

# **Методы решения задач со звуком**

**Автор:**

Гринберг Петр

# Вспомним про звук в компьютере

Сэмплирование

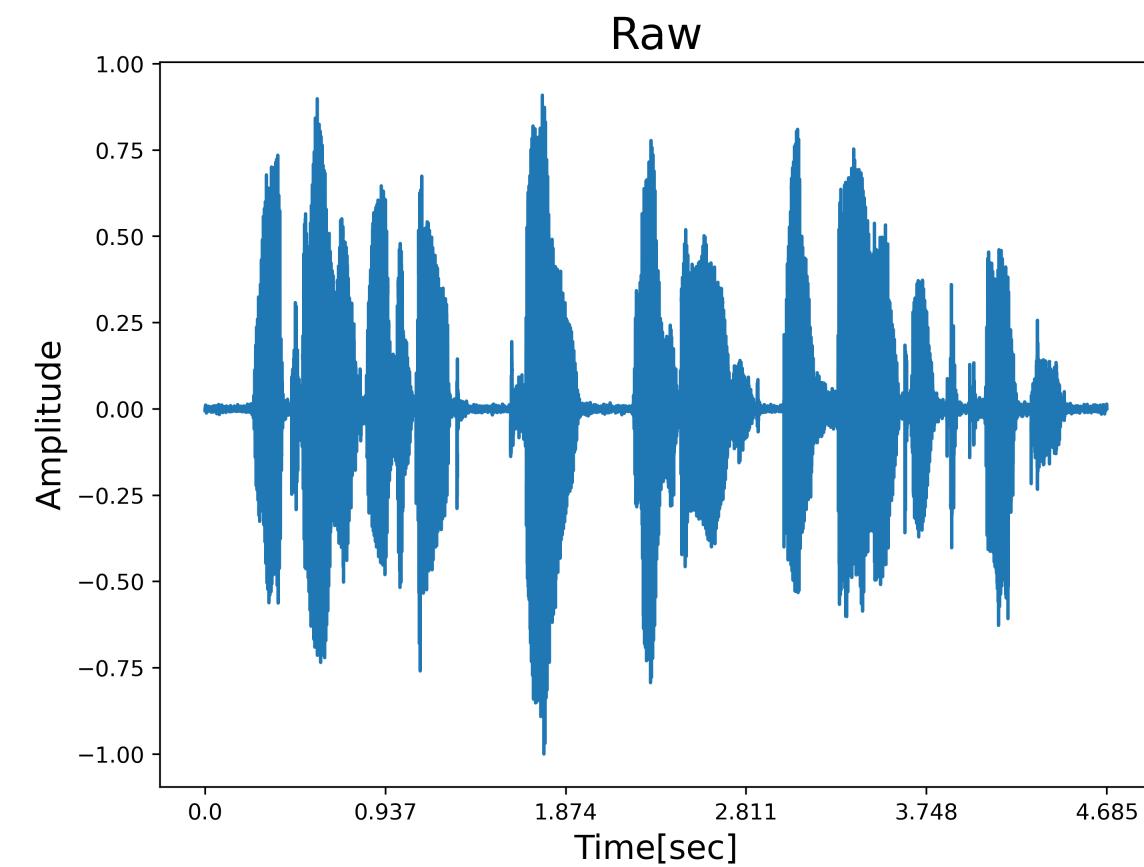
Квантизация

Клиппирование

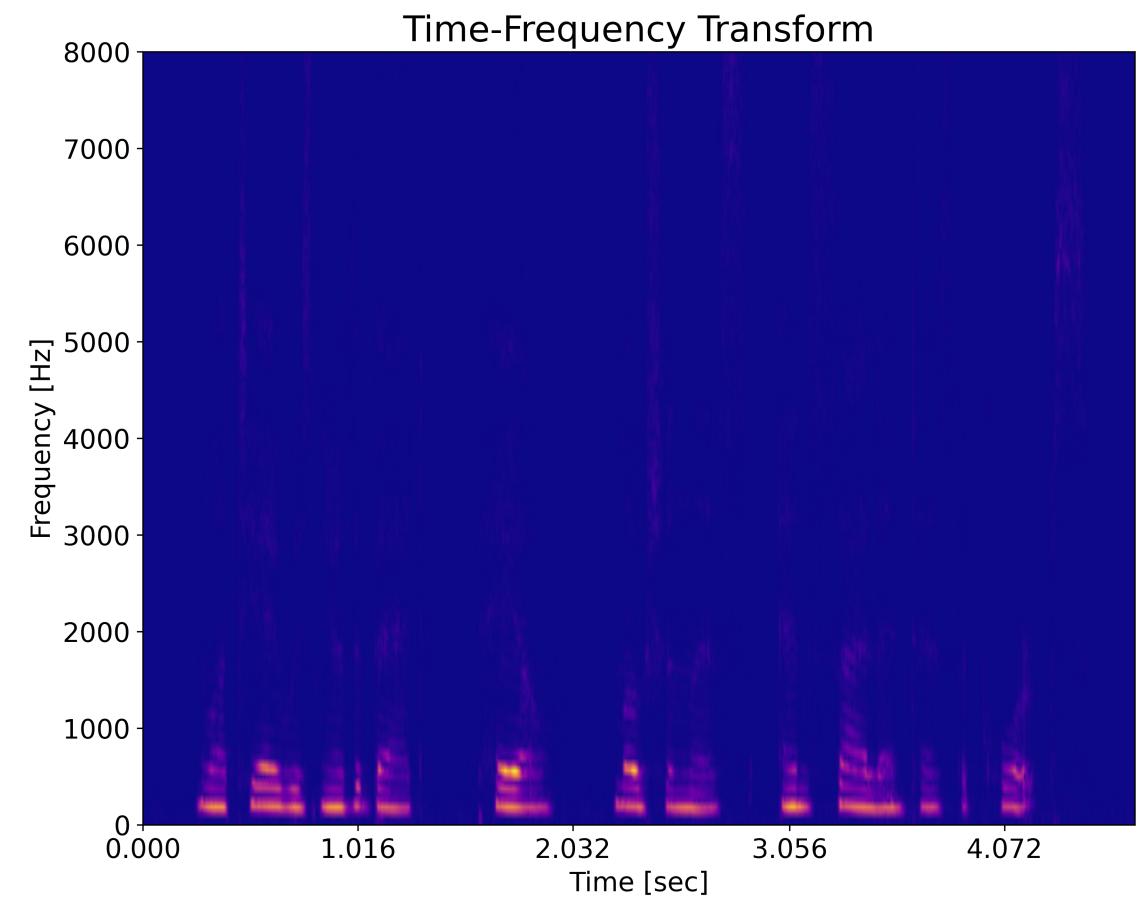
$$x_a(t) \implies x[n] = x_a(nT)$$

Представление не единственное и зависит от  $T$ .

# Представления звука на практике



$x[n]$



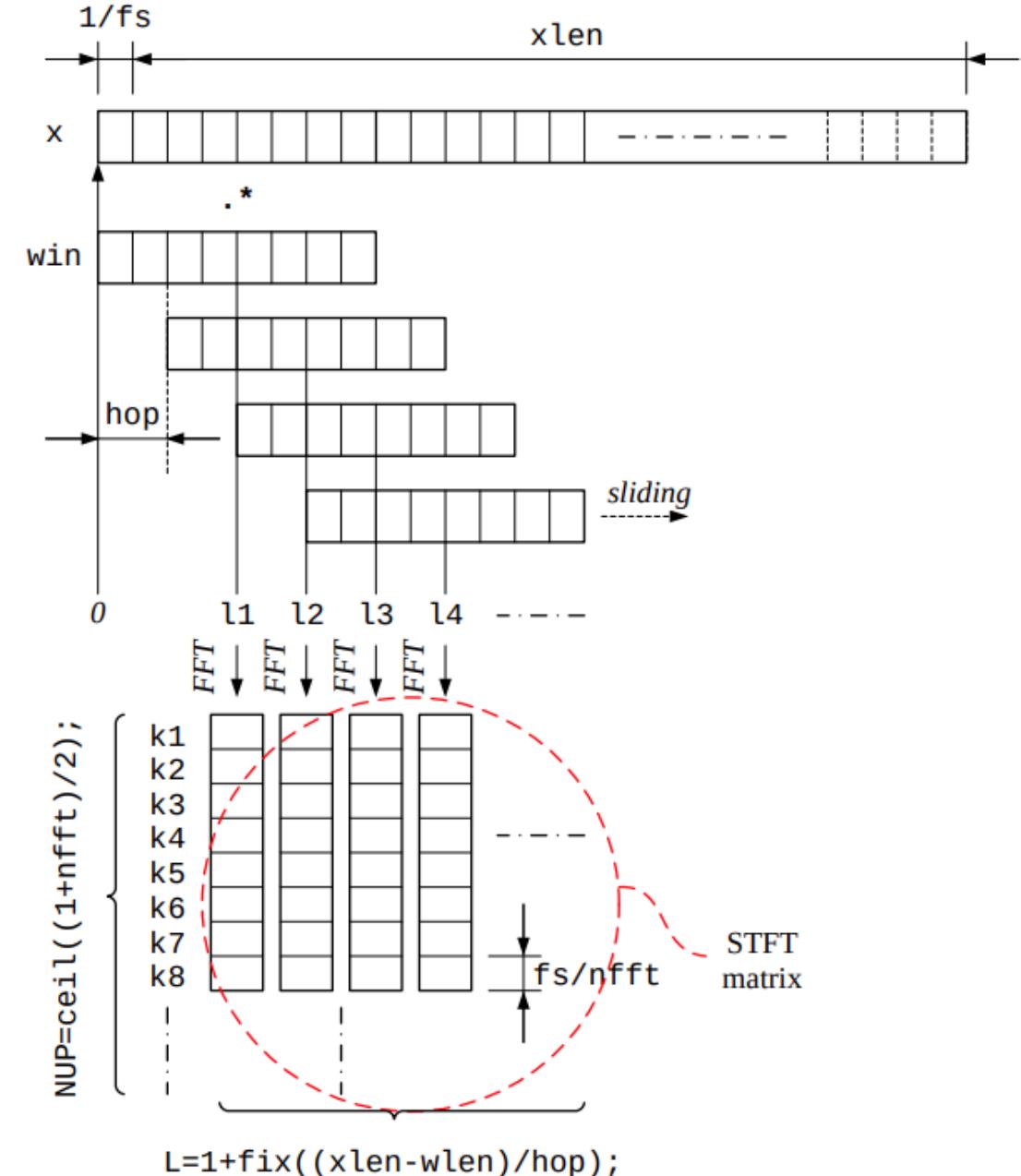
$X[f, t]$

# Частотно-временные преобразования

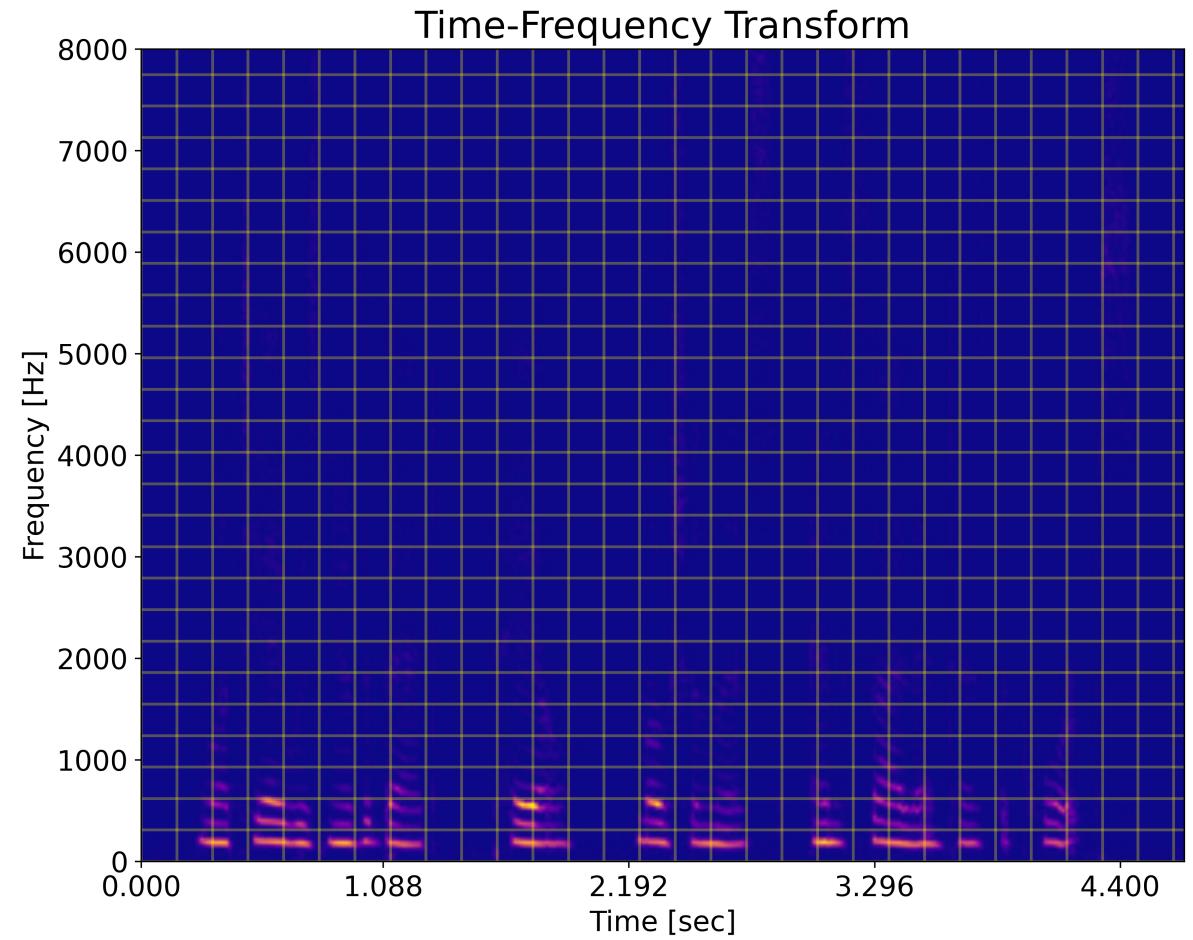
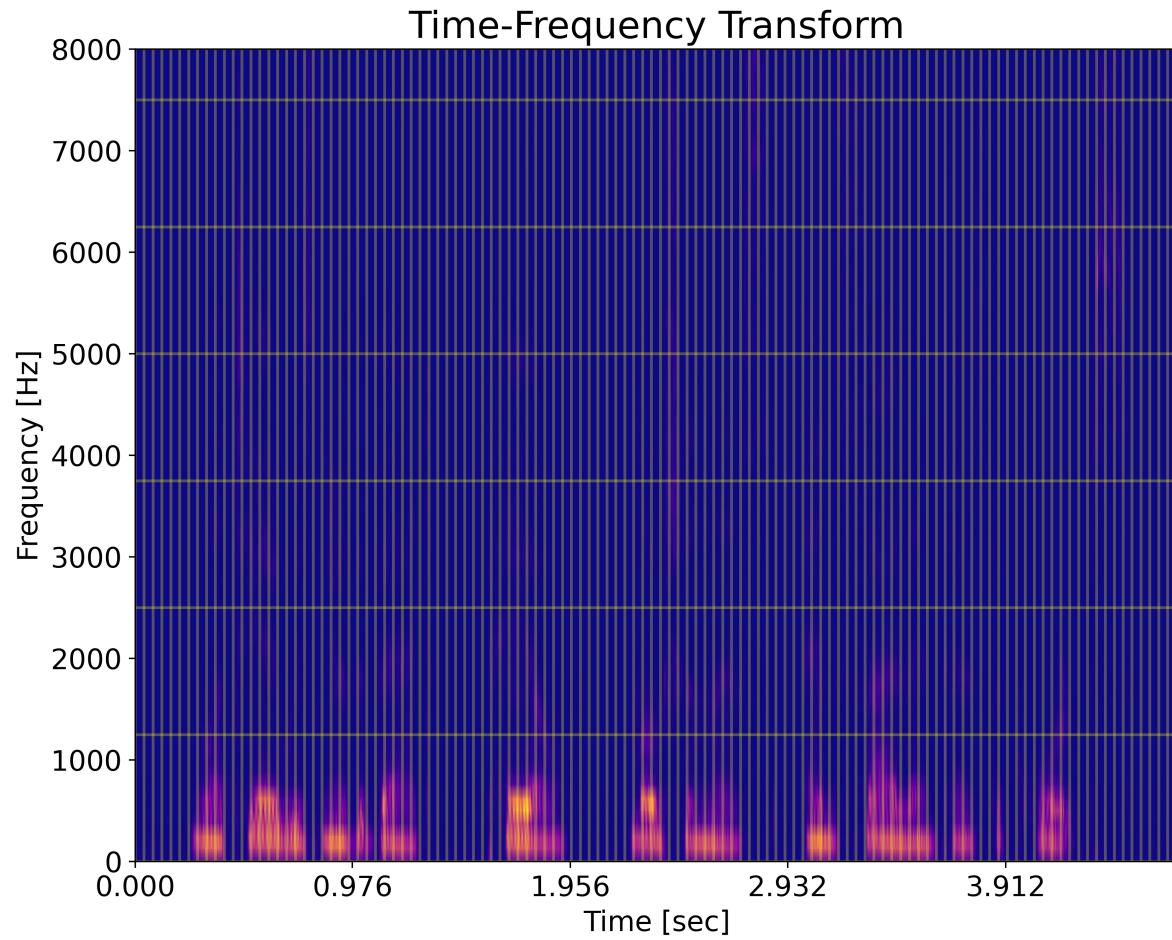
Бывают разные:

- STFT
- MEL
- CQT

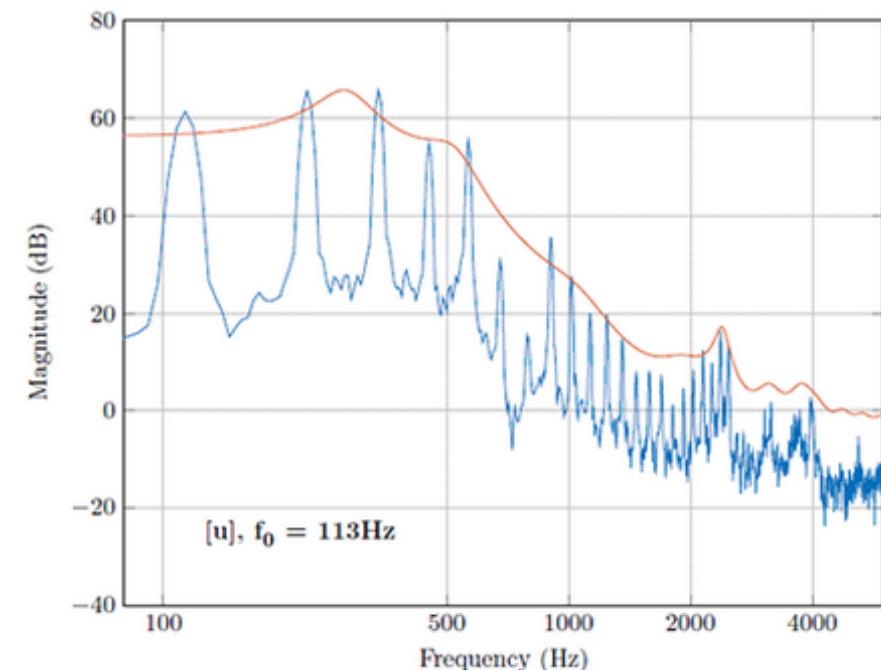
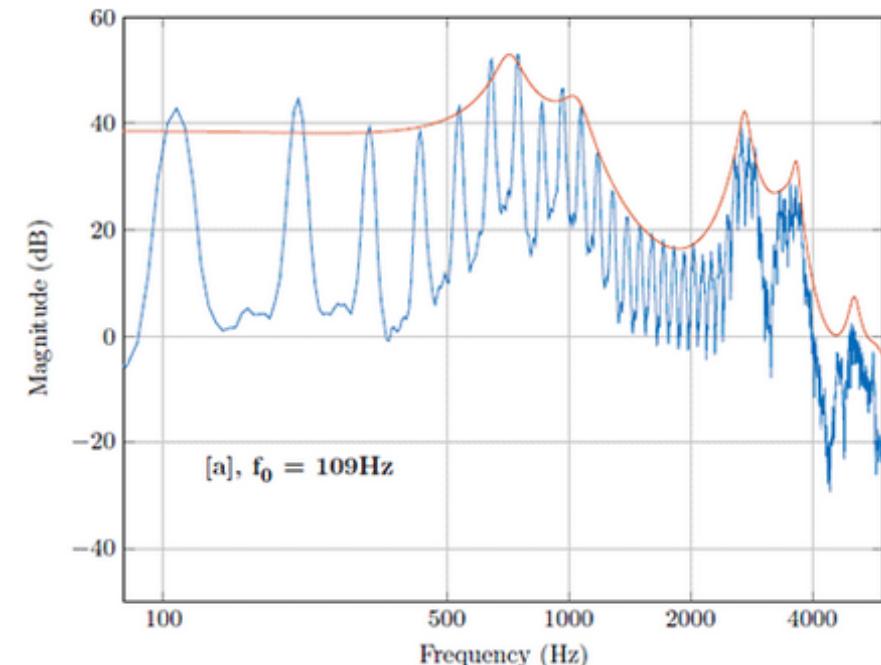
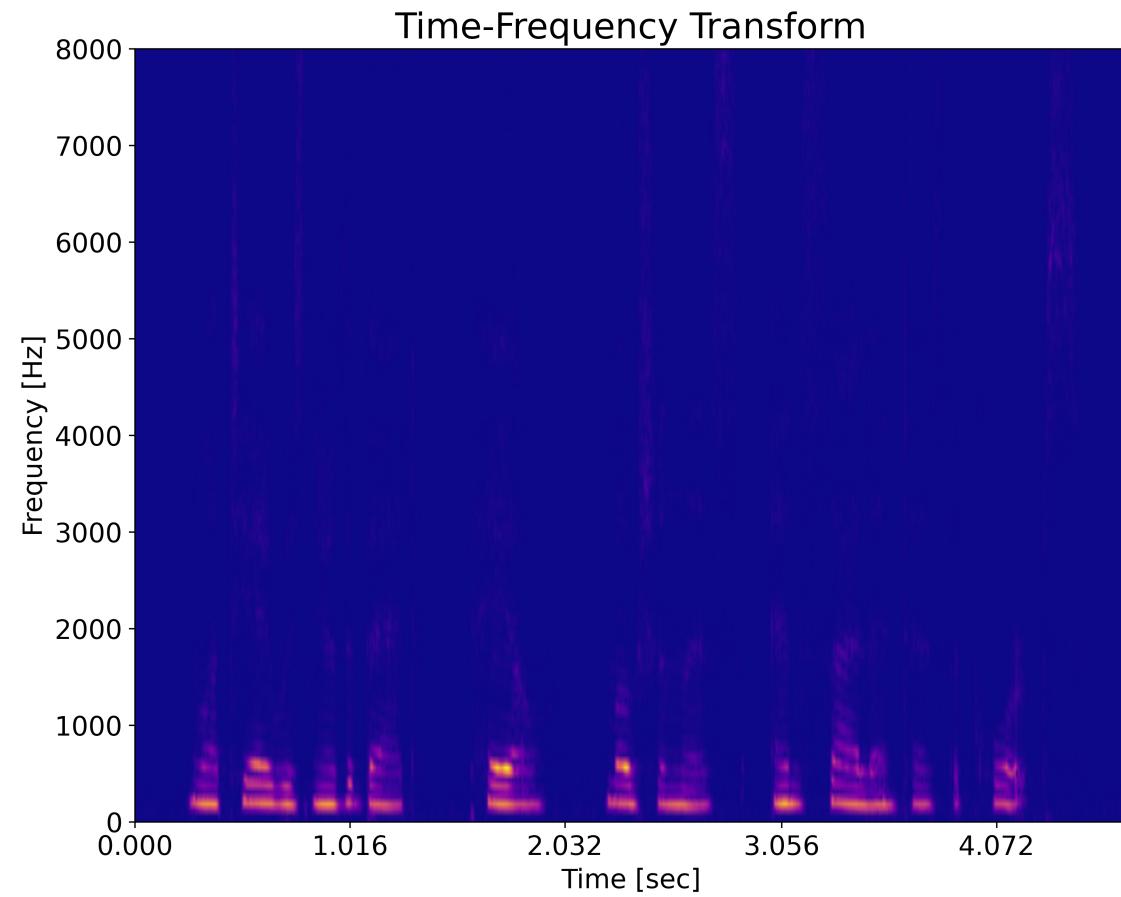
Может быть обратимо при выполнении условий OLA



# Разрешение ЧВ-преобразования



# Зачем оно надо

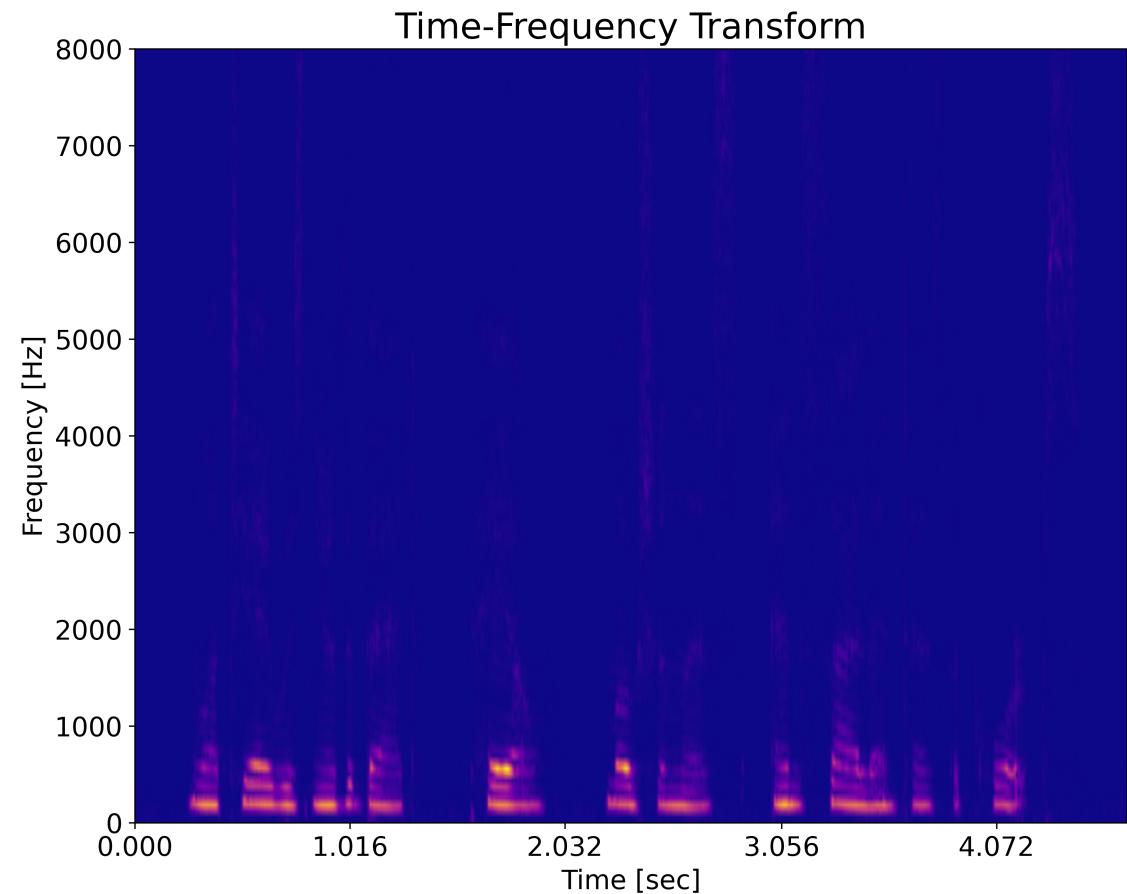


# Пример

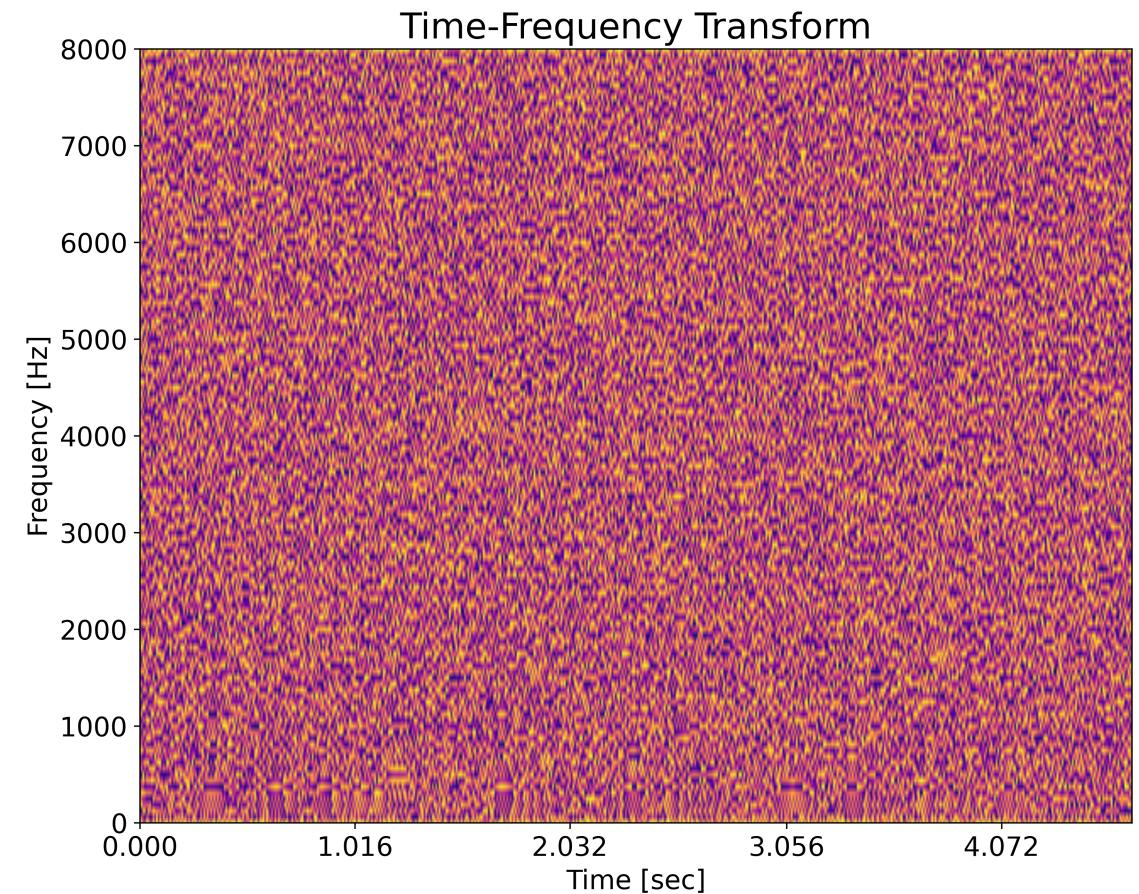
Хотим убрать шумы из сигнала с речью

- Берем любую сетку из CV для сегментации по картинке.  
Какой-нибудь encoder-decoder, типа U-net.
- Подаем на вход спектrogramму  $X$
- Получаем маску  $G$
- Выдаем "чистую" спектrogramму  $Y = G \odot X$
- Делаем обратное преобразование к Y

# Магнитуда, Фаза

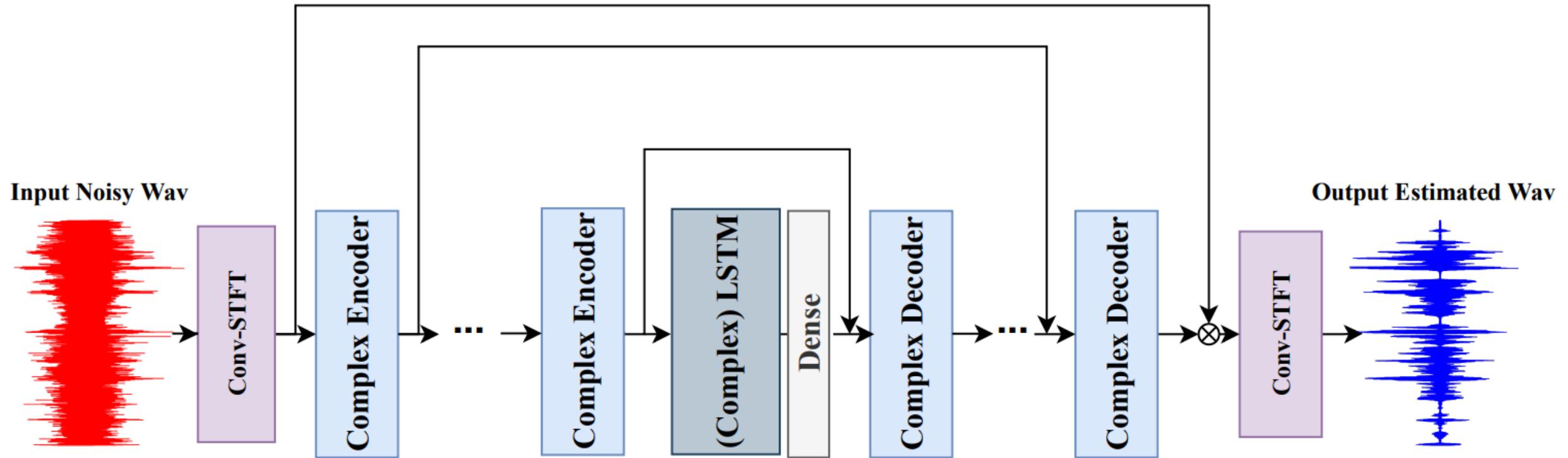


$$|X[f, t]|$$



$$\angle X[f, t]$$

# DCCRN



DCCRN: Deep Complex Convolution Recurrent Network for Phase-Aware Speech Enhancement

# DCCRN

Комплексные свертки:

$$F_{out} = (X_r * W_r - X_i * W_i) + j(X_r * W_i + X_i * W_r)$$

Комплексный LSTM:

$$F_{ab} = \text{LSTM}_a(X_b), \text{ где } a, b \in \{r, i\}$$

$$F_{out} = (F_{rr} - F_{ii}) + j(F_{ri} + F_{ir})$$

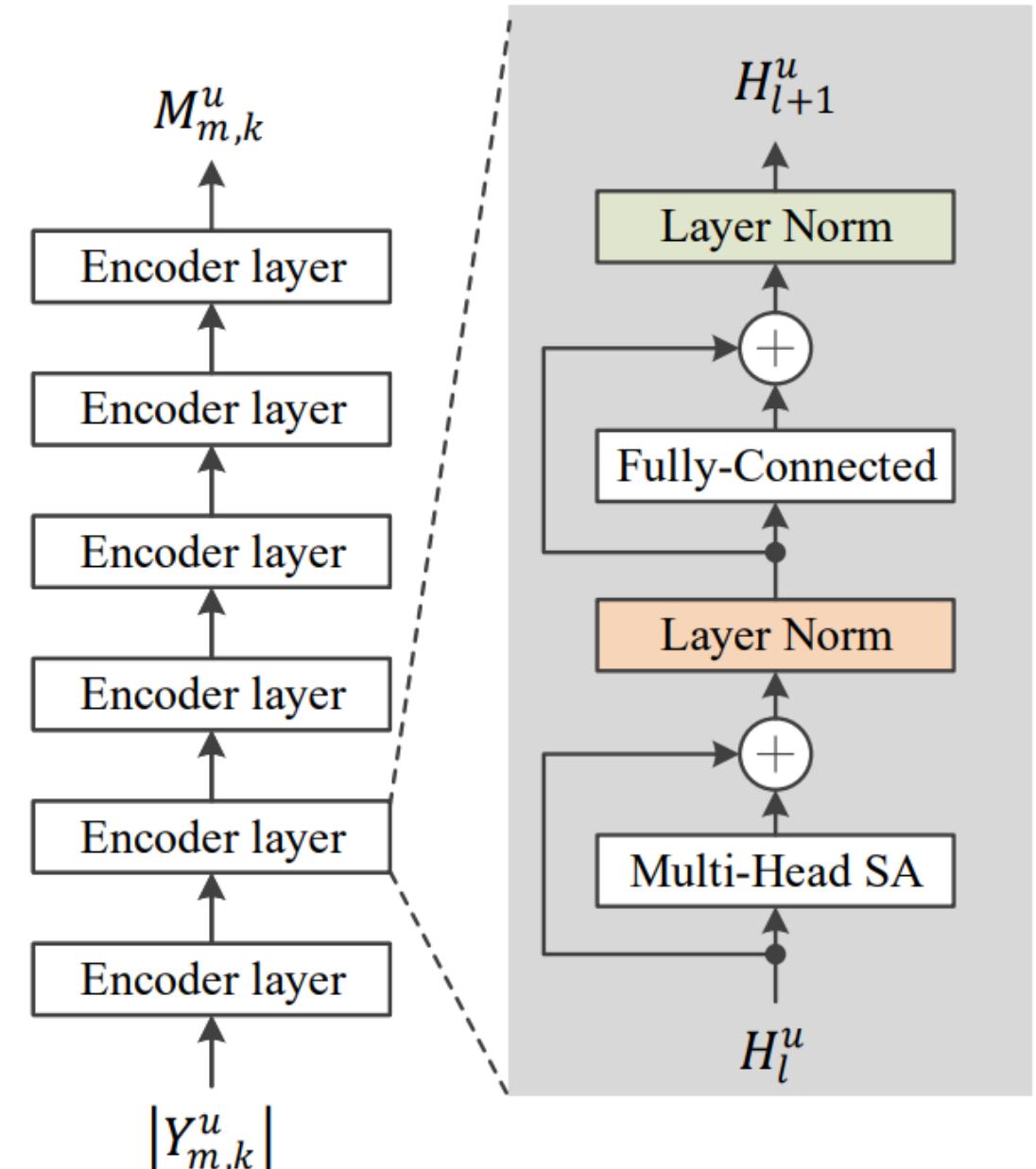
# T-GSA (Real)

Формула для Multi-Head SA:

$$G_l = (g_{ij}^l), g_{ij}^l = e^{\frac{-|i-j|^2}{\sigma_l^2}}$$

$$S_l = G_l \odot \left( \frac{Q_l(K_l)^T}{\sqrt{d}} \right)$$

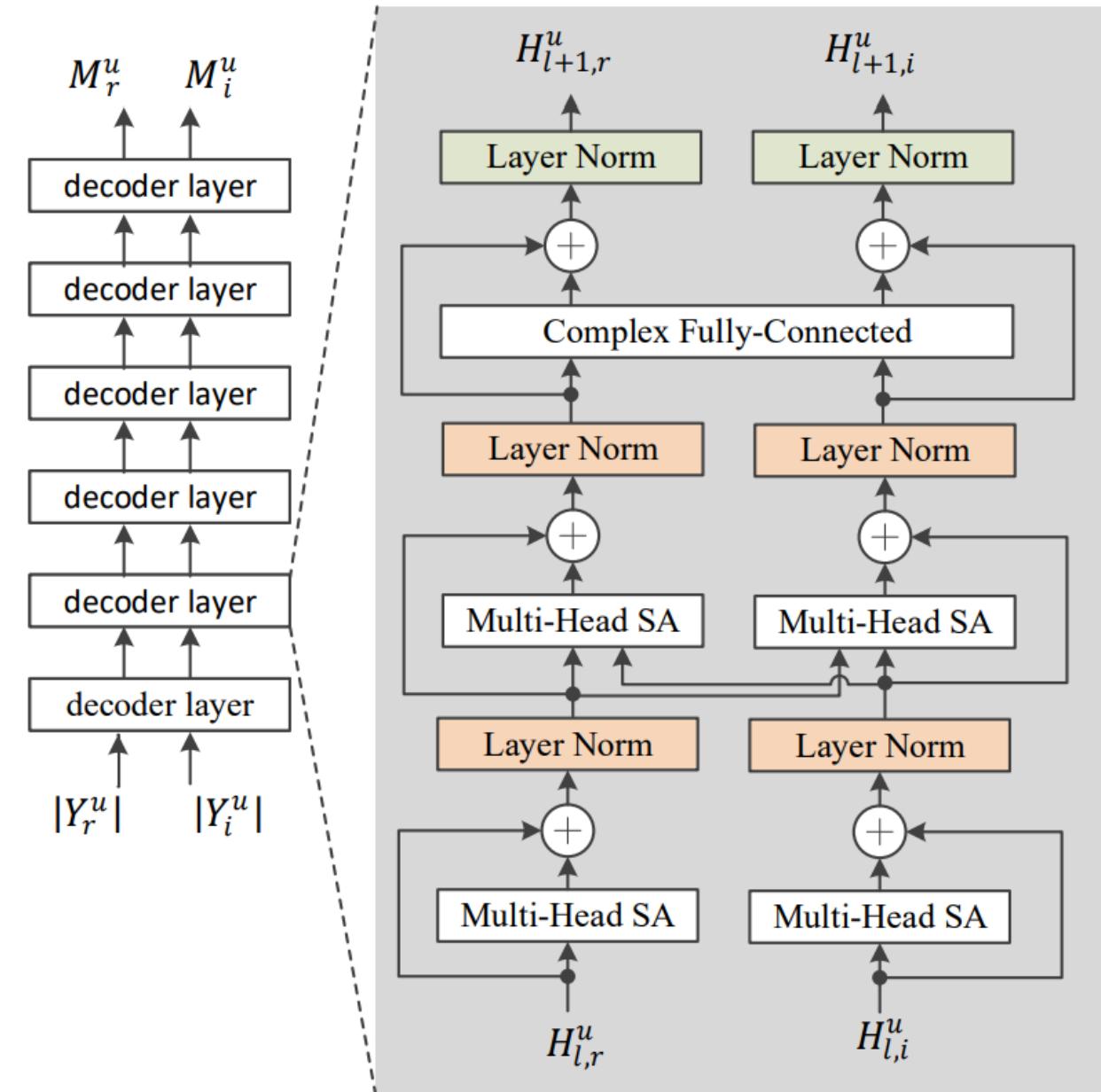
$$O_l = \text{SoftMax}(|S_l|)V_l$$



T-GSA: Transformer with Gaussian-weighted self-

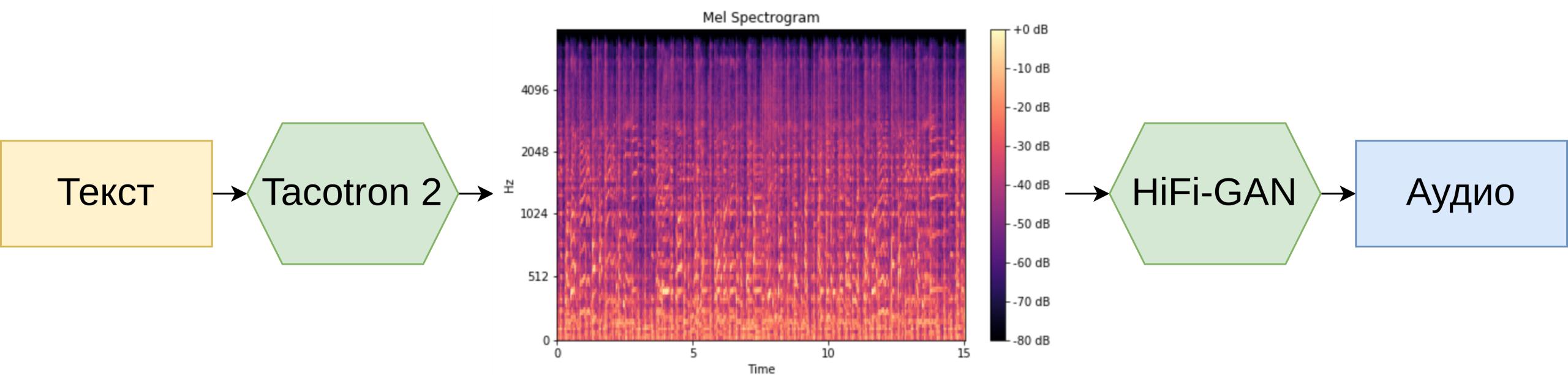
# T-GSA (Complex)

- Первые Multi-Head SA независимы
- Последующие получают K, V с одной стороны, Q с другой



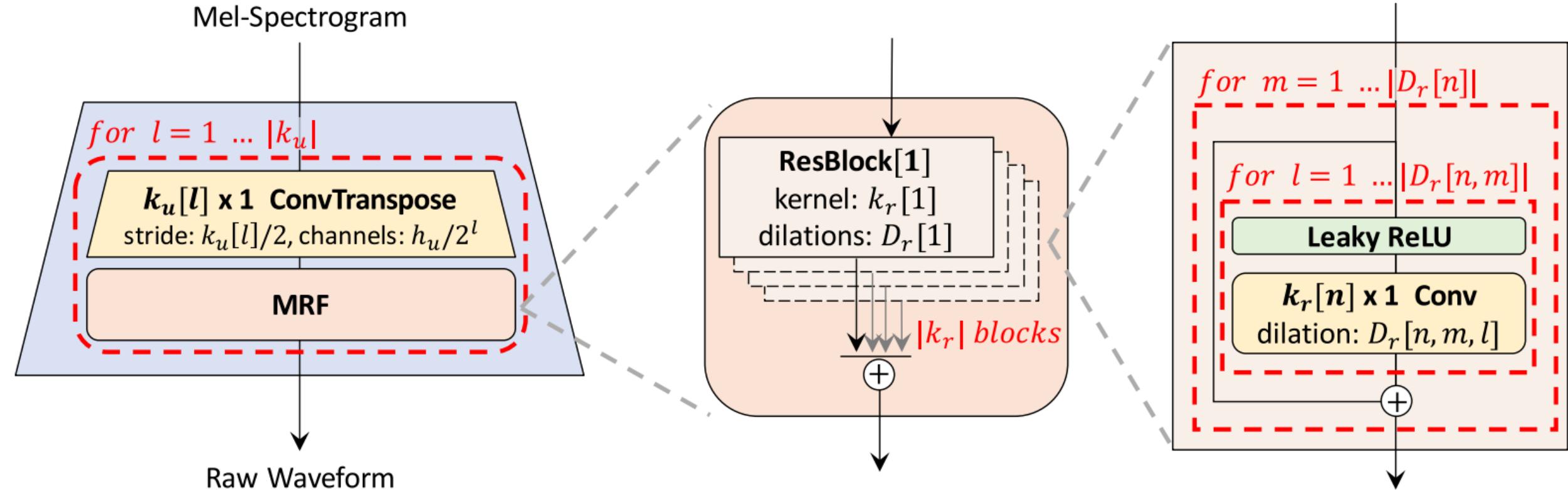
-attention for speech enhancement

# Другая задача

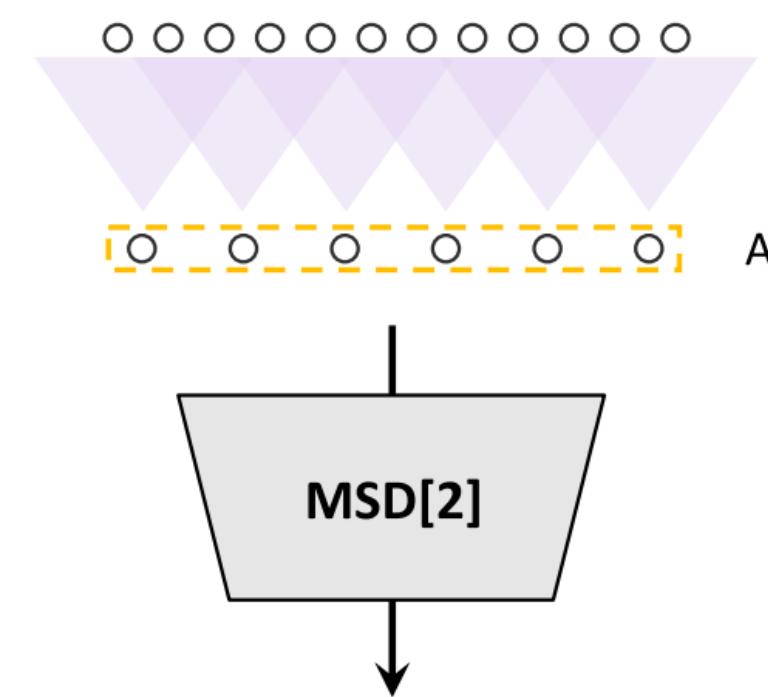


Одной Mel спектрограммой не обойтись:  
Не те размеры, нет фазы, невозможнo преобразовать обратно

# HiFi-GAN: Generator



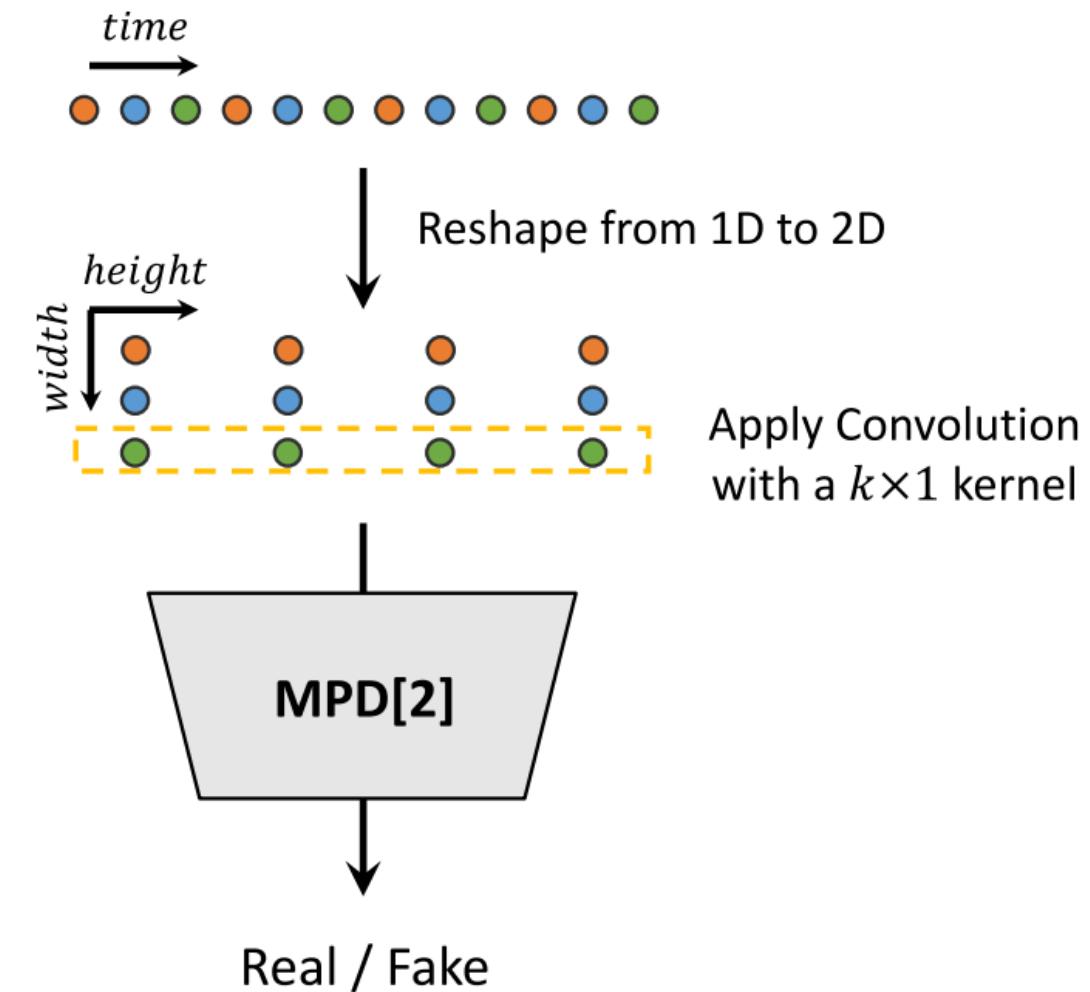
# HiFi-GAN: Discriminator



Average Pooling  
Apply Convolution

**MSD[2]**

Real / Fake



height  
width

Reshape from 1D to 2D

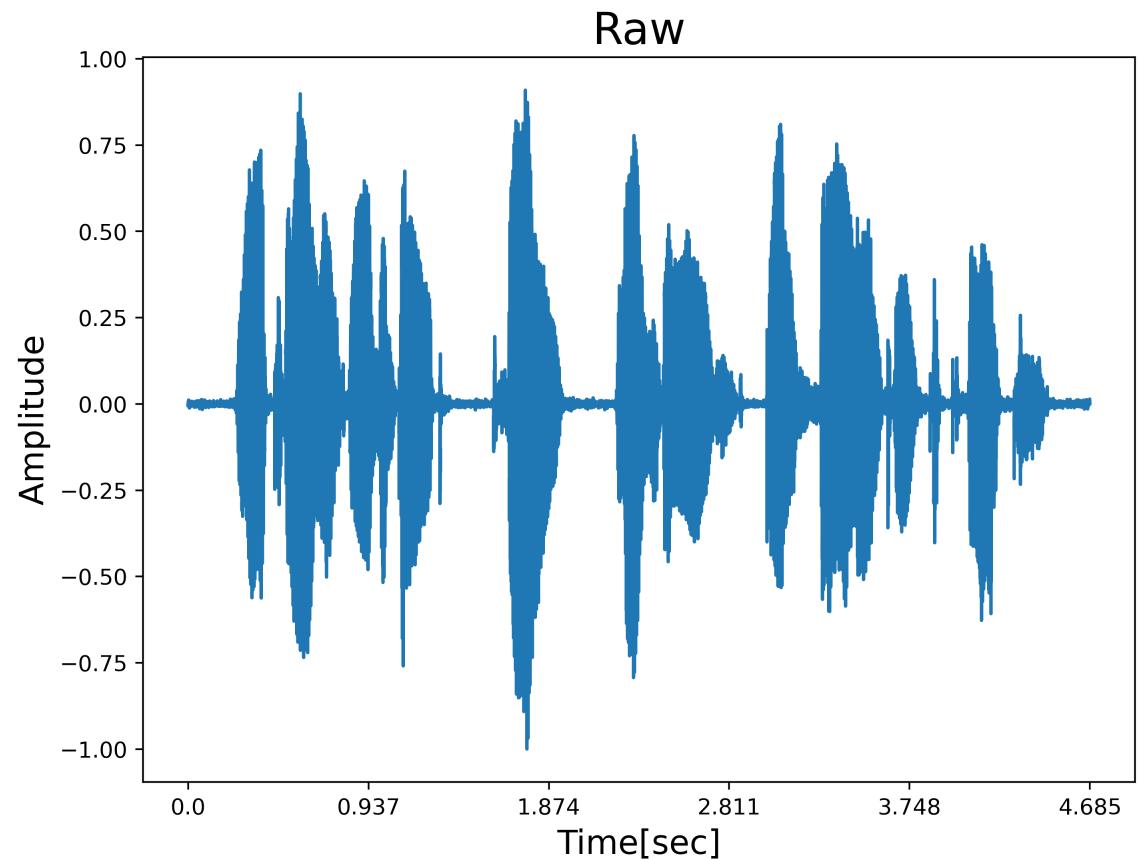
Apply Convolution  
with a  $k \times 1$  kernel

**MPD[2]**

Real / Fake

# А что можно делать с $x[n]$ ?

- В ранее рассмотренной задаче шумоподавления:
  - Вариации LSTM
  - Трансформеры
- В других задачах:
  - Одномерные свертки



# Sinc-layer

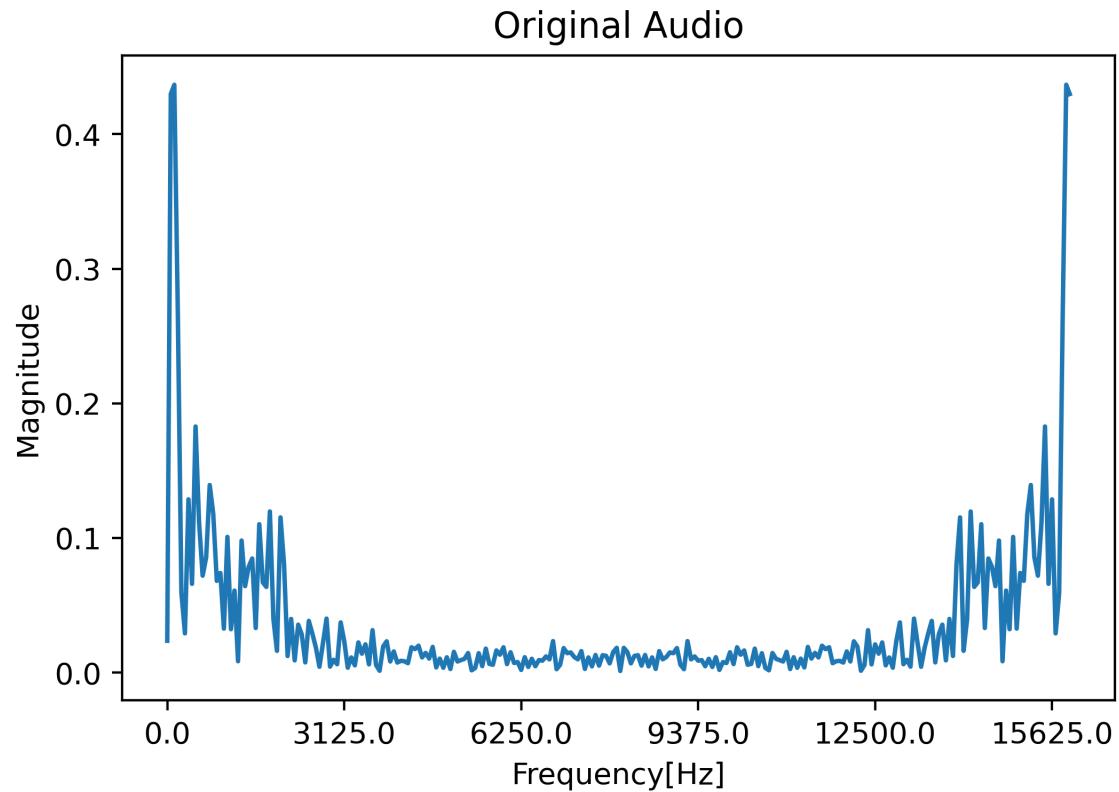
Обычная свертка в CNN (L - размер фильтра):

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l]h[n-l]$$

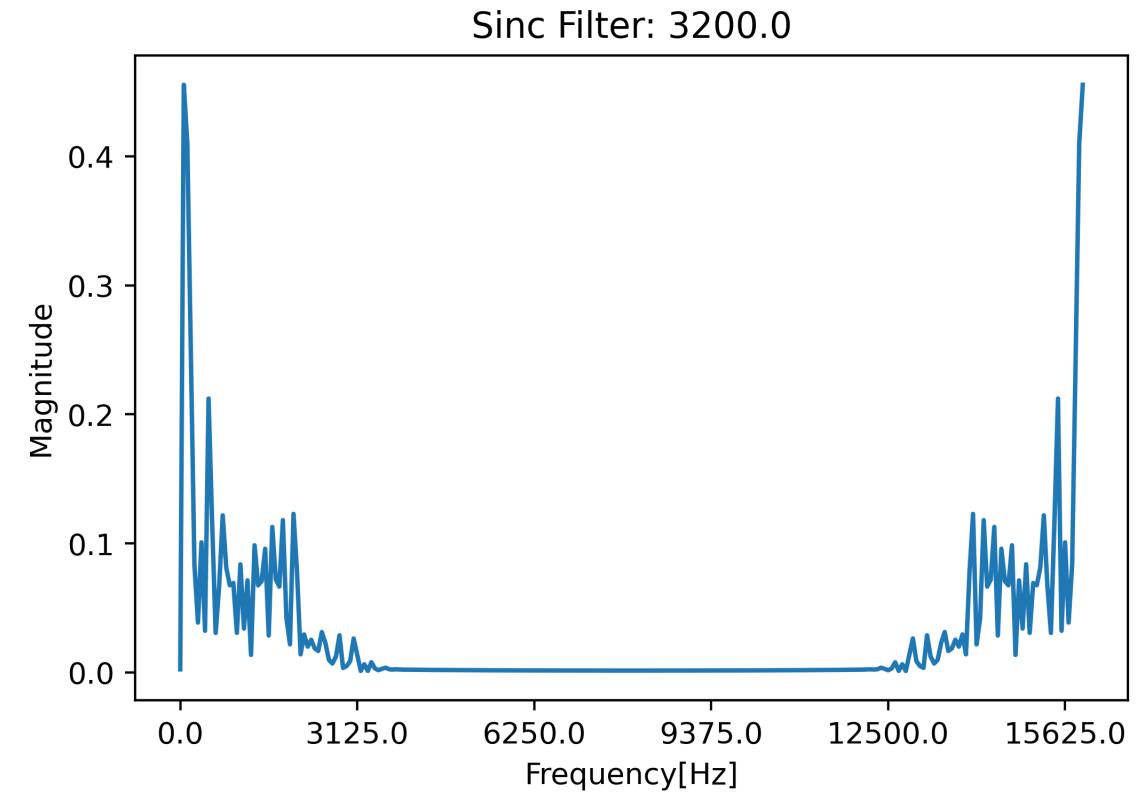
Вместо  $h$  с  $L$  обучаемыми параметрами возьмем следующее  $g$ :

$$g[n, f_{max}, f_{min}] = 2f_{max}\text{sinc}(2\pi f_{max}n) - 2f_{min}\text{sinc}(2\pi f_{min}n)$$

# Фильтрация

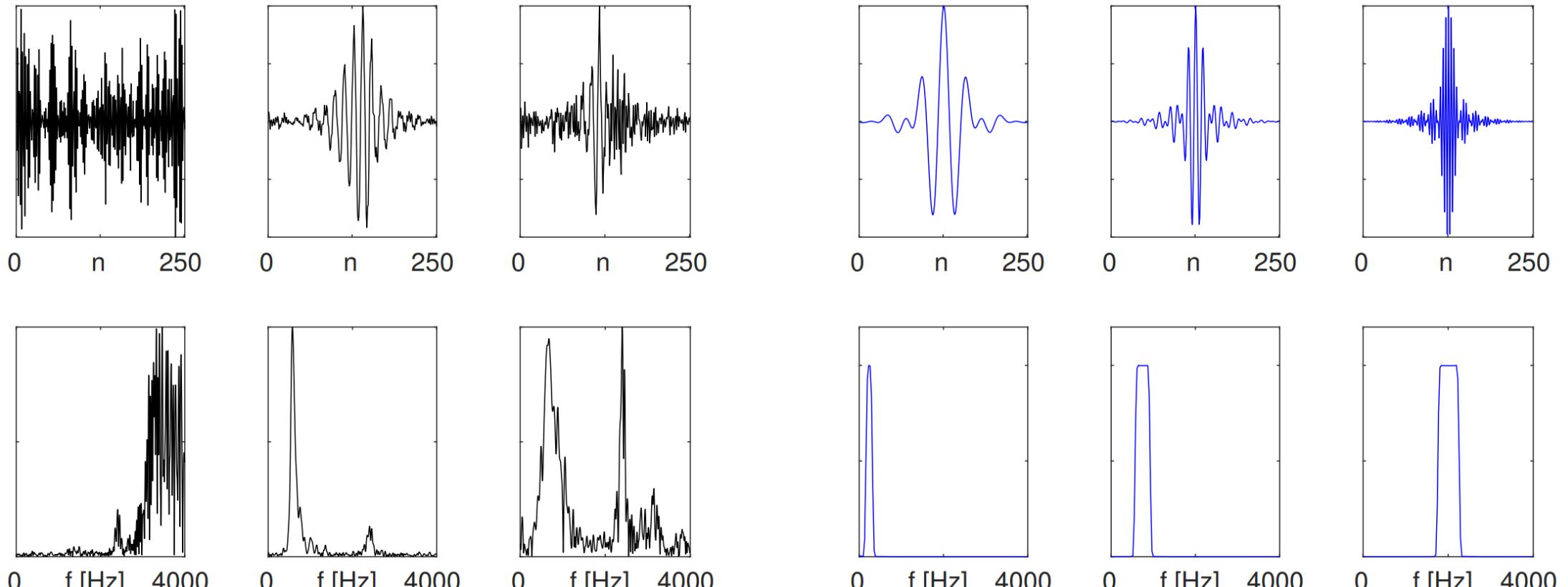


$x[n]$



$x[n] * \text{filter}[n]$

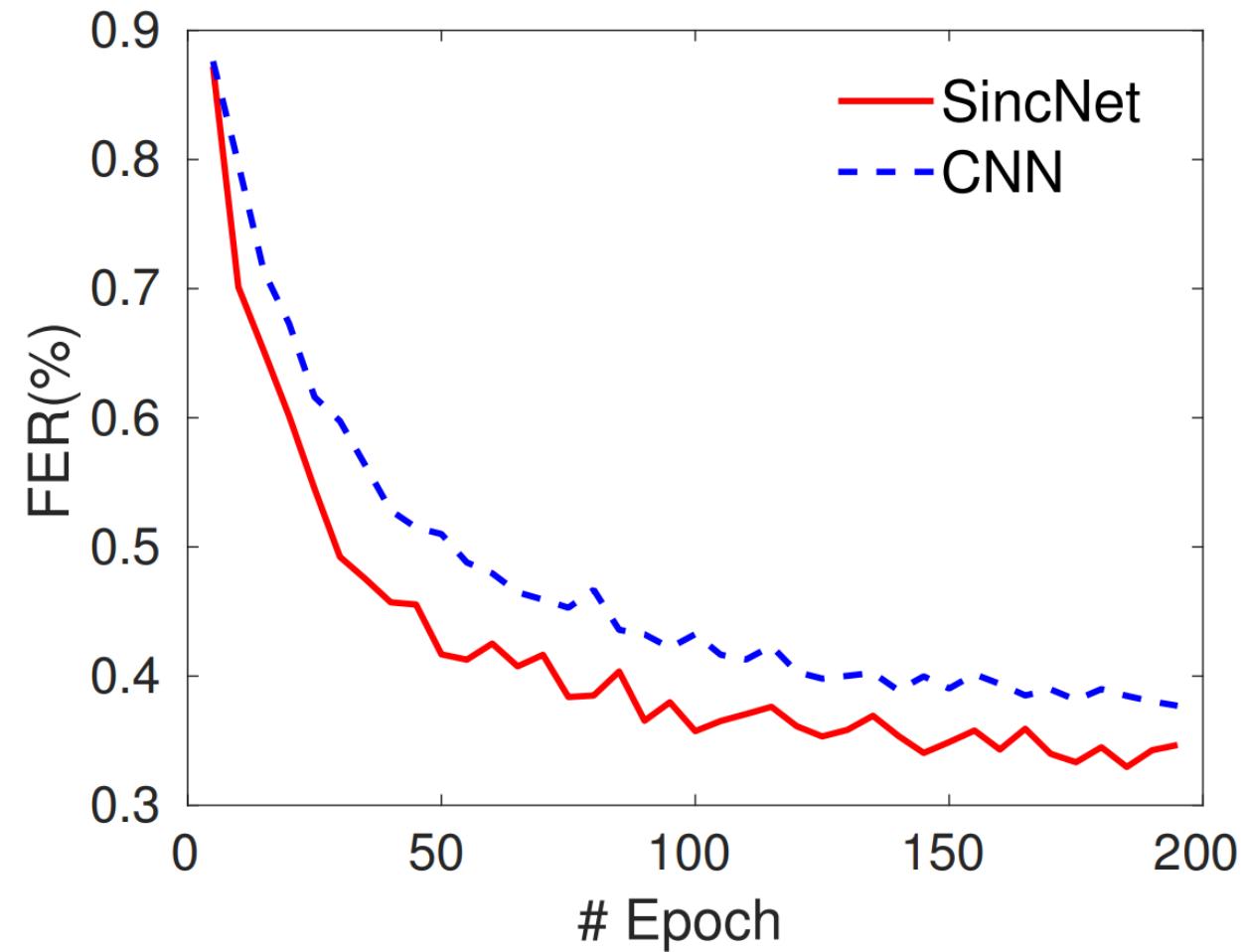
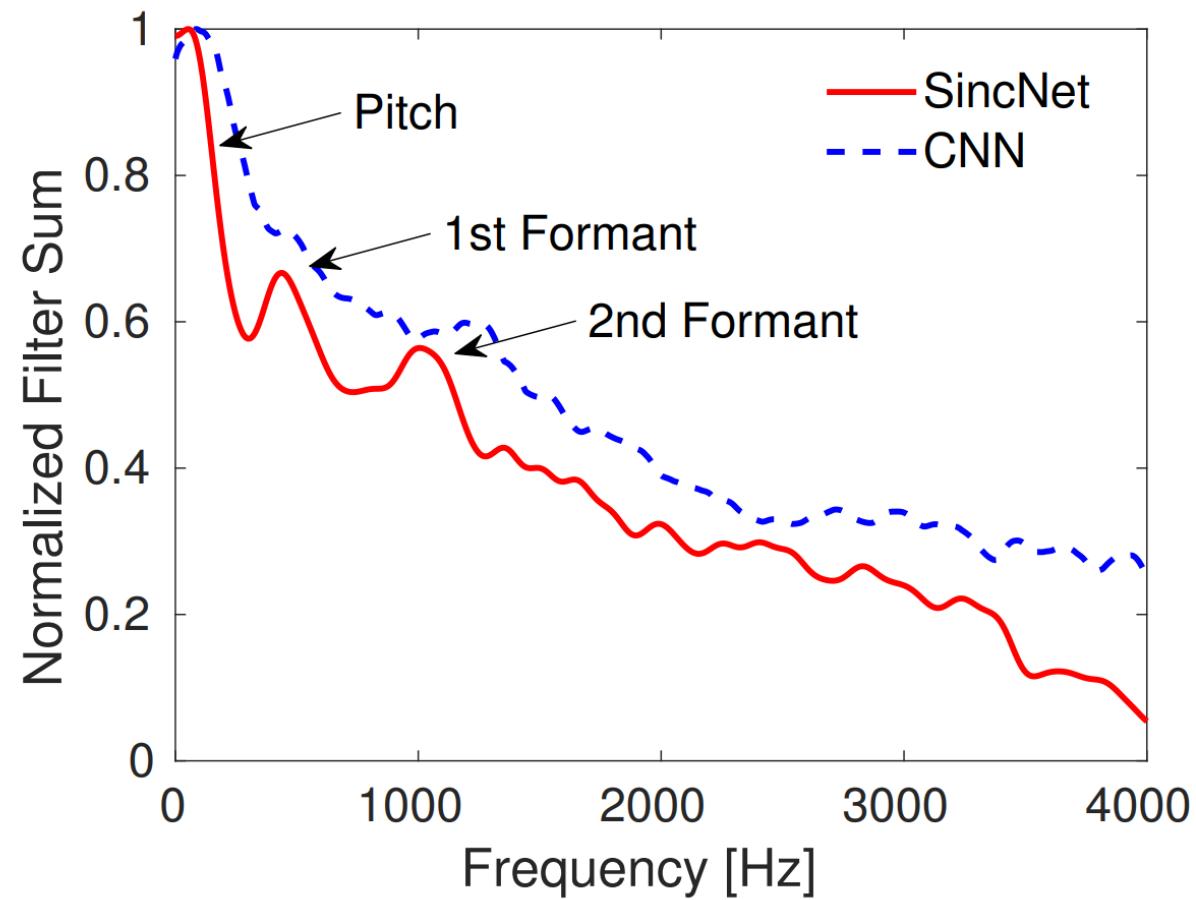
# CNN-layer vs Sinc-layer



(a) CNN Filters

(b) SincNet Filters

# CNN vs SincNet



# Преимущества Sinc-layer

- Меньше параметров  $LF$  против  $2F$  где  $F$  - кол-во фильтров
- Понятная физическая интерпретация
- Быстрее сходится и выучивает более естественные признаки