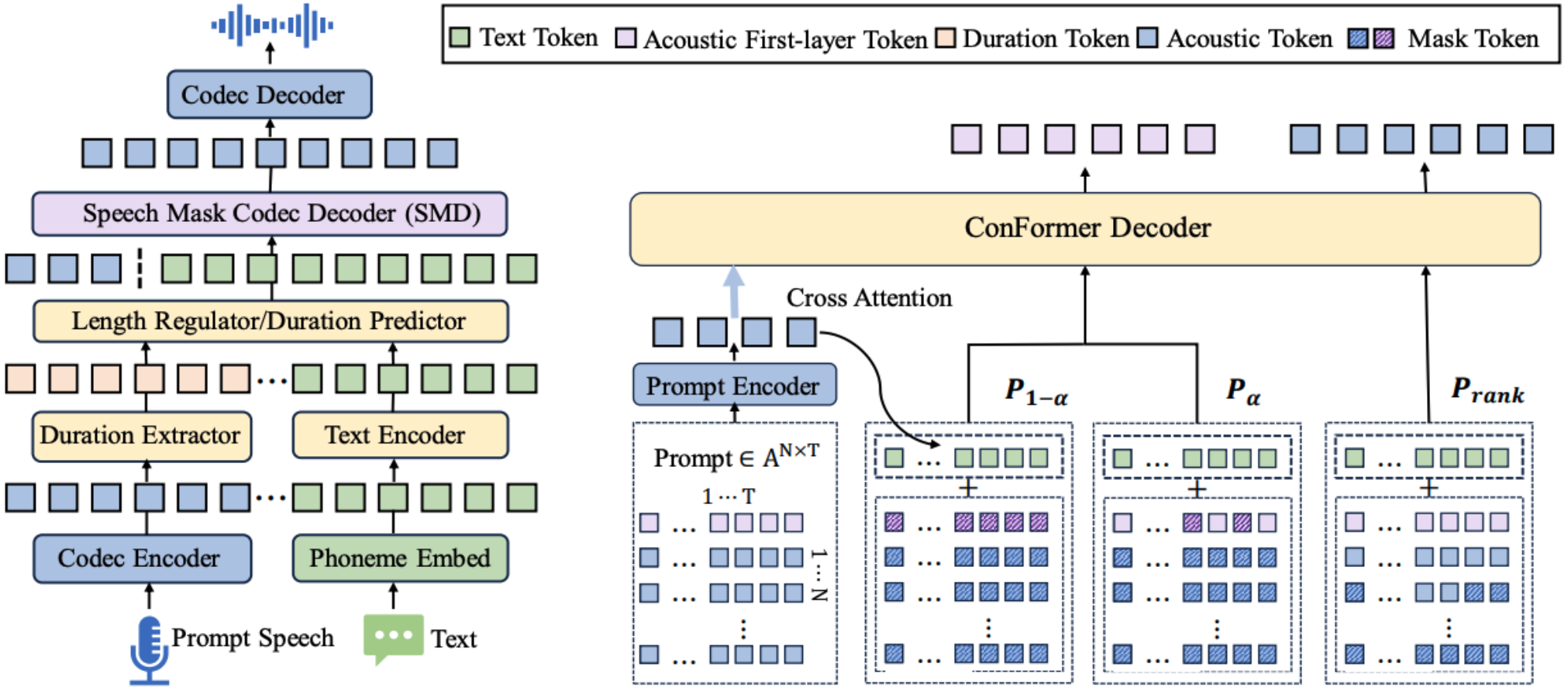
MobileSpeech: A Fast and High-Fidelity Framework for Mobile Zero-Shot Text-to-Speech

2024-03-06

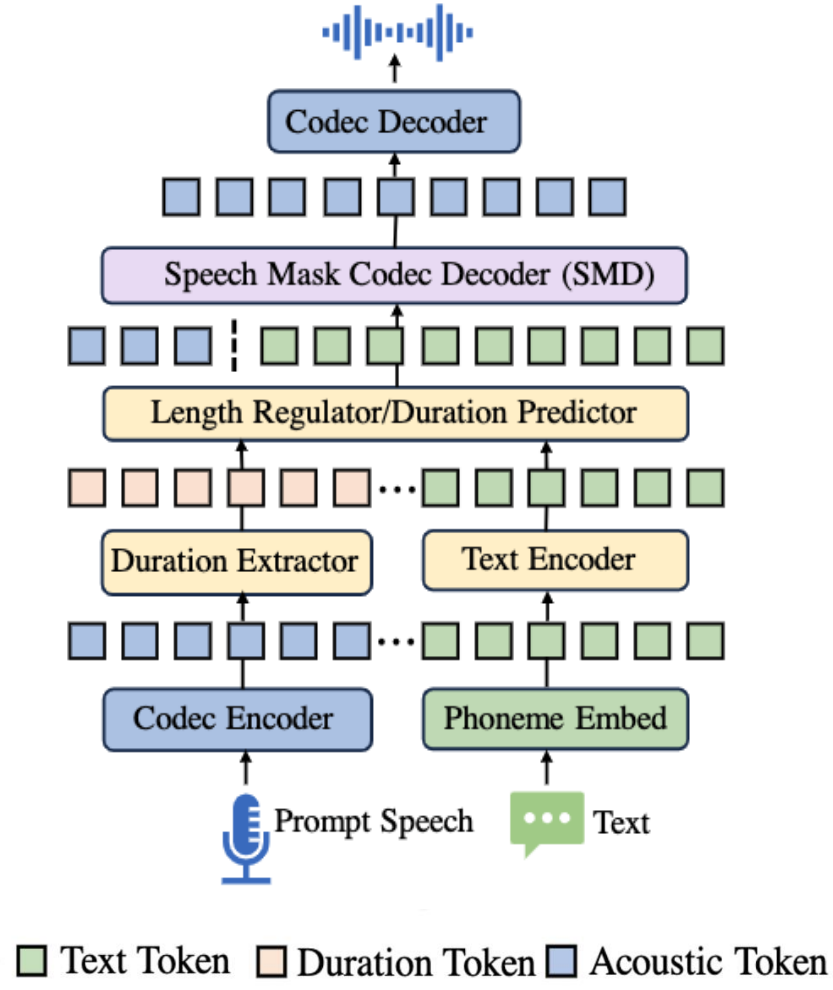
Review: Two-Stage Zero-Shot TTS

- **基于离散 token 的 TTS**
  - s2 建模：百家争鸣
    - GPT (AudioLM/Spear-TTS)
    - Diffusion (Tortoise-TTS/ X-TTS)
    - Flow (GPT-SoVITS)
    - MaskGIT (SoundStorm)
  - s1 建模：一枝独秀
    - GPT 自回归
    - 优势：隐式学习文本与 semantic token 间对齐关系；采样的结果多样性高
    - 不足：合成不够稳健，丢字/重复现象难以根除；GPT 自回归生成速度较慢
- **回顾：Tacotron → FastSpeech**
  - 引入 duration predictor，将输入输出长度不等的生成任务改为非自回归任务
  - 顾虑：这一过程中丢失了什么：韵律多样性？表现力?

MobileSpeech: Faster and Better



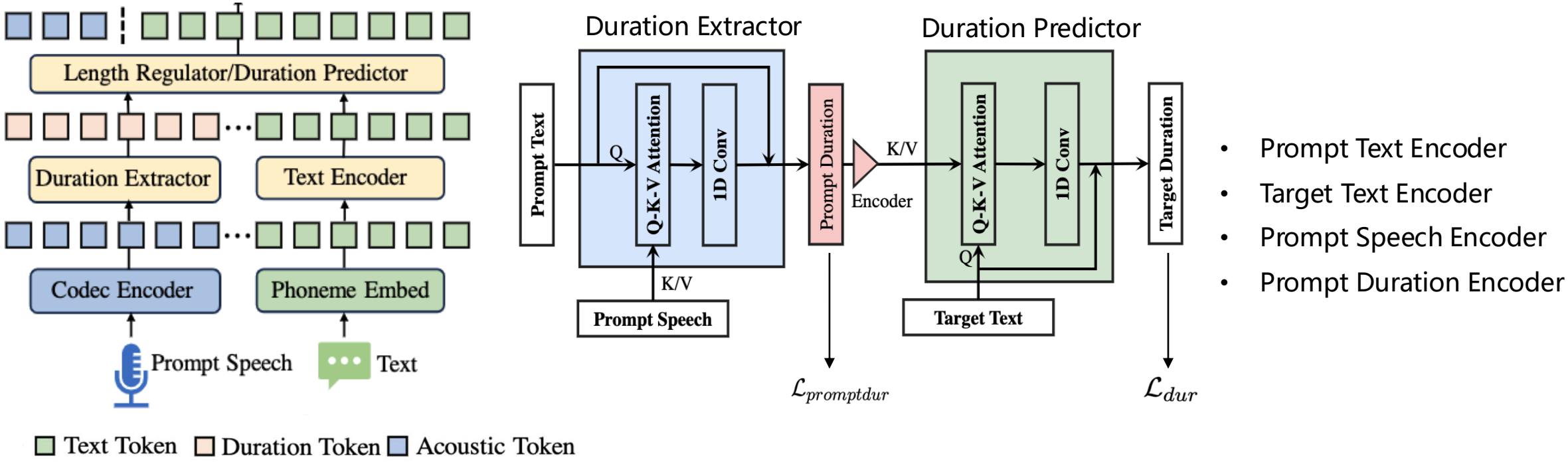
MobileSpeech: Faster and Better



- FastSpeech 作为 backbone，非自回归结构
- 不变的部分
  - 采用端到端方式训练，直接预测 target，不区分 s1 和 s2
  - 保留了 duration predictor 和 length regulator
- 创新点
  - 1. 预测目标从连续的 Mel 特征变为 Encodec 多层 codec (分类任务)
    - 基于 mel 的模型，生成效果 diversity 和 quality 都不如 codec？
  - 2. duration predictor 引入 prompt speech 作为 condition
    - 提出一种新的 duration in-context learning 结构
    - 验证了比 NaturalSpeech2 的方式更有效
  - 3. 非自回归 Decoder 改为 SoundStorm 的 MaskGIT 结构
    - 通过采样提高合成结果的多样性

MobileSpeech: Faster and Better

In-Context Duration Modeling



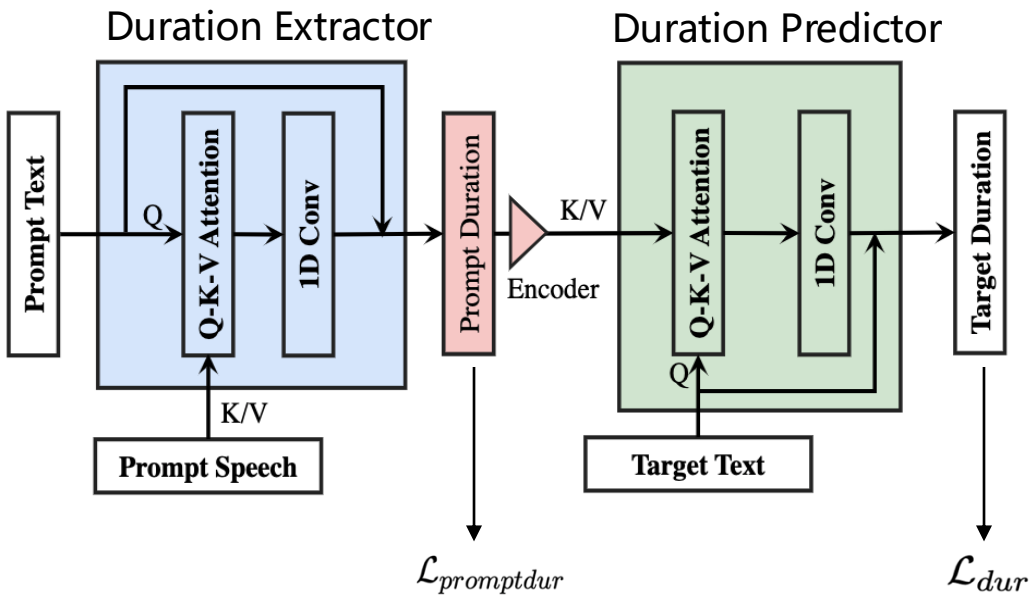
- Prompt Text Encoder
- Target Text Encoder
- Prompt Speech Encoder
- Prompt Duration Encoder

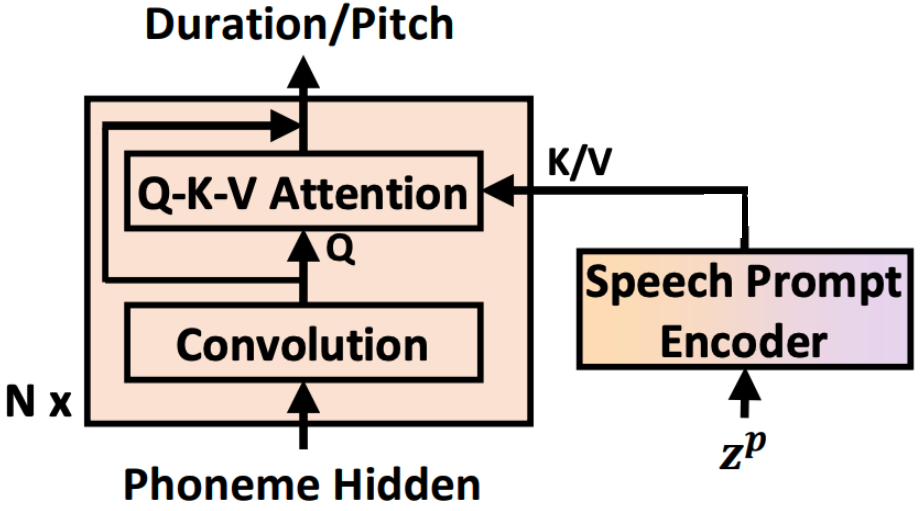
说明： Duration Extractor 相当于对齐模块，在推理时可以根据 prompt 的 text 和 speech 得到 duration

疑问： 如果已经有 duration 信息（Mega-TTS），是否直接引入到 duration predictor 即可?

MobileSpeech: Faster and Better

In-Context Duration Modeling





NaturalSpeech2 的 duration predictor

MobileSpeech: Faster and Better

Speech Codec Mask Decoder (SMD)

- 实际上只是在 SoundStorm 基础上做了点微小的优化
- 同一条长音频，拆分为 prompt 和 target 两部分
  - 文本输入通过 length regulator 规整到和 codec 相同的长度
  - 类比 SoundStorm：输入是 semantic token 的 embedding

- 问题定义

已知 $Y_{prompt} = C_{1:k,1:N}, X_{target} = X_{k:T}$ 预测：$C_{k:T,1:N}$

求解目标：最大化条件概率 $P(C_{k:T,1:N} \mid C_{1:k,1:N}, X_{k:T}; \theta)$

prompt 随机选取 $\frac{T}{3} \leq k \leq \frac{2 \times T}{3}$

MobileSpeech: Faster and Better

Speech Codec Mask Decoder (SMD)

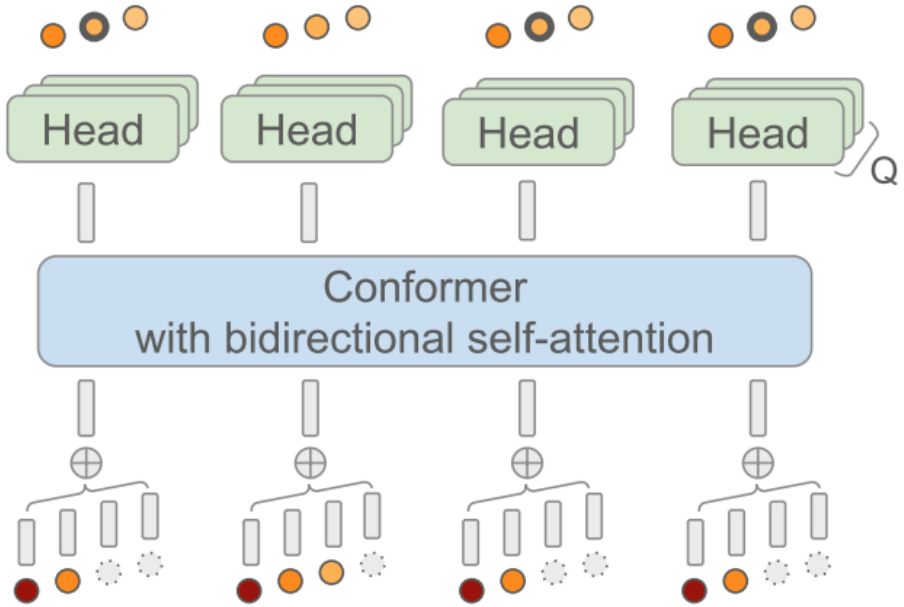
- 第一层：需要更强的建模（VALL-E 的思想）

$$P(C_{k:T,1} \mid C_{1:k,1:N}, X_{k:T}; \theta)$$

$$P(M_1 C_{k:T,1} \mid C_{1:k,1:N}, X_{k:T}, \bar{M}_1 C_{k:T,1}; \theta)$$

- 剩余其他层：
  - 训练时：随机选择某一层 j
  - 推理时：逐层 greedy search 生成

$$P(M_j C_{k:T,j} \mid C_{1:k,1:N}, X_{k:T}, C_{k:T,<j}, \bar{M}_j C_{k:T,j}; \theta)$$

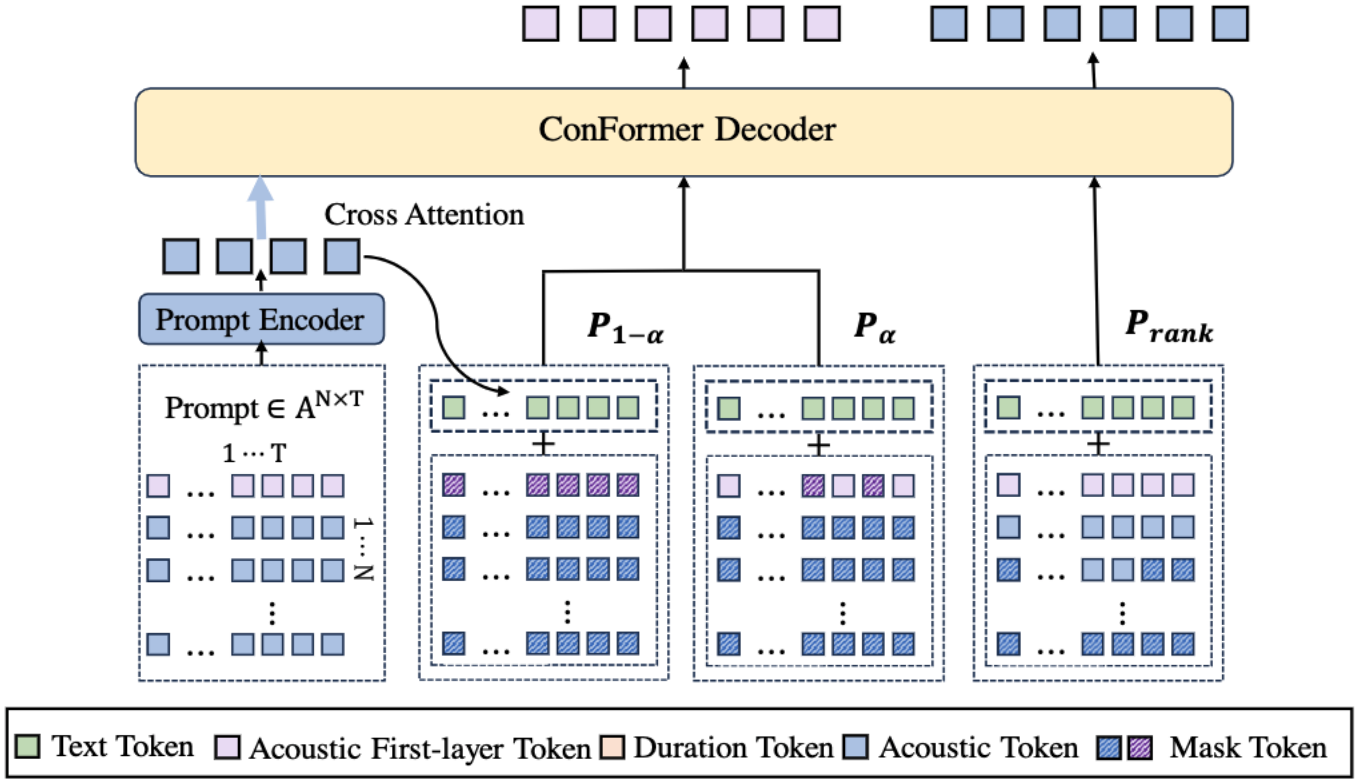


SoundStorm

MobileSpeech: Faster and Better

SMD 相比于 SoundStorm 的三点小改动

- 1. 第一层的训练过程中，以一定概率强制 mask 全部 target
  - 动机：迫使模型对预测结果更加置信，减少推理时所需的 iteration 次数



$$P(C_{k:T,1} \mid C_{1:k,1:N}, X_{k:T}; \theta)$$
$$= \alpha P(M_1 C_{k:T,1} \mid C_{1:k,1:N}, X_{k:T}, \bar{M}_1 C_{k:T,1}; \theta)$$
$$+(1-\alpha) P(M_1 C_{k:T,1} \mid C_{1:k,1:N}, X_{k:T}; \theta)$$

α = 0.6

MobileSpeech: Faster and Better

SMD 相比于 SoundStorm 的三点小改动

- 2. P_rank 策略
  - 根据 Encodec 的信息逐层递减的先验，在采样训练哪一层时，越高的层，采样权重越低
  - 动机：更关注靠近底层的 codec 层，花费越多的精力预测准确

Algorithm 1 P_{rank} algorithm

Require: The channels set $\{c_i\}_{i=2}^{N}$ and decreasing corresponding weights $\{w_i\}_{i=2}^{N}$; where $w_j > w_{j+1}, j \in \{2, 3, \cdots, N-1\}$

Initialize $\mathcal{C}_U = \emptyset$;

for each c_i **do**

  Repeat c_i for w_i times;

  Add repeated c_i to $\mathcal{C}_U$;

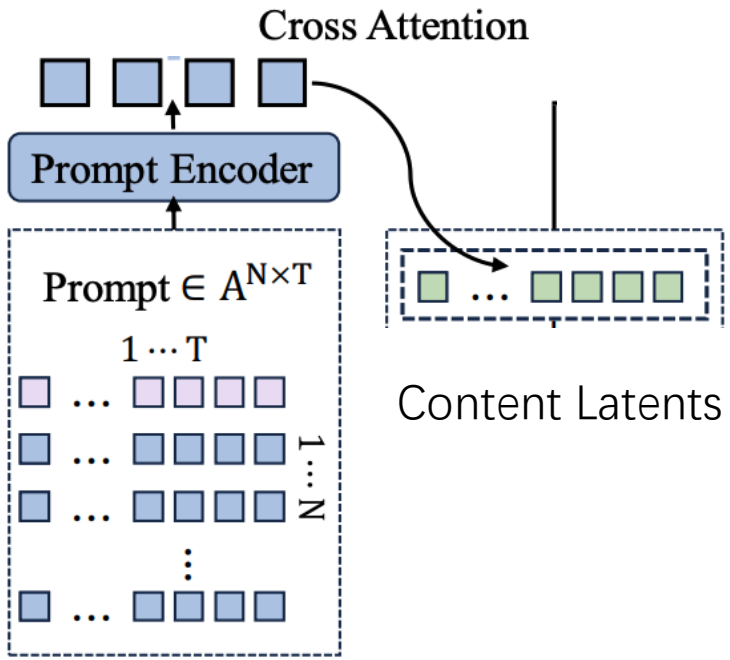
end for

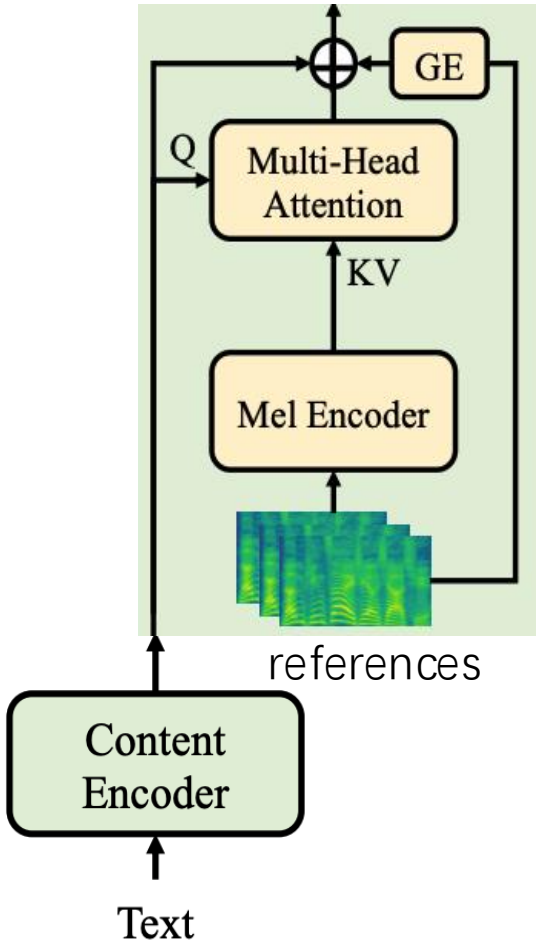
Random sample a channel c from $\mathcal{C}_U$;

MobileSpeech: Faster and Better

SMD 相比于 SoundStorm 的三点小改动

- 3. 引入类似 MRTE 的 cross-attention 模块
  - 区别：在帧级别的 content latents 上，与 prompt 做 cross-attention
  - 动机：融合一些 prompt acoustic token 的信息，比如音色等

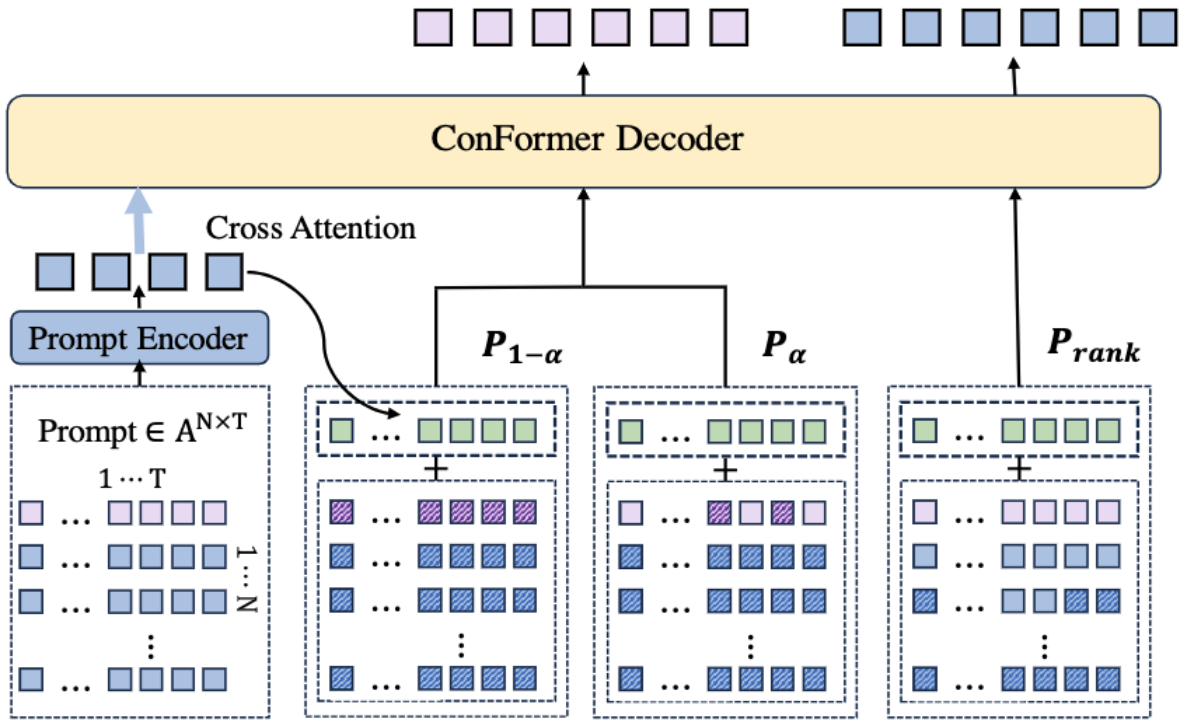




MobileSpeech: Faster and Better

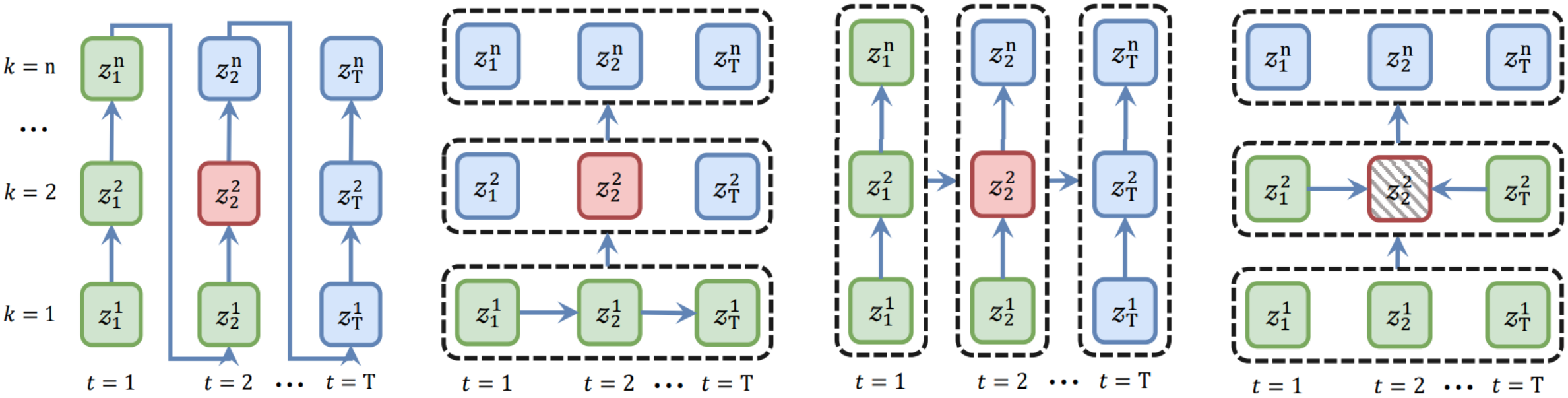
Speech Codec Mask Decoder (SMD)

- 结构上细节：prompt 部分最好也加入 prompt text 对应的 embedding?



MobileSpeech: Faster and Better

基于 RVQ 的 codec：多种建模方式



MobileSpeech: Faster and Better

实验一：英文

- 数据集：小数据量 – LibriTTS – 580 小时
- Baseline: YourTTS, VALL-E, NaturalSpeech2, Mega-TTS
- 英文在 test-clean 测试集上的效果

Model	Data	WER ↓	SPK ↑	RTF ↓	MOS-Q ↑	MOS-P ↑	MOS-S ↑
GT Codec	-	2.4	0.871	-	4.41±0.08	4.28±0.10	4.45±0.06
YourTTS	640	7.7	0.504	0.22	3.71±0.11	3.59±0.14	3.68±0.12
VALL-E	60000	5.9	**0.751**	0.94	**4.22±0.06**	4.09±0.13	**4.16±0.07**
VALL-E-Continue	60000	3.8	0.734	0.94	4.12±0.12	**4.13±0.10**	4.11±0.11
NaturalSpeech2-Continue	580	4.6	0.581	0.35	3.85±0.11	3.69±0.14	3.76±0.08
MegaTTS-Continue	580	5.8	0.615	0.39	3.93±0.08	3.85±0.09	3.89±0.09
MobileSpeech-Continue	**580**	**3.1**	**0.688**	**0.09**	**4.06±0.07**	**4.02±0.08**	**4.05±0.10**

实验二：中文

- 数据集：大数据量 – 内部数据 – 4万小时
- 对比：火山开放的音色克隆接口

Model	CMOS-Q ↑	CMOS-P ↑	CMOS-S ↑
GT codec	+0.07	+0.15	+0.03
MegaTTS 2	-0.16	-0.05	-0.24
MobileSpeech	**0.00**	**0.00**	**0.00**

MobileSpeech: Faster and Better

消融实验

1. MobileSpeech w/o durprompt:

- Duration Predictor 直接用 Prompt Acoustic Token 作为条件输入
- SMD 模块 text latents 不与 Prompt Encoder 输出做 cross-attention

2. MobileSpeech w/o onechannel

- 随机选择一层进行预测?

Model	WER ↓	SPK ↑
MobileSpeech w/o durprompt	3.6	0.638
MobileSpeech w/o onechannel	4.2	0.614
MobileSpeech	**3.1**	**0.688**

Table 3: The ablation experiments of the SMD module and the Speaker Prompt module were conducted to test SPK and WER.

Model	MOS-Q ↑	MOS-P ↑	MOS-S ↑
MobileSpeech w/o durprompt	4.03±0.05	3.90±0.11	3.97±0.12
MobileSpeech w/o onechannel	3.98±0.12	3.95±0.09	3.92±0.11
MobileSpeech	**4.06±0.07**	**4.02±0.08**	**4.05±0.10**

Table 4: The ablation experiments of the SMD module and the Speaker Prompt module were conducted to test MOS.

MobileSpeech: Faster and Better

消融实验

3. 第一层的采样次数对效果的影响

iterations	RTF ↓	WER ↓	SPK ↑
24	0.20	3.1	0.698
16	0.15	3.0	0.696
8	**0.09**	**3.1**	**0.688**
4	0.07	4.2	0.615
1	0.05	5.7	0.483

Table 5: The impact of different iterations on the first channel of MobileSpeech.

MobileSpeech: Faster and Better

总结

- 从 GPT 改回使用 Duration Predictor 的非自回归方案
  - Duration Predictor 的结构设计值得尝试
  - 避免自回归方案生成速度较慢的问题
  - 支持在移动端部署，同时效果强于 Mega-TTS2

- 可能的改进点
  - 顾虑：直接从文本预测 acoustic token 可能难度较大
  - 改动：将 MobileSpeech 仍拆分 s1 + s2 两部分
    - s1: 基于非自回归预测时长 + semantic token
    - s2: 套用 semantic token → acoustic token 的建模方式 (加上本文的改进点)

SoundStorm 近期改进

Pheme:

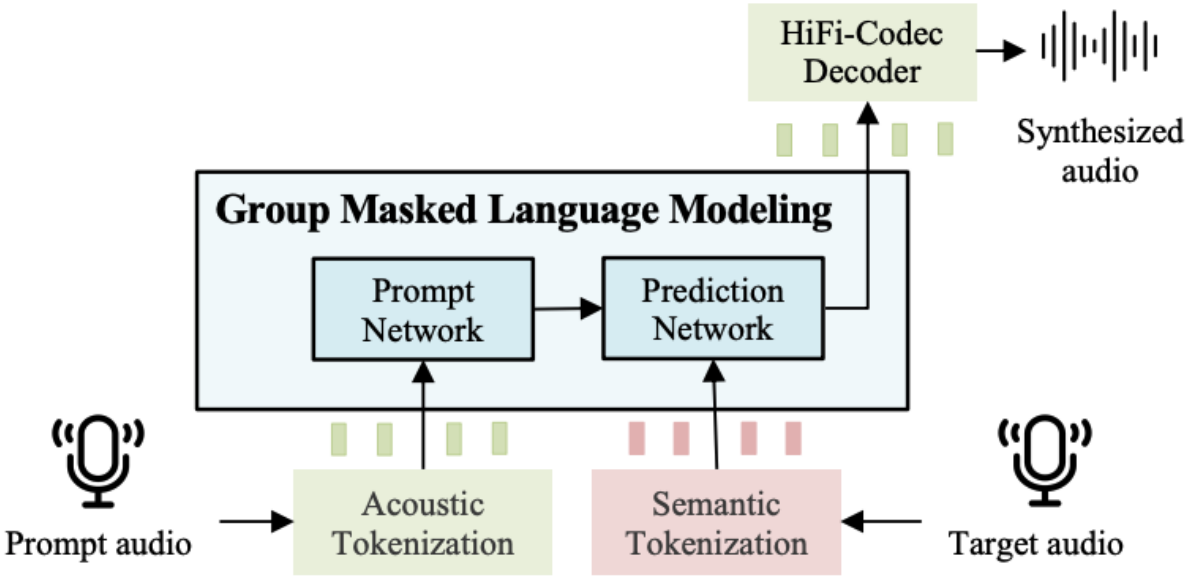
- https://arxiv.org/pdf/2401.02839.pdf
- 基于 SpeechTokenzier 的 SpearTTS-s1 + SoundStorm
- 可参考点：在 SoundStorm 中加入 prompt 提取的 speaker embedding (每个时间步)
  - 显式提供宏观的 speaker 信息，对音色相似度有明显提升

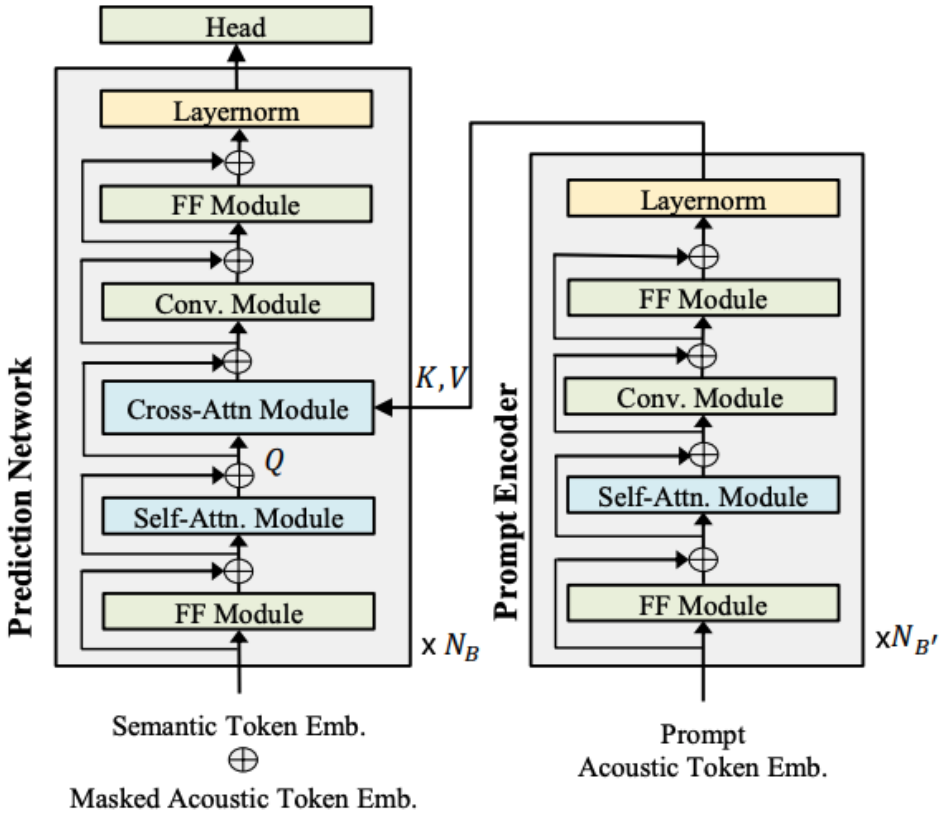
Model	WER ↓	SSS ↑	MCD ↓	FID ↓
MQTTS (100M)	14.2	**0.682**	9.568	19.690
PHEME-SMALL (100M)	**12.4**	0.594	**8.838**	**20.349**
PHEME-SMALL (100M), w/o SE	16.3	0.492	8.893	20.608
PHEME-LARGE (300M)	11.9	0.549	8.671	19.675

SoundStorm 近期改进

SoundGroup:

- https://arxiv.org/pdf/2401.01099.pdf
- 为 SoundStorm 提供了一种新的 prompt 建模方式





- 不需要 prompt 的 semantic token
- K-V 可以缓存，不需要重复计算