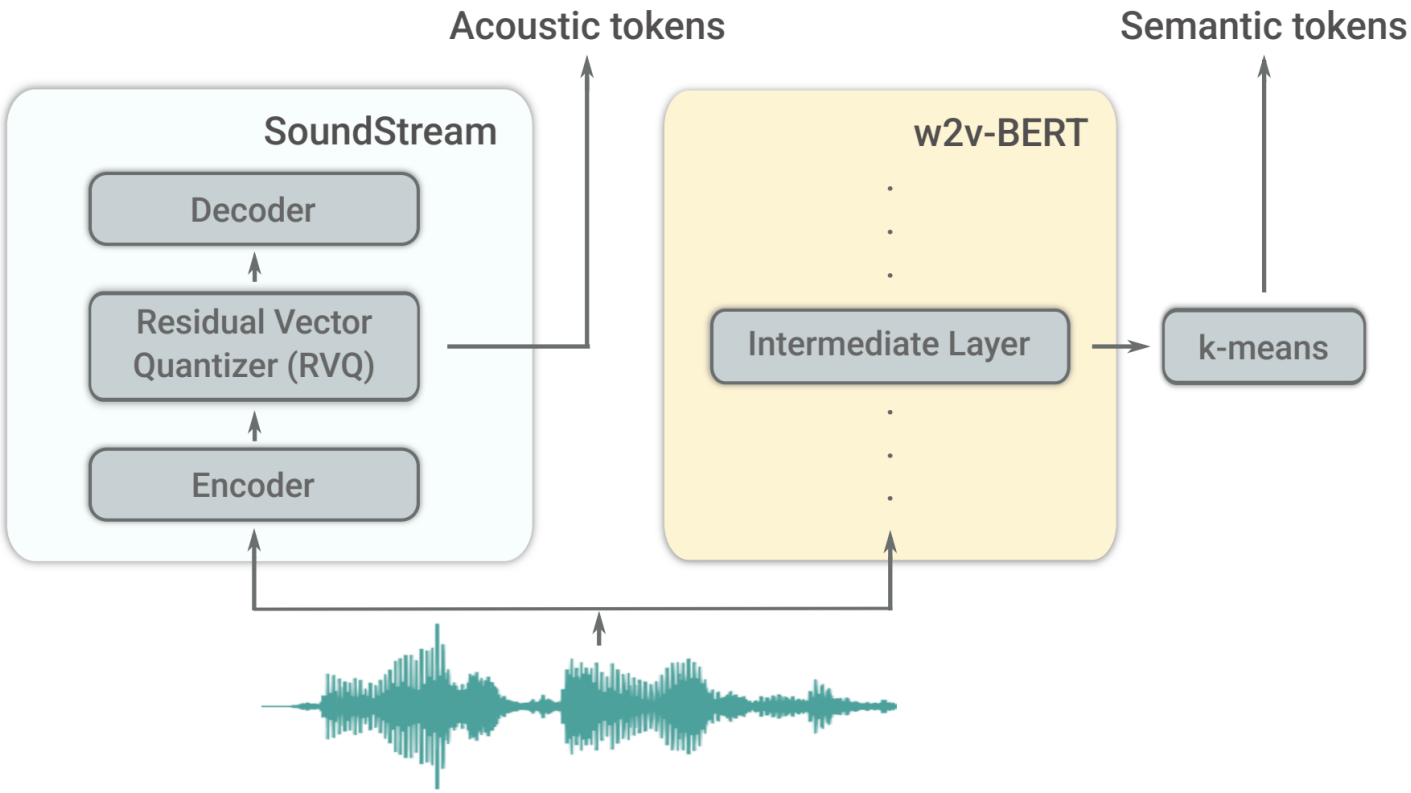
SoundStorm: Efficient Parallel Audio Generation

Бекян Артём

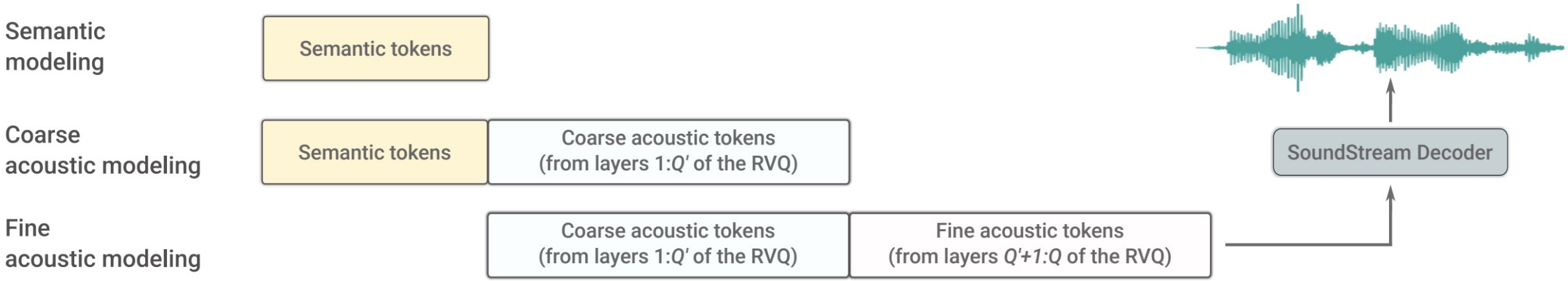
Предисловие. AudioLM и его токены

- Semantic tokens – хорошо захватывают данные о глобальной структуре, но плохо о локальной
- Acoustic tokens – захватывают акустические локальные детали аудио, чтобы генерация была более высокого качества



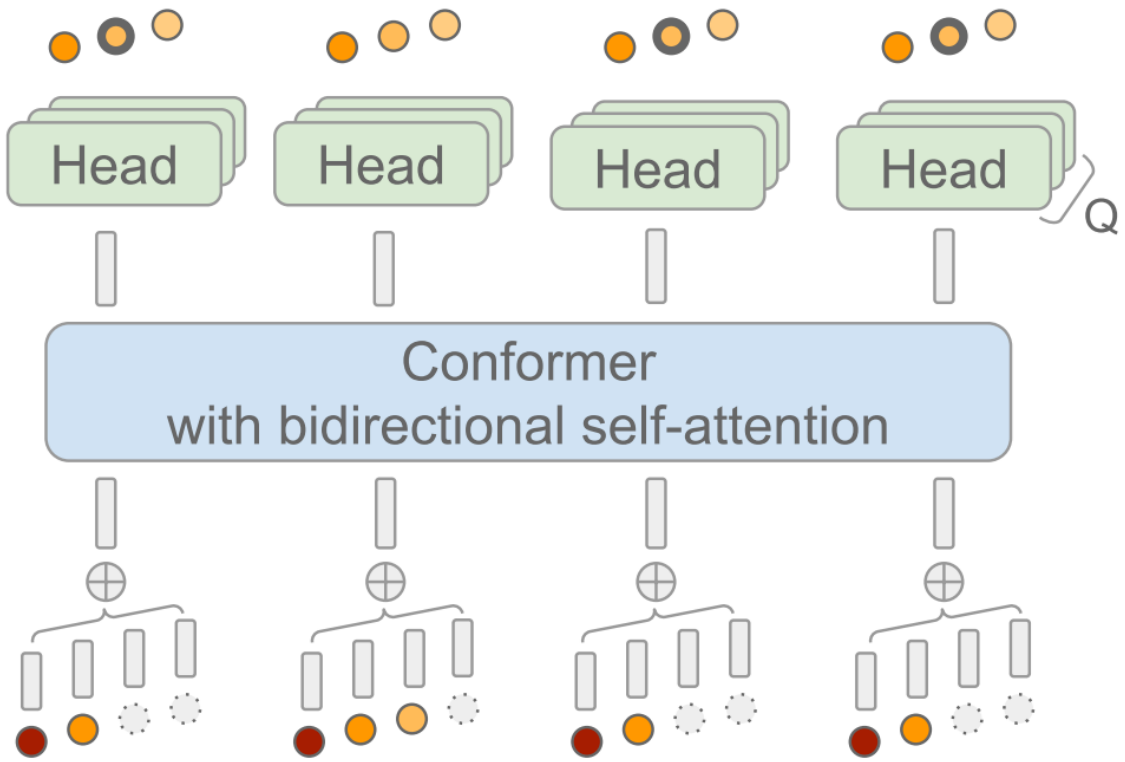
Предисловие. А куда SoundStorm?

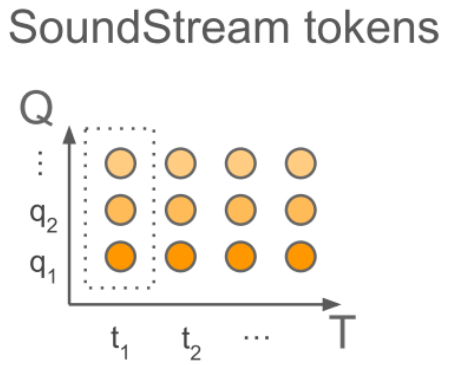
SoundStorm предлагается использовать вместо *Coarse acoustic model* и *Fine acoustic model*

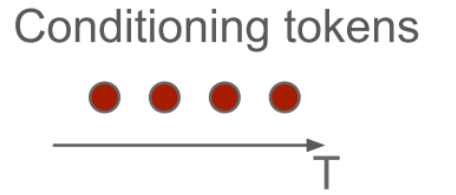


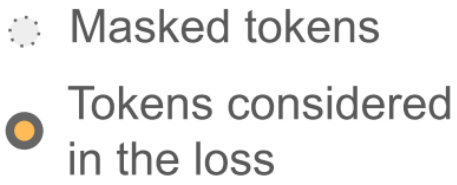
Архитектура

Предсказываются RVQ токены, выбирается q, все токены q+1 … Q маскируются, некоторые токены q тоже маскируются, чтобы по ним считать лосс









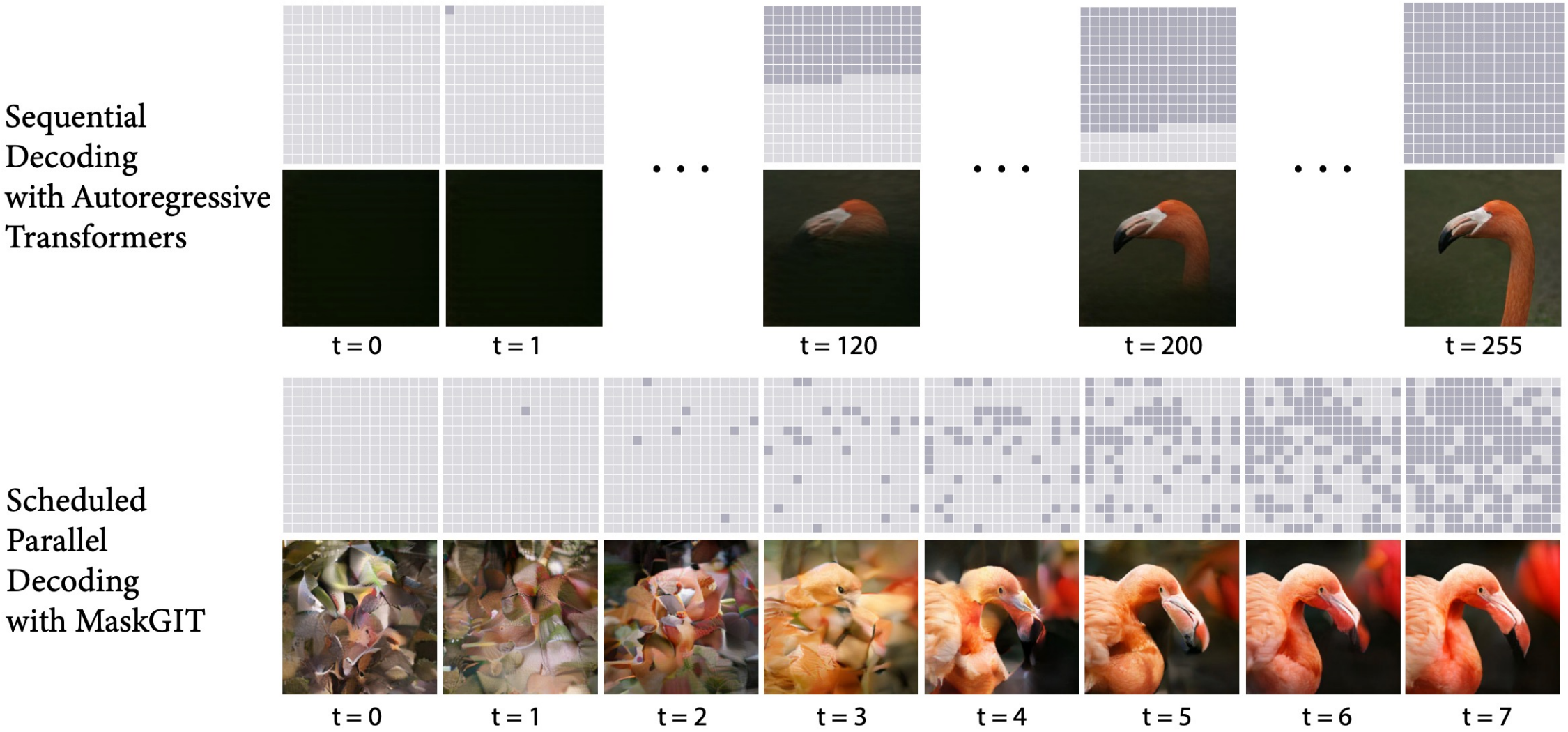
На картинке Q=3, q=2

Архитектура. Декодинг в MaskGIT

Изначально все токены маскированы. Алгоритм декодирования на итерации t:

1. Для каждого маскированного токена предсказываем вероятность $p^{(t)} \in \mathbb{R}^{N \times K}$ параллельно.
2. Семплируем токен, основываясь на вероятностях, а величину $p_i^{(t)}$ считаем уверенностью предсказания
3. В соответствии с функцией γ (в статье это косинус) маскируем обратно $n = \left\lceil \gamma\left(\frac{t}{T}\right) N \right\rceil$, где T – общее количество итераций, при помощи сортировки уверенностей по убыванию

Архитектура. Декодинг в MaskGIT



Архитектура. Маскирование

Для обучения:

1. (Если хотим использовать промпты) Равновероятно семплируем таймстеп для промпта от 0 до T-1
2. Семплируем q (уровень RVQ) равновероятно от 1 до Q
3. Семплируем маску по косинусному расписанию, то есть $u \sim \mathcal{U}[0, \pi/2], \quad p = \cos(u), \quad M_i = Bernoulli(p)$
4. Маскируем не промптовые токены, которые выбрали или которые на >q уровнях RVQ

Архитектура. Параллельный декодинг

По сути такой же, как в MaskGIT, только использующийся для каждого уровня RVQ с 1 до Q (coarse-to-fine)

Отличие: при последнем проходе на каждом уровне RVQ используется жадное декодирование вместо confidence-based семплинга

Интересный момент: поскольку чем дальше в RVQ, тем локальнее становятся токены, то со временем T уменьшается

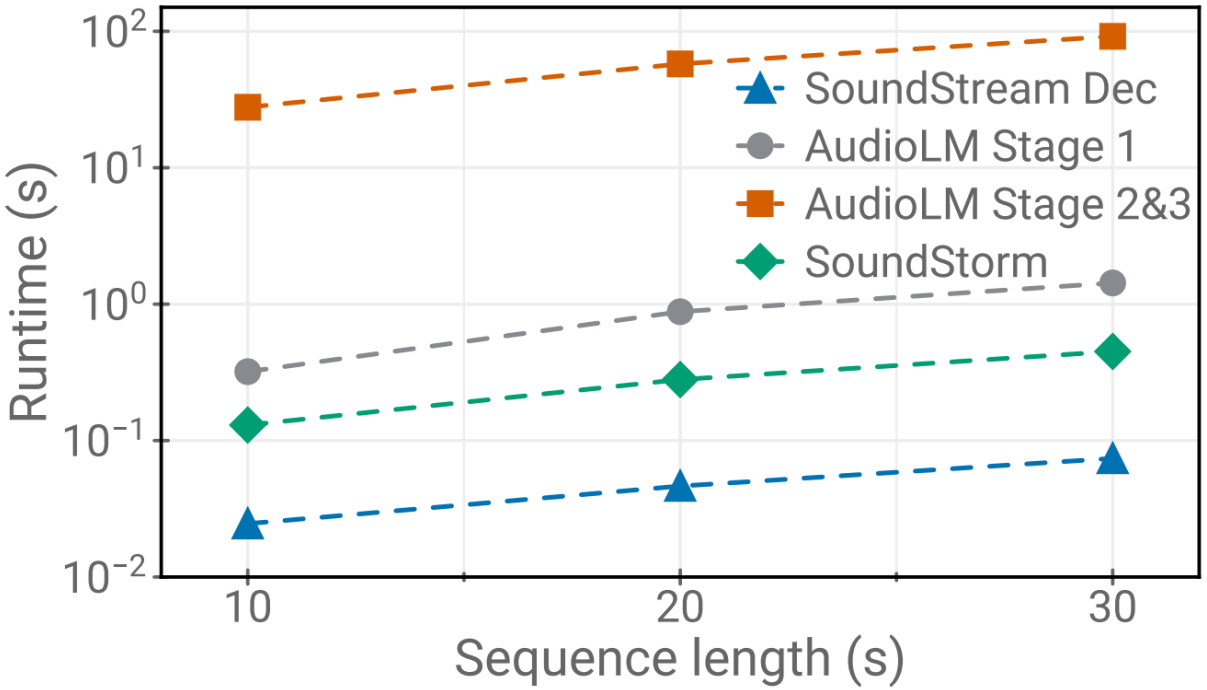
Результаты

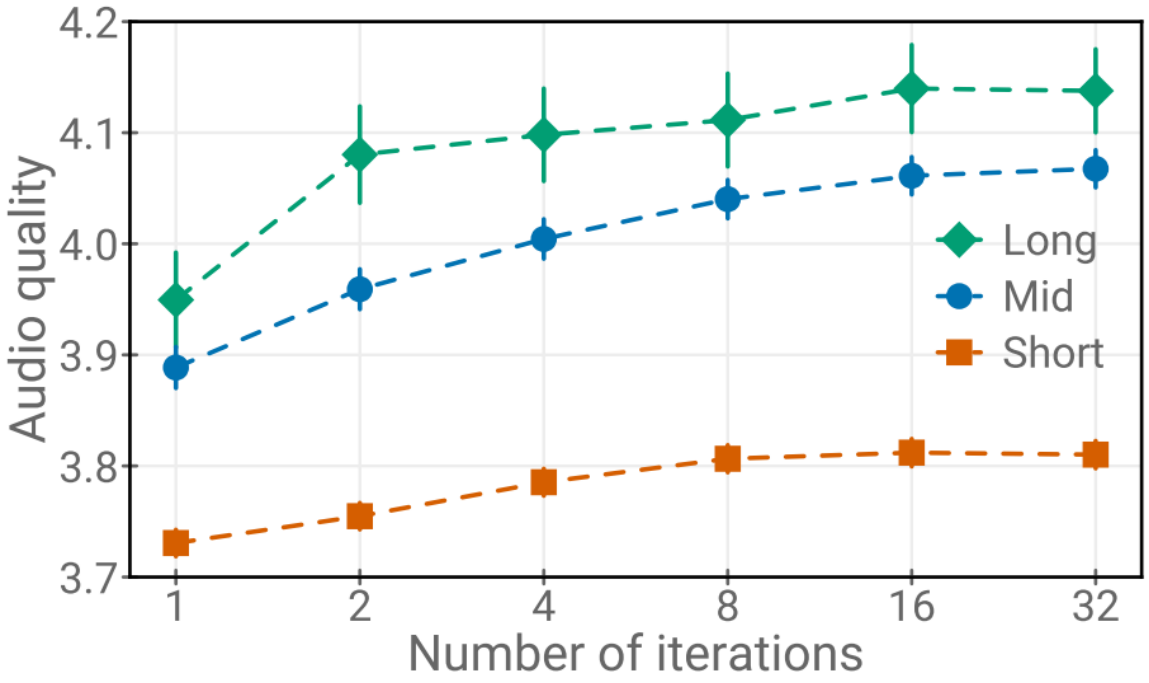
Рассчитаны с помощью ASR модели | MOS | Одинаковость акустических свойств

	WER↓			CER↓			Audio quality↑			Voice preservation↑			Acoustic consistency↑		
	short	mid	long	short	mid	long	short	mid	long	short	mid	long	short	mid	long
Original SoundStream rec.	2.62	1.95	2.20	0.89	0.55	0.69	3.72	3.91	3.99	0.63	0.65	0.66	0.97	0.95	0.93
Without a speaker prompt															
AudioLM	4.65	3.59	4.79	2.15	1.57	2.30	3.93	4.04	4.08	–	–	–	–	–	–
SoundStorm	**3.48**	**2.55**	**3.33**	**1.39**	**0.89**	**1.29**	**4.01**	**4.16**	**4.20**	–	–	–	–	–	–
With a speaker prompt															
AudioLM	3.77	3.40	3.75	1.50	1.47	1.54	**3.91**	**4.06**	4.10	0.46	0.48	0.48	**0.96**	0.91	0.86
SoundStorm	**2.99**	**2.43**	**3.36**	**1.10**	**0.81**	**1.24**	3.81	**4.05**	**4.15**	**0.57**	**0.59**	**0.59**	**0.96**	**0.94**	**0.91**

Сохранение личности спикера
Рассчитано с помощью ASV модели

Парочка аблейшнов





Только для 1 уровня RVQ, остальные декодируются жадным алгоритмом, так лучше выходит