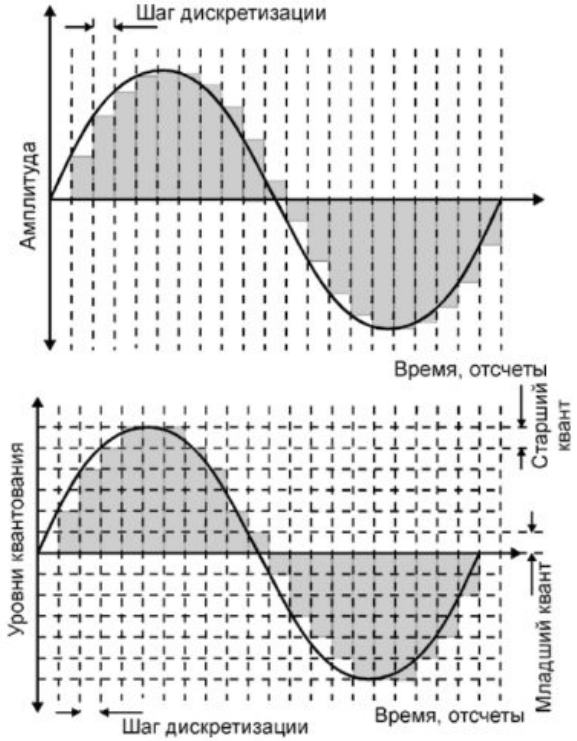
Encodec

High Fidelity Neural Audio Compression

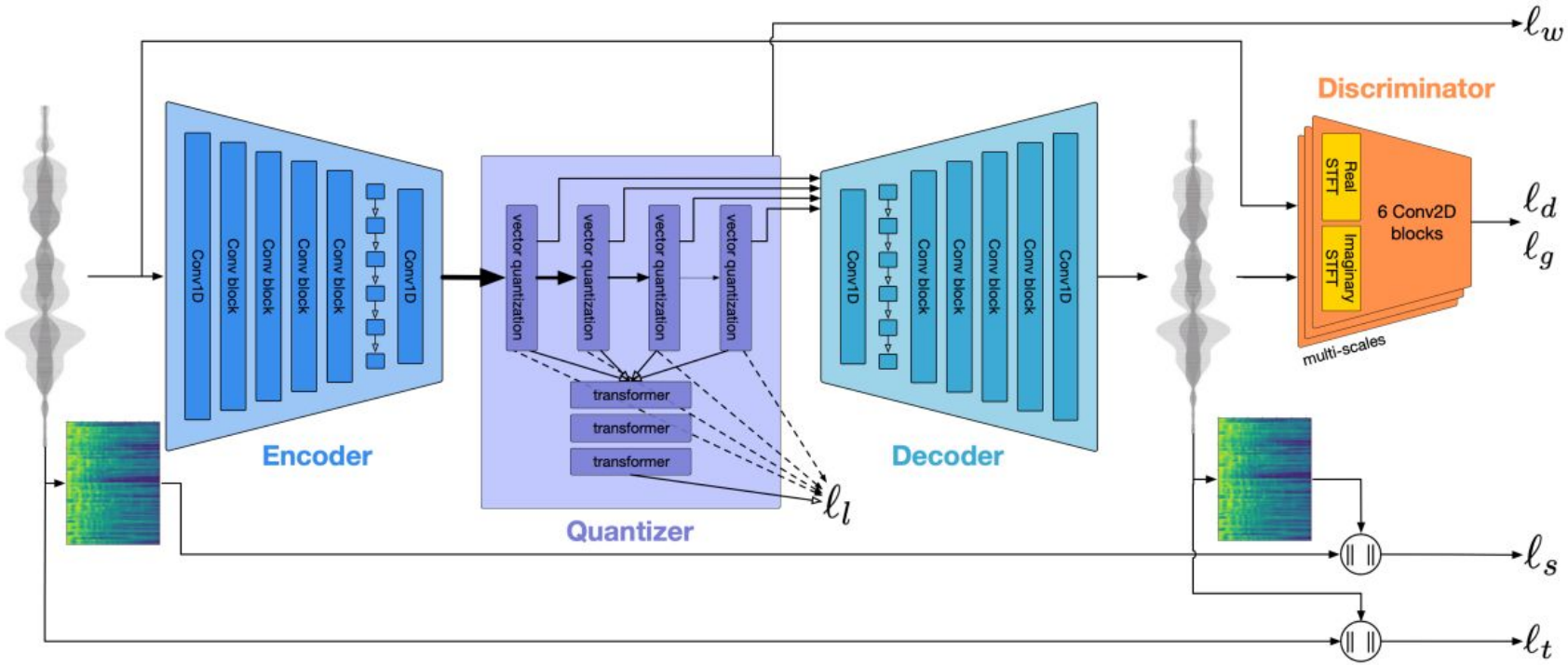
Филатов Егор, БПМИ213

Что такое аудио кодек и как им сжимать?



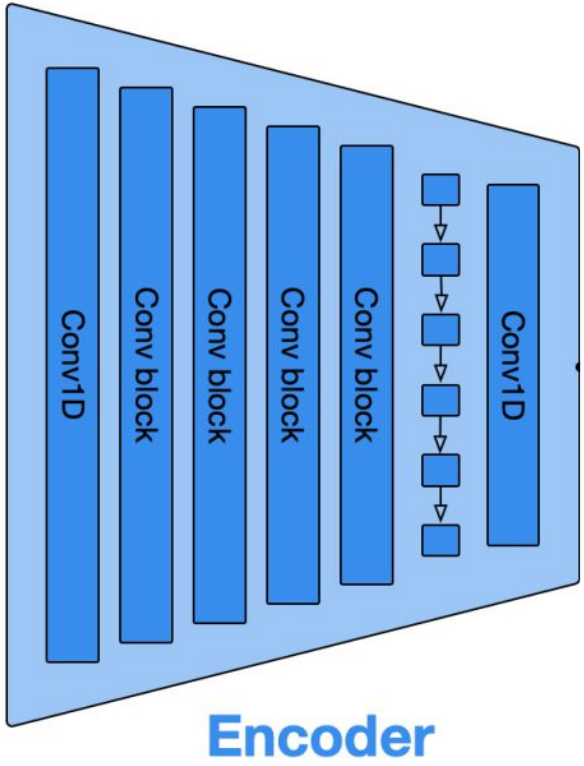
Например WAV, MP3, FLAC

Encodec — кодируем с помощью нейросетей



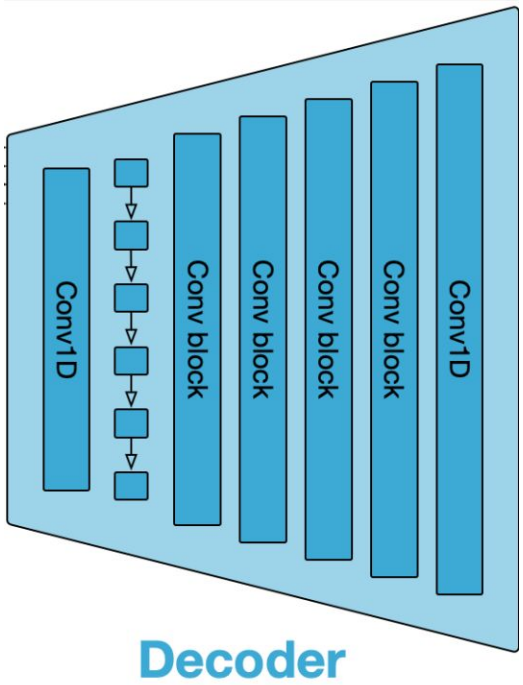
Энкодер

- Conv block - 1D свертка со skip-connection и strided 1D свертка для уменьшения длины
- Количество каналов увеличивается каждый conv block, на выходе длина сокращается в 320 раз
- Предпоследний слой - LSTM

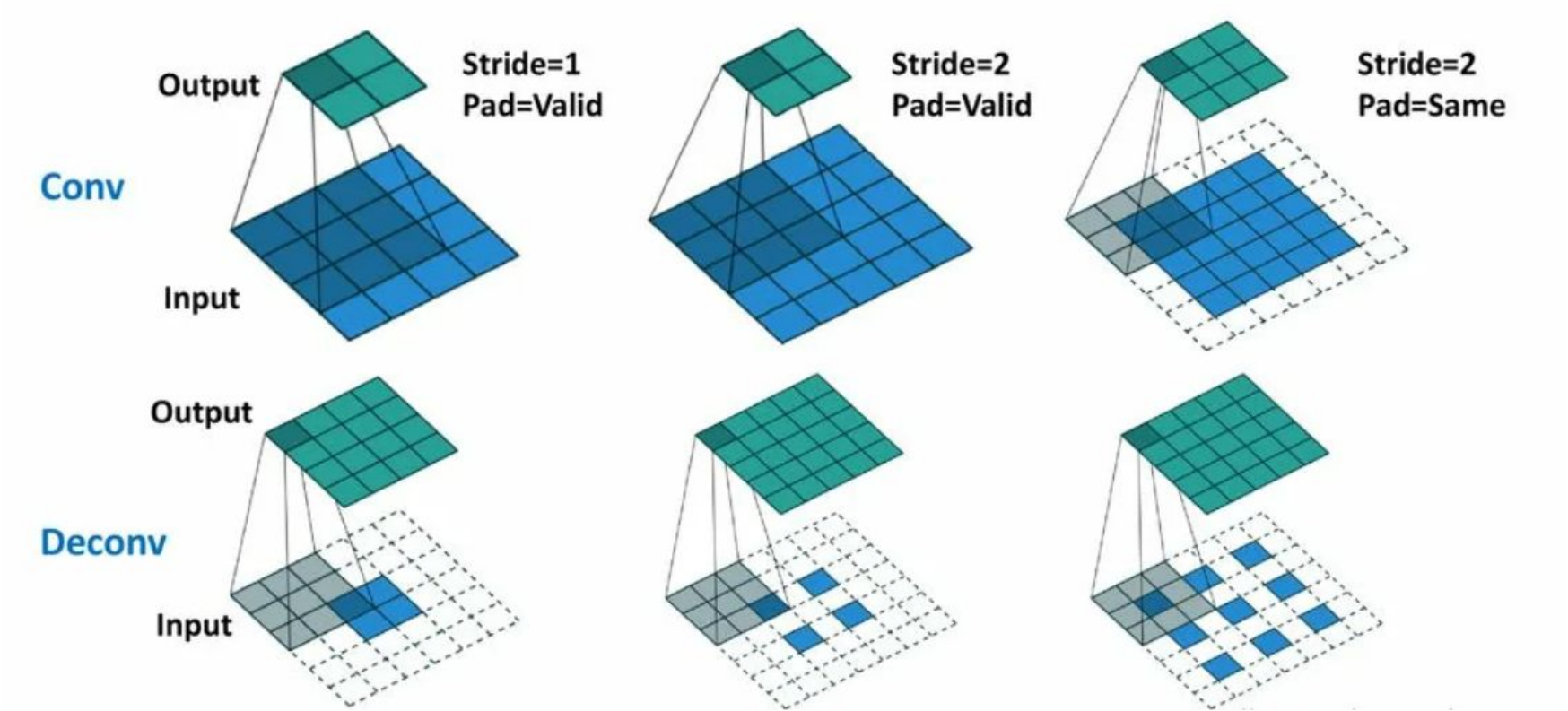


Декодер

- Здесь все то же самое, что и в энкодере, только в обратном порядке, и используются transposed 1D свертки

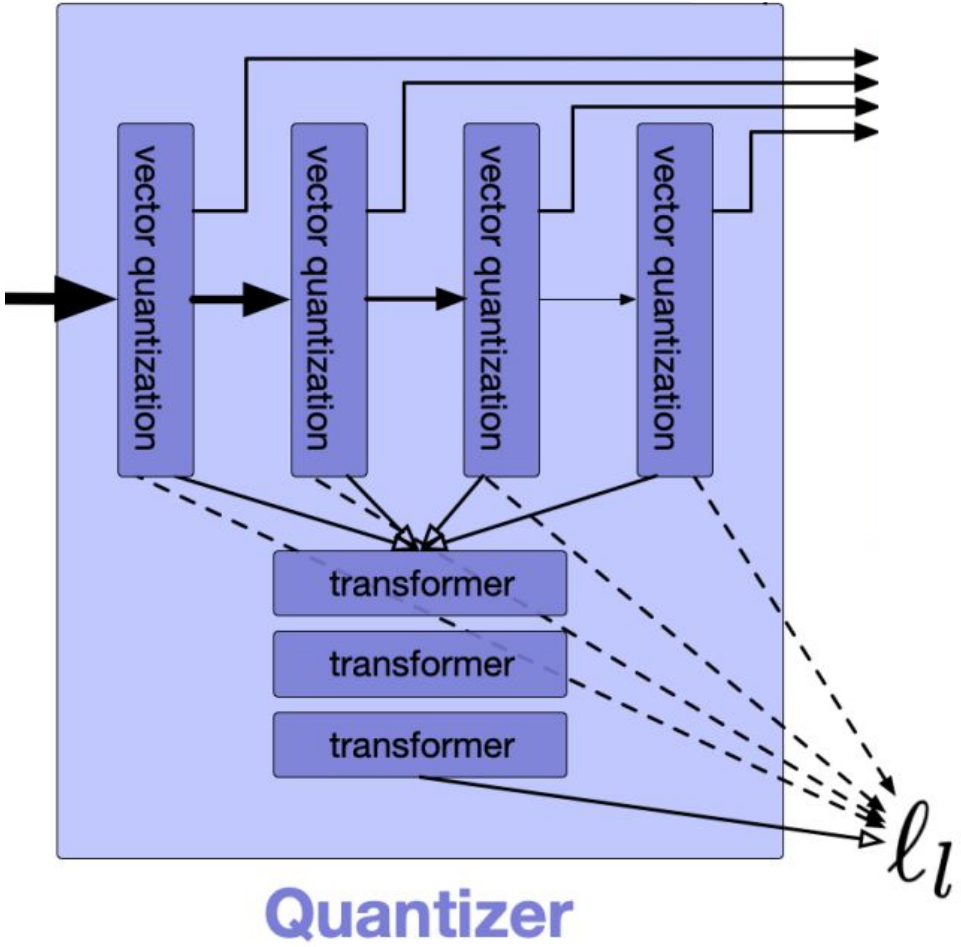


Transposed Convolution (Deconvolution)

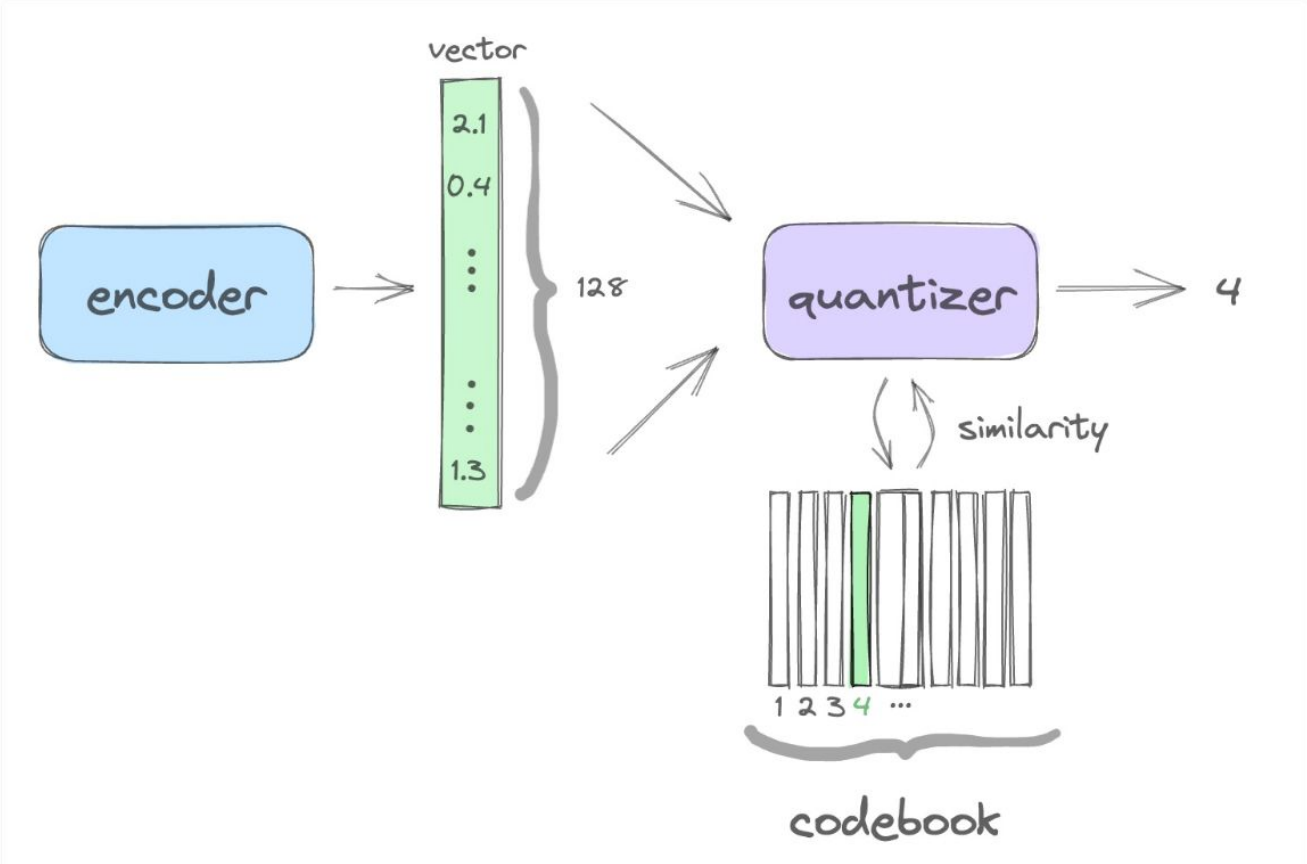


Квантизатор

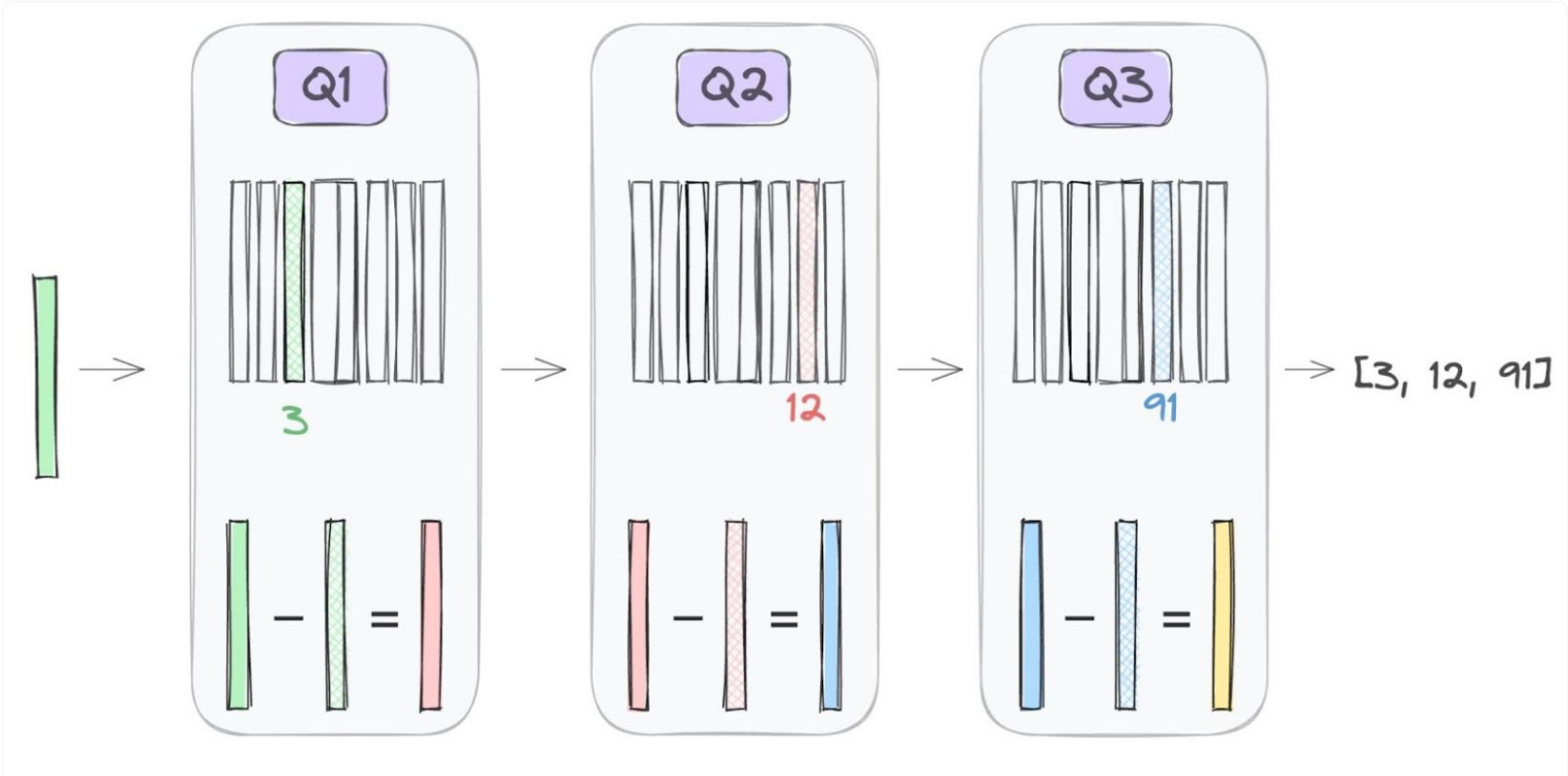
- Residual Vector Quantization
- Range-based arithmetic coder через Transformer



Residual Vector Quantization



Residual Vector Quantization

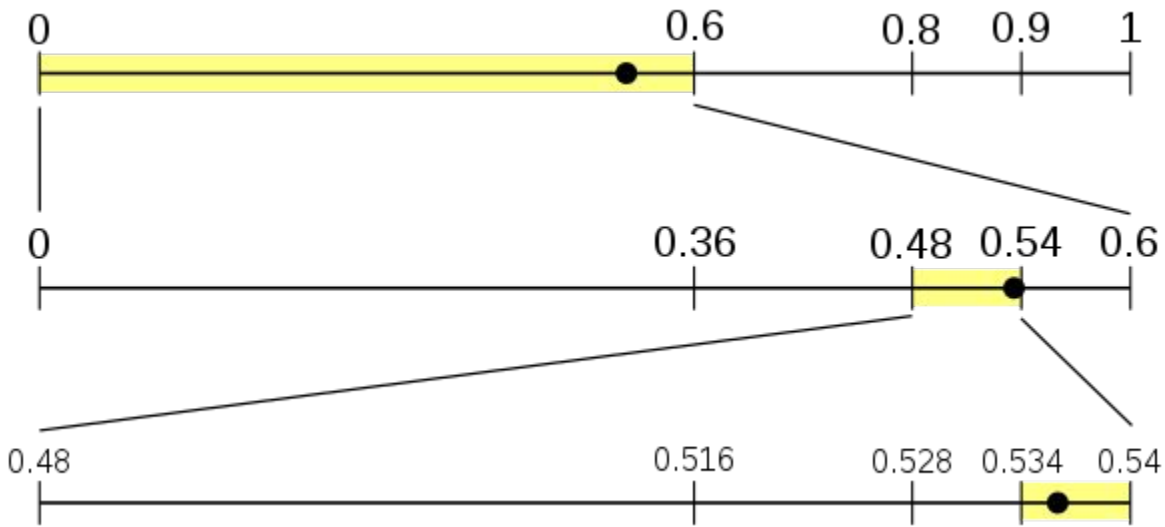


Arithmetic Coding

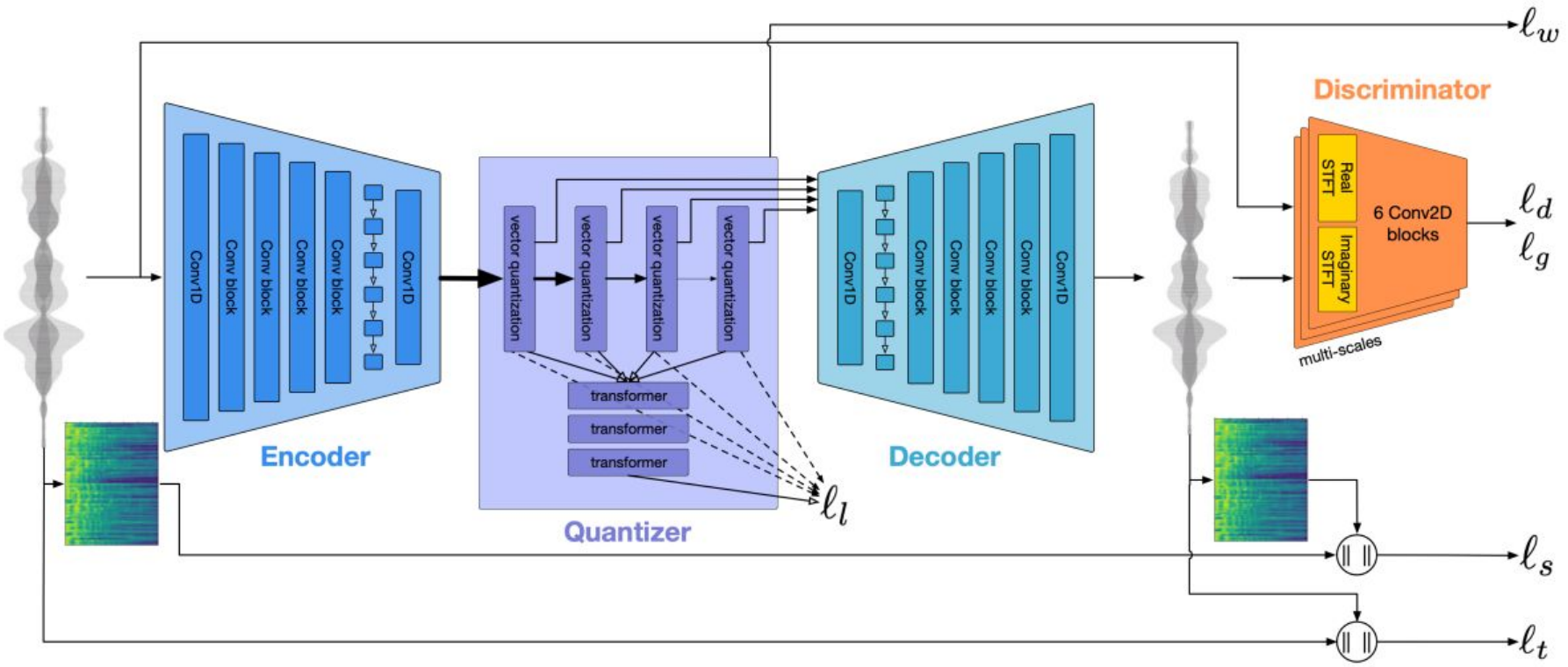
Пусть мы хотим закодировать последовательности из 4 возможных элементов - A, B, C, D

P(A) = 0.6
P(B) = 0.2
P(C) = P(D) = 0.1

Пусть мы хотим закодировать ACD



Encodec


ℓ_w
Discriminator
Real STFT
Imaginary STFT
6 Conv2D blocks
ℓ_d
ℓ_g
multi-scales
Conv1D
Conv block
Conv block
Conv block
Conv block
Conv block
Conv1D
Encoder
vector quantization
vector quantization
vector quantization
vector quantization
transformer
transformer
transformer
Quantizer
ℓ_l
Conv1D
Conv block
Conv block
Conv block
Conv block
Conv1D
Decoder
ℓ_s
ℓ_t

Как это учится?

Reconstruction Loss

$$\ell_t(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \left\| \boldsymbol{x} - \hat{\boldsymbol{x}} \right\|_1$$

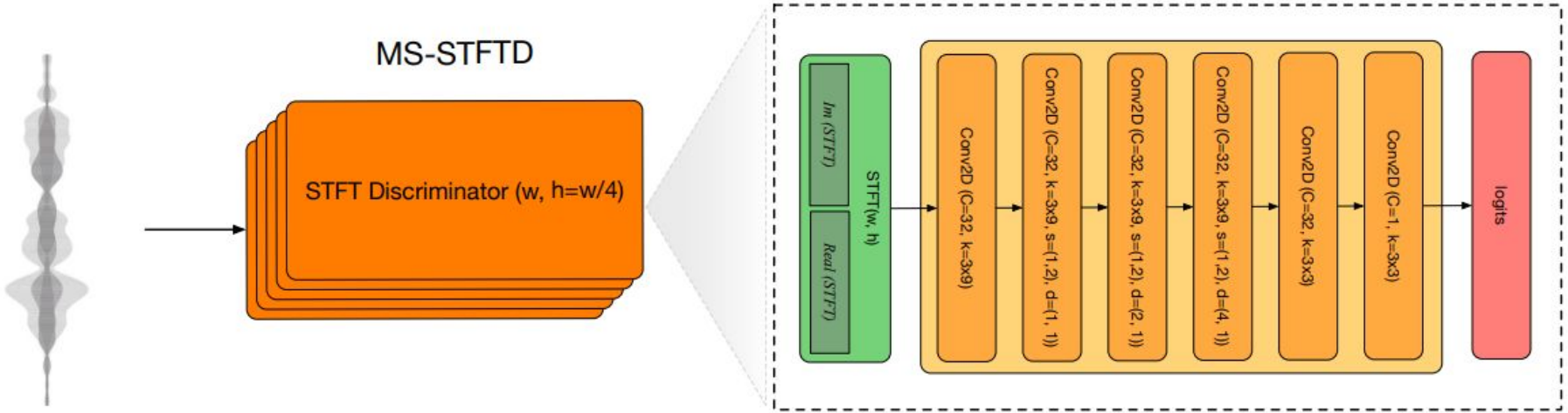
$$\ell_f(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{|\alpha| \cdot |s|} \sum_{\alpha_i \in \alpha} \sum_{i \in e} \|\mathcal{S}_i(\boldsymbol{x}) - \mathcal{S}_i(\hat{\boldsymbol{x}})\|_1 + \alpha_i \|\mathcal{S}_i(\boldsymbol{x}) - \mathcal{S}_i(\hat{\boldsymbol{x}})\|_2,$$

Как это учится?

Vector Quantization Loss

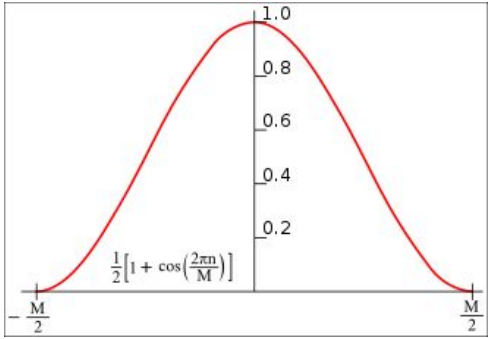
$$l_w = \sum_{c=1}^{C} \|\boldsymbol{z}_c - q_c(\boldsymbol{z}_c)\|_2^2$$

Дискриминатор

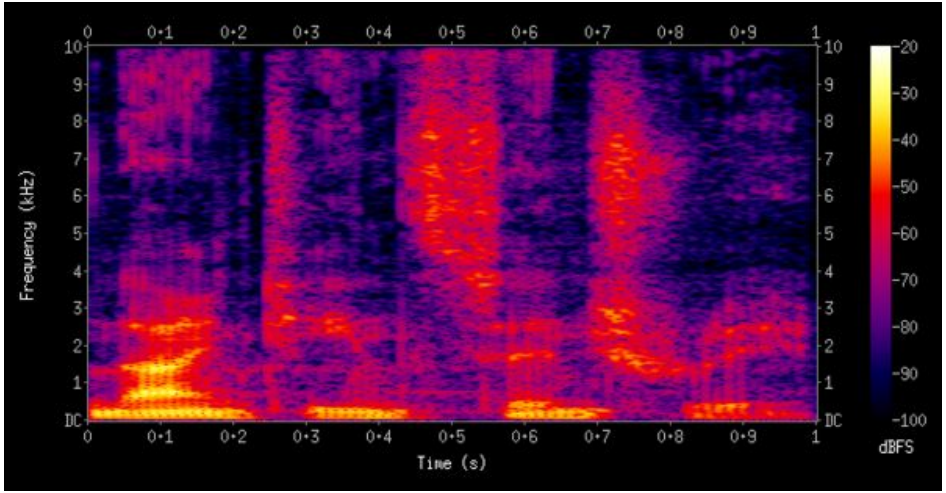


Оконное преобразование Фурье (STFT)

$$\mathbf{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-i\omega n}$$



$$\text{spectrogram}\{x(t)\}(\tau,\omega) \equiv |X(\tau,\omega)|^2$$



Как это учится?

Discriminative Loss

$$\ell_g(\hat{\boldsymbol{x}}) = \frac{1}{K}\sum_k \max(0, 1 - D_k(\hat{\boldsymbol{x}})))$$

$$L_d(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{K}\sum_{k=1}^{K} \max(0, 1 - D_k(\boldsymbol{x})) + \max(0, 1 + D_k(\hat{\boldsymbol{x}}))$$

$$\ell_{feat}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{KL}\sum_{k=1}^{K}\sum_{l=1}^{L} \frac{\|D_k^l(\boldsymbol{x}) - D_k^l(\hat{\boldsymbol{x}})\|_1}{\text{mean}\left(\|D_k^l(\boldsymbol{x})\|_1\right)}$$

Как это учится?

$$L_G = \lambda_t \cdot \ell_t(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda_f \cdot \ell_f(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda_g \cdot \ell_g(\hat{\boldsymbol{x}}) + \lambda_{feat} \cdot \ell_{feat}(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda_w \cdot \ell_w(w)$$

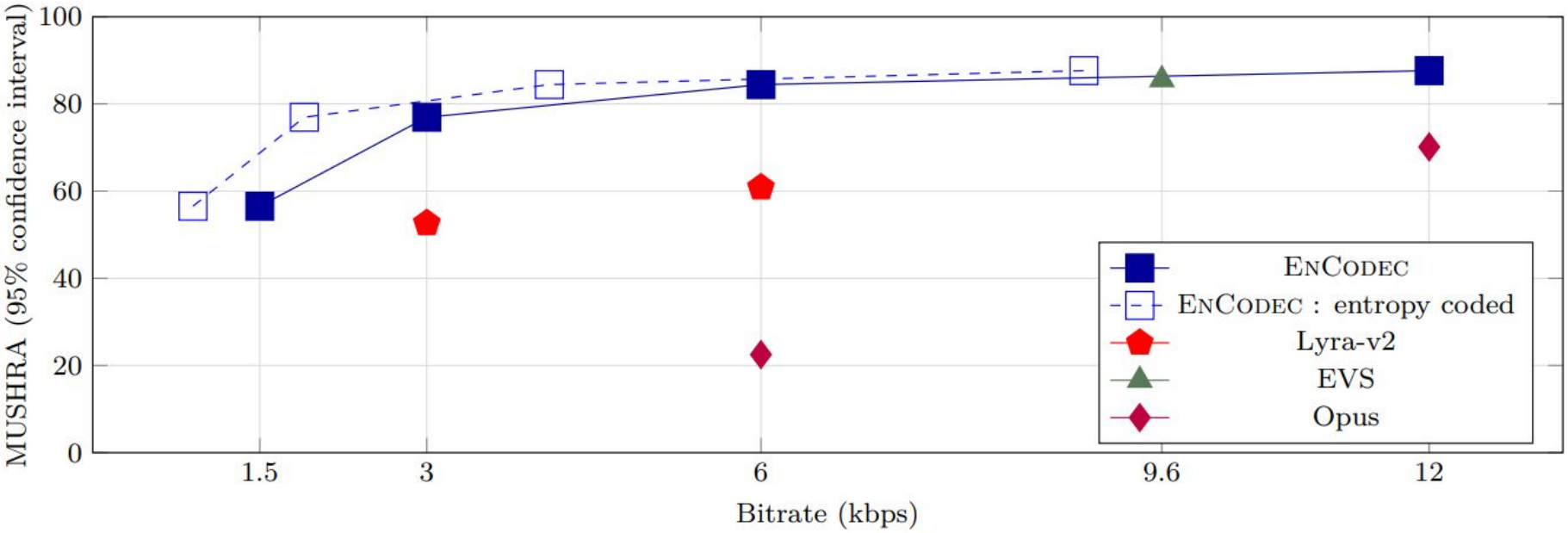
Loss Balancer

$g_i = \frac{\partial \ell_i}{\partial \hat{x}}$ — градиенты лоссов

$\langle \|g_i\|_2 \rangle_\beta$ — экспоненциальное скользящее среднее

$\tilde{g}_i = R \frac{\lambda_i}{\sum_j \lambda_j} \cdot \frac{g_i}{\langle \|g_i\|_2 \rangle_\beta}$ — вот это бэкпропогейтим

Результаты


MUSHRA (95% confidence interval)
100
80
60
40
20
0
1.5
3
6
9.6
12
Bitrate (kbps)
EnCodec
EnCodec : entropy coded
Lyra-v2
EVS
Opus

Результаты

MUSHRA

Model	Bandwidth	Entropy Coded	Clean Speech	Noisy Speech	Music Set-1	Music Set-2
Reference	-	-	95.5±1.6	93.9±1.8	93.2±2.5	97.1±1.3
Opus	6.0 kbps	-	30.1±2.8	19.1±5.9	20.6±5.8	17.9±5.3
Opus	12.0 kbps	-	76.5±2.3	61.9±2.1	77.8±3.2	65.4±2.7
EVS	9.6 kbps	-	84.4±2.5	80.0±2.4	89.9±2.3	87.7±2.3
Lyra-v2	3.0 kbps	-	53.1±1.9	52.0±4.7	69.3±3.3	42.3±3.5
Lyra-v2	6.0 kbps	-	66.2±2.9	59.9±3.3	75.7±2.6	48.6±2.1
EnCodec	1.5 kbps	0.9 kbps	49.2±2.4	41.3±3.6	68.2±2.2	66.5±2.3
EnCodec	3.0 kbps	1.9 kbps	67.0±1.5	62.5±2.3	89.6±3.1	87.8±2.9
EnCodec	6.0 kbps	4.1 kbps	83.1±2.7	69.4±2.3	92.9±1.8	91.3±2.1
EnCodec	12.0 kbps	8.9 kbps	90.6±2.6	80.1±2.5	91.8±2.5	92.9±1.2

Результаты

Model	Streamable	SI-SNR	ViSQOL
Opus	✓	2.45	2.60
EVS	✓	1.89	2.74
EnCodec	✓	6.67	4.35
EnCodec	✗	7.46	4.39

$$\mathbf{s}_{target} = \frac{\langle \hat{\mathbf{s}}, \mathbf{s} \rangle \mathbf{s}}{\|\mathbf{s}\|^2}$$

$$\mathbf{e}_{noise} = \hat{\mathbf{s}} - \mathbf{s}_{target}$$

$$\text{SI-SNR} := 10\, log_{10} \frac{\|\mathbf{s}_{target}\|^2}{\|\mathbf{e}_{noise}\|^2}$$

ViSQOL — метрика, которая сравнивает, насколько похожи спектрограммы оригинала и восстановленного звука

Спасибо за внимание!

Источники

- High Fidelity Neural Audio Compression