

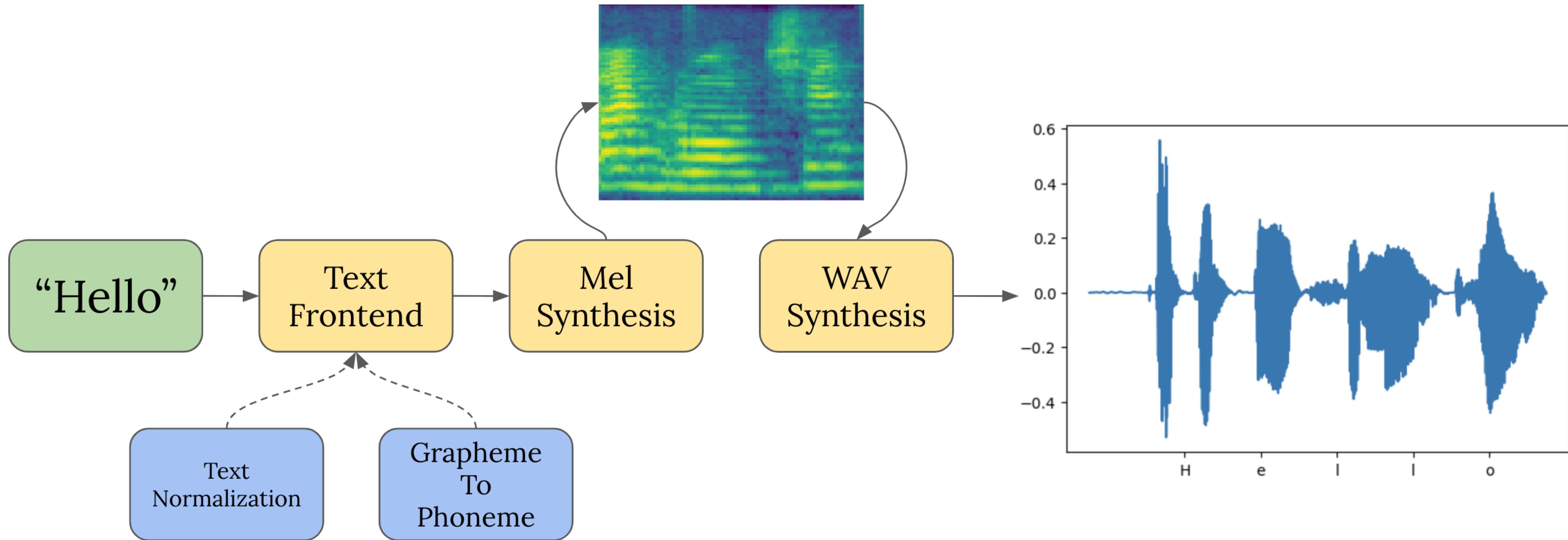
SoundStorm

Efficient Parallel Audio Generation

Ekaterina Kozlova



Text-to-speech: classical pipeline



Text-to-speech: seq2seq approach

- хотим прикрутить трансформер к TTS
- благодаря **neural codecs** можем получить некоторое дискретное представление для аудио и свести задачу к seq2seq
- чтобы при помощи токенов из neural codec генерировать качественные треки, нужно увеличить «rate of discrete representation» – то есть либо увеличить размер codebook, либо генерировать очень длинные последовательности из токенов
- сильно увеличить codebook нет возможности (не влезем по памяти) => нужно генерировать длинные последовательности токенов
- размер attention-матрицы квадратично зависит от длины последовательности:
trade-off между тем чтобы сгенерировать быстро и сгенерировать качественно

Soundstream: an end-to-end neural audio codec

(actually, EnCodec has the same architecture)

- encoder: конвертируем waveform фиксированной длины в эмбеддинг
- quantizer: прогонаем эмбеддинг через RVQ (Residual Vector Quantization), получаем некоторое сжатое представление
- decoder: берём сжатое представление, пытаемся предсказать оригинальную waveform
- discriminator: сравниваем оригинальную и восстановленную waveform-ы
- зачем нам знать про neural codecs? они служат в качестве backbone у AudioLM

Soundstream: an end-to-end neural audio codec (actually, EnCodec has almost the same architecture)

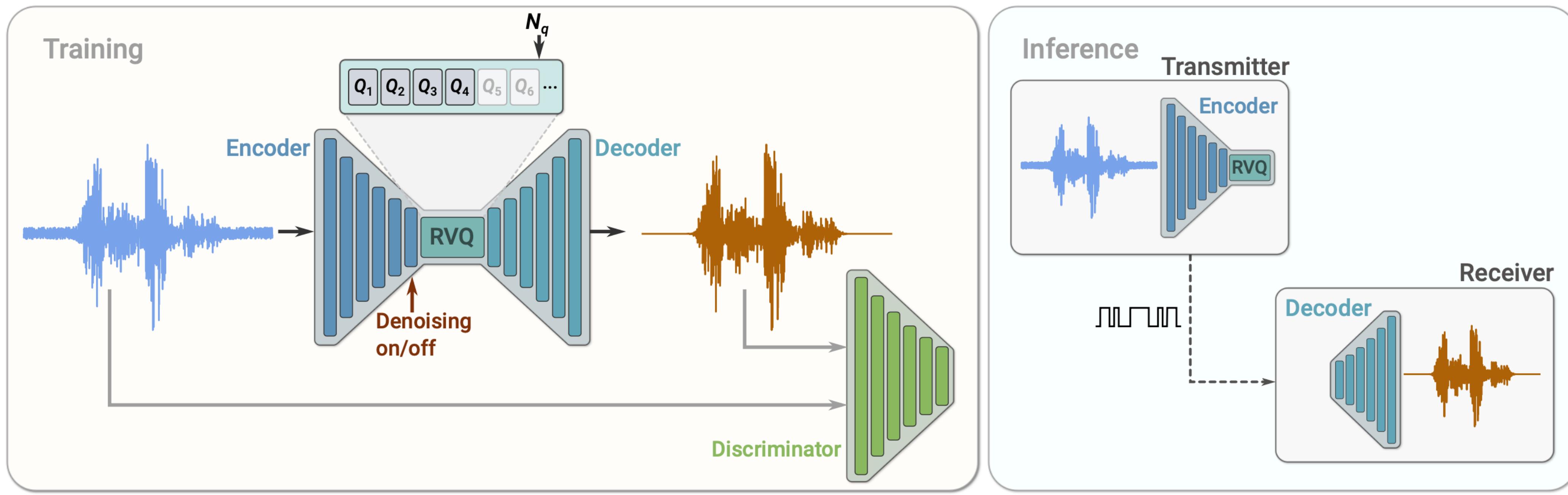
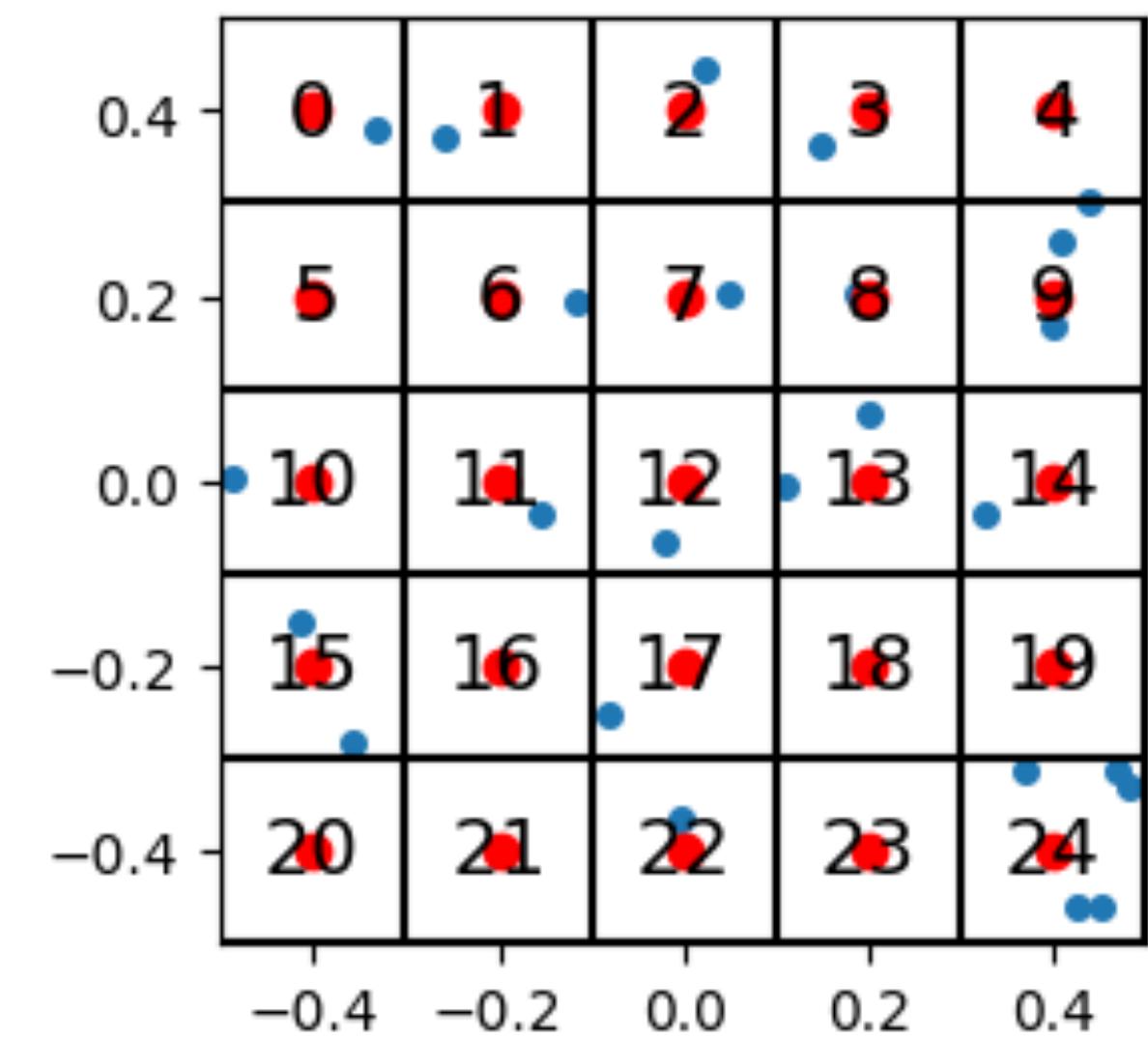


Fig. 2: *SoundStream* model architecture. A convolutional encoder produces a latent representation of the input audio samples, which is quantized using a variable number n_q of residual vector quantizers (RVQ). During training, the model parameters are optimized using a combination of reconstruction and adversarial losses. An optional conditioning input can be used to indicate whether background noise has to be removed from the audio. When deploying the model, the encoder and quantizer on a transmitter client send the compressed bitstream to a receiver client that can then decode the audio signal.

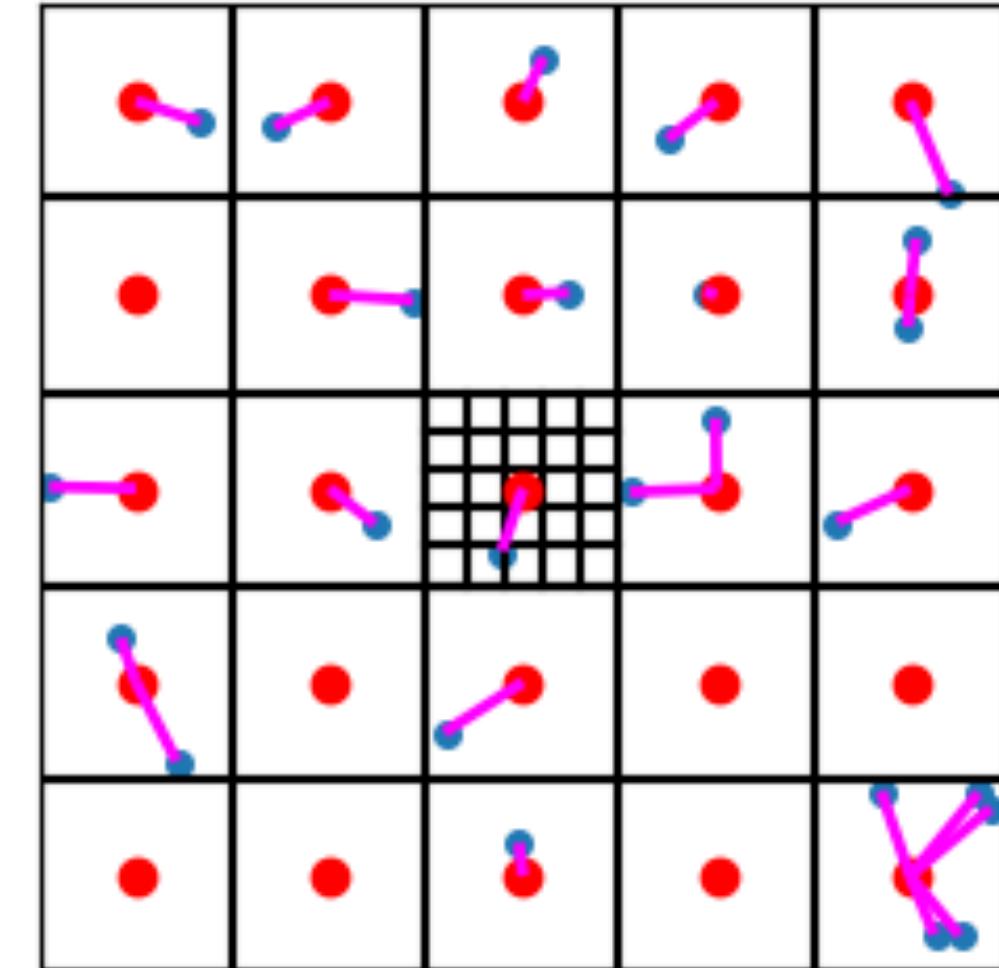
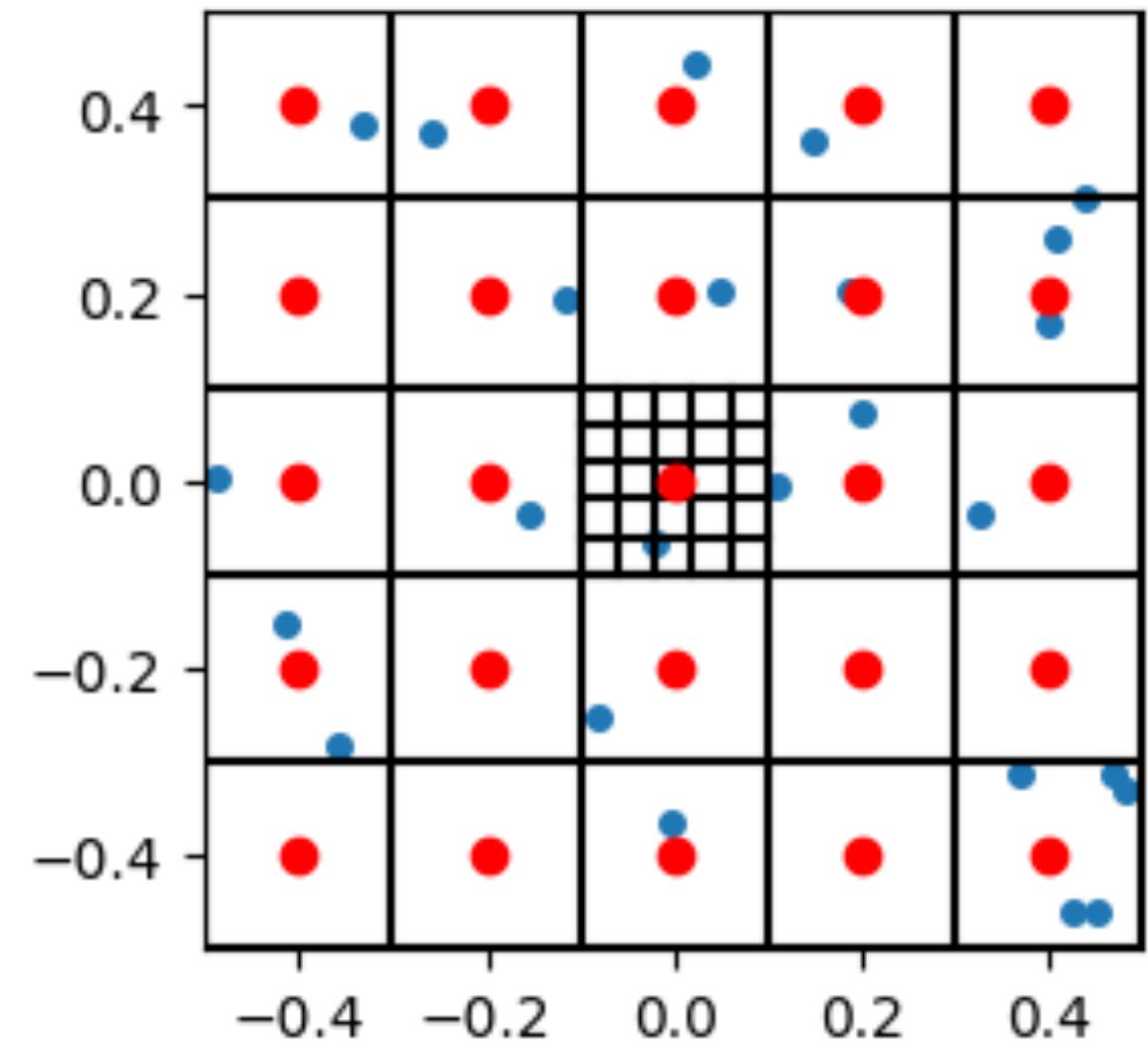
Vector Quantization (пока не residual)

- vector quantization – это процесс разбиения пространства точек на некоторое число регионов
- точку, которая попала в некоторый регион, мы будем представлять центроидом этого региона
- codebook отображает номер региона в координаты центроида
- чем меньше размер региона, тем меньше будет reconstruction loss, но тем больше будет становиться размер codebook



Residual Vector Quantization

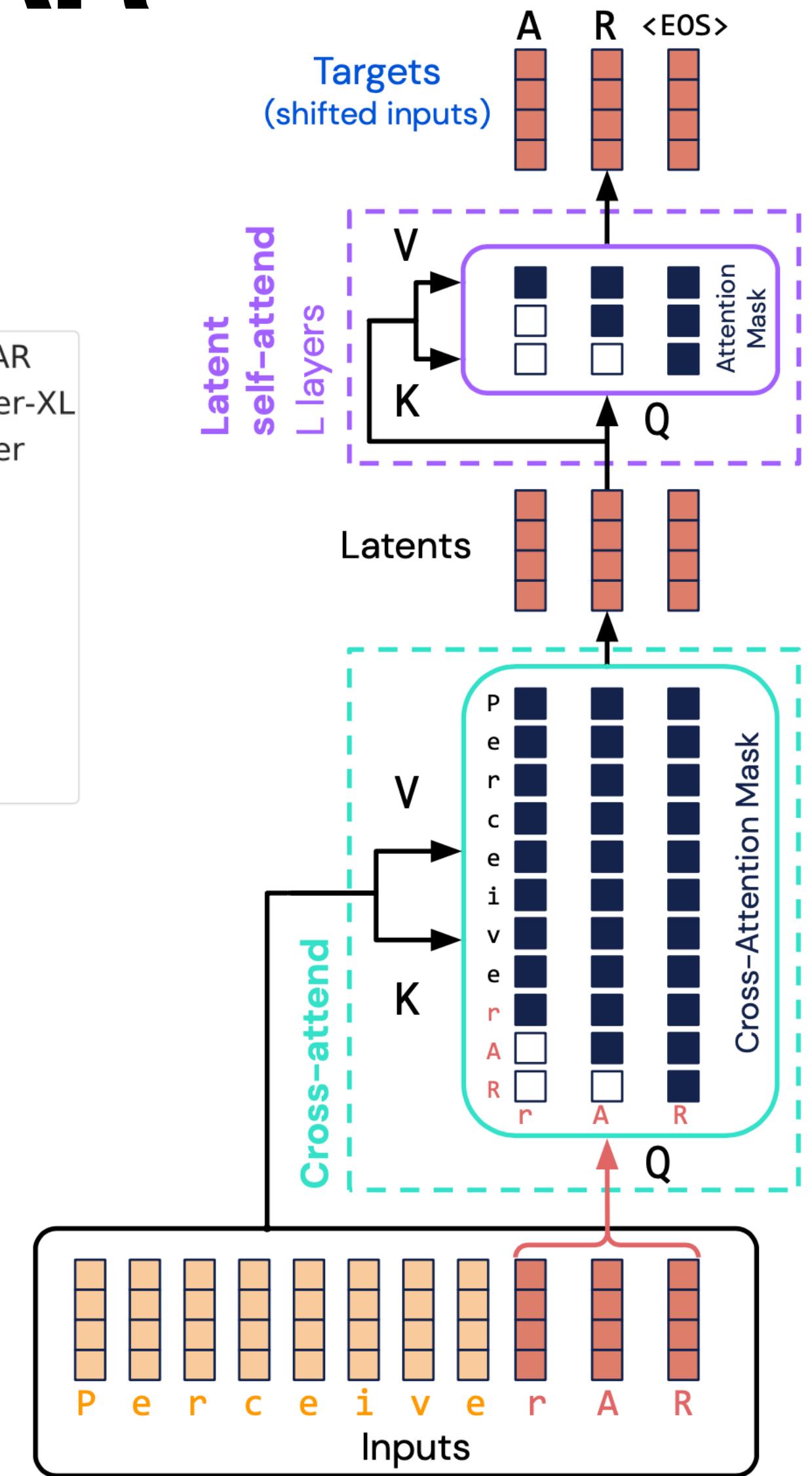
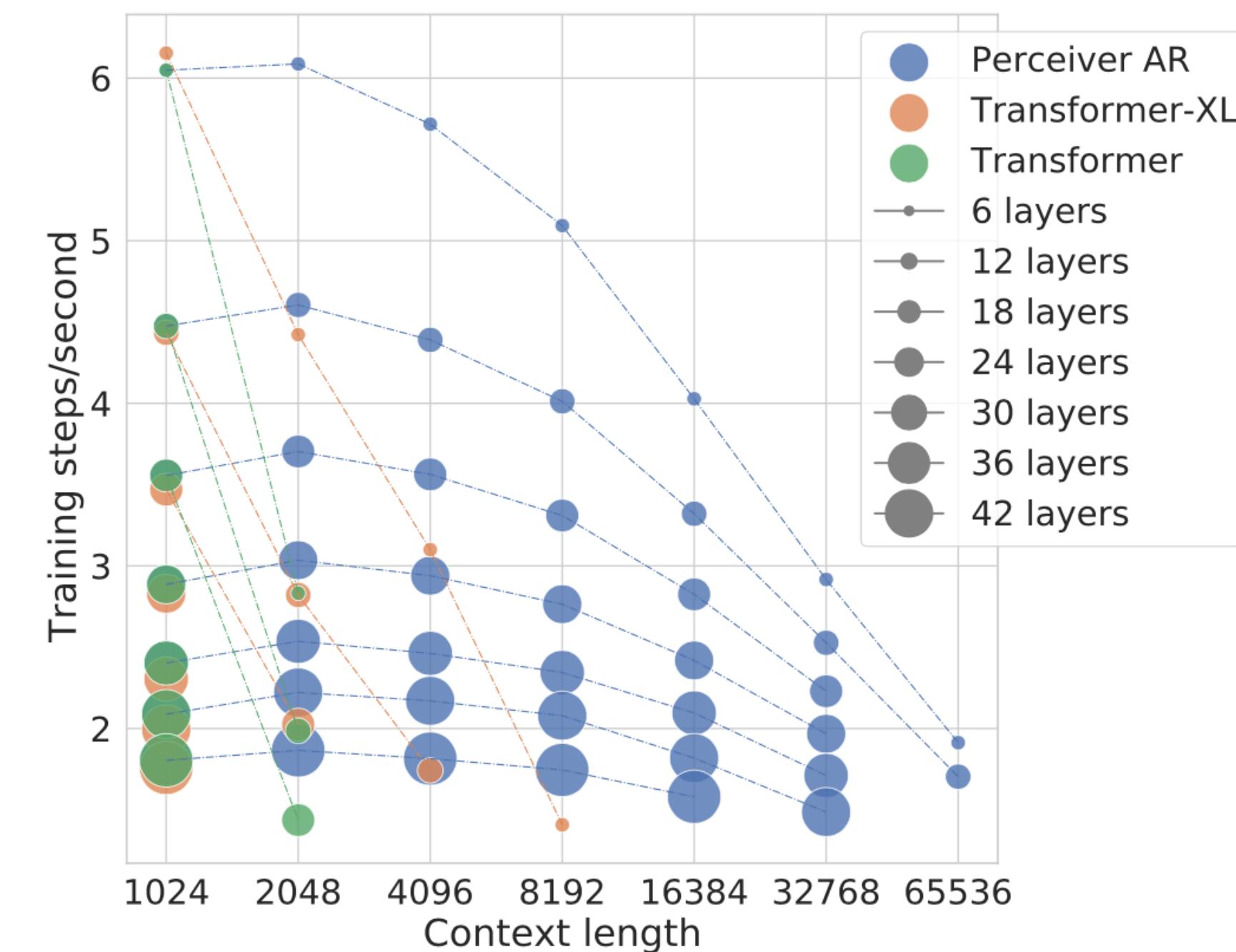
- «codebooks in codebooks» вместо одного codebook с большим разрешением
- residual: разность координат между исходной точкой и центроидом её региона
- residual тоже квантизуем (делаем разбивку на регионы и codebook, но уже для residual-ов)
- **residual codebook одинаковый для всех регионов**
- если бы мы использовали один большой codebook 25×25 , сложность вычислений была бы $(5^5) * (5^5) = 625$, здесь же получим $2 * (5^5) = 50$



Hacks with attention: Perceiver AR

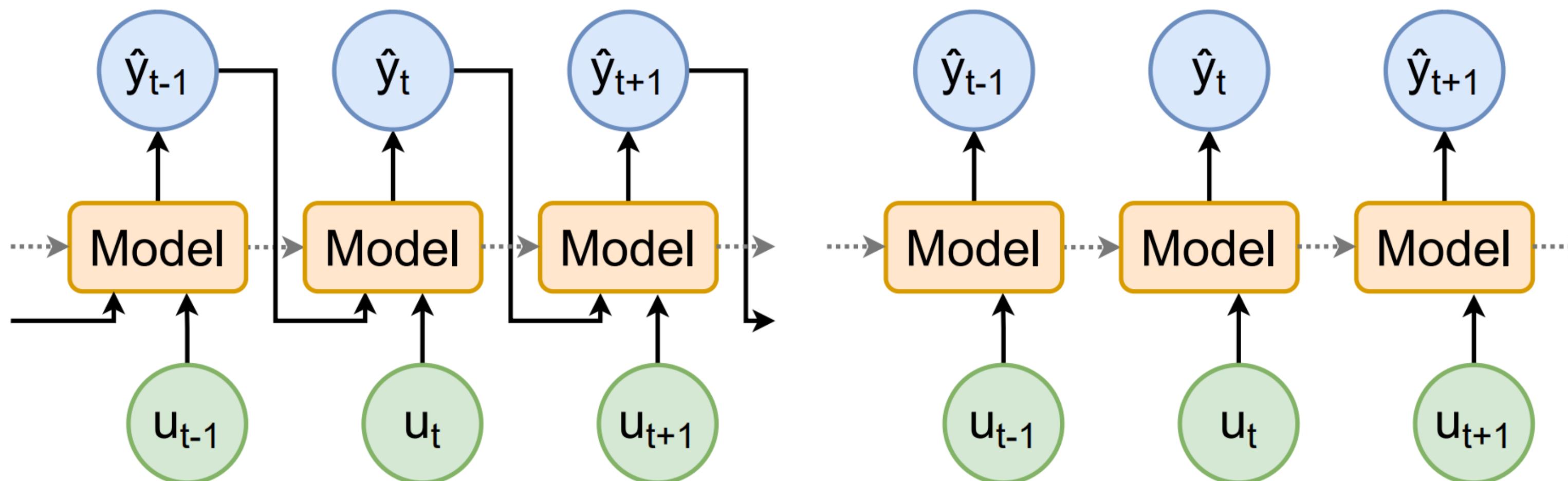
AR: AutoRegressive

- решаем проблему квадратичной сложности
- авторегрессивная modality-agnostic архитектура, основанная на cross attention
- через cross attention переводим input array в latent array, дальше все операции с attention проводим уже в латентном пространстве



Non-Autoregressive models

- autoregressive: output_t зависит от $\text{output}_{\{t - 1\}}$
- non-autoregressive быстрее, потому что можно распараллелить вычисления

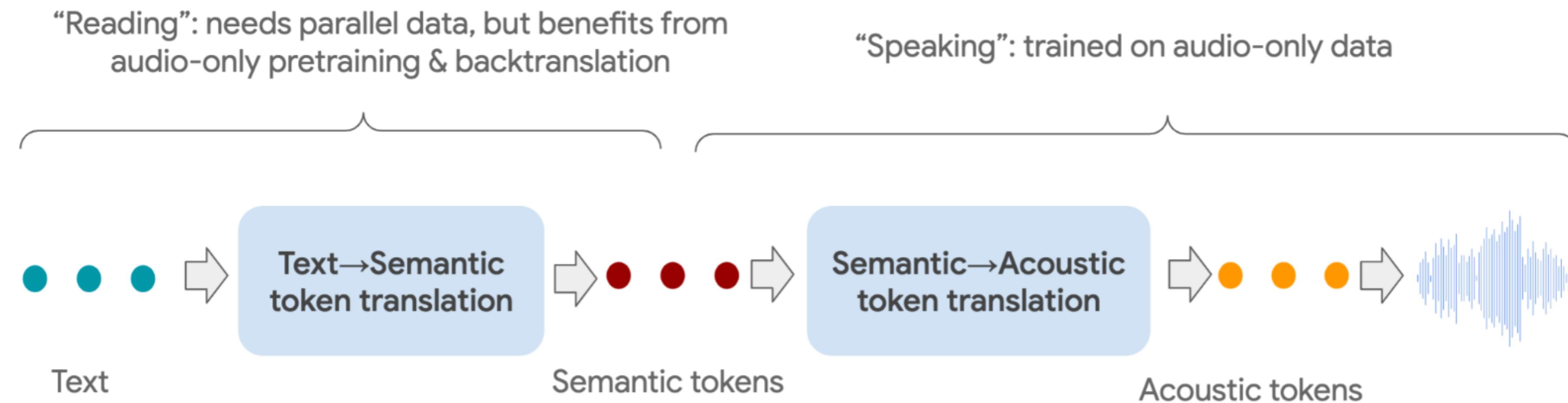


(a) Autoregressive model

(b) Non-autoregressive model

Spear TTS

yet another paper by Zalán Borsos



- two-stage модель, на обоих этапах решается задача seq2seq
- acoustic tokens превращаются в waveform при помощи декодера SoundStream
- спойлер: главная часть — text-to-semantic-token

AudioLM

- semantic tokens нужны, чтобы сохранить long-term структуру
- acoustic tokens из neural codec нужны для качественного синтеза
- решается проблема trade-off, о которой говорили в начале

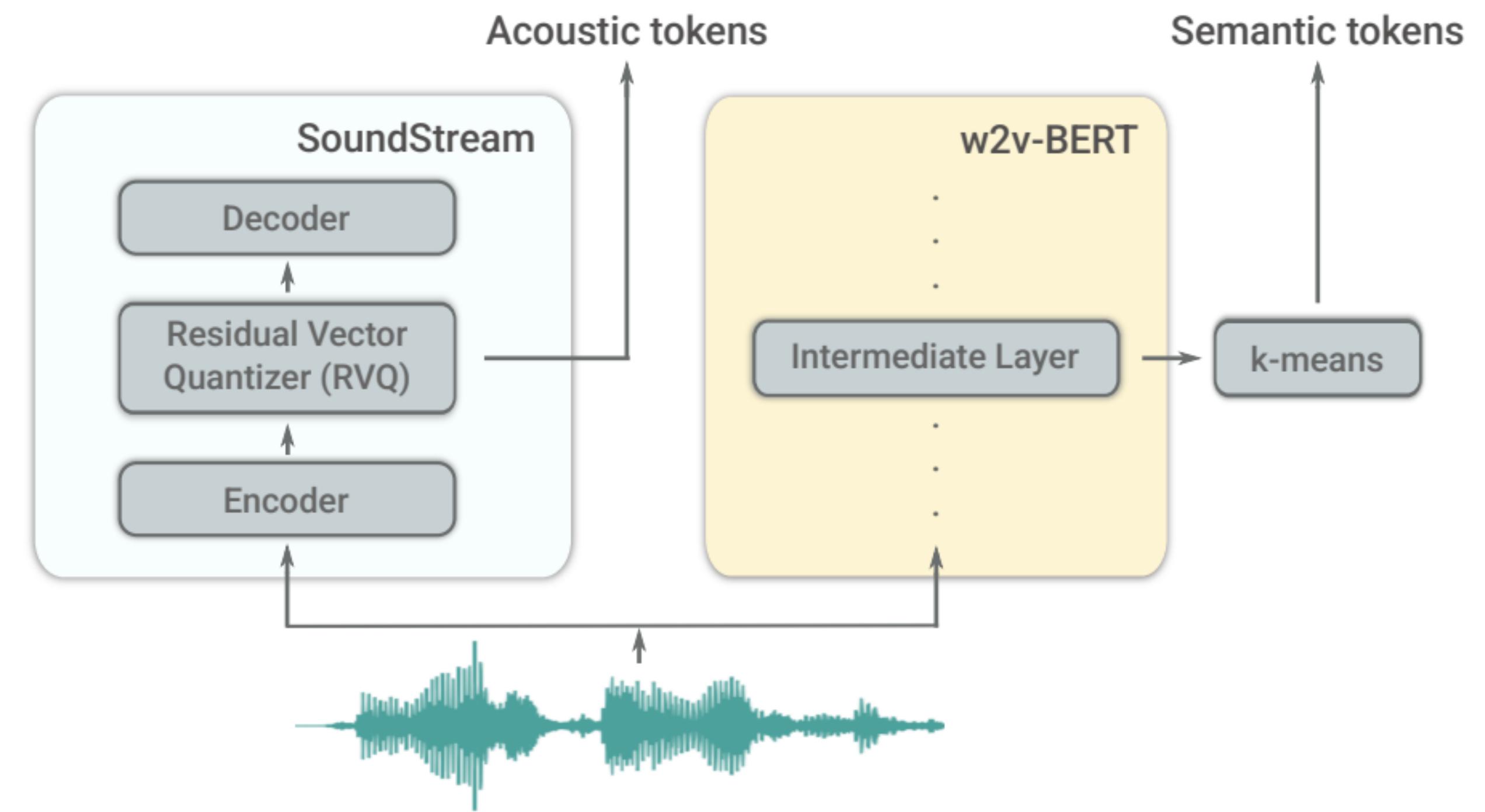


Fig. 1. Overview of the tokenizers used in AudioLM. The acoustic tokens are produced by SoundStream [16] and enable high-quality audio synthesis. The semantic tokens are derived from representations produced by an intermediate layer of w2v-BERT [17] and enable long-term structural coherence.

AudioLM

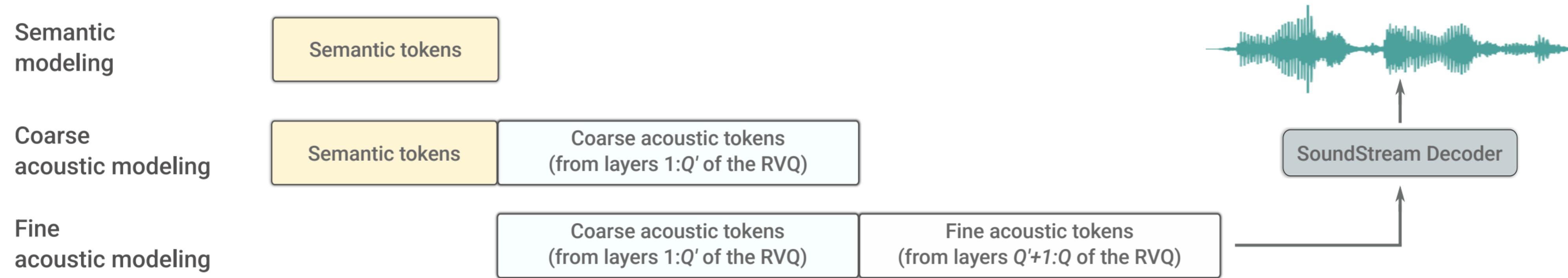


Fig. 2. The three stages of the hierarchical modeling of semantic and acoustic tokens in AudioLM: i) semantic modeling for long-term structural coherence, ii) coarse acoustic modeling conditioned on the semantic tokens and iii) fine acoustic modeling. With the default configuration, for every semantic token there are $2Q'$ acoustic tokens in the second stage and $2(Q - Q')$ tokens in the third stage. The factor of 2 comes from the fact that the sampling rate of SoundStream embeddings is twice as that of the w2v-BERT embeddings.

- три этапа, на каждом — свой decoder-only трансформер для предсказания следующего токена (*semantic tokens* обуславливают предсказание *acoustic tokens*)
- у *coarse acoustic tokens* иерархическая структура (вспоминаем RVQ)
- на третьем шаге избавляемся от артефактов, возникших при генерации, и в целом улучшаем качество

SoundStorm

- SoundStorm может служить как acoustic generator (то есть этапы 2 и 3) у AudioLM
- при этом в AudioLM генератор был autoregressive, а SoundStorm non-autoregressive => выигрываем по скорости
- SoundStorm, объединённый со Spear-TTS (частью про text-to-semantic token) может синтезировать диалоги по транскрипту, сохраняя особенности голосов у разных спикеров

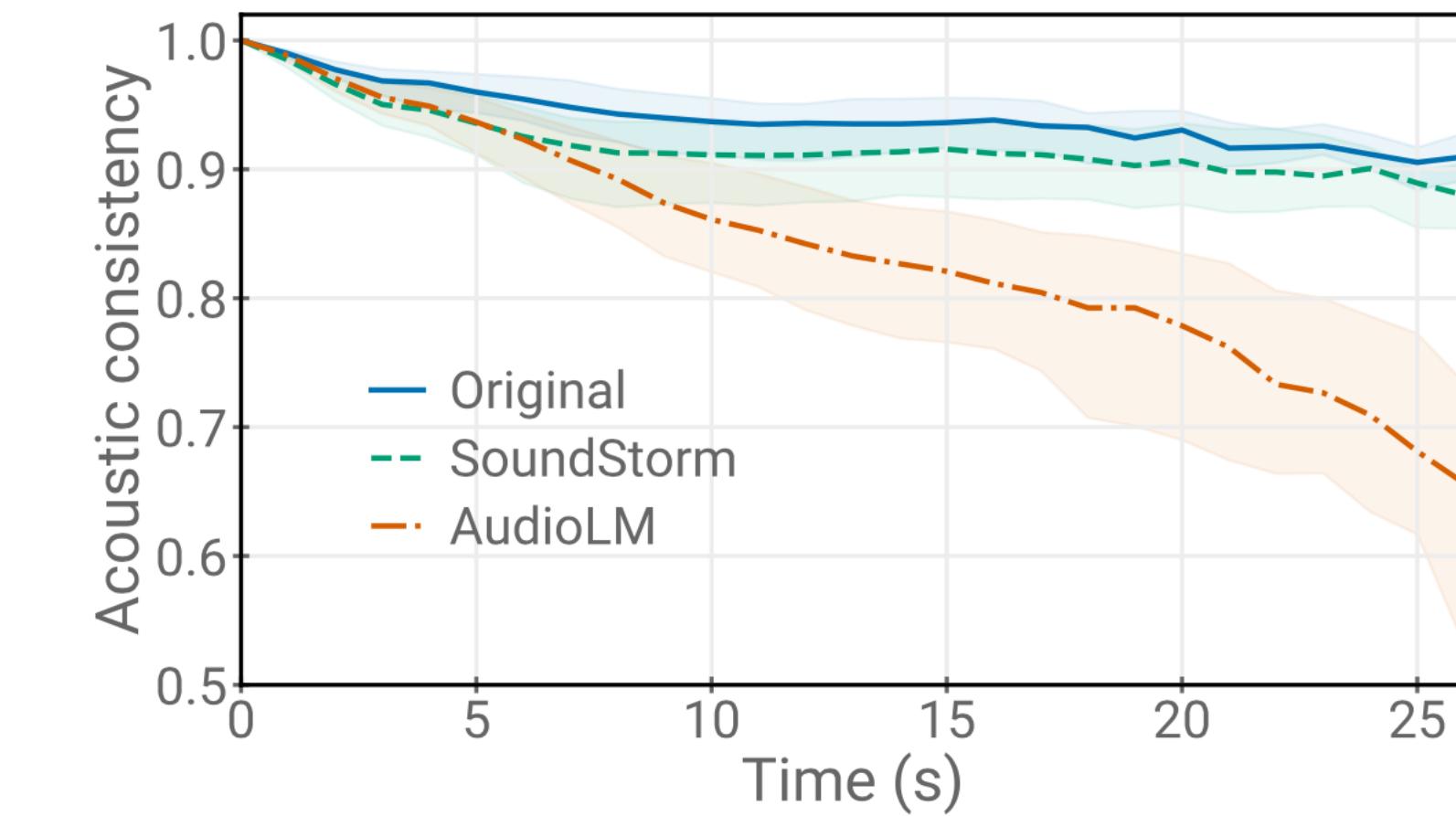


Figure 2. Acoustic consistency between the prompt and the generated audio for the samples in the ‘long’ split of LibriSpeech test-clean. The shaded area represents the inter-quartile range.

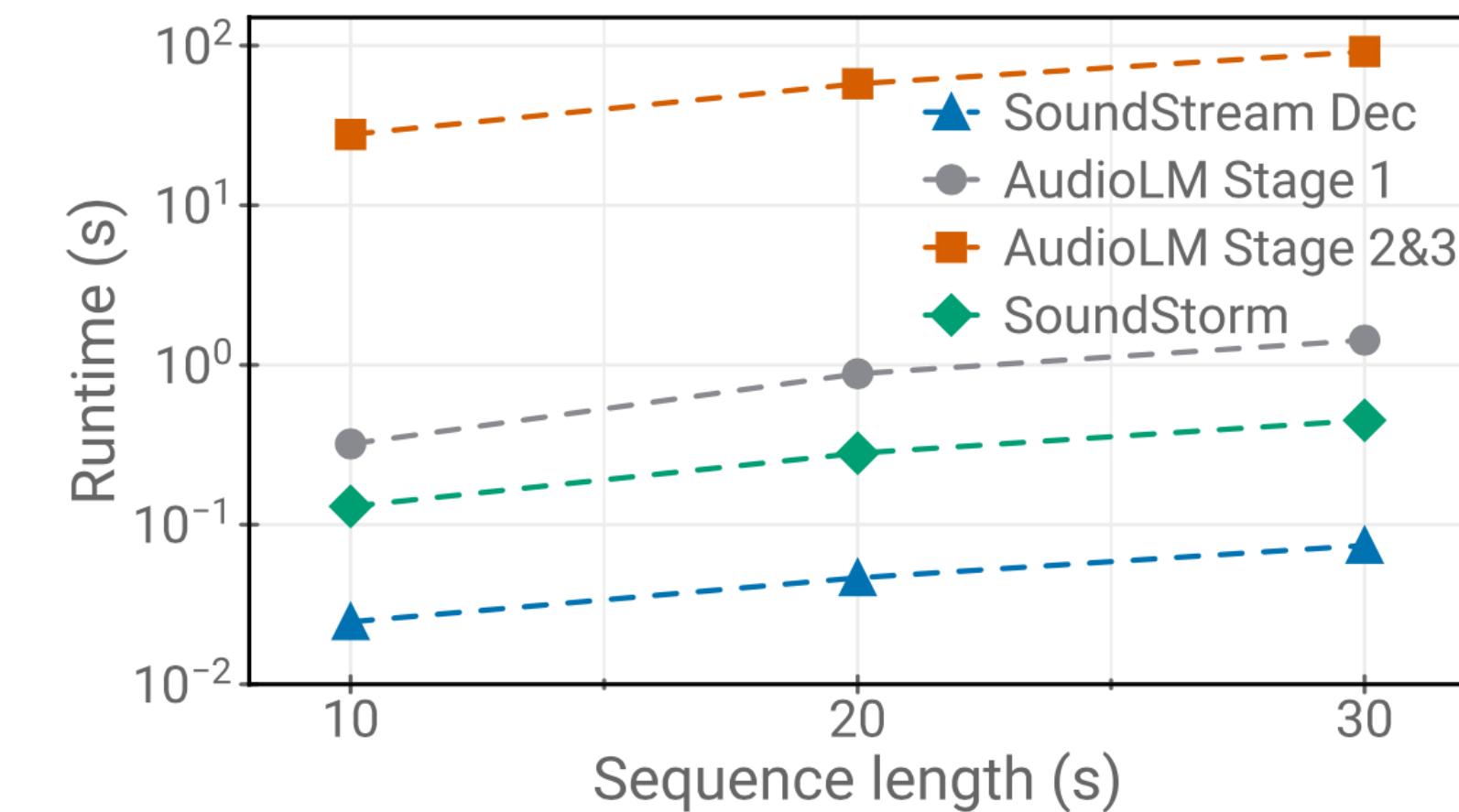
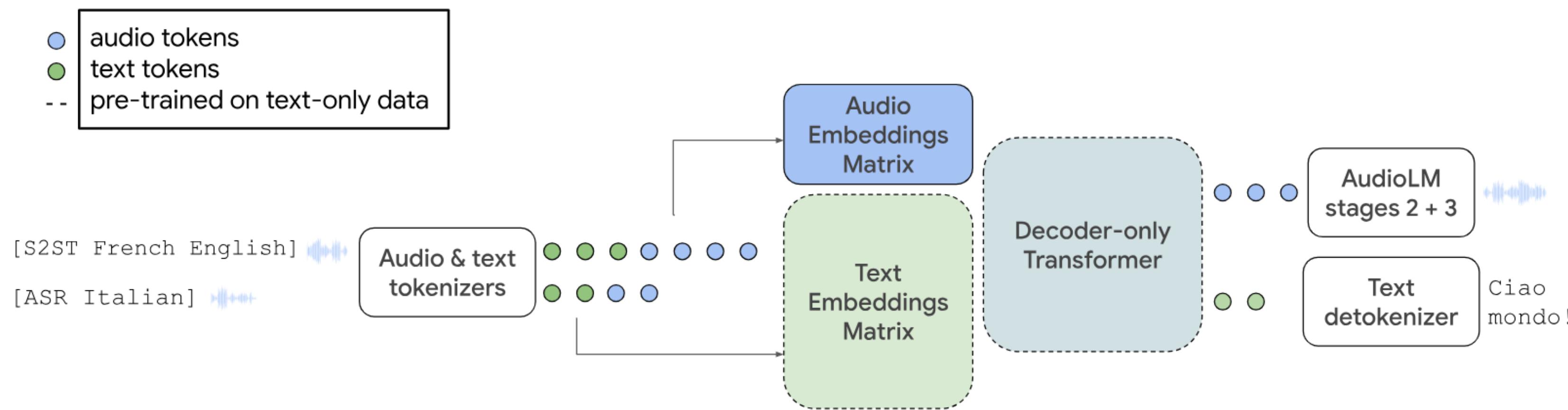


Figure 3. Runtimes of SoundStream decoding, SoundStorm and different stages of AudioLM on a TPU-v4.

Appendix: AudioPaLM

yet another paper by our beloved Zalán Borsos (as a co-author)

- A Large Language Model That Can Speak and Listen (ASR + speech2speech translation)



- смесь PaLM-2 и AudioLM, заставляем предобученную text-only модель (пунктирная линия) расширить свою матрицу эмбеддингов на аудио-токены
- остальной пайплайн такой же: получаем звук из токенов через этапы 2-3 AudioLM

<https://google-research.github.io/seanet/audiopalm/examples/>

Zalán Borsos @ Google Research



Zalán Borsos

<https://zalanborsos.com/>

Selected Publications

SoundStorm: Efficient Parallel Audio Generation

Zalán Borsos, Matt Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, Marco Tagliasacchi
arXiv:2305.09636, 2023
[\[paper\]](#) [\[blog post\]](#)

Speak, Read and Prompt: High-Fidelity Text-to-Speech with Minimal Supervision

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, Neil Zeghidour
arXiv:2302.03540, 2023
[\[paper\]](#)

MusicLM: Generating Music From Text

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, Christian Frank
arXiv:2301.11325, 2023
[\[paper\]](#)

AudioLM: a Language Modeling Approach to Audio Generation

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, Neil Zeghidour
TASLP, 2023
[\[paper\]](#) [\[blog post\]](#)

NO BENCHMARKS?

Benchmarks

Text-To-Speech Synthesis on LJSpeech

Leaderboard

Dataset

View by for

