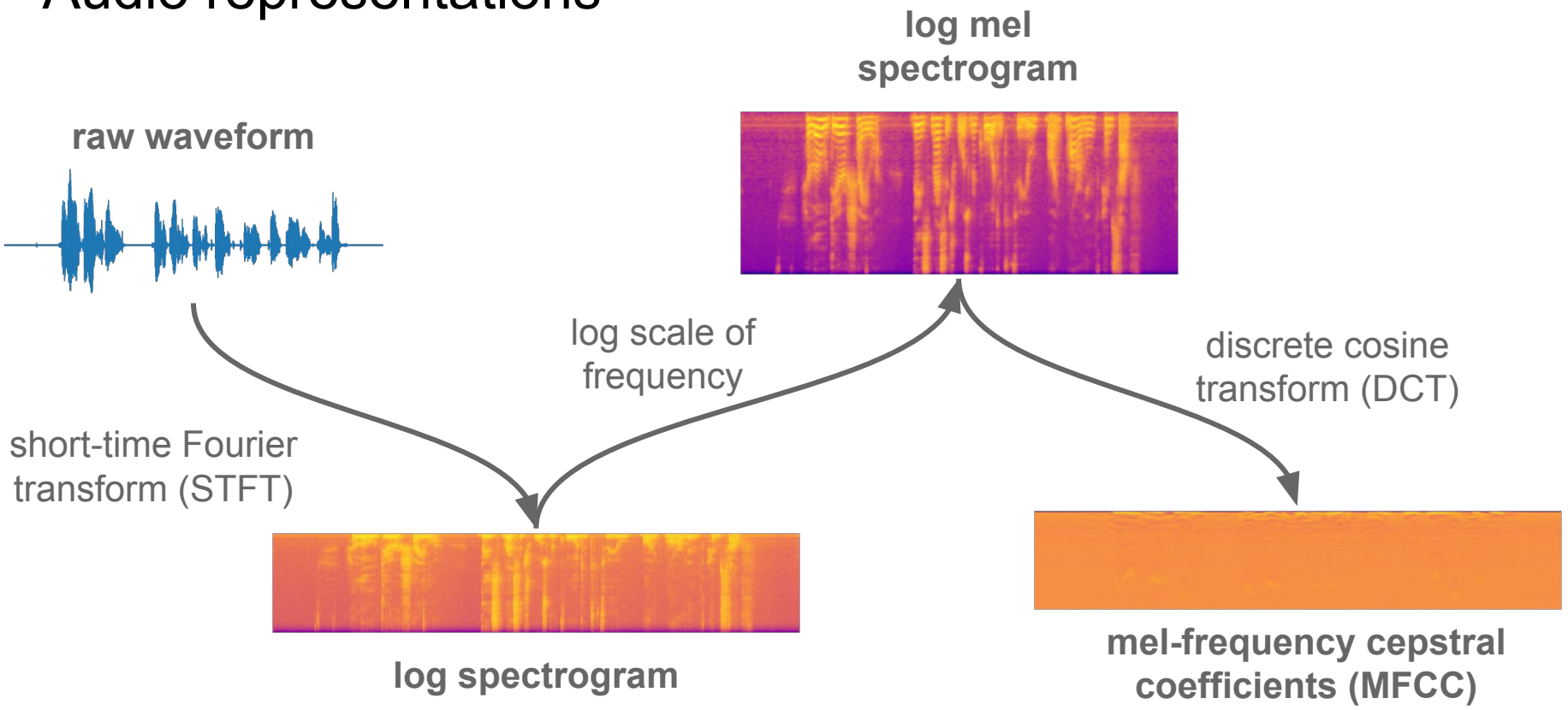
LSDL Lecture 12
Large scale audio processing

Ildus Sadrtdinov, 02.12.24

What makes audio different?

- Audio signal is continuous (same as images)
- Audio signal is sequential (same as texts)
- Audio signal has high-frequency (even more than images)
- There are many informative feature representations available for audio

Audio representations



Tasks in audio processing

- Automatic Speech Recognition (**ASR**)
- Text-to-Speech (**TTS**)
- Multiple source separation
- Audio enhancement (e.g., denoising, accent correction)
- Audio classification (speaker identification, music genre detection)
- Per-frame classification (Voice Activity Detection, speaker diarization)

- seq2seq tasks
- classification tasks

Plan

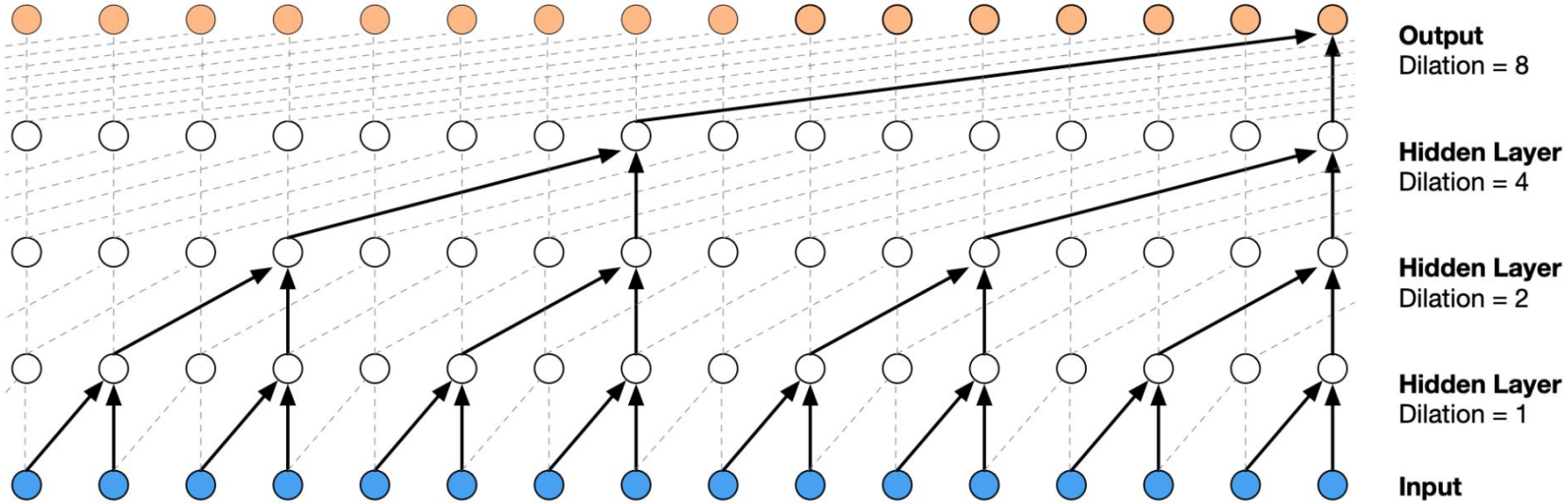
- **Audio/speech generation (incl. Text-to-Speech)**
  - **WaveNet, Tacotron 2, WaveGlow, HiFi-GAN**
- Automatic Speech Recognition (ASR)
  - Jasper, Whisper
- Self-supervised learning for audio
  - CPC, Wav2Vec 2.0, HuBERT,
    Multi-format CL, BYOL-A, CLAP

van der Oord et al., 2016, Google Deepmind

WaveNet

- Autoregressive model for raw waveforms
- Dilated causal convolutions

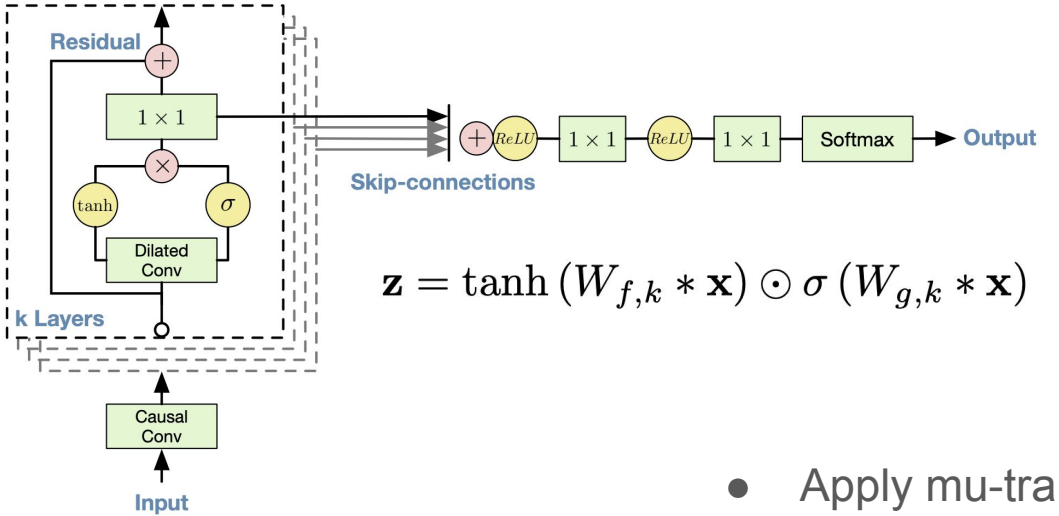
$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$



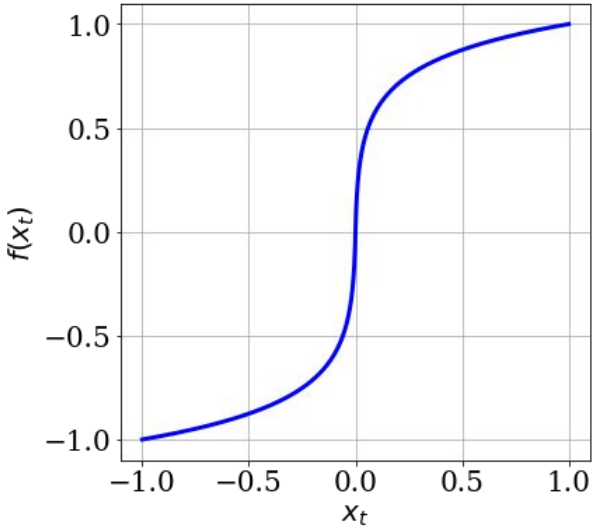
van der Oord et al., 2016, Google Deepmind

WaveNet

- Gated activation unit



$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x}\right) \odot \sigma\left(W_{g,k} * \mathbf{x}\right)$$



$$f\left(x_t\right) = \operatorname{sign}(x_t)\frac{\ln\left(1+\mu\left|x_t\right|\right)}{\ln\left(1+\mu\right)} \qquad \mu = 255$$

- Apply mu-transform to waveform, quantize and use categorical distribution for output prediction

van der Oord et al., 2016, Google Deepmind

Conditional WaveNet

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

- Global conditioning
  (e.g. speaker embedding)

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x} + V_{f,k}^T \mathbf{h}\right) \odot \sigma\left(W_{g,k} * \mathbf{x} + V_{g,k}^T \mathbf{h}\right)$$

- Local conditioning
  (e.g. text for TTS)

Condition sequence h_t → Upsampled sequence y_t

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

Le Paine et al., 2016

Generation in WaveNet

- Fast to train (convs are parallel) but very slow in generation
- Solution:

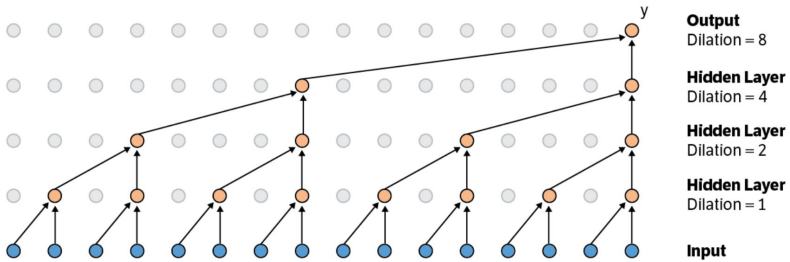


Figure 1: Naïve implementation of generation process. Notice that generating a single sample requires $O(2^L)$ operations.

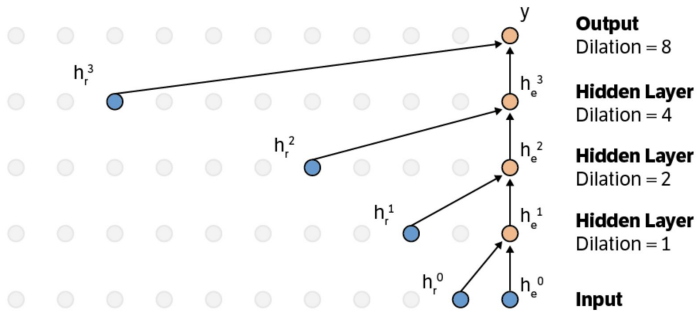
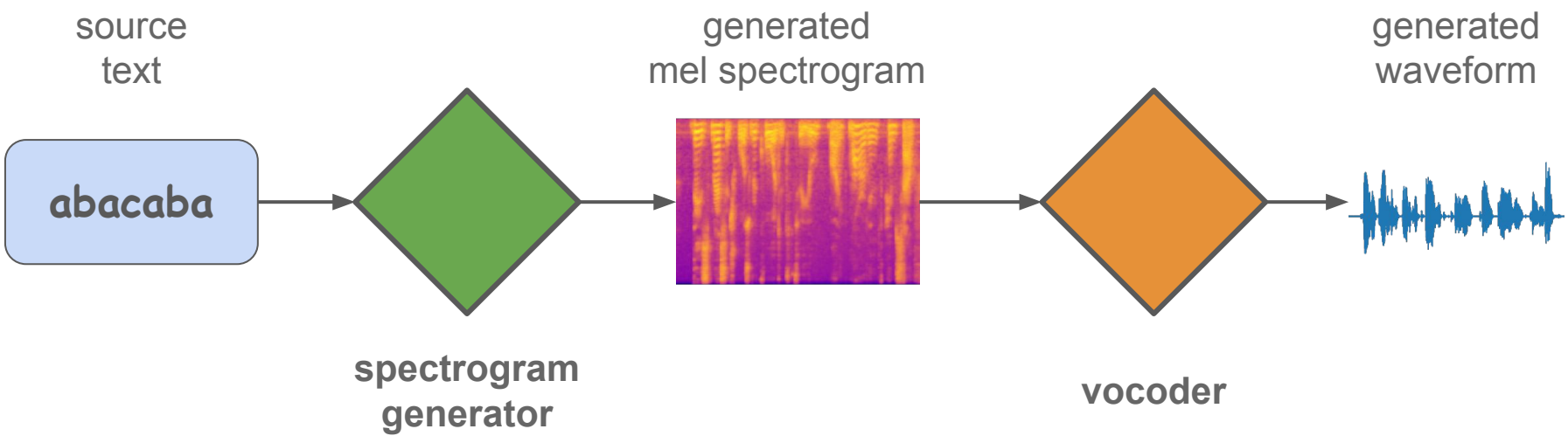


Figure 2: Simplified computation graph produced by our Fast Wavenet method. Now for a single output, the computational complexity is $O(L)$ where L is number of layers in the network.

Two-staged TTS



Shen et al., 2017, Google

Tacotron 2

- Encoder-decoder architecture
- Location Sensitive Attention between decoder and encoder
- Decoder predicts next frame and probability of stopping
- Pre-Net to process frames before inputting to LSTM
- Post-Net to smooth the resulting spectrogram

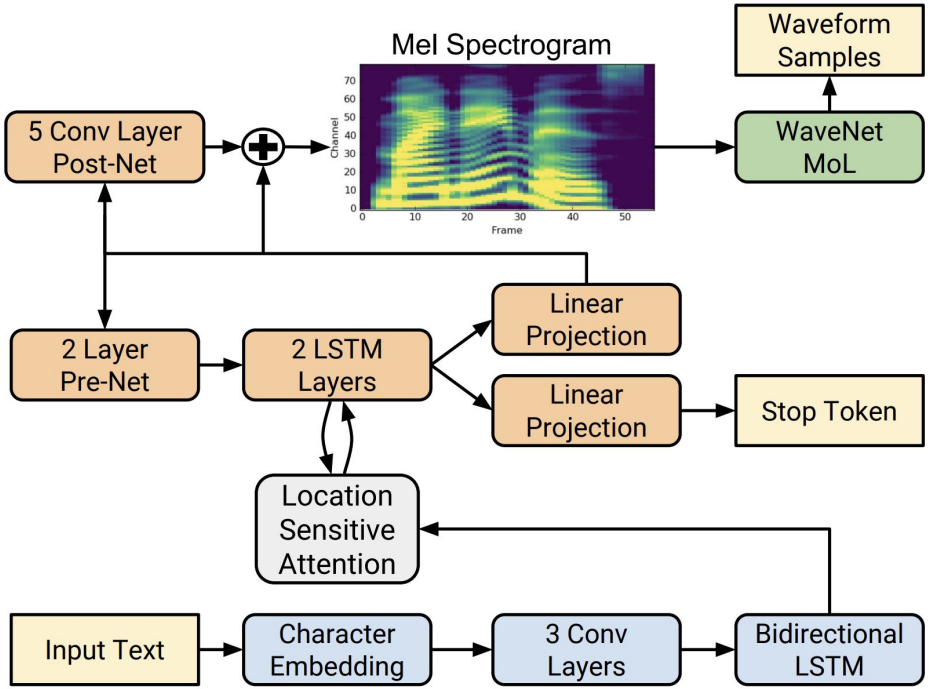


Fig. 1. Block diagram of the Tacotron 2 system architecture.

Prenger et al., 2018, NVIDIA

WaveGlow

- Generative flow vocoder

$$z \sim \mathcal{N}(z; 0, I)$$

$$x = f_0 \circ f_1 \circ \dots f_k(z)$$

$$\log p_\theta(x) = \log p_\theta(z) + \sum_{i=1}^{k} \log \left| \det(J(f_i^{-1}(x))) \right|$$

$$z = f_k^{-1} \circ f_{k-1}^{-1} \circ \dots f_0^{-1}(x)$$

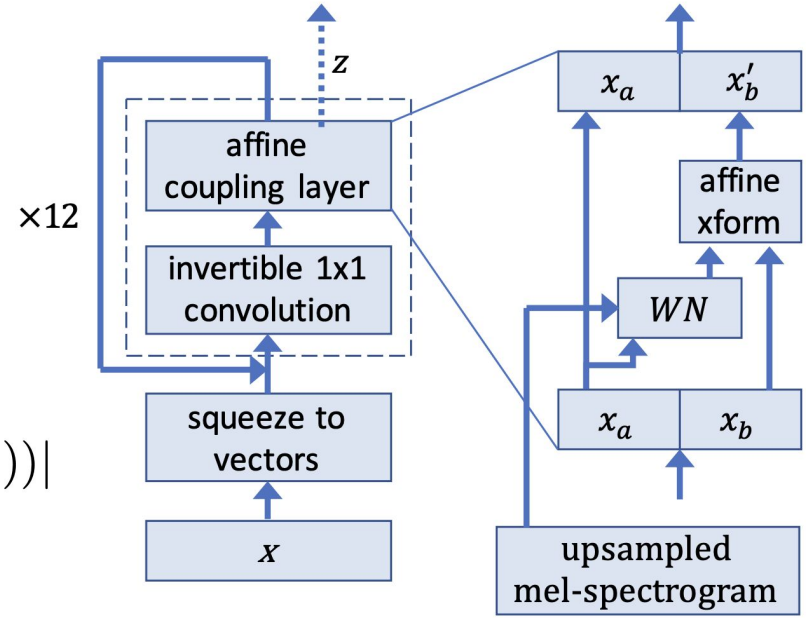


Fig. 1: WaveGlow network

Prenger et al., 2018, NVIDIA

WaveGlow

- Affine coupling layer

$$\boldsymbol{x}_a, \boldsymbol{x}_b = split(\boldsymbol{x})$$
$$(\log \boldsymbol{s}, \boldsymbol{t}) = WN(\boldsymbol{x}_a, mel\text{-}spectrogram)$$
$$\boldsymbol{x}_b\prime = \boldsymbol{s} \odot \boldsymbol{x}_b + \boldsymbol{t}$$
$$\boldsymbol{f}^{-1}_{coupling}(\boldsymbol{x}) = concat(\boldsymbol{x}_a, \boldsymbol{x}_b\prime)$$

$$\log|\det(\boldsymbol{J}(\boldsymbol{f}^{-1}_{coupling}(\boldsymbol{x})))| = \log|\boldsymbol{s}|$$

- Invertible 1x1 convolution

$$\boldsymbol{f}^{-1}_{conv} = \boldsymbol{W}\boldsymbol{x}$$
$$\log|\det(\boldsymbol{J}(\boldsymbol{f}^{-1}_{conv}(\boldsymbol{x})))| = \log|\det \boldsymbol{W}|$$

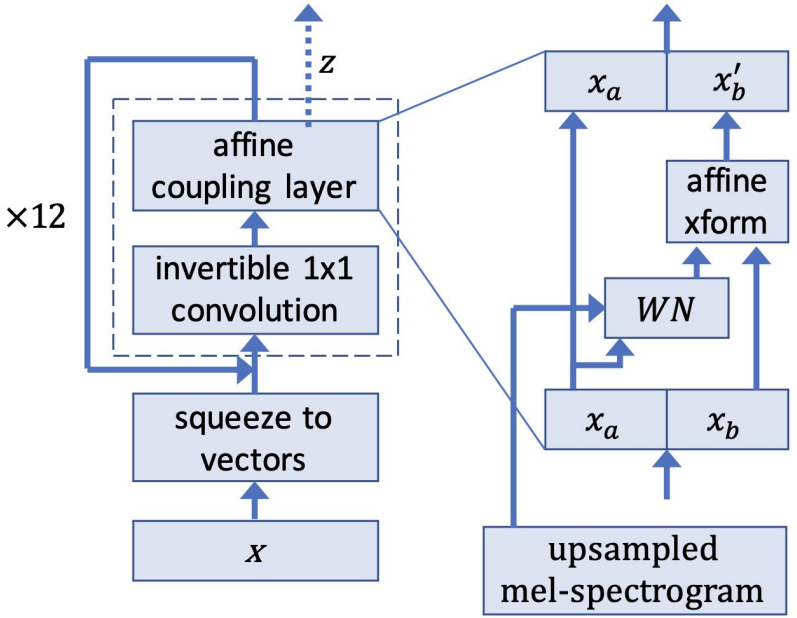


Fig. 1: WaveGlow network

Kong et al., 2020, Kakao Enterprise

HiFi-GAN

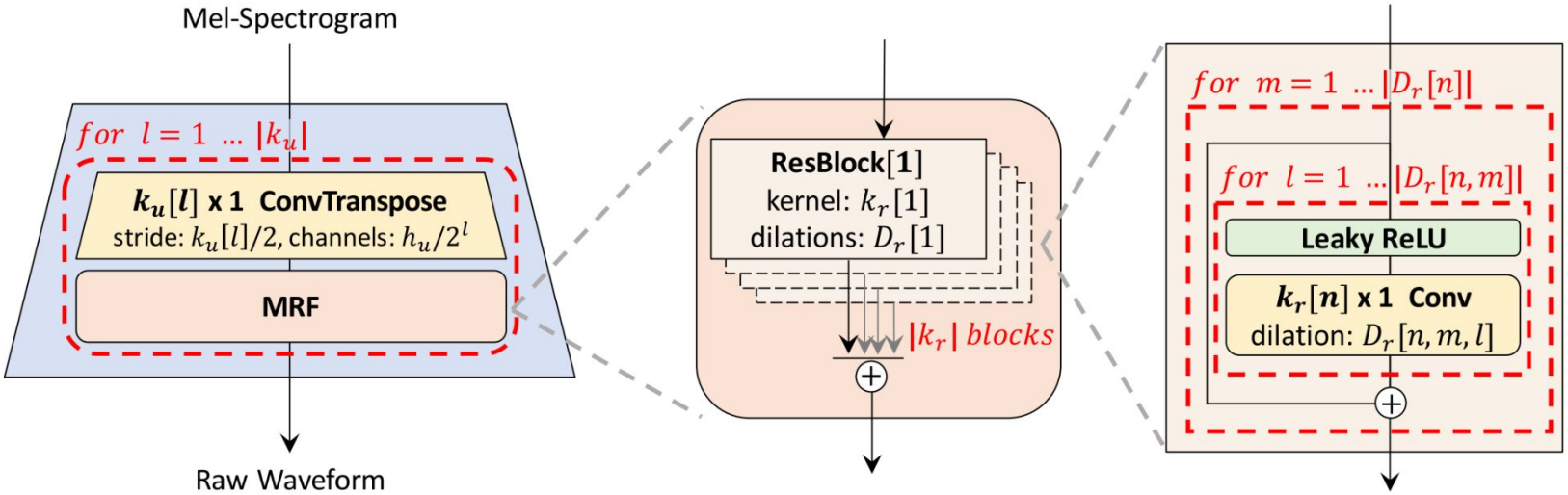


Figure 1: The generator upsamples mel-spectrograms up to $|k_u|$ times to match the temporal resolution of raw waveforms. A MRF module adds features from $|k_r|$ residual blocks of different kernel sizes and dilation rates. Lastly, the n-th residual block with kernel size $k_r[n]$ and dilation rates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

HiFi-GAN

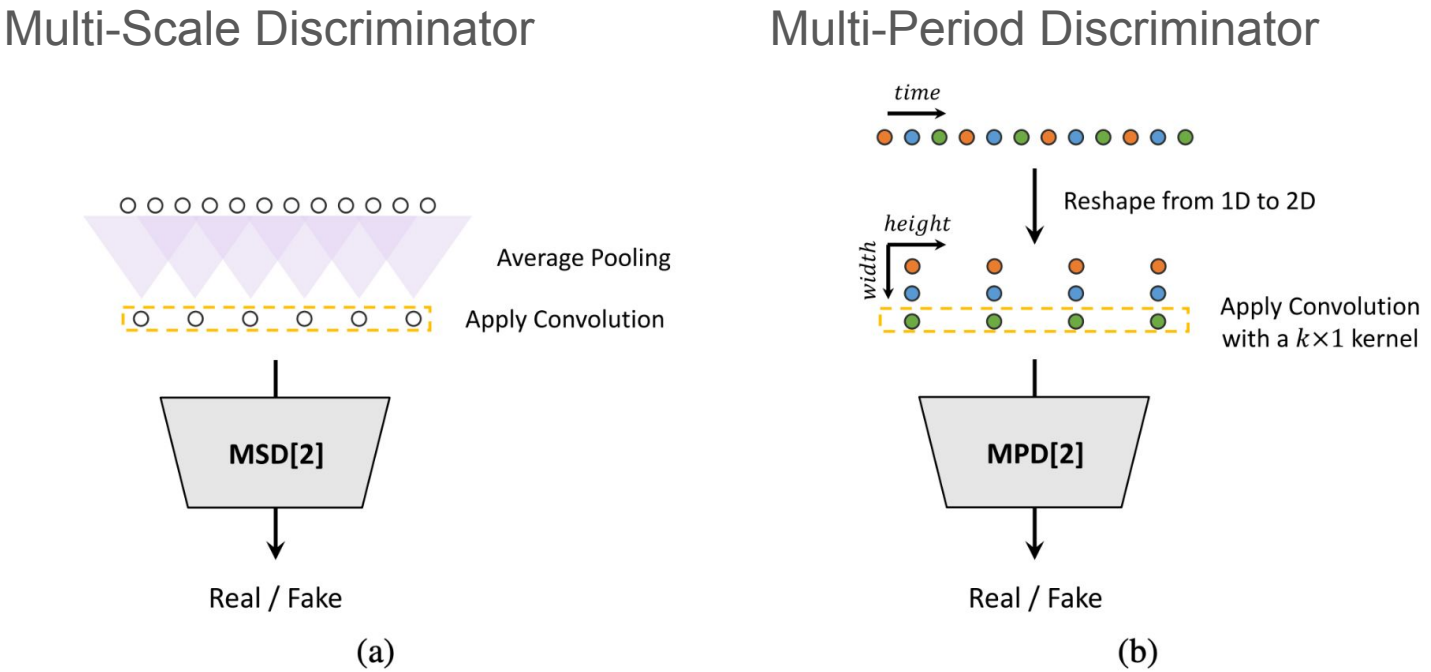


Figure 2: (a) The second sub-discriminator of MSD. (b) The second sub-discriminator of MPD with period 3.

HiFi-GAN

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

$$\mathcal{L}_D = \sum_{k=1}^{K} \mathcal{L}_{Adv}(D_k; G)$$

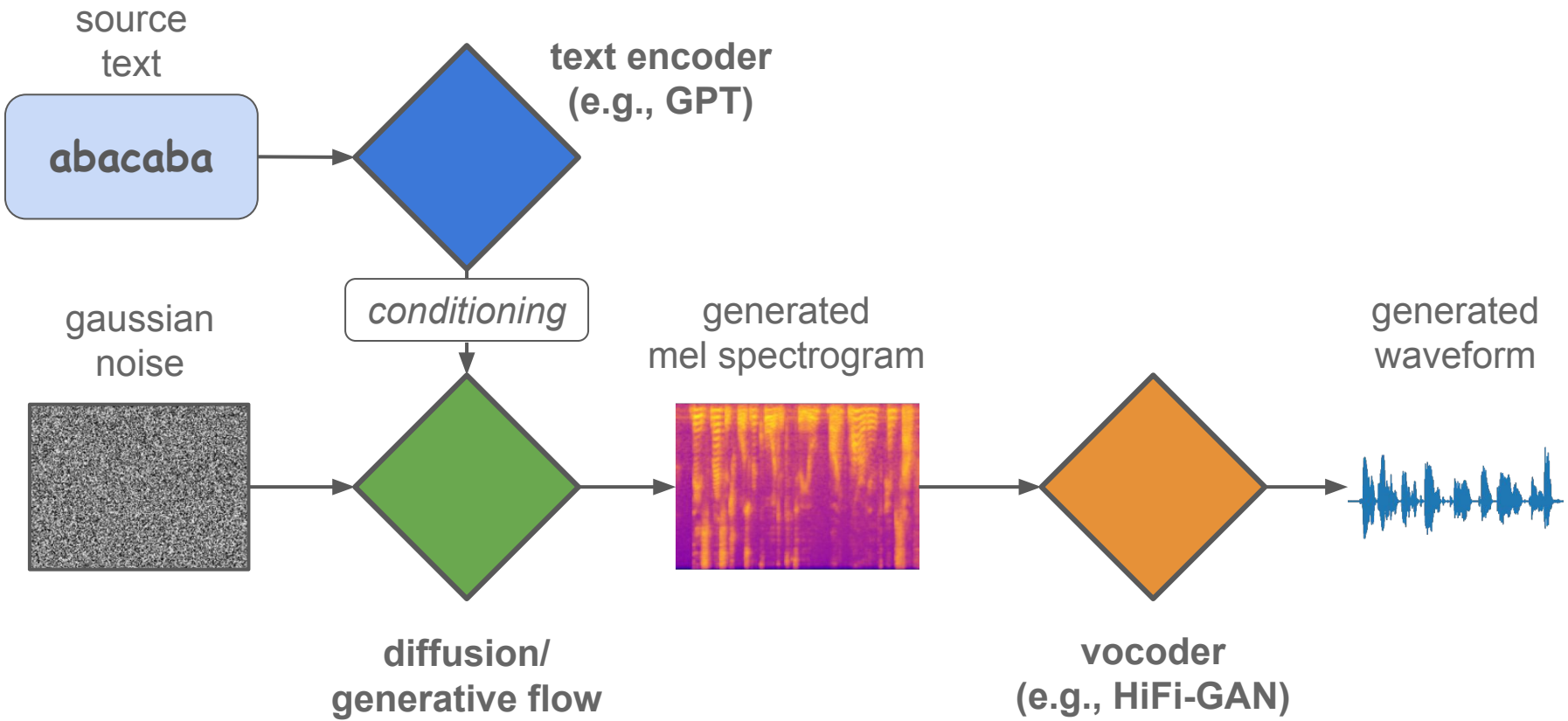
$$\mathcal{L}_{Adv}(D; G) = \mathbb{E}_{(x,s)} \left[(D(x) - 1)^2 + (D(G(s)))^2 \right]$$

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

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

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

Overview of modern approaches

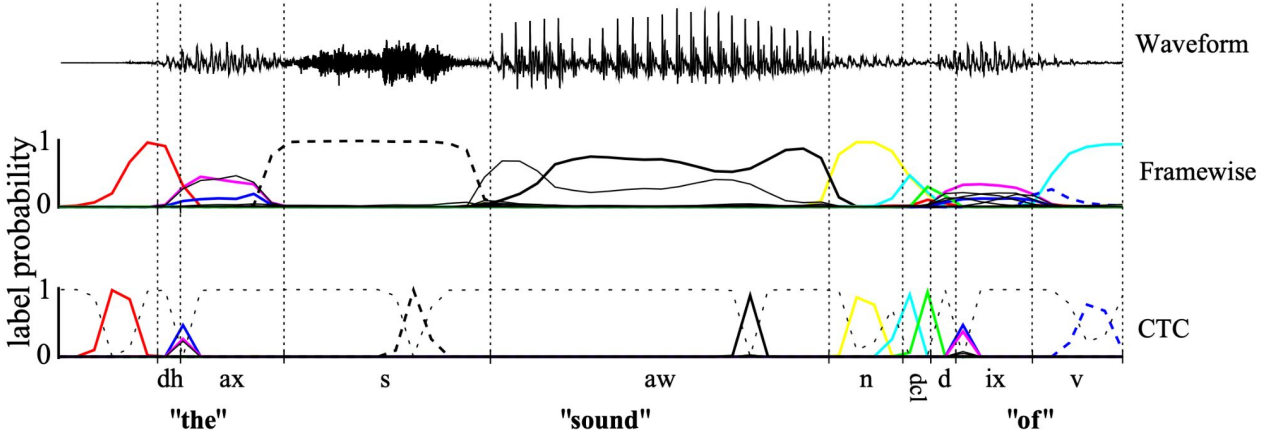


Plan

- Audio/speech generation (incl. Text-to-Speech)
  - WaveNet, Tacotron 2, WaveGlow, HiFi-GAN
- **Automatic Speech Recognition (ASR)**
  - **Jasper, Whisper**
- Self-supervised learning for audio
  - CPC, Wav2Vec 2.0, HuBERT,
    Multi-format CL, BYOL-A, CLAP

Graves et al., 2006

Connectionist Temporal Classification (CTC) loss



$$p(\pi|\mathbf{x}) = \prod_{t=1}^{T} y^t_{\pi_t}$$

$$\mathcal{B}(a-ab-) = \mathcal{B}(-aa--abb) = aab$$

$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} p(\pi|\mathbf{x})$$

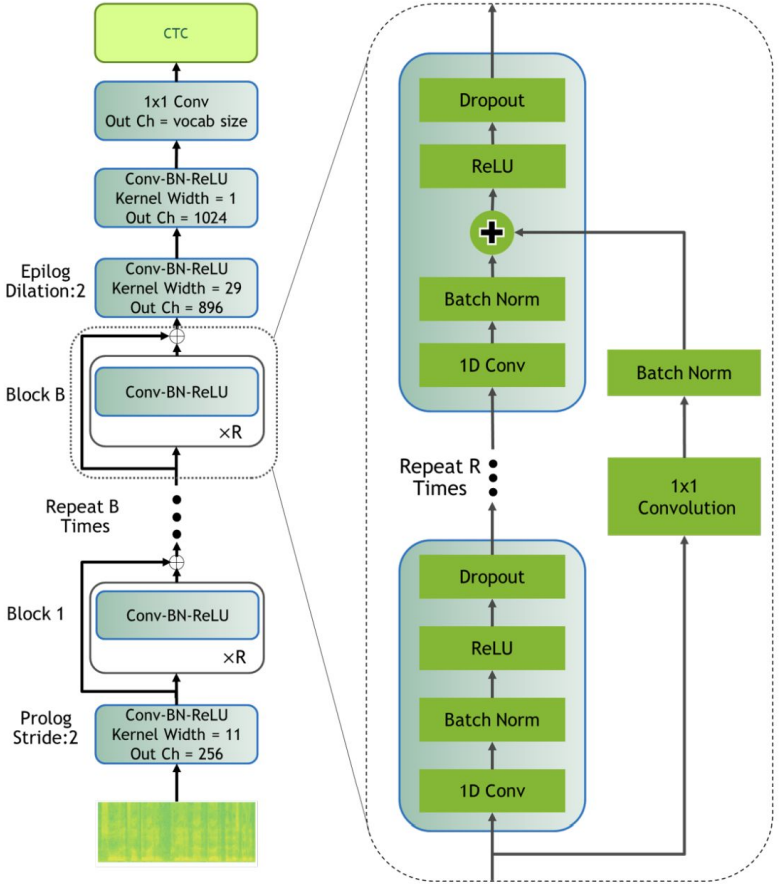
$$h(\mathbf{x}) = \arg\max_{\mathbf{l} \in L^{\leq T}} p(\mathbf{l}|\mathbf{x})$$

Calculated with dynamic programming

Li et al., 2019, NVIDIA

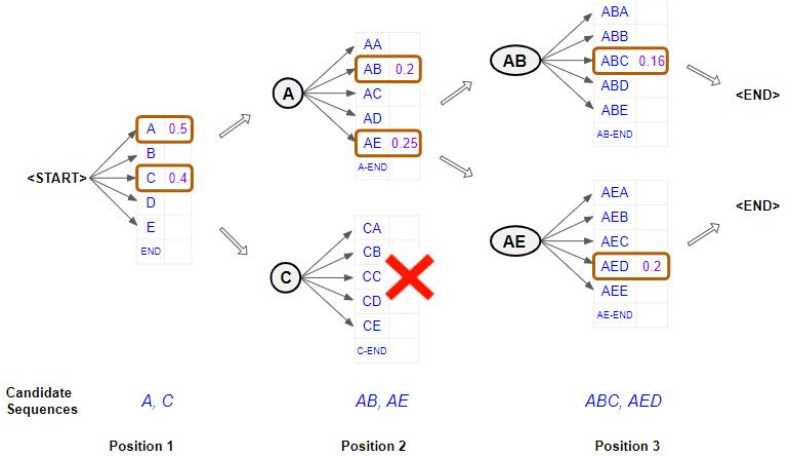
Jasper

- Purely 1D convolutional model
- Takes mel spectrograms as inputs
- Trained with CTC loss
- Uses a trained Language Model in beam search



ASR and Language Models

$$p\left(w_t \middle| w_{\leq t-1}\right) \propto p_{\mathrm{ASR}}\left(w_t \middle| w_{\leq t-1}, a\right) \cdot p_{\mathrm{LM}}\left(w_t \middle| w_{\leq t-1}\right)$$



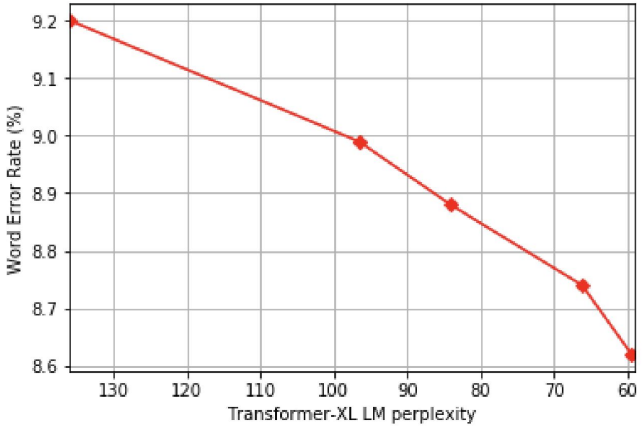


Figure 3: *LM perplexity vs WER. LibriSpeech dev-other. Varying perplexity is achieved by taking earlier or later snapshots during training.*

Model	E2E	LM	dev-clean	dev-other	test-clean	test-other
Jasper DR 10x5	Y	-	3.64	11.89	3.86	11.95
Jasper DR 10x5	Y	6-gram	2.89	9.53	3.34	9.62
Jasper DR 10x5	Y	Transformer-XL	2.68	8.62	**2.95**	8.79
Jasper DR 10x5 + Time/Freq Masks [4]	Y	Transformer-XL	2.62	7.61	2.84	7.84

LibriSpeech, WER (%)

Radford et al., 2022, Open AI

Whisper

Multitask training data (680k hours)

English transcription

"Ask not what your country can do for ···"

Ask not what your country can do for ···

Any-to-English speech translation

"El rápido zorro marrón salta sobre ···"

The quick brown fox jumps over ···

Non-English transcription

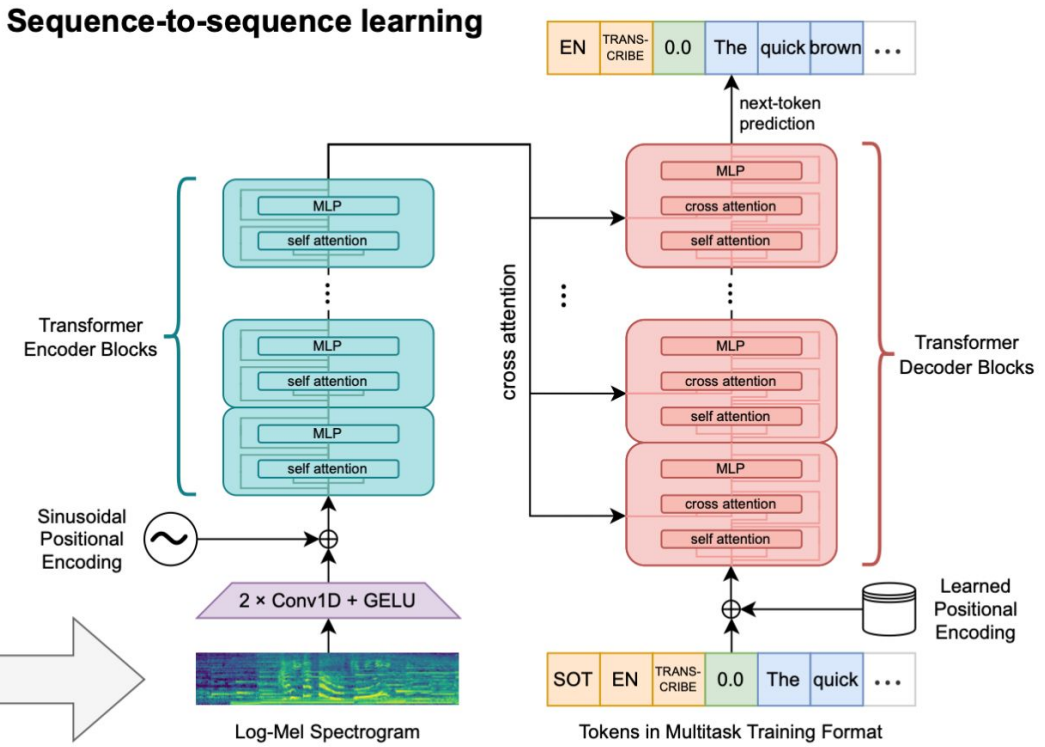
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···"

언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···

No speech

(background music playing)

Ø



Radford et al., 2022, Open AI

Whisper

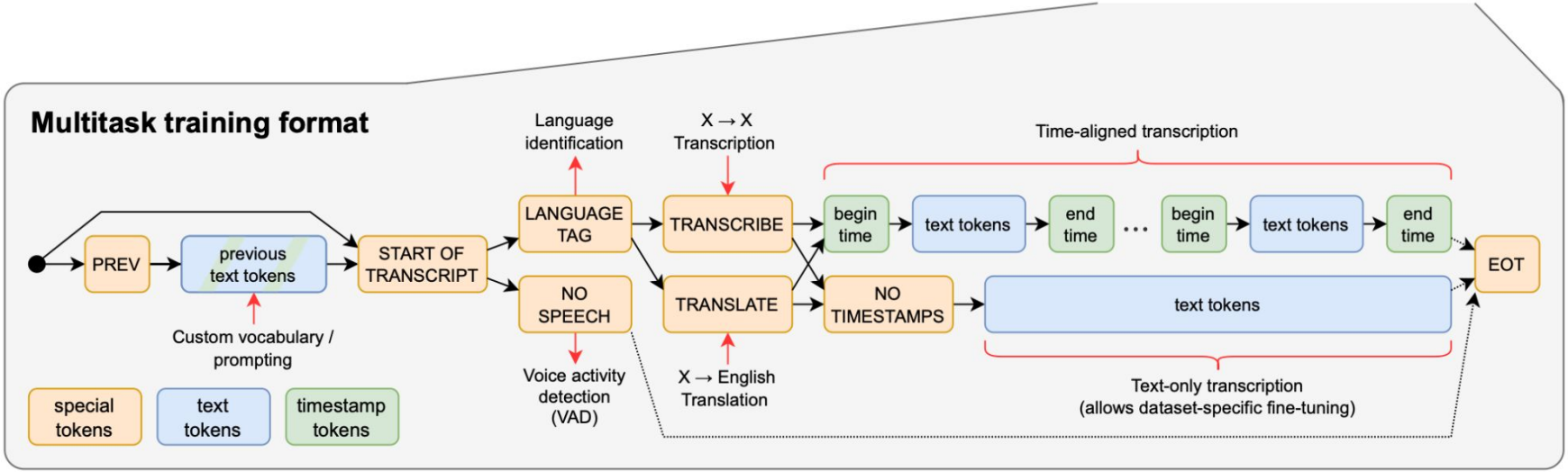


Figure 1. **Overview of our approach.** A sequence-to-sequence Transformer model is trained on many different speech processing tasks, including multilingual speech recognition, speech translation, spoken language identification, and voice activity detection. All of these tasks are jointly represented as a sequence of tokens to be predicted by the decoder, allowing for a single model to replace many different stages of a traditional speech processing pipeline. The multitask training format uses a set of special tokens that serve as task specifiers or classification targets, as further explained in Section 2.3.

Plan

- Audio/speech generation (incl. Text-to-Speech)
  - WaveNet, Tacotron 2, WaveGlow, HiFi-GAN
- Automatic Speech Recognition (ASR)
  - Jasper, Whisper
- **Self-supervised learning for audio**
  - **CPC, Wav2Vec 2.0, HuBERT, Multi-format CL, BYOL-A, CLAP**

Approaches to self-supervised learning

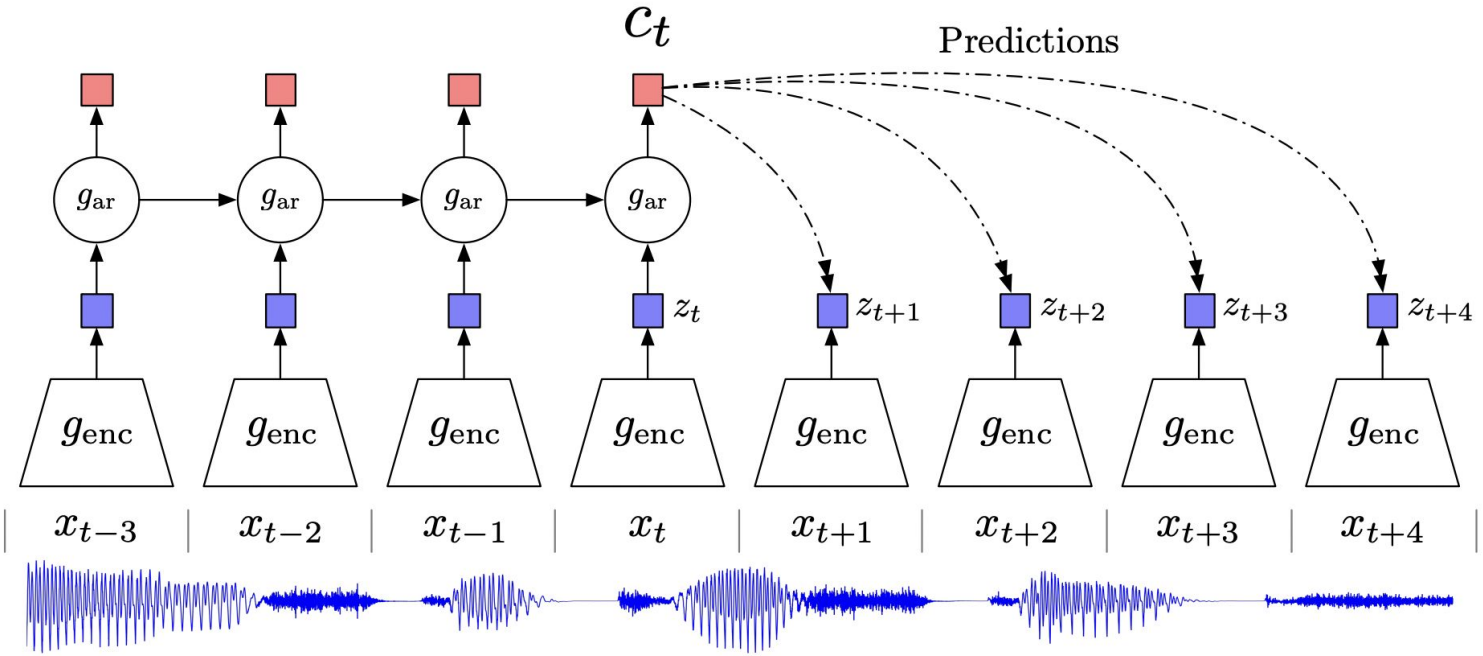
- Autoregressive (GPT-family)
- Contrastive (SimCLR, BYOL, CLIP)
- Masked modelling (*BERT, MAE)

SSL methods for audio

Model	Speech	Input format	Framework	Encoder	Loss	Inspired by
LIM [36]	✓	raw waveform	(d)	SincNet	BCE, MINE or NCE loss	SimCLR
COLA [36]	✗	log mel-filterbanks	(d)	EfficientNet	InfoNCE loss	SimCLR
CLAR [33] (semi)	✗	raw waveform log mel-spectrogram	(d)	1D ResNet-18 ResNet-18	NT-Xent + cross-entropy	SimCLR
Fonseca et al. [36]	✗	log mel-spectrogram	(d)	ResNet, VGG, CRNN	NT-Xent loss	SimCLR
Wang et al. [88]	✗	raw waveform + log mel-filterbanks	(d)	CNN ResNet	NT-Xent loss + cross-entropy	SimCLR
BYOL-A [89]	✗	log mel-filterbanks	(b)	CNN	MSE loss	BYOL
Speech2Vec [48]	✓	mel-spectrogram	(a)	RNN	MSE loss	Word2Vec
Audio2Vec [91]	✓✗	MFCCs	(a)	CNN	MSE loss	Word2Vec
Carr [67]	✓	MFCCs	(a)	Context-free network	Fenchel-Young loss	-
Ryan [68]	✗	constant-Q transform spectrogram	(a)	AlexNet	Triplet loss	- -
Mockingjay [92]	✓	mel-spectrogram	(a)	Transformer	L1 loss	BERT
TERA [93]	✓	log mel-spectrogram	(a)	Transformer	L1 loss	BERT
Audio ALBERT [94]	✓	log mel-spectrogram	(a)	Transformer	L1 loss	BERT
DAPC [95]	✓	spectrogram	(a)	Transformer	Modified MSE loss + orthogonality penalty	BERT
PASE [96]	✓	raw waveform	(a)	SincNet + CNN	L1, BCE loss	BERT
PASE+ [97]	✓	raw waveform	(a)	SincNet + CNN + QRNN	MSE, BCE loss	BERT
CPC [40]	✓	raw waveform	(a)	ResNet + GRU	InfoNCE loss	-
CPC v2 [59]	✓	raw waveform	(a)	ResNet + Masked CNN	InfoNCE loss	-
CPC2 [98]	✓	raw waveform	(a)	ResNet + LSTM	InfoNCE loss	-
Wav2Vec [84]	✓	raw waveform	(a)	1D CNN	Contrastive loss	-
VQ-Wav2Vec [85]	✓	raw waveform	(a)	1D CNN + BERT	Contrastive loss	BERT
Wav2Vec 2.0 [81]	✓	raw waveform	(a)	1D CNN + Transformer	Contrastive loss	BERT
HuBERT [99]	✓	raw waveform	(c)	1D CNN + Transformer	Contrastive loss	BERT

van der Oord et al., 2018, Google Deepmind

Contrastive Predictive Coding (CPC)



$$f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})} \qquad f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right) \qquad \mathcal{L}_N = -\mathop{\mathbb{E}}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

van der Oord et al., 2018, Google Deepmind

Contrastive Predictive Coding (CPC)

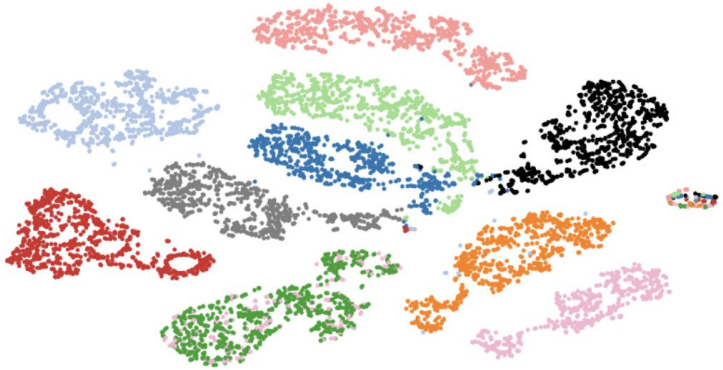


Figure 2: t-SNE visualization of audio (speech) representations for a subset of 10 speakers (out of 251). Every color represents a different speaker.

Method	ACC
Phone classification	
Random initialization	27.6
MFCC features	39.7
CPC	64.6
Supervised	74.6
Speaker classification	
Random initialization	1.87
MFCC features	17.6
CPC	97.4
Supervised	98.5

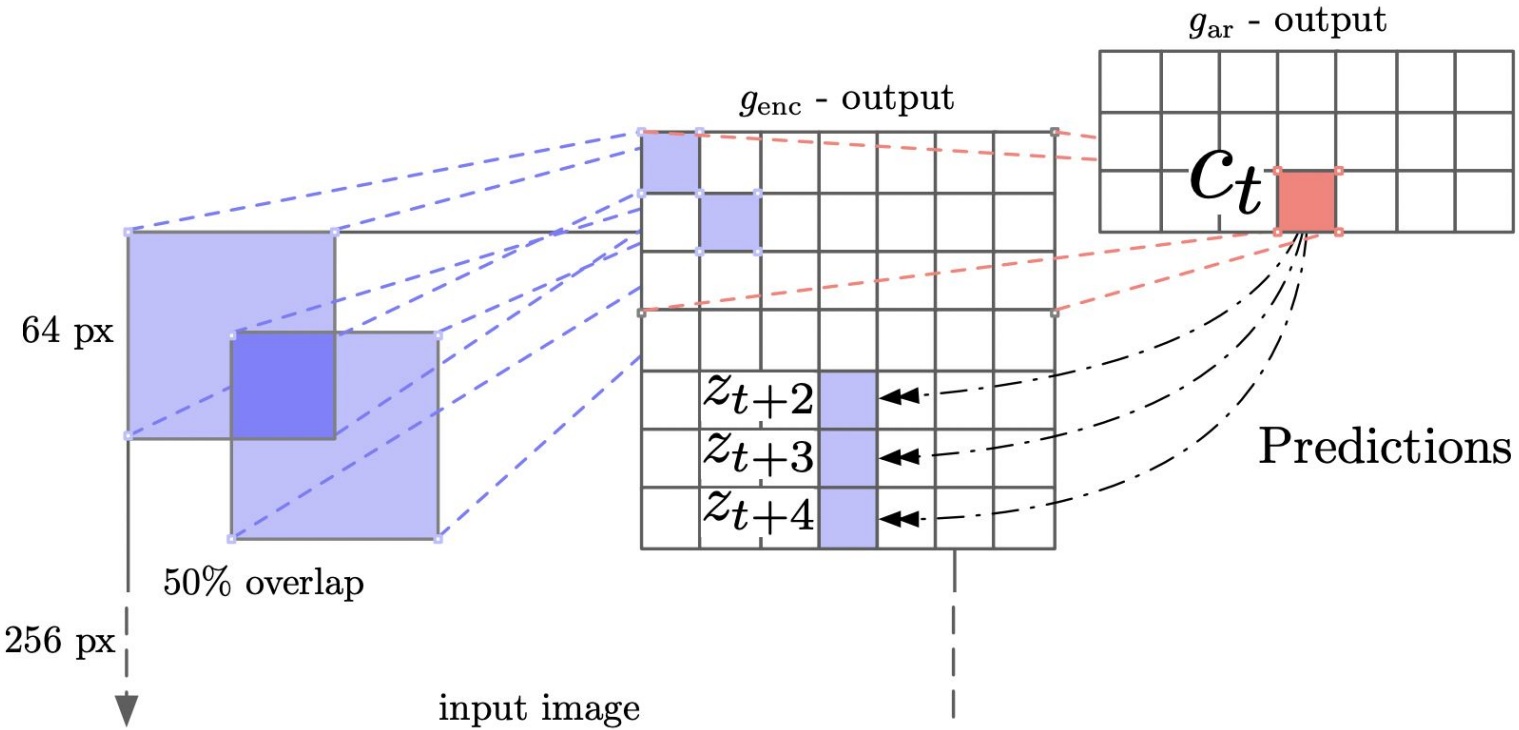
Table 1: LibriSpeech phone and speaker classification results. For phone classification there are 41 possible classes and for speaker classification 251. All models used the same architecture and the same audio input sizes.

Method	ACC
#steps predicted	
2 steps	28.5
4 steps	57.6
8 steps	63.6
12 steps	64.6
16 steps	63.8
Negative samples from	
Mixed speaker	64.6
Same speaker	65.5
Mixed speaker (excl.)	57.3
Same speaker (excl.)	64.6
Current sequence only	65.2

Table 2: LibriSpeech phone classification ablation experiments. More details can be found in Section 3.1.

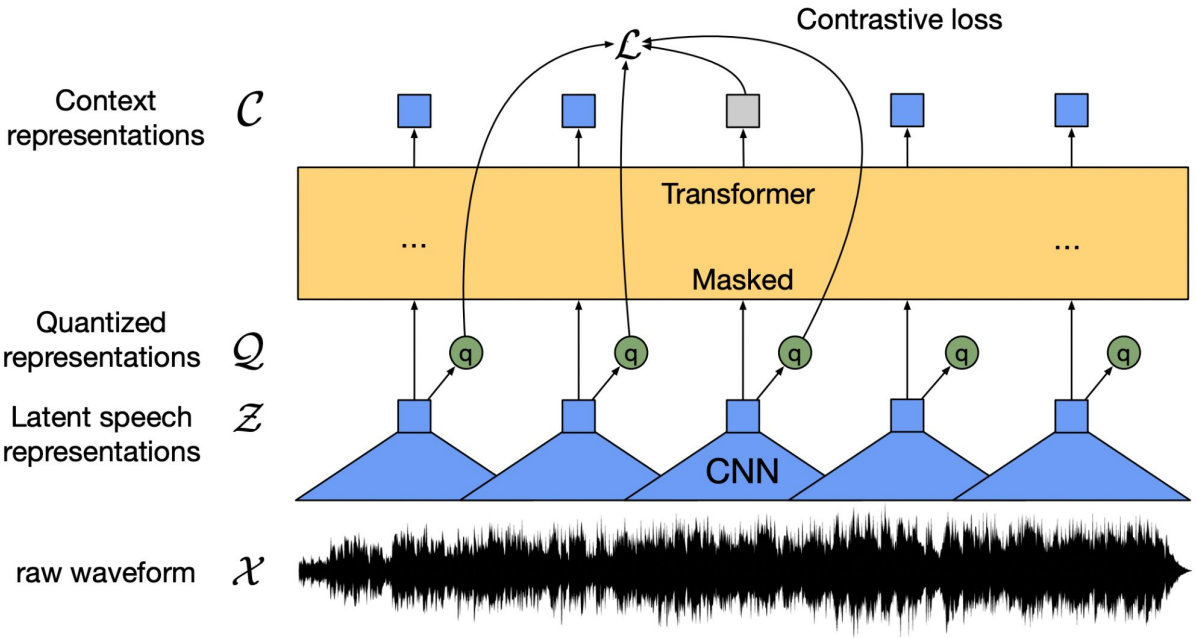
van der Oord et al., 2018, Google Deepmind

Contrastive Predictive Coding (CPC)



[Baevski et al., 2020](), Facebook AI

Wav2Vec 2.0

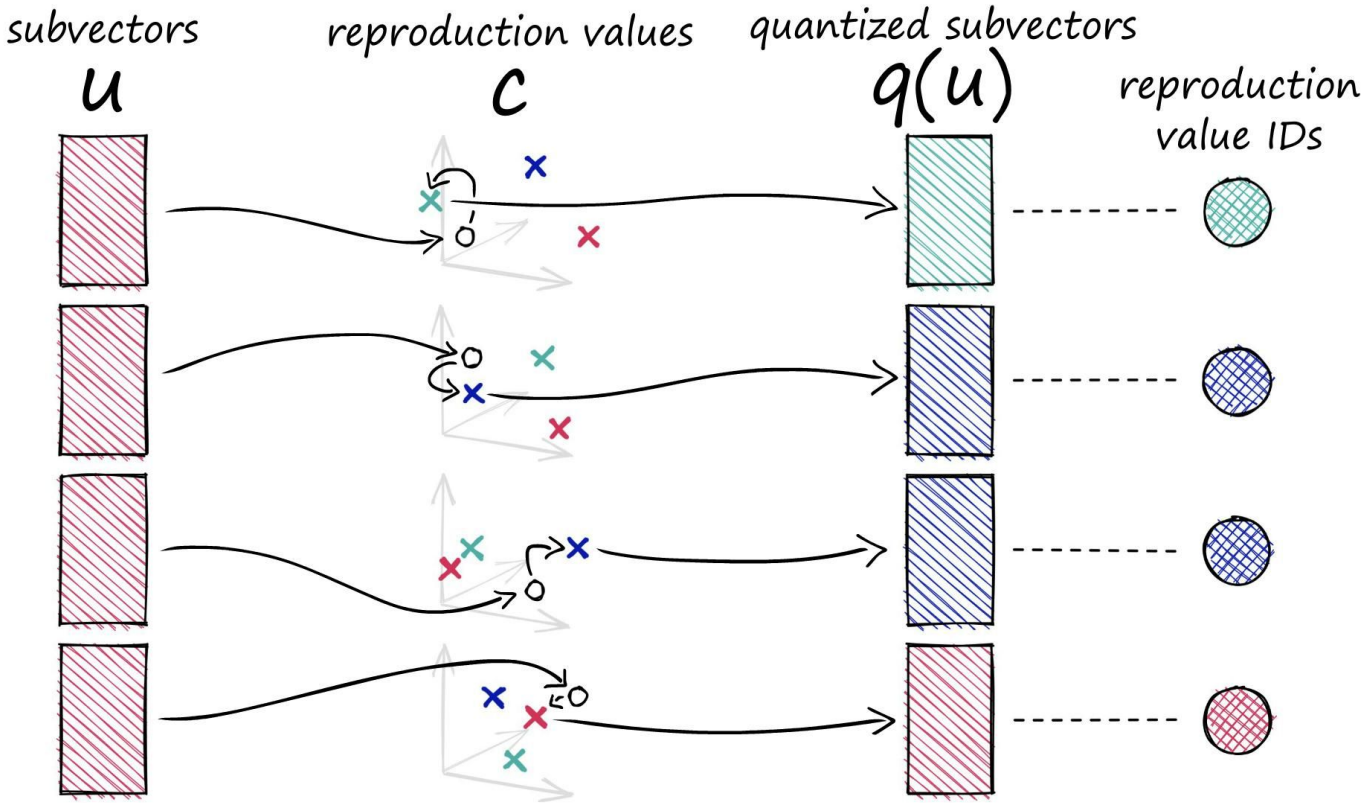


$$\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d$$

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)}$$

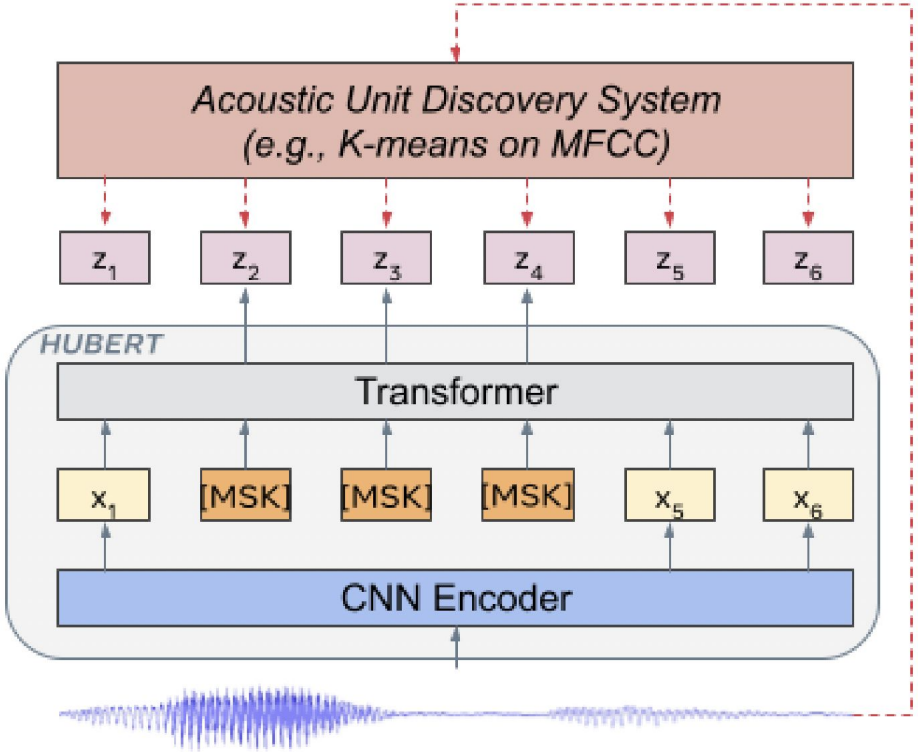
https://www.pinecone.io/

Product Quantization



Hsu et al., 2021

HuBERT

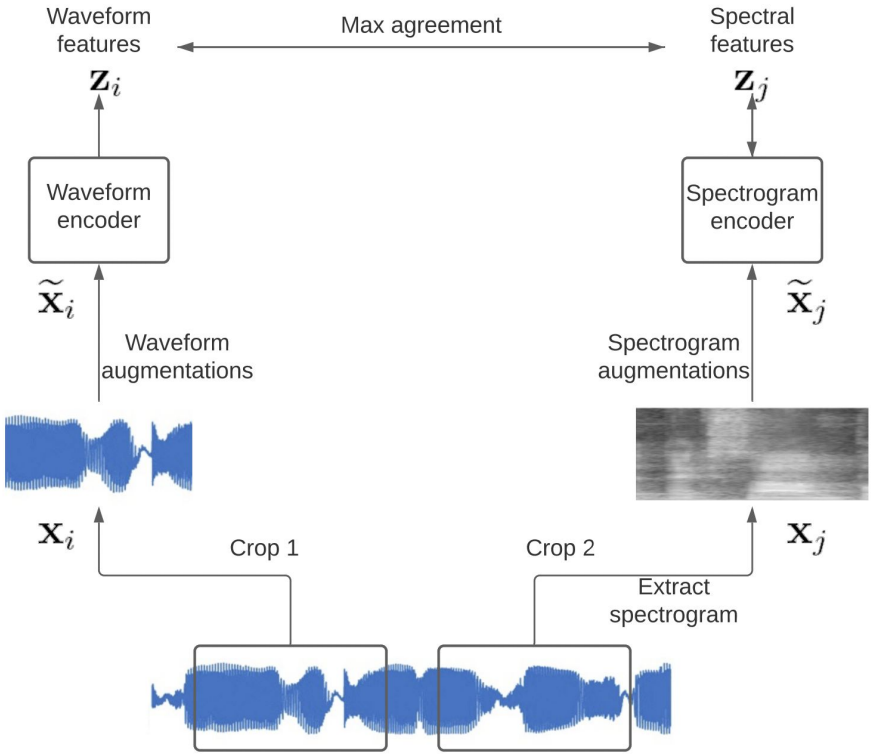


$$L = \alpha L_m + (1-\alpha) L_u$$

$$p_f^{(k)}(c \mid \tilde{X}, t) = \frac{\exp(\mathrm{sim}(A^{(k)} o_t, e_c)/\tau)}{\sum_{c'=1}^{C} \exp(\mathrm{sim}(A^{(k)} o_t, e_{c'})/\tau)}$$

Wang and van der Oord., 2021, Google Deepmind

Multi-format contrastive learning



$$L_{i,j} = -\log \frac{\exp\left(\text{sim}\left(\mathbf{z}_i, \mathbf{z}_j\right)/\tau\right)}{\sum_{k \neq i} \exp\left(\text{sim}\left(\mathbf{z}_i, \mathbf{z}_k\right)/\tau\right)}$$

Wang and van der Oord., 2021, Google Deepmind

Multi-format contrastive learning

Audio mixing Small additive noise of any sort will not alter the original categories of the audio. Given two audio clips $\mathbf{x_1}$ and $\mathbf{x_2}$, the mixed-up version is

$$\hat{\mathbf{x}}_\mathbf{1} = \alpha \mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2 \tag{2}$$

where $\hat{\mathbf{x}}_\mathbf{1}$ inheritances labels from $\mathbf{x_1}$. In this work, α is samples from $\beta(5, 2)$ distribution. This simulates various realistic noise conditions.

Time masking t consecutive time steps $[t_0, t_0 + t)$ of the audio can be dropped out and it should not change the event classes, where t_0 is randomly sampled. This can be applied both to raw audio and spectrograms.

Frequency masking A small amount of f frequency components $[f_0, f_0 + f)$ on the spectrogram can be masked out without losing semantic information.

Frequency shift One can apply the truncated shift in frequency to the spectrograms by an integer number sampled from $[-F, F]$, where F is the maximum shift size. Missing values after the shift are set to zero energy. Intuitively, this is a less expensive alternative of changing the pitch of the audio.

Wang and van der Oord., 2021, Google Deepmind

Multi-format contrastive learning

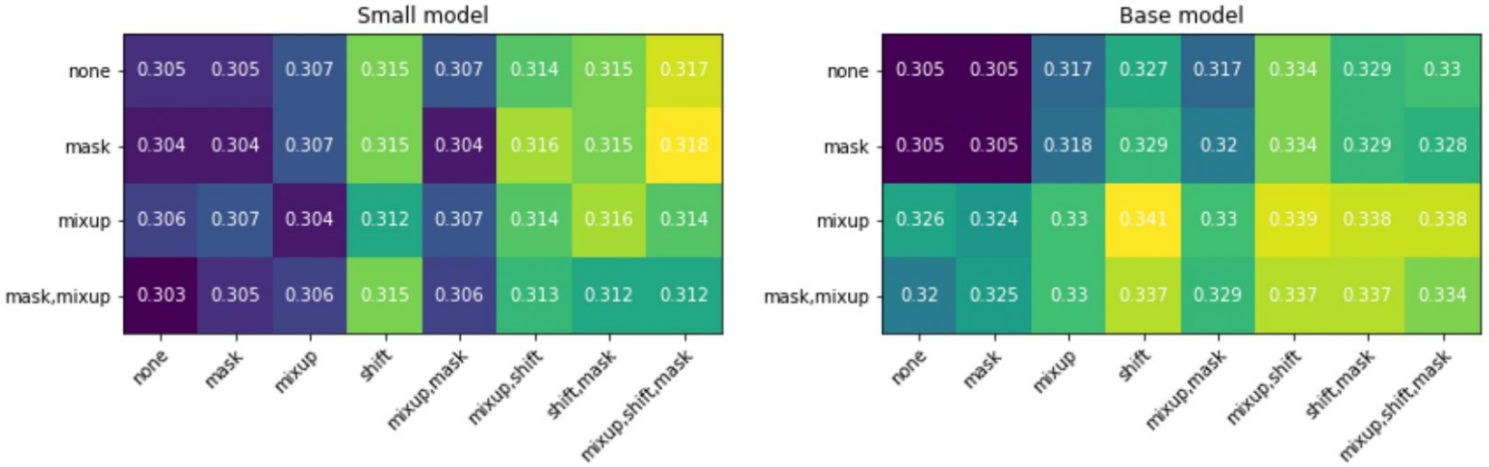


Figure 3: Validation mAP of the raw-audio-vs-log-mel models with different combinations of raw audio (along rows) and spectrogram (along columns) augmentations.

Wang and van der Oord., 2021, Google Deepmind

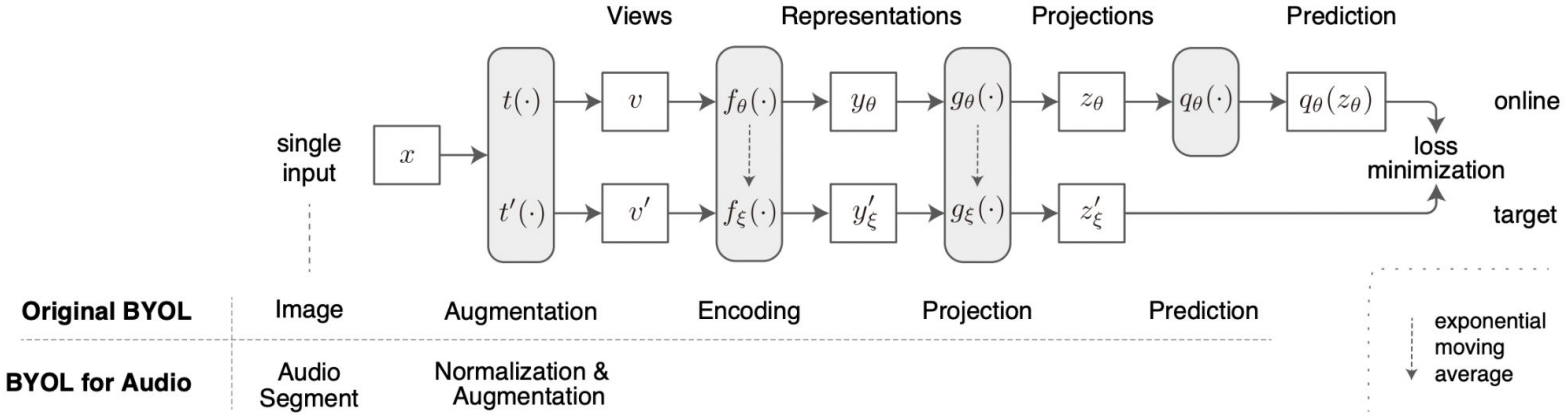
Multi-format contrastive learning

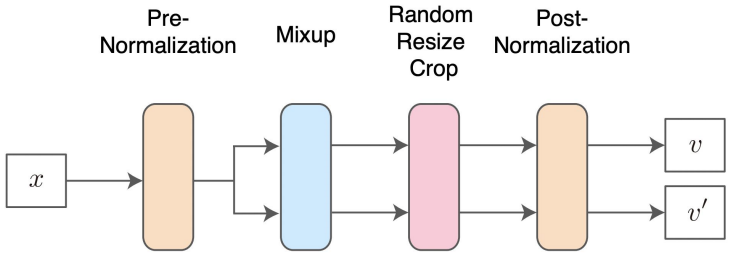
Table 3: Test performance of shallow model classification on AudioSet with fixed representations.

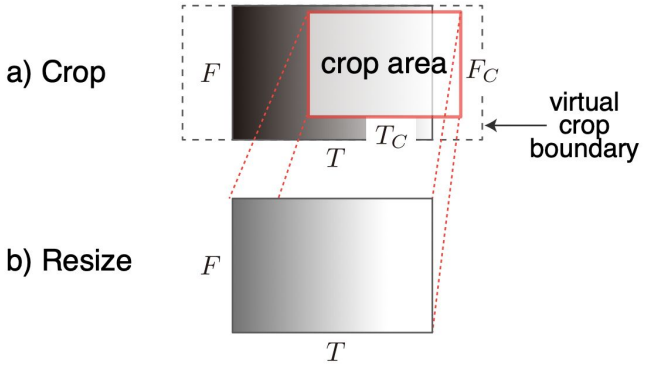
Model	Train inputs	Eval inputs	Test mAP
Triplet [20]	log-mel	log-mel	0.244
L^3 [22]	log-mel + video	log-mel	0.249
CPC [21]	waveform	waveform	0.277
C^3 [26]	log-mel + video	log-mel	0.285
MMV [28]	log-mel + video + text	log-mel	0.309
Ours	log-mel	log-mel	0.329
Ours	waveform	waveform	0.336
Ours	waveform + log-mel	log-mel	0.368
Ours	waveform + log-mel	waveform	0.355
Ours	waveform + log-mel	waveform + log-mel	**0.376**
Supervised [19]	waveform + log-mel	waveform + log-mel	0.439

Niizumi et al., 2021, NTT Corporation

BYOL-A







Niizumi et al., 2021, NTT Corporation

BYOL-A

TABLE II
ABLATIONS OF BYOL-A AUGMENTATION MODULE WITH ACCURACY RESULTS, PRETRAINED WITH 1/10 AUDIOSET

Augmentation blocks used	NS	US8K	VC1	VF	SPCV2/12	SPCV2	Average	Degradation
Mixup+RRC (BYOL-A)	**71.2%**	77.0%	31.0%	83.1%	**84.5%**	87.2%	**72.3%**	
Mixup+Gaussian+RRC	69.5%	74.3%	25.2%	**84.0%**	82.8%	**87.4%**	70.5%	BYOL-A -1.8
Gaussian+RRC	69.7%	73.1%	29.2%	83.1%	78.0%	83.1%	69.3%	BYOL-A -3.0
RRC	69.4%	**77.1%**	**34.5%**	80.3%	71.4%	77.4%	68.4%	BYOL-A -3.9
Mixup	55.6%	69.4%	22.3%	78.3%	75.8%	82.0%	63.9%	BYOL-A -8.4
Gaussian	29.5%	31.2%	0.9%	57.9%	9.4%	10.3%	23.2%	BYOL-A -49.1

TABLE III
ABLATIONS OF NORMALIZATION BLOCKS WITH AVERAGE ACCURACY RESULTS, PRETRAINED ON 1/10 AUDIOSET

Method	Average	Degradation
BYOL-A	**72.3%**	
w/o Post-Norm	72.1%	BYOL-A -0.2
w/o Pre-Norm (mixup $\alpha = 0.05$)	70.5%	BYOL-A -1.8
w/o Pre-Norm (mixup $\alpha = 0.1$)	70.3%	BYOL-A -2.0
w/o Pre-Norm (mixup $\alpha = 0.4$)	68.9%	BYOL-A -3.4

Elizalde et al., 2022, Microsoft

CLAP: Contrastive Language Audio Pre-training