Recent Advances in Adaptive Text-to-Speech

2022-05-26

Outline

- Background
  - Adaptive TTS
  - FastSpeech 1/2
  - AdaSpeech
- AdaSpeech 2: Adaptive Text to Speech with Untranscribed Data
- AdaSpeech 3: Adaptive Text to Speech for Spontaneous Style
- AdaSpeech 4: Adaptive Text to Speech in Zero-Shot Scenarios
- Conclusion

Background

2022-05: https://arxiv.org/pdf/2205.04421.pdf

NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality

微软全华班放出语音炸弹！NaturalSpeech语音合成首次达到人类水平

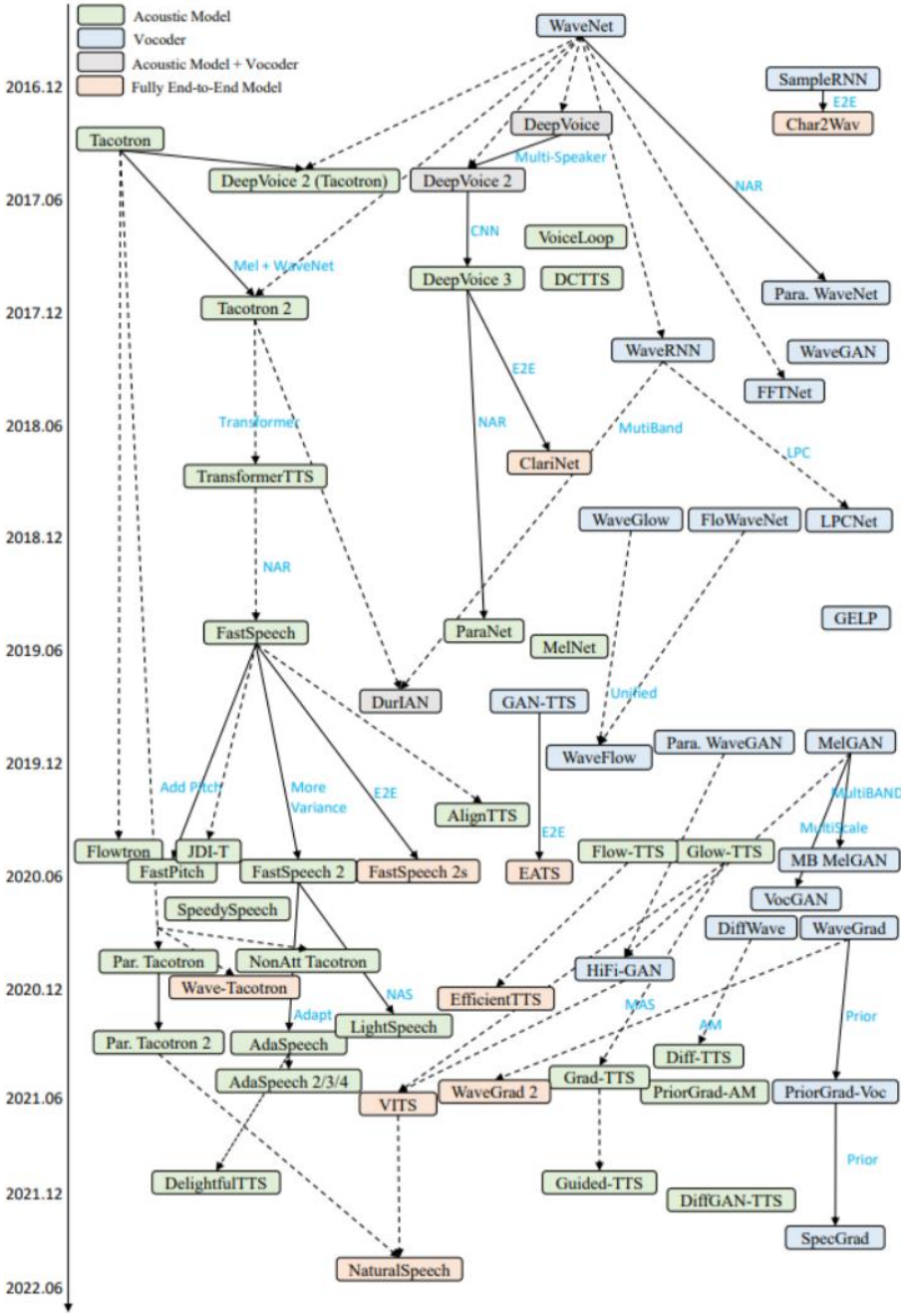
新智元 新智元 2022-05-17 13:11 发表于北京

Table 2: MOS comparison between NaturalSpeech and human recordings. Wilcoxon rank sum test is used to measure the p-value in MOS evaluation.

Human Recordings	NaturalSpeech	Wilcoxon p-value
4.58 ± 0.13	4.56 ± 0.13	0.7145

Table 3: CMOS comparison between NaturalSpeech and human recordings. Wilcoxon signed rank test is used to measure the p-value in CMOS evaluation.

Human Recordings	NaturalSpeech	Wilcoxon p-value
0	−0.01	0.6902



LJSpeech 数据集特点:

- 类型/风格：**阅读**文章
- 训练语音时长：**24** 小时，**1** 个说话人

Background

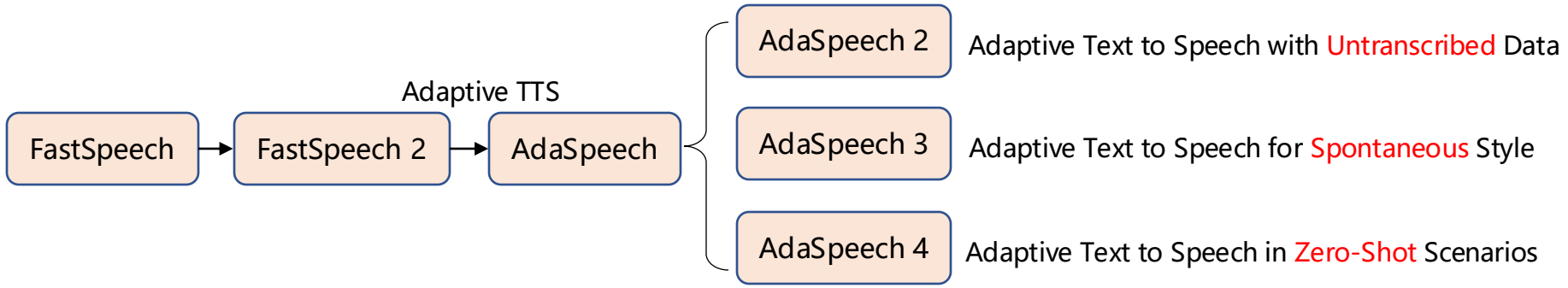
一般方法

- Train from scratch 从头训练：将目标说话人数据和所有多说话人数据一起训练多说话人 TTS 模型
- Adaptation 预训练-调优：用很多说话人数据训练 base 模型，再用一些目标说话人的数据调优

目标：已有一个训练好的多说话人 TTS 模型，如何在将模型 Adapt 到另一个领域/风格或说话人，使其达到满意的合成效果？

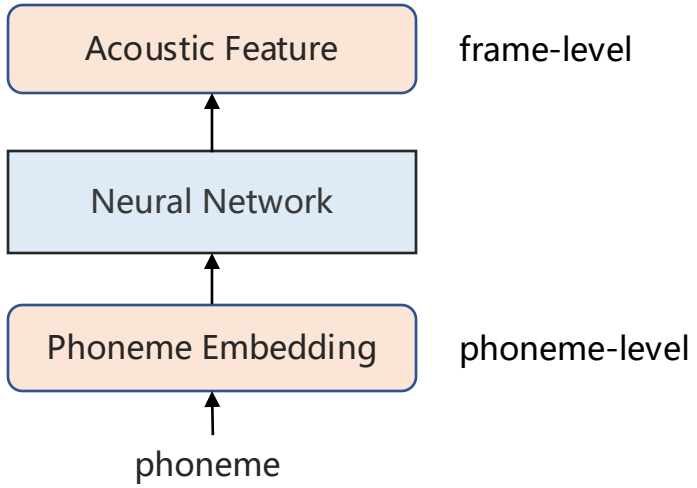
TTS 实际应用中的问题 → Adaptive TTS

- 新的目标说话人训练数据不足
  - 无标注语音比较充足 → Untranscribed Data → AdaSpeech 2
  - 无标注语音只有一条 → Zero-Shot TTS → AdaSpeech 4
- 自然口语语音合成 → Spontaneous Speech → AdaSpeech 3

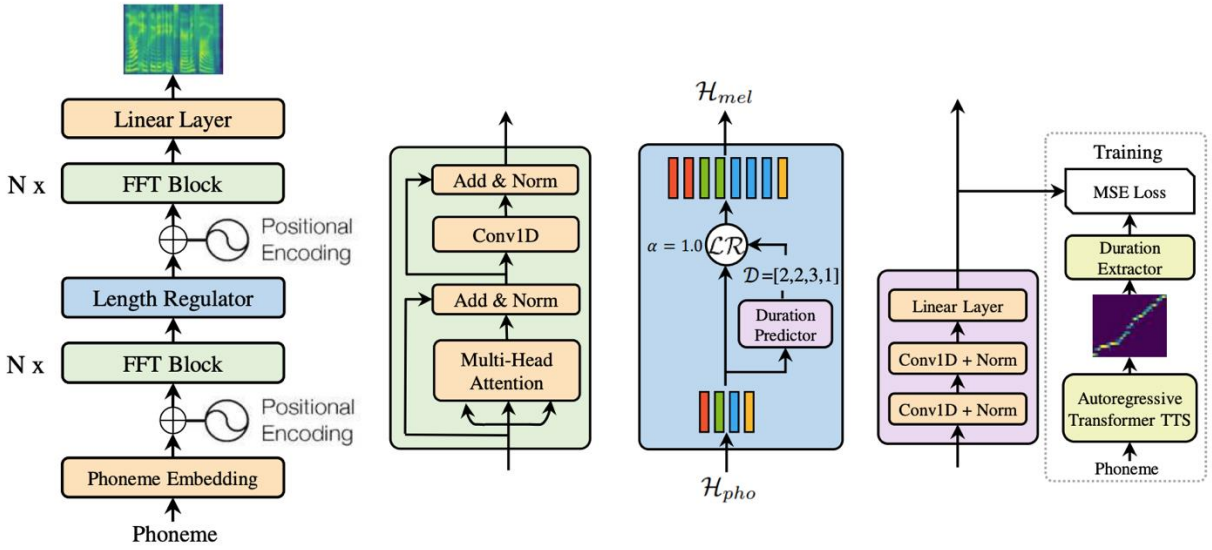


Background

The three key components in neural TTS

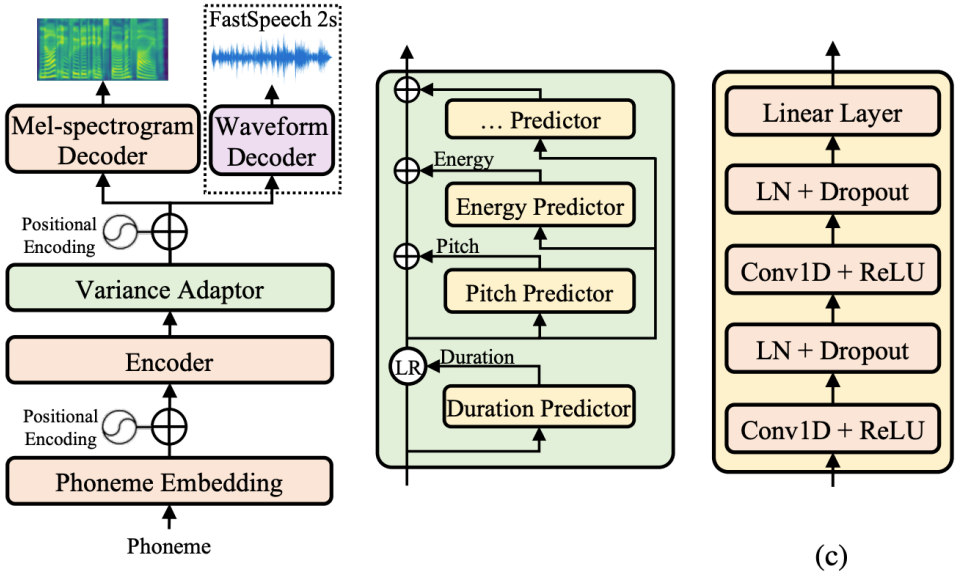


FastSpeech 1/2



(a) Feed-Forward Transformer (b) FFT Block (c) Length Regulator (d) Duration Predictor

FastSpeech



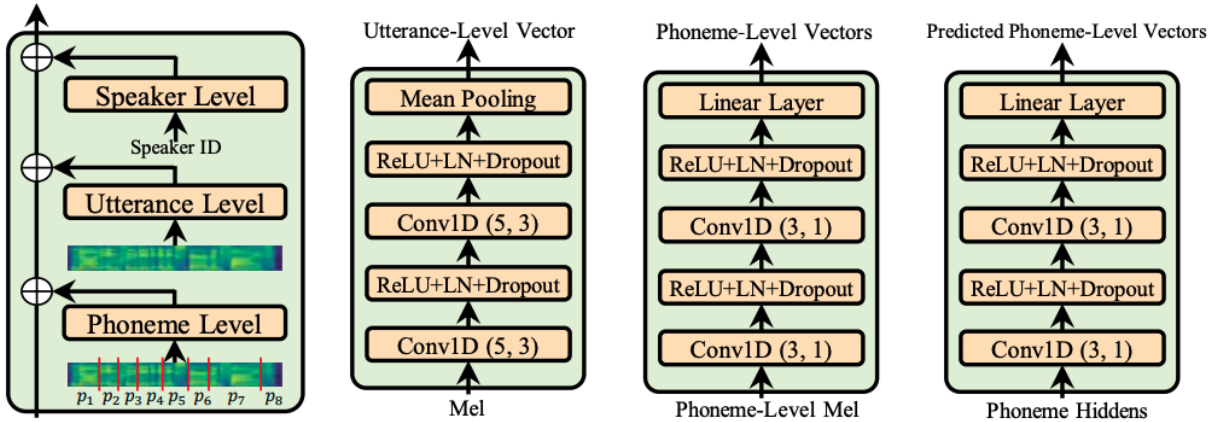
(a) FastSpeech 2 (b) Variance adaptor (c) Duration/pitch/energy predictor

FastSpeech 2

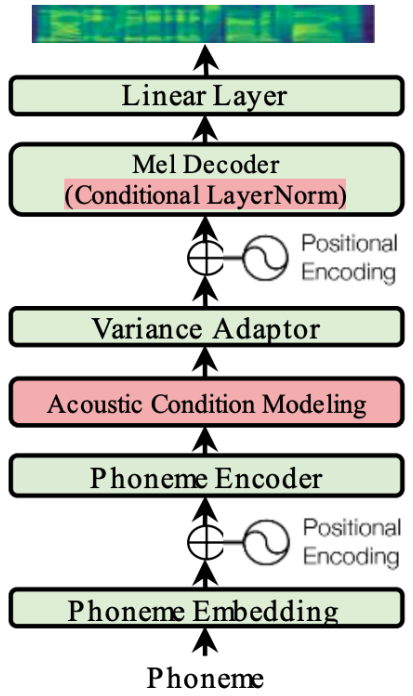
AdaSpeech

- **Acoustic Condition Modeling**
  - 对说话人、句子、音素等不同粒度声学条件进行建模
  - 提高模型对声学条件的泛化能力

- **Conditional Layer Normalization (CLN)**
  - 只微调 CLN 中 Condition Network 的参数
  - 微调的参数量数目和声音质量之间的 trade off，希望在微调参数量较少的情况下，达到更好的合成效果



(a) Overall. (b) Utterance level. (c) Phoneme level. (d) Phoneme level.



$$LN(x) = \gamma \frac{x-\mu}{\sigma} + \beta$$

LayerNorm

- μ 是隐含向量的均值
- σ 是隐含向量的方差
- γ 是可学习的 scale 参数
- β 是可学习的 bias 参数

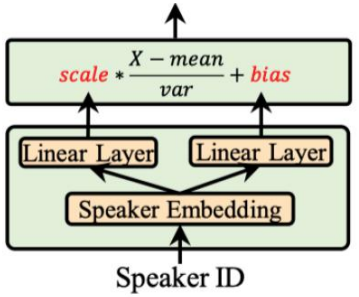


Figure 3: Conditional LayerNorm.

AdaSpeech 2: Adaptive Text to Speech with Untranscribed Data

Yuzi Yan, Xu Tan, Bohan Li, Tao Qin, Sheng Zhao, Yuan Shen, Tie-Yan Liu

ICASSP 2021

https://arxiv.org/pdf/2104.09715.pdf

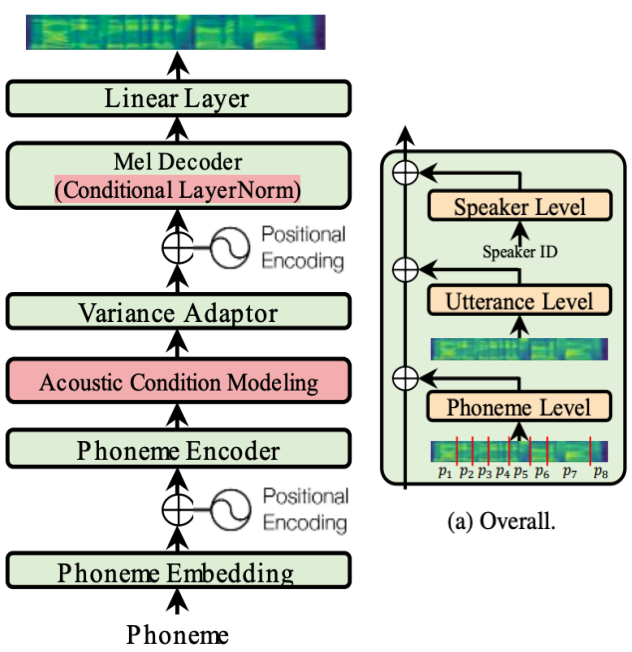
AdaSpeech 2

Adaptive TTS

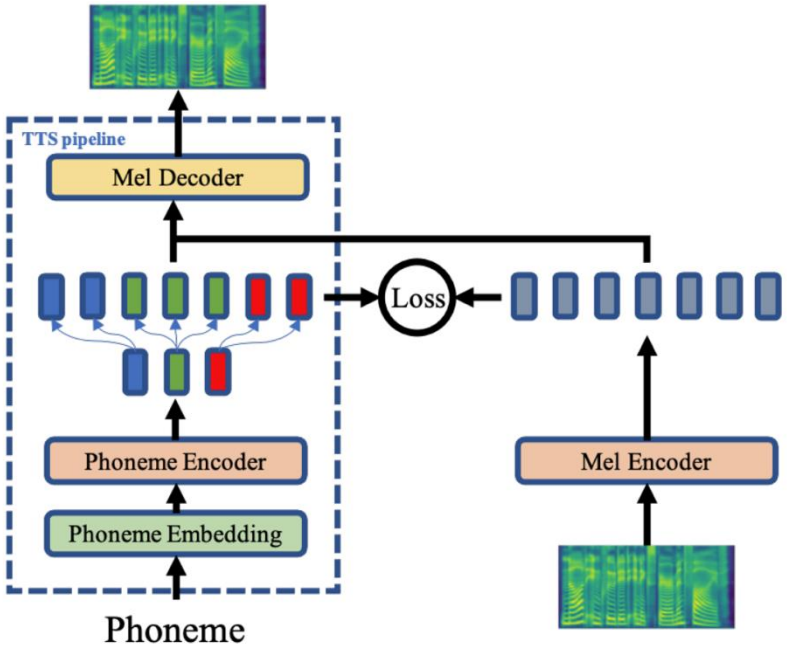
- Adaptation 预训练-调优：更多的说话人数据训练种子模型，再使用一定量的目标说话人数据调优
- 如果只有目标说话人未标注的语音，如何 Adaptation?

直观思路：使用 ASR 识别出文本/音素序列，得到文本-语音数据对

- 问题一：需要训练 ASR 模型　　问题二：ASR 结果存在错误

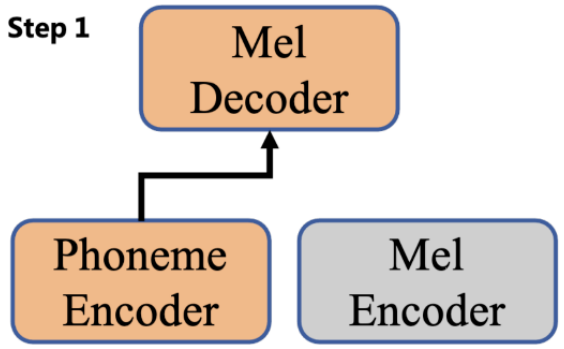


AdaSpeech



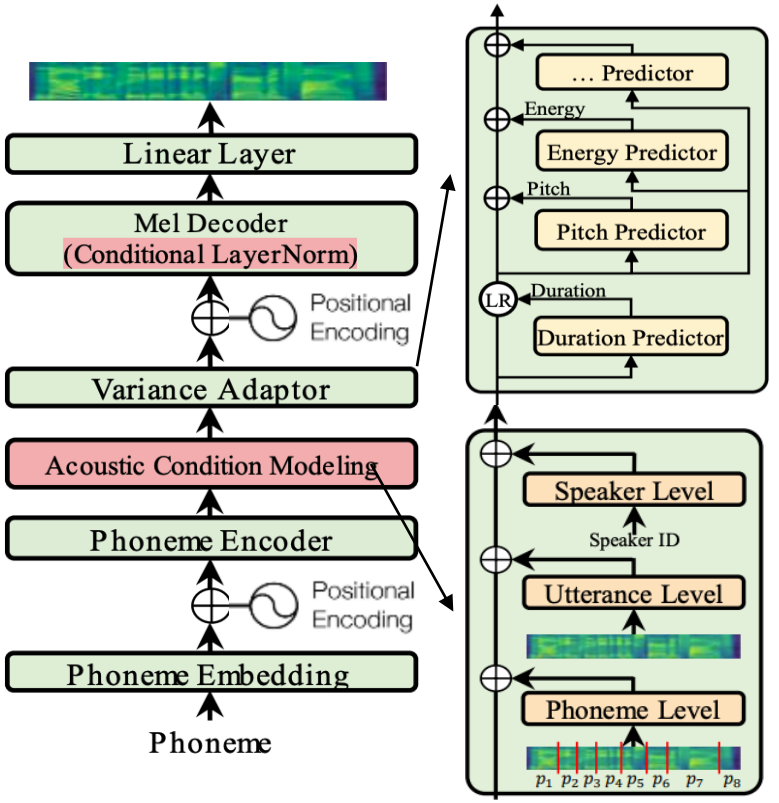
AdaSpeech 2

AdaSpeech 2



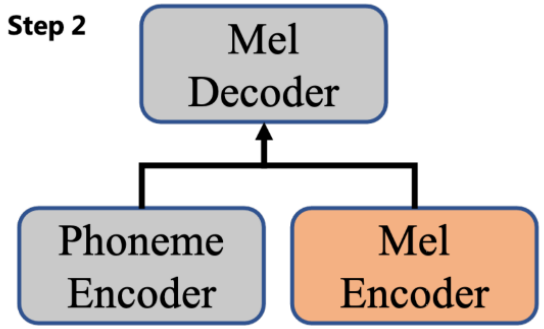
Step 1：先训练 Multi-Speaker TTS 模型结构

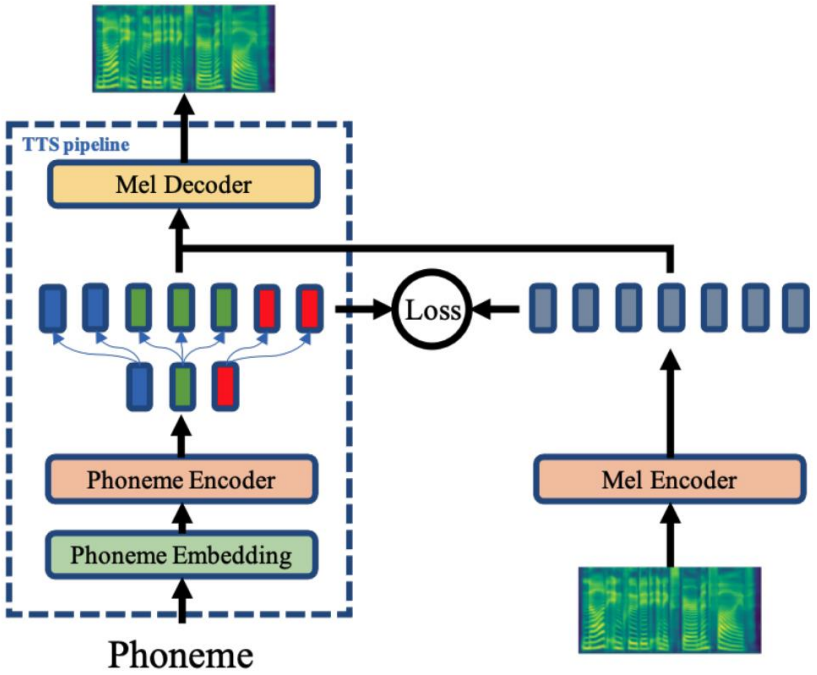
- 训练 AdaSpeech
  - Acoustic Condition Modeling
  - Conditional LayerNorm (CLN)
  - Duration Predictor：来源 MFA



AdaSpeech

AdaSpeech 2

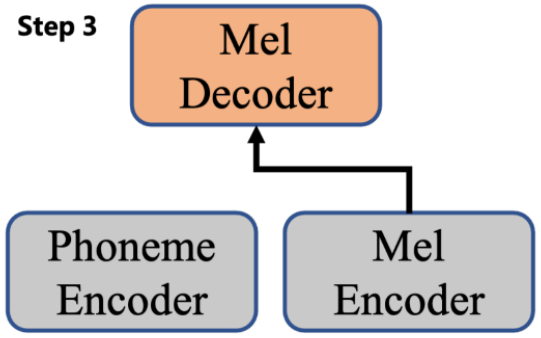


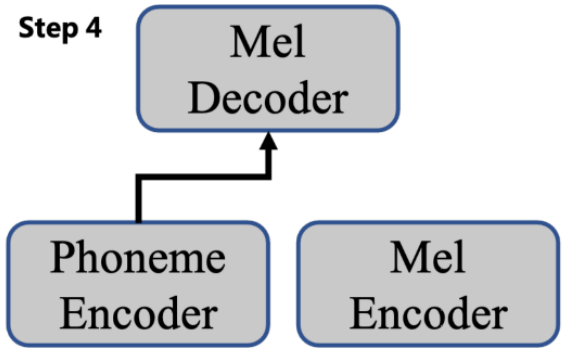


Step 2：训练 Mel Encoder (Mel Encoder Aligning)

- 训练数据：有标注的数据
- 训练目标：Mel Encoder 输出和 Phoneme Encoder 的输出相同
- 损失函数：L2 Alignment Loss + Speech Reconstruction Loss
- 思考：Phoneme Encoder 和 Mel Decoder 参数固定，因为从有标注数据中学习到的参数已经足够有效且可靠，只需要更新 Mel Encoder 使其学习到和 Phoneme Encoder 类似的作用

AdaSpeech 2





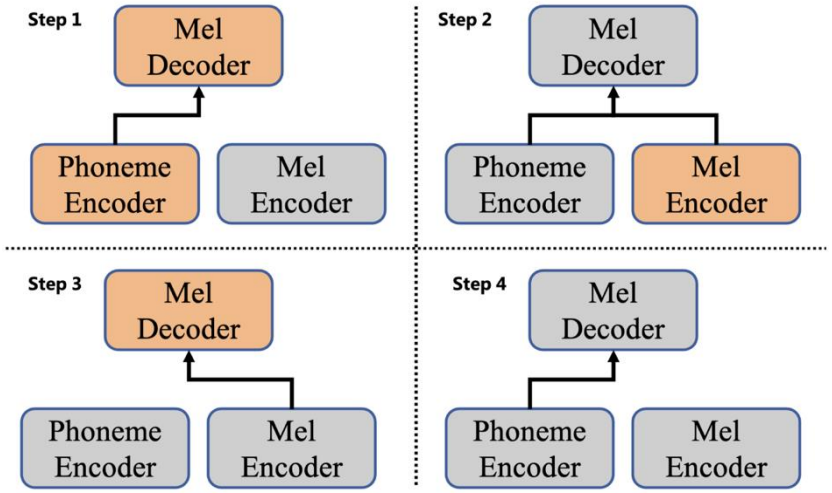
Step 3：Untranscribed Speech Adaptation

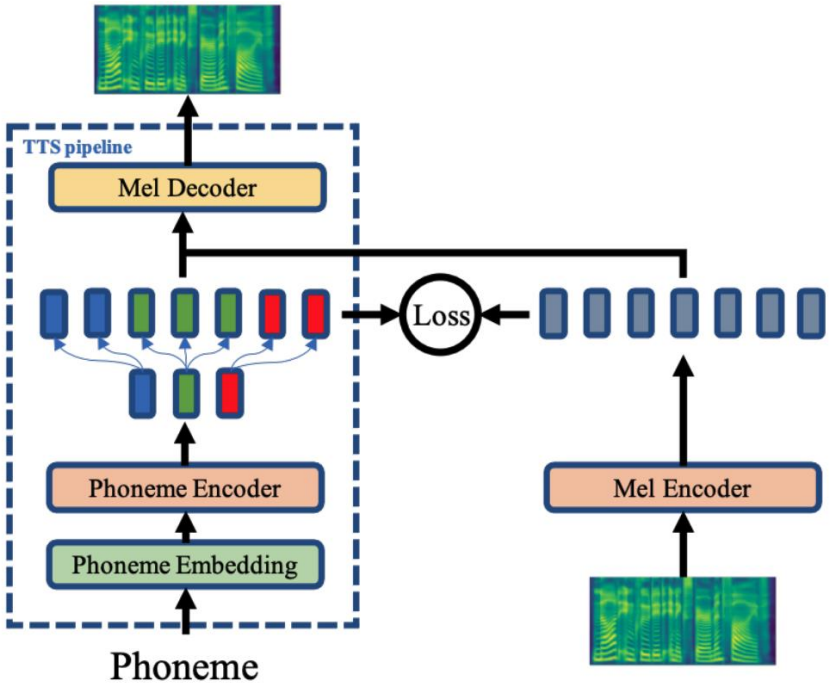
- 无标注语音输入 Mel Encoder
- 训练目标：Mel Decoder 输出与 Mel Encoder 输入相近
- 训练过程中微调 Mel Decoder，采用 AdaSpeech 的思想，只微调 Mel Decoder 的 CLN 部分

Step 4：Inference

- 推理时输入的音素序列，前向预测 Mel
  - Phoneme Encoder Step 1 训练后参数一直不变
  - Mel Decoder 针对说话人进行了微调

AdaSpeech 2





1. Step 1：先训练 Multi-Speaker TTS 模型
2. Step 2：训练 Mel Encoder，L2 Alignment Loss
3. Step 3 : 未标注数据 adapt Mel Decoder 的 CLN
4. Step 4：Phoneme Encoder + Mel Decoder 推理

AdaSpeech 2

实验信息

- Source Multi-Speaker TTS
  - 训练数据：LibriTTS, 586 小时，2456 个说话人
- Adaptation
  - 数据1：VCTK, 44 小时 108个说话人
  - 数据2：LJSpeech，24 小时 1个说话人
  - 数据3：内部的 spontaneous speech
- 模型训练配置
  - 音频采样率：16kHz
  - 前端：文本使用 g2p 转成音素
  - Phoneme / Mel Encoder + Mel Decoder
    - embedding size, hidden size 都是 256
    - Multi-head: 2 head
    - Conv1d: filter size 1024, kernel size 9
  - Mel Decoder 输出: 256
  - Linear: 输出 80, target 对应于 80 维的 Mel 特征

训练配置

- Step 1: 100000 steps, Source Multi-Speaker TTS
- Step 2 : 10000 steps, Mel Encoder Aligning
- Step 3 : 2000 steps, 微调 Mel Decoder

训练说明

- Step 1 + Step 2：使用的是有标注的数据
- Step 3：只用了目标说话人无标注的数据
- 对于新说话人的无标注数据，不用重新训练整个网络，只需执行 Step 3

AdaSpeech 2

对比实验

1. GT：真实的人声
2. GT Mel + Vocoder：给出声码器 MelGAN 对 MOS 的影响
3. joint-training (联合训练)：Step 2 中 Phoneme Encoder 和 Mel Encoder 同时训练
4. 基于 PPG 的方法
    - 将 Mel Encoder 替换为 PPG Encoder，输入为：用 ASR 声学模型提取的 PPG
    - PPG Encoder 结构与 Mel Encoder 相同，具体配置
        - 输入为 512 维的 PPG (Senone-PPG)，线性投影到 256 维
        - 优势：基于 ASR 模型引入了靠谱的 phoneme 级别的信息
5. AdaSpeech (天花板)
    - AdaSpeech 2 中 finetune 使用的无标注语音， AdaSpeech 中用的是带有标注的语音

- Adaptation 的测试集说话人
    - VCTK： 3 男 3 女 + LJSpeech：1 女
    - Adaptation 数据量：50 条无标注语音
- 测试文本： 15 条　　评价人数： 20 人
- 评价指标：MOS / SMOS (Similarity MOS) / CMOS (Comparison MOS)

AdaSpeech 2

实验结论

MOS 值

- 达到了和 PPG / AdaSpeech 相近的程度
- 比 joint-training 的效果更好

SMOS 值

- 比 AdaSpeech 差，与 PPG 方案 持平
- 但比 joint-training 更好

CMOS 值

- 相比于 AdaSpeech 的 CMOS = -0.012，略差一点

相比于 PPG 方案的优势

- 不需要另外的可靠的 ASR 模型帮助提取 PPG 信息

Metric	Setting	VCTK	LJSpeech
MOS	*GT*	3.58 ± 0.12	3.63 ± 0.11
	GT mel+Vocoder	3.42 ± 0.12	3.49 ± 0.11
	Joint-training	2.91 ± 0.09	2.89 ± 0.12
	PPG-based	3.39 ± 0.11	3.44 ± 0.12
	AdaSpeech	3.39 ± 0.10	3.45 ± 0.11
	AdaSpeech 2	3.38 ± 0.12	3.42 ± 0.12
SMOS	*GT*	4.20 ± 0.12	4.24 ± 0.09
	GT mel+Vocoder	4.06 ± 0.08	4.02 ± 0.11
	Joint-training	3.71 ± 0.13	3.19 ± 0.16
	PPG-based	3.82 ± 0.11	3.51 ± 0.15
	AdaSpeech	3.94 ± 0.12	3.59 ± 0.12
	AdaSpeech 2	3.84 ± 0.08	3.51 ± 0.12

AdaSpeech 2

消融实验

改变一：去掉训练 Mel Encoder 时的 L2 损失函数，其他保持不变，Step 2 只包含 Speech Reconstruction 的损失函数。

原因：无法保证 Phoneme Encoder 和 Mel Encoder 输出在相同空间

改变二：Step 3 不只是调优 Mel Decoder 的 CLN，而是联合 Mel Encoder 一起调优。

原因：改变了 Mel Encoder 在 Step 2 学习到的与 Phoneme Encoder 相同的输出空间，从而影响 Mel Decoder 导致 Step 4 出现不匹配问题。

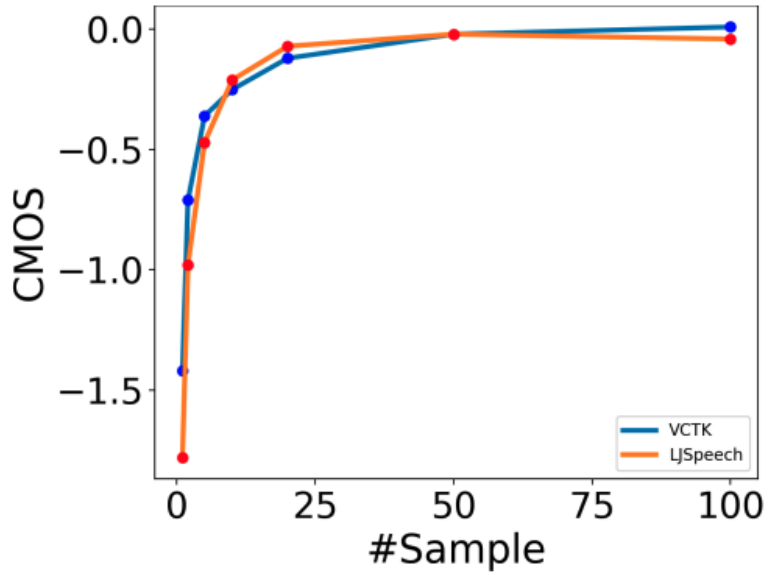
Table 2: Analyses on the adaptation strategy. *Origin* represents the original pipeline that constrains the mel encoder and the phoneme encoder with an L2 loss, and only fine-tune a part of parameters in the mel decoder.

Setting	CMOS	SMOS
Origin	0	3.89 ± 0.12
Without L2 loss constraint	-0.112	3.82 ± 0.12
Fine-tune mel encoder & decoder	-0.132	3.79 ± 0.12

AdaSpeech 2

实验：Adaptation 的数据量

- 数据量从 1- 20 条的时候，CMOS 提升显著
- 20 条之后，增加 Adaptation 的数据量，效果没有明显更大提升



AdaSpeech 3: Adaptive Text to Speech for Spontaneous Style

Yuzi Yan, Xu Tan, Bohan Li, Guangyan Zhang, Tao Qin, Sheng Zhao,

Yuan Shen, Wei-Qiang Zhang, Tie-Yan Liu

Interspeech 2021

https://arxiv.org/pdf/2107.02530.pdf

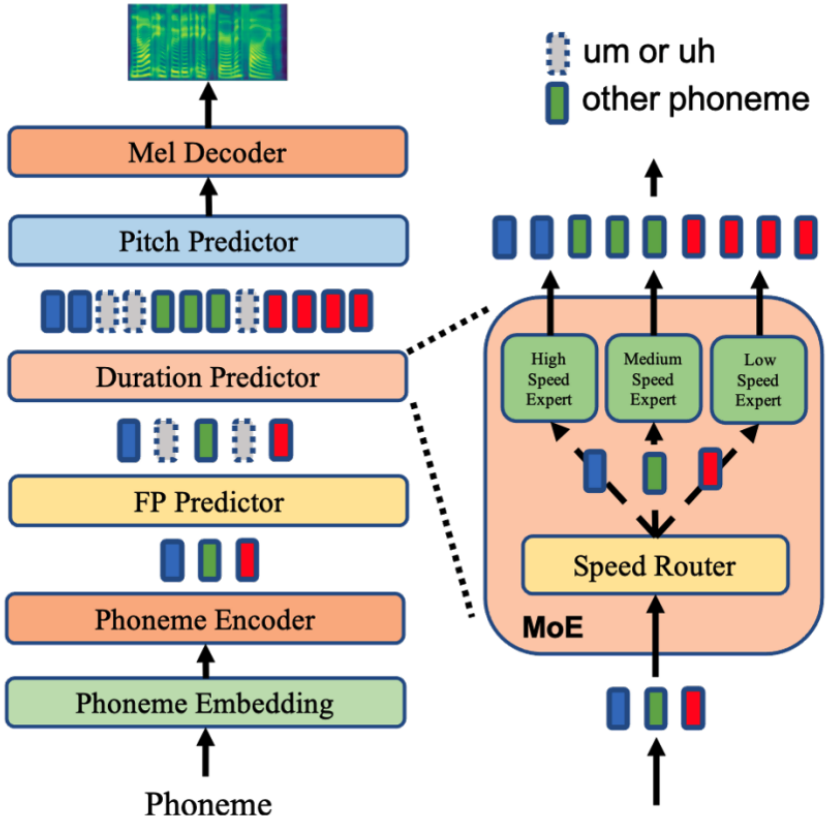
AdaSpeech 3

- 目标
  - 以阅读风格的 TTS 模型为基础
  - 最终效果：口语化风格 + 目标说话人音色

- 优化一：针对**停顿语气词 Filled Pauses → FP Predictor**

- 优化二：针对**口语语音的 Rhythm → Rhythm Adaptation**

- 优化三：针对**说话人音色 Timbre → Speaker Adaptation**



AdaSpeech 3

AdaSpeech 3

解决口语语音数据缺失的问题

- 口语数据构建，archive.org 爬取 28 小时播客，并转录
  - 包含口语停顿词 (um/uh)
  - 将文本使用 g2p 转换为音素序列
  - 将音频切分为 7 – 10 秒的片段

- 构建数据集
  - SPON-FP，用于 FP Predictor
    - 338 um: ah m
    - 2614 uh: ah
  - SPON-RHYTHM，用于 Pitch/Duration Predictor
  - SPON-TIMBRE，用于说话人 Adaptation

Table 2: *The statistics of the three datasets mined for spontaneous speech.*

Adaptation Step	Name	Sentence Num
FP prediction	SPON-FP	2952
Rhythm fine-tuning	SPON-RHYTHM	14273
Speaker adaptation	SPON-TIMBRE	50

AdaSpeech 3

Step 0:

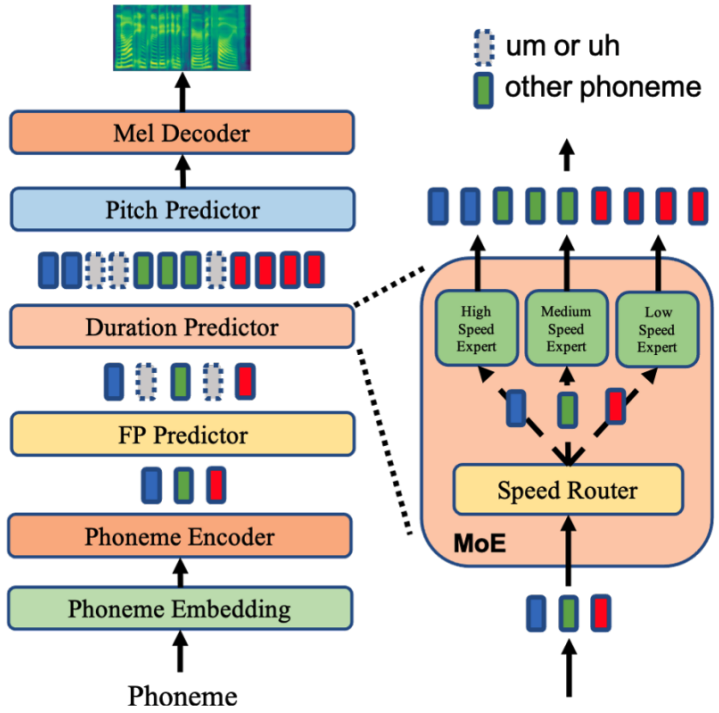
- 首先训练一个 Multi-Speaker TTS 模型 Adaspeech

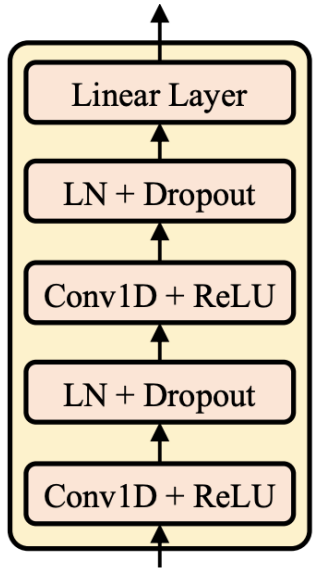
Step 1: FP Predictor

- 位置：在 Phoneme Encoder 之后增加 FP Predictor
- 输入：Phoneme Encoder 的输出
- 输出：预测每个音素后面是否有 FP 以及是哪个 FP
- 模型：三分类: 0 (No FP), 1 (uh), 2 (um)

Table 1: *An example of text-FP data pair. Phoneme w/o FP is the phoneme sequence with filled pauses (FP) removed.*

Raw text	*It's called* ***um*** *right* ***uh*** *apple*
Raw phomeme	*ih t s k ao l d* ***ah m*** *r ay t* ***ah*** *ae p ax l*
Phomeme w/o FP	*ih t s k ao l d r ay t ae p ax l*
FP tag	0, 0, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0





AdaSpeech 3

Step 1: FP Predictor

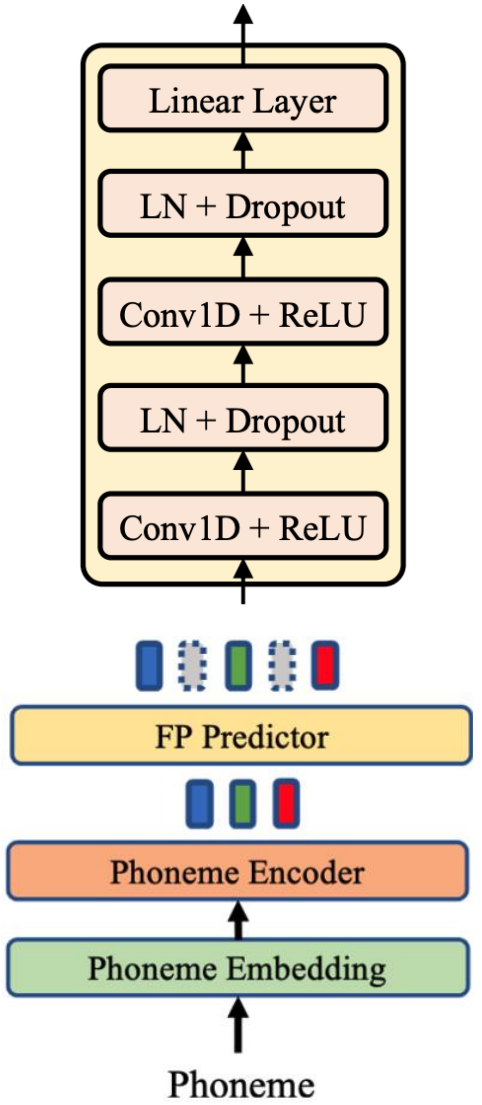
问题：1(uh)/2(um) 类别的数据太少，数据不均衡，倾向于全部预测为 No FP

一些方案：

1. 只使用包含了 uh/um 的句子 (SPON-FP) 作为训练数据
2. 加权损失函数，提高类别 1 和 2 的 class weight，减轻数据不均衡问题

$$L = -y_0 \log s_0 - \sigma \sum_{i=1}^{2} y_i \log s_i$$

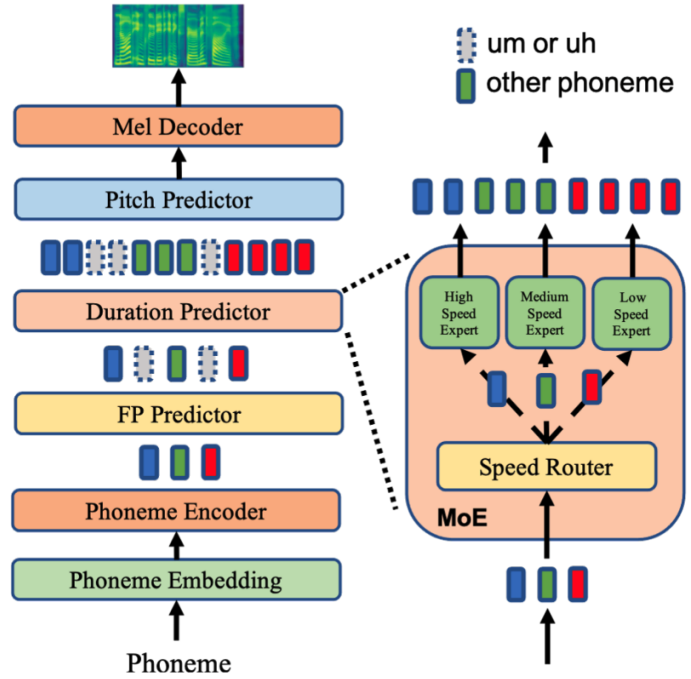
3. 预测时，可以通过调整阈值控制 FP 预测的强度
   - s0, s1, s2 分别代表 no FP, uh, um 三者的概率
   - s0 > T 时，预测 no FP
   - s0 <= T 时，选概率更大的类别
   - 阈值越低，越容易预测出 no FP
   - 阈值越高，越容易预测出 FP，插入到音素序列中
   - 加入方式：将 FP 的 Embedding 添加到相应的 Phoneme 隐含向量序列中



AdaSpeech 3

Step 2: Rhythm Adaptation

- Pitch Adaptation: 使用 SPON-RHYTHM 的数据微调 **Pitch Predictor**
- Duration Adaptation: 基于 MoE 的 **Duration Predictor**



MoE（Mix-of-Experts）的 Duration Predictor

- Router：得到每个 Expert 的权重
  - 将 Phoneme 根据 duration 分为 low, medium, high 三类语速标签
  - FP Predictor 的输出和对应 Phoneme 的语速类别，训练分类器
  - 分类器模型和 Duration Predictor 结构相同，输出三分类的 softmax 后的概率值
- Expert：每个 Expert 是一个单独的 Duration Predictor
  - 初始化：使用的是 Multi-Speaker TTS 模型的 Duration Predictor
  - 训练：分别用各自语速类别的数据调优 Duration Predictor
  - 预测：最终预测的 duration 是三个 Expert 输出的加权平均，权重来自 Speed Router

Step 3: Speaker Timbre Adaptation 与 AdaSpeech 相同

AdaSpeech 3

实验信息

- Source Multi-Speaker TTS
  - 训练数据：LibriTTS　586 小时，2456 个说话人
- **模型训练配置**
  - 音频采样率：16kHz
  - 前端：文本使用 g2p 转成音素
  - Phoneme / Mel Encoder + Mel Decoder
    - Embedding Size, Hidden Size 都是 256
    - Multi-head: 2 head
    - Conv1d: filter size 1024, kernel size 9
  - Mel Decoder 输出: 256
  - Linear: 输出 80, target 对应于 80 维的 Mel 特征
- **训练策略**
  - Step 0: 100000 steps, Multi-Speaker TTS 模型训练
  - Step 1: 4000 steps, 调优 FP Predictor
  - Step 2: 4000 steps, Rhythm Adaptation，即调优 MoE Duration Predictor 和 Pitch Predictor
  - Step 3: 2000 steps, Speaker Timbre Adaptation

AdaSpeech 3

对比实验

- GT： 真实的人声
- GT Mel + Vocoder：给出声码器 MelGAN 对 MOS 的影响
- AdaSpeech：没有 FP / Rhythm 的 Adaptation
- AdaSpeech 3: FP + Rhythm /Speaker Adaptation

- Adaptation 的说话人：VCTK 中的 2 男 2 女
- Adaptation 数据量：每人 50 条
- 测试文本： SPON-TIMRE 的 15 条文本
- 评测人数： 20 人
- 评价指标：MOS / SMOS / CMOS

Table 3: *The SMOS results of AdaSpeech 3 and the baseline.*

Setting	SMOS
GT	4.33 ± 0.14
GT mel+Vocoder	4.07 ± 0.14
AdaSpeech	3.45 ± 0.18
AdaSpeech 3	3.75 ± 0.16

Table 4: *The MOS results of AdaSpeech 3 and the baseline in terms of different evaluation aspects.*

Setting	Naturalness	Pause	Speaking Rate
GT	4.14 ± 0.06	4.01 ± 0.06	3.04 ± 0.06
GT mel+Voc	3.84 ± 0.06	3.78 ± 0.06	3.06 ± 0.08
AdaSpeech	3.21 ± 0.06	3.36 ± 0.06	2.66 ± 0.08
AdaSpeech 3	3.45 ± 0.06	3.53 ± 0.06	2.79 ± 0.06

AdaSpeech 3

实验分析

FP prediction 的作用

- 对比实验：随机插入相同数量 FP，和预测的 FP 进行 CMOS 评测。
- FP predictor 的 CMOS 比随机的高 0.156
- 基线有点弱？用文本标点是否可以作为更好的基线

Rhythm Adaptation

- 对比实验一：不用 MoE，只用一个普通的 Duration Predictor
  同样用 SPON-RYHTHM 数据调优
- 对比实验二：不用 SPON-RYHTHM 数据调优 Duration Predictor
- 对比实验三：MoE 基础上，不用 SPON-RYHTHM 数据 Pitch Predictor
- 对比实验四：Pitch/Duration Predictor 都不调优

实验结果：在 FP Predictor 的基础上，Rhythm Adaptation 体现了更大的优势

Table 5: *The CMOS results in the ablation studies of rhythm fine-tuning. w/o MoE: fine-tuning with the single duration predictor instead of MoE; w/o duration adaptation: using the single duration predictor without fine-tuning. w/o pitch adaptation: using the pitch predictor without fine-tuning. w/o pitch/duration adaptation: using the pitch/duration predictor without fine-tuning.*

Setting	with FP	without FP
AdaSpeech 3	/	/
w/o MoE	−0.332	−0.274
w/o duration adaptation	−0.356	−0.289
w/o pitch adaptation	−0.157	−0.112
w/o pitch/duration adaptation	−0.384	−0.308

AdaSpeech 3

实验分析

目标说话人的口语语音合成效果

- Step 0: 100000 steps, Multi-Speaker TTS 模型训练
- Step 1: 4000 steps, 调优 FP Predictor
- Step 2: 4000 steps, Rhythm Adaptation
- **Step 3: 2000 steps, Speaker Timbre Adaptation**

跨说话人的口语风格迁移实验

- Step3 的 Speaker Timbre Adaption 步骤中，使用 VCTK 的 reading-style 的数据做 Adaptation
- 和 AdaSpeech 进行对比，CMOS 分别平均超出了 0.175 (2男) 和 0.205 (2女)
- 验证了跨说话人场景下能够具有更好的口语风格语音合成效果

AdaSpeech 3

总结

Step 1+2：针对自然口语合成效果的改进 (插入语气词、复杂多样的语调和语速)

Step 3： 说话人音色的迁移，不需要口语风格的语音，使用阅读风格的就可以

- Step 0: Multi-Speaker TTS 模型训练
- Step 1: 调优 FP Predictor
- Step 2: Rhythm Adaptation
- Step 3: Speaker Timbre Adaptation

AdaSpeech 3 的优势

- 数据使用更高效，要求更低
  - Base 模型是阅读风格的多说话人 TTS 模型
  - 少量口语风格数据用于 FP Predictor 和 Rhythm Adaptation
- 跨说话人的语音风格迁移
  - 说话人迁移时，不要求目标说话人的口语风格数据，只需要阅读类的语音用于微调说话人音色
- 模型有一定的可控性
  - FP Predictor 可以通过调整阈值来控制插入语气词的稀疏度

AdaSpeech 4: Adaptive Text to Speech in Zero-Shot Scenarios

Yihan Wu, Xu Tan, Bohan Li, Lei He, Sheng Zhao, Ruihua Song, Tao Qin, Tie-Yan Liu

Interspeech 2022

https://arxiv.org/pdf/2204.00436.pdf

AdaSpeech 4

Zero-Shot：没有目标说话人的标注数据，进行说话人适应

通常的 Speaker Adaptation

- 较多数据训练出多说话人的 TTS 模型
- 对于新的目标说话人，使用有标注训练数据微调网络

目标说话人的有标注训练数据很少时，可以分为：

- Few-shot (有标注数据的量级比较小)
  - 9 句 / 20 句，100 句(已经比较多)
- Zero-shot (没有标注数据，只有 reference speech，无标注的参考语音)
- One-shot (只有一条有标注数据)

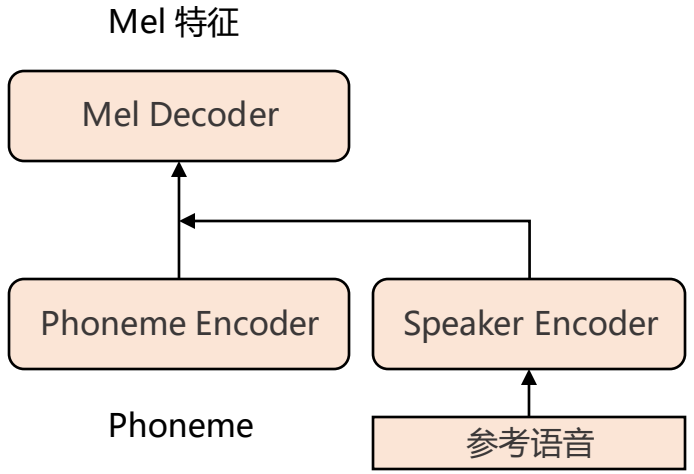
AdaSpeech 4

Zero-Shot 的核心思想

- 提高 Multi-Speaker TTS 模型在说话人层面上的建模能力，增强对新的未见过的说话人的泛化性

一般做法

- 训练多说话人模型时，建模语音的说话人表征 (Speaker/Reference Encoder)
- 模型设计
  - 从参考语音提取说话人表征作为 Condition 输入，捕捉语音和说话人的关系
  - 拼接或者加到 Phoneme Encoder 的输出上，作为 Decoder 的输入信息
- 推理时，对于新的目标说话人，基于参考语音获取说话人表征，不微调网络的前提下达到说话人适应的效果

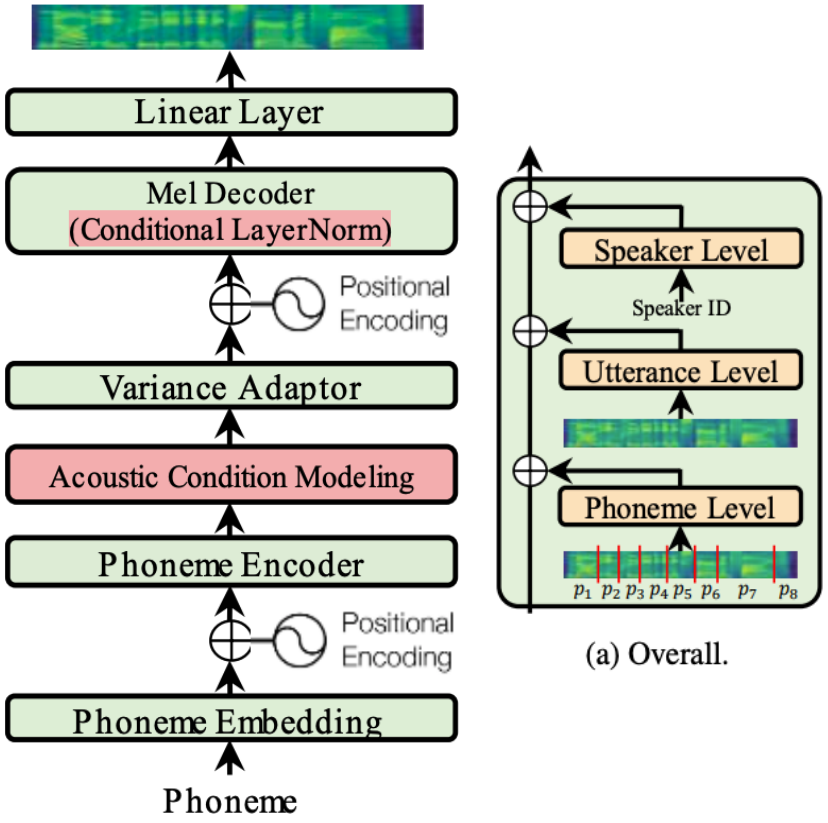


问题一：如何提取泛化能力更强的说话人特征表示?

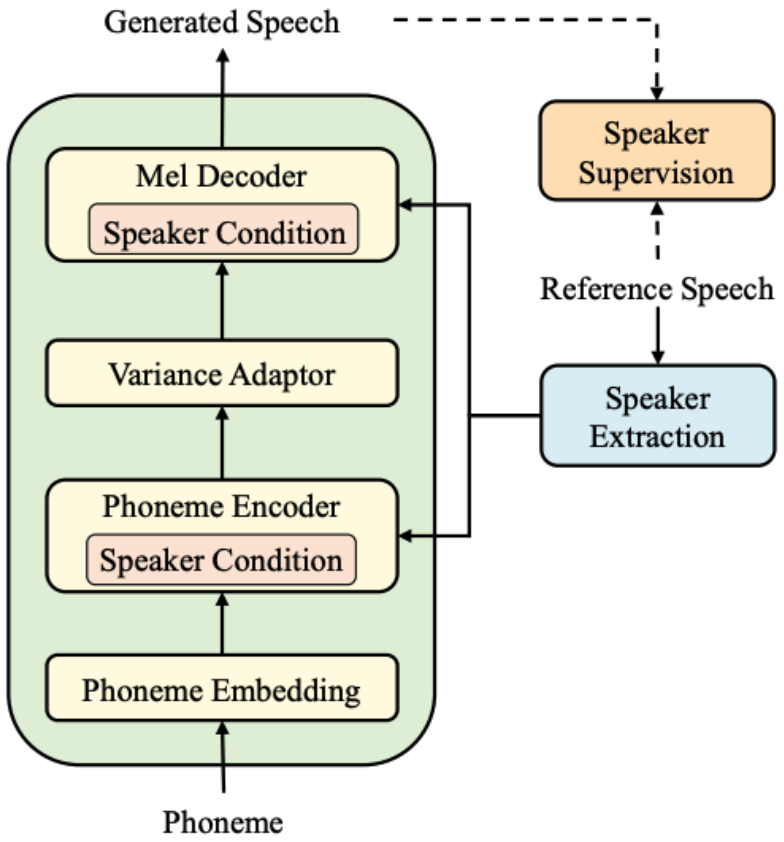
问题二：如何让说话人特征更好地融入到 TTS 模型中?

AdaSpeech 4

模型结构



AdaSpeech



AdaSpeech 4

AdaSpeech 4

模块一：Speaker Extraction

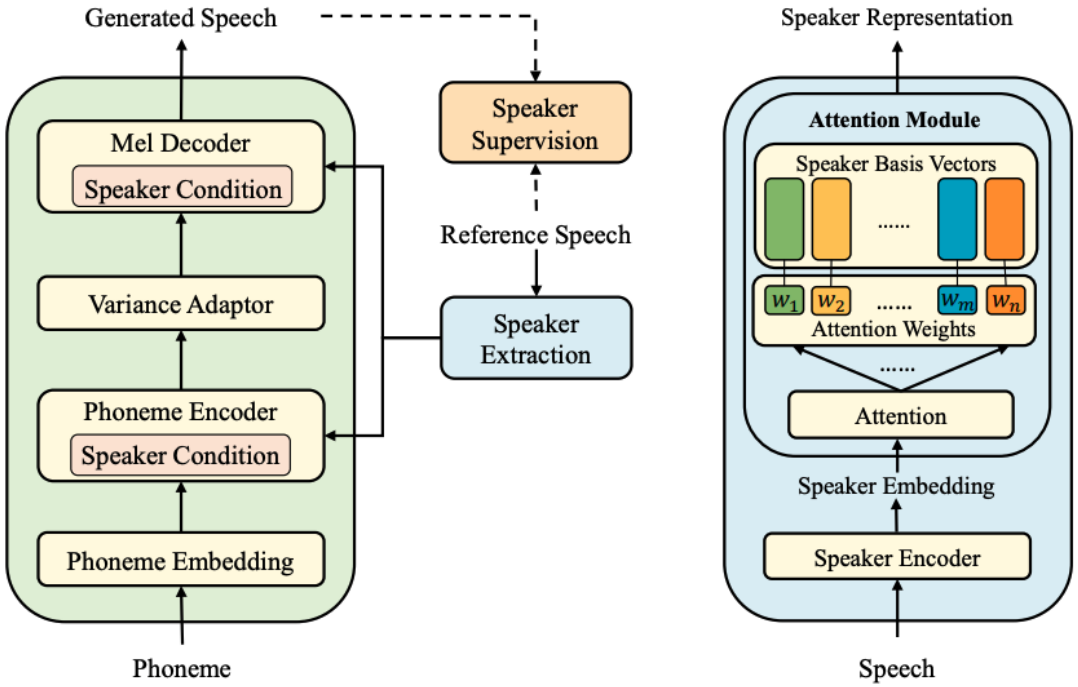
如何提取泛化能力更强的说话人特征表示?

方法：

- 参考 GST (Global Style Token) 的思想，将说话人特征分解为一些基向量的加权和
- 从训练数据中学习基向量，权重使用 Attention 计算

问题一：如何学习更好的基向量？

问题二：如何得到基向量的权重？



(a) AdaSpeech 4　　(b) Speaker Extraction

AdaSpeech 4

模块一：Speaker Extraction

问题一：如何学习更好的基向量？

1. 预训练 Speaker Encoder

训练带有 Speaker Encoder 的多说话人 TTS 模型

2. 提取 Speaker Embedding

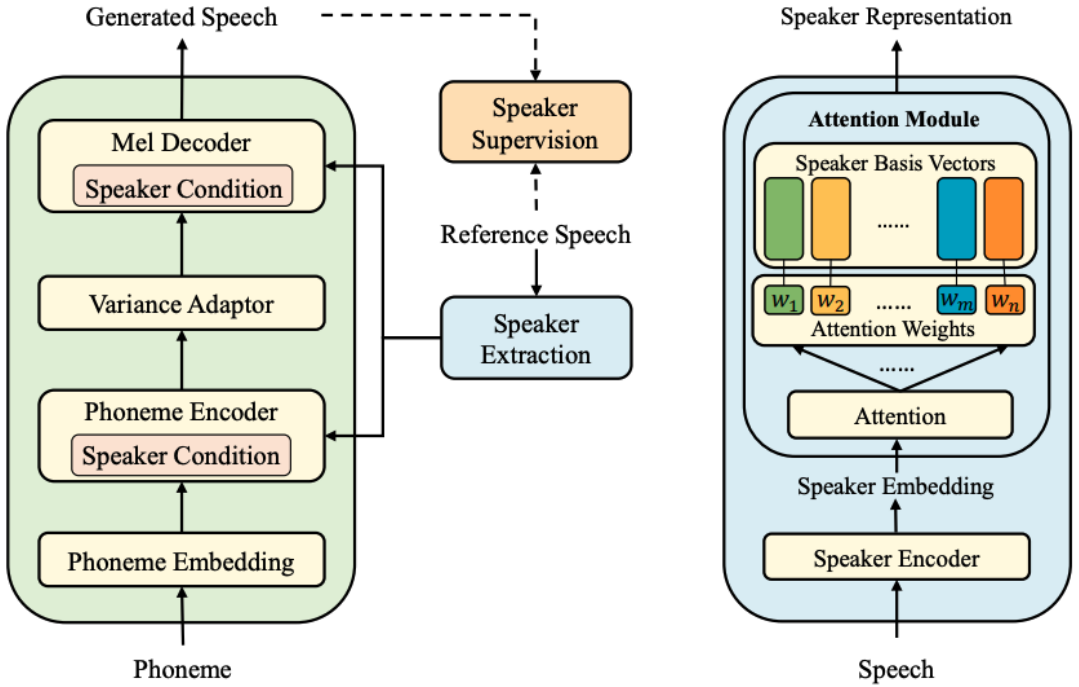
通过 Speaker Encoder 提取全部训练数据的 Speaker Embedding

3. k-means 聚类 Speaker Embedding

- 聚类的个数 N 是超参数，等于基向量的个数
- 将 N 个聚类的中心向量作为基向量初始值，初始化基向量矩阵 B

4. 矩阵 B 训练优化

- 目的：让基向量之间的相似度更低，接近正交基向量
- 针对基向量增加正则化惩罚项，最小化基向量之间的相似度



(a) AdaSpeech 4　　(b) Speaker Extraction

$$\mathcal{L}_{reg} = \frac{1}{N(N-1)} \sum_{i}^{N} \sum_{j,j \neq i}^{N} cos < b_i, b_j >$$

AdaSpeech 4

模块一：Speaker Extraction

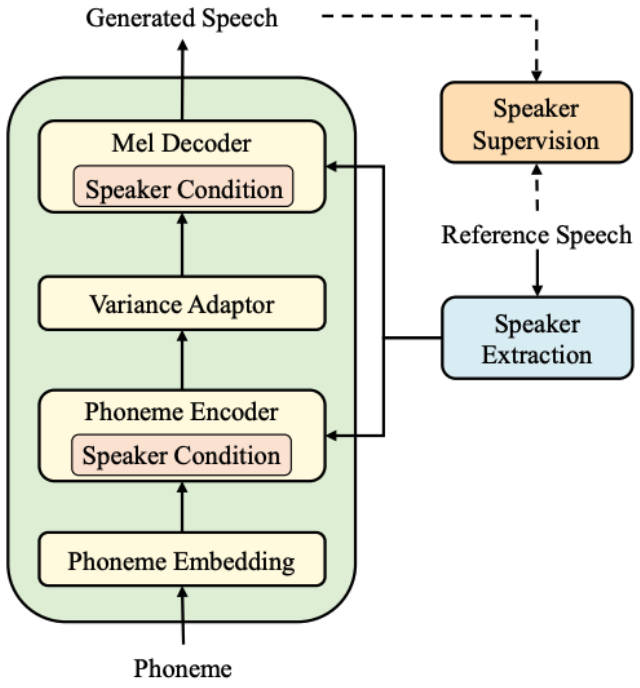
问题二：如何得到基向量的权重？

Speaker Embedding 和基向量 Attention

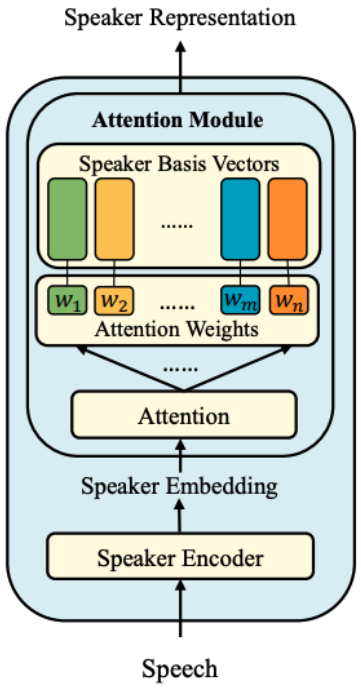
- 用 Speaker Encoder 输出的 Embedding S 作为 Query
- 基向量 B 作为 Key 和 Value
- Attention 之后相当于 Value=B 各基向量的加权和
- 计算得到的 E 即为输出的说话人表征

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V$$

$$E = Attention(SW^Q, BW^K, BW^V)$$



(a) AdaSpeech 4

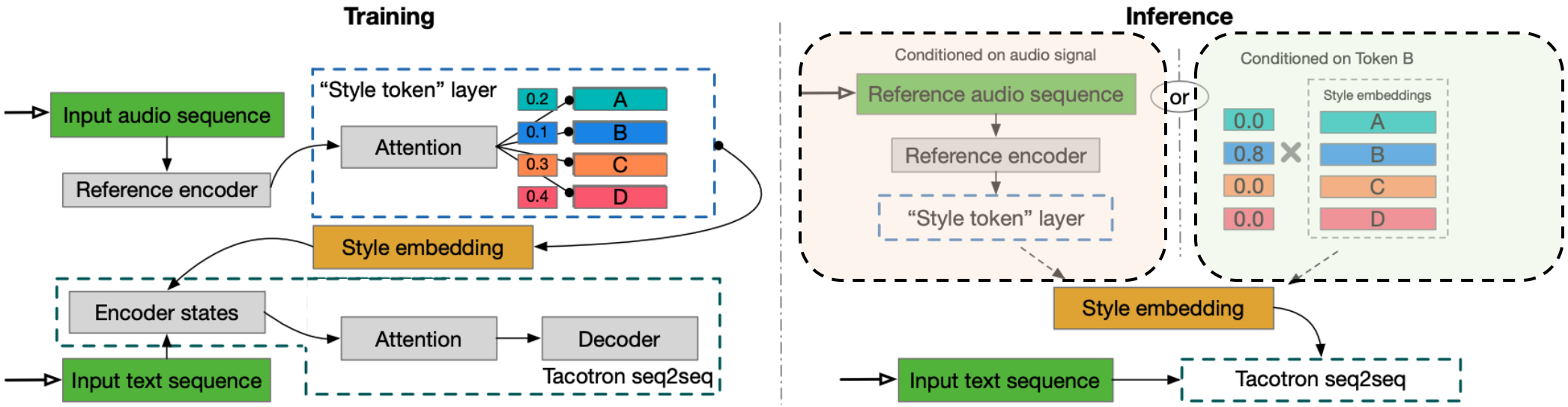


(b) Speaker Extraction

AdaSpeech 4

模块一：Speaker Extraction

和 GST (Global Style Tokens) 的对比



AdaSpeech 4

模块二：Speaker Condition

如何让说话人特征更好地融入到 TTS 模型中?

- **改进一**：从 Reference Speech 提取类似于 GST Style Embedding 的说话人表征，而不再只用一个简单的 Speaker Encoder，送入 CLN
- **改进二**：Phoneme Encoder 也增加了 Speaker Condition 模块，因为 Phoneme Encoder 的输出用于预测 Pitch 等和说话人相关的信息，与说话人相关，所以也可以加入 CLN 模块

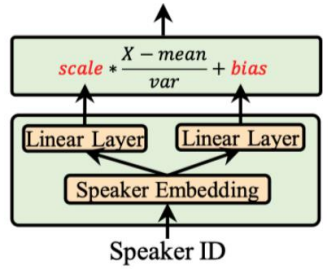
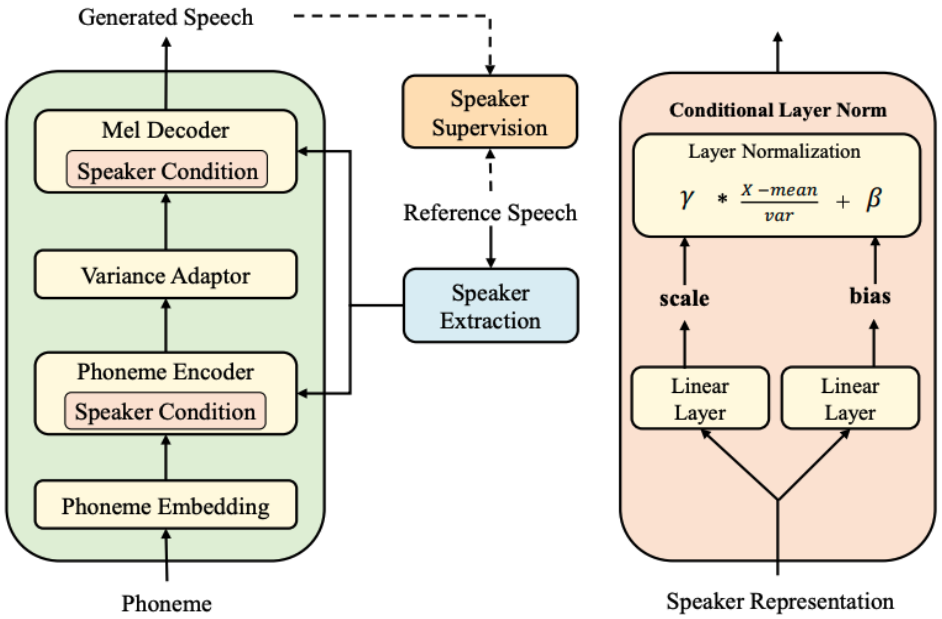


Figure 3: Conditional LayerNorm.

$$\gamma_c^s = E^s * W_c^\gamma, \quad \beta_c^s = E^s * W_c^\beta,$$

- E^s: speaker embedding
- W_c^γ, W_c^β: simple linear layers
- s: speaker ID
- c ∈ [C] denotes there are C CLN in the decoder

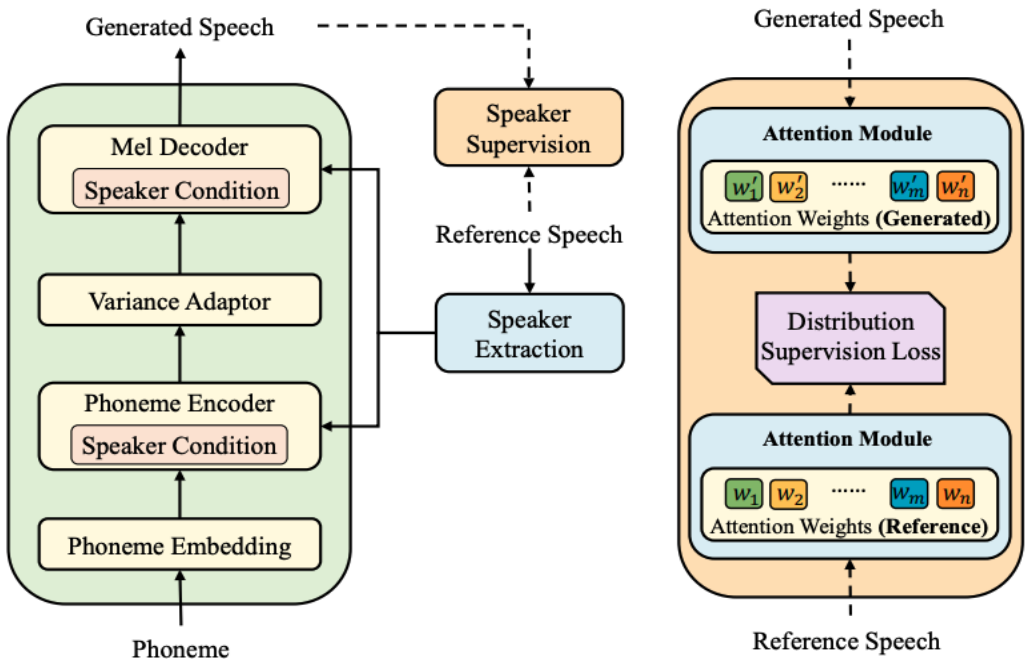


AdaSpeech 4

模块三：Speaker Supervision

强调合成语音和参考语音在说话人层面的相似性

- 训练阶段增加额外的 Speaker Supervision 模块
- 两个语音都经过 Speaker Representation 得到各基向量 b_i 对应的 attention 权重 w_i
- Distribution Loss：用于最大化说话人表征的相似度
- 两个 Attention 权重之间的 KL 散度作为相似度的衡量，希望通过最小化 KL 散度，使得说话人越相似



$$w_i = softmax(\frac{SW^Q(b_iW^K)^T}{\sqrt{d}})$$

$$\mathcal{L}_{dist} = \sum_{i=1}^{N} w_i log \frac{w_i}{w_i'}$$

AdaSpeech 4

完整流程-三阶段

1. 预训练带有 Speaker Encoder 的多说话人 TTS

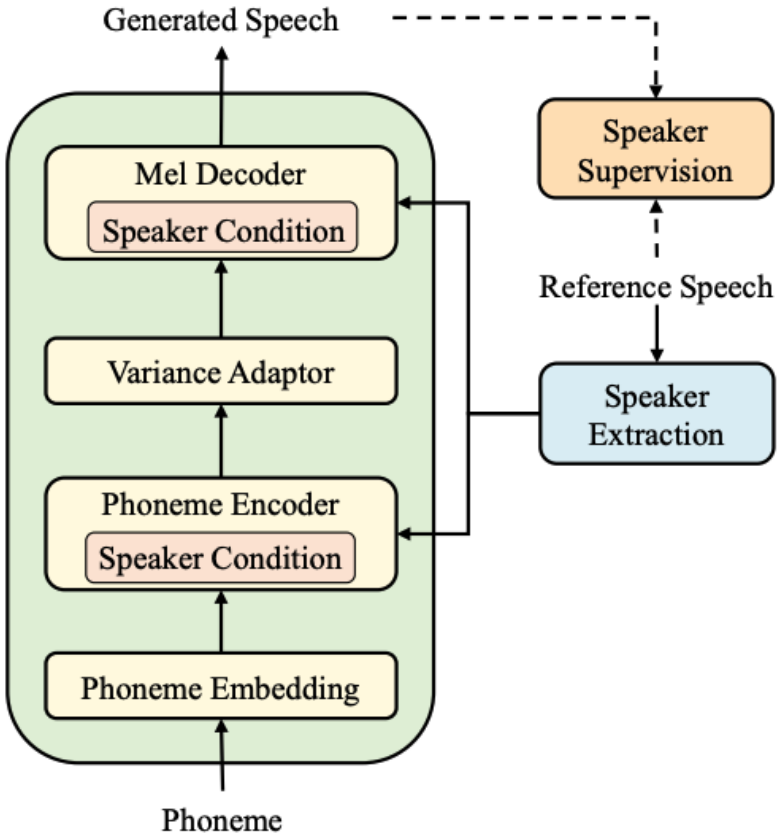
- 对于训练集中的所有语音，提取 Speaker Embedding
- K-means 聚类成 N=2000 个类
- 每个类的中心向量作为 N=2000 个基向量的初始值

2. 增加 Speaker Extraction 和 Speaker Condition 后继续训练，相当于用泛化更好的说话人表征对模型做 Condition 训练

$$\mathcal{L}_{reg} = \frac{1}{N(N-1)} \sum_{i}^{N} \sum_{j,j \neq i}^{N} cos < b_i, b_j >$$

3. 增加 Speaker Supervision，增加 Distribution Loss，用来确保说话人相似度

$$\mathcal{L}_{dist} = \sum_{i=1}^{N} w_i log \frac{w_i}{w_i'}$$



AdaSpeech 4

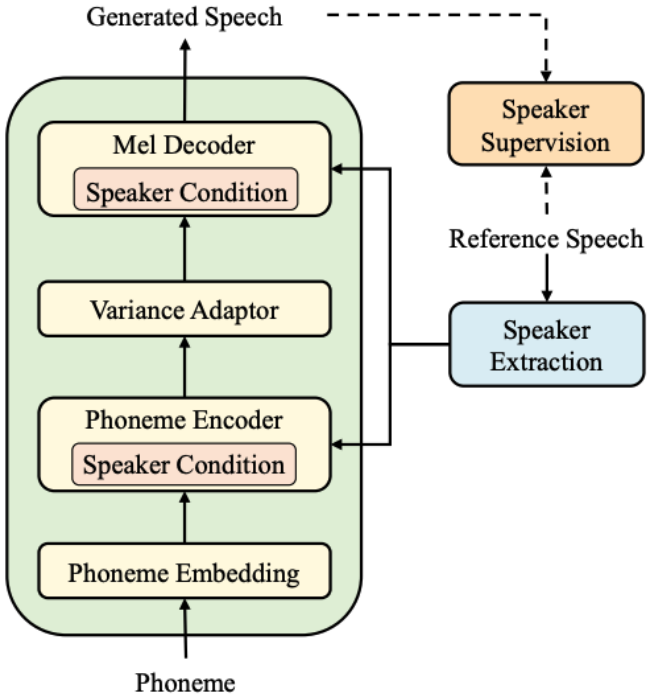
实验信息

- **Source Multi-Speaker TTS**
  - 训练数据：LibriTTS, 586 小时，2456 个说话人
  - 数据集划分：测试集的说话人在训练集里都没见过

- **Zero-Shot Adaptation**
  - LibriTTS-test + VCTK：随机选择 10 个说话人 (5 男 5 女)
  - LJSpeech：1 个说话人
  - 随机从每个说话人的语音中选择一个作为参考语音 (未使用文本输入)

- **模型训练配置**
  - Speech Encoder
    - 6 层 2D 卷积
    - kernel size=(3, 3)，stride=(2, 2), padding="same"
    - ReLU + BatchNorm
    - Output Channel = (32, 32, 64, 64, 128, 128)
  - Speaker Extraction 模块的基向量个数 N = 2000

- 音频采样率：16kHz
- 前端：文本使用 g2p 转成音素
- Phoneme / Mel Encoder + Mel Decoder
  - embedding size, hidden size 都是 256
  - Multi-head: 2 head
  - Conv1d: filter size 1024, kernel size 9
- Mel Decoder 输出: 256
- Linear: 输出 80, target 对应于 80 维的 Mel 特征



AdaSpeech 4

对比实验

- GT： 真实的人声
- GT Mel + Vocoder：给出声码器 HiFi-GAN 对 MOS 的影响
- FastSpeech 2 (vanilla)：Speaker Encoder 的输出作为 Speaker Embedding
- FastSpeech 2 (d-vector)：d-vector 作为 Speaker Embedding
- AdaSpeech (zero shot)：只有 Acoustic Condition Modeling
- 测试：15 条文本　　评测：20 人　　指标：MOS / SMOS / CMOS

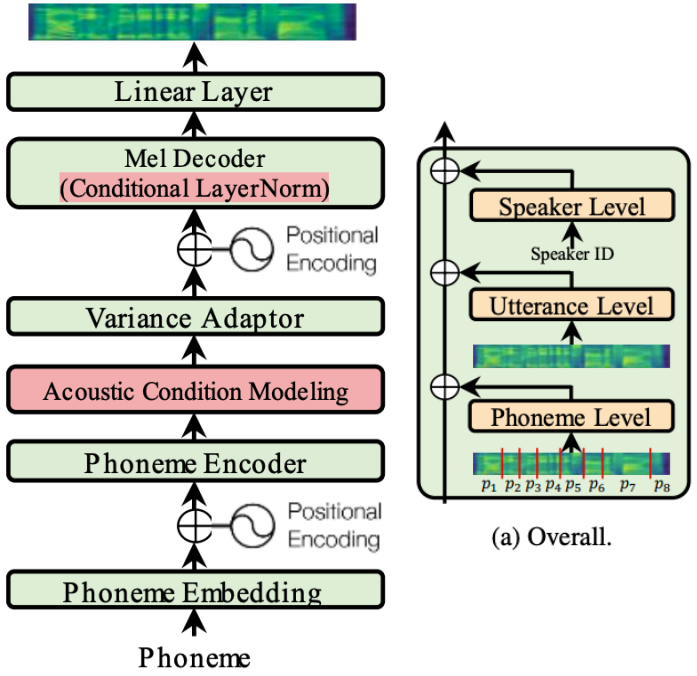


Table 1: *The MOS and SMOS scores with 95% confidence on LibriTTS, VCTK, and LJSpeech.*

Metric	**SMOS** (↑)			**MOS** (↑)		
Dataset	**LibriTTS**	**VCTK**	**LJSpeech**	**LibriTTS**	**VCTK**	**LJSpeech**
GT	4.09 ± 0.11	4.17 ± 0.10	4.08 ± 0.10	3.45 ± 0.12	3.69 ± 0.13	3.67 ± 0.13
GT mel + Vocoder	3.97 ± 0.10	4.13 ± 0.09	4.03 ± 0.09	3.42 ± 0.12	3.68 ± 0.14	3.62 ± 0.12
FastSpeech 2 (vanilla) [5]	3.33 ± 0.13	3.23 ± 0.12	3.26 ± 0.13	3.12 ± 0.14	3.54 ± 0.14	3.16 ± 0.12
FastSpeech 2 (d-vector) [5]	3.12 ± 0.14	2.98 ± 0.12	2.80 ± 0.12	3.08 ± 0.14	2.63 ± 0.13	3.53 ± 0.14
AdaSpeech (zero-shot) [10]	3.62 ± 0.14	3.79 ± 0.10	3.54 ± 0.12	3.22 ± 0.13	3.63 ± 0.14	3.35 ± 0.17
AdaSpeech 4	$\mathbf{3.88 \pm 0.11}$	$\mathbf{3.86 \pm 0.10}$	$\mathbf{3.68 \pm 0.10}$	$\mathbf{3.34 \pm 0.12}$	$\mathbf{3.66 \pm 0.13}$	$\mathbf{3.37 \pm 0.11}$

AdaSpeech 4

消融实验：各模块的作用分析

1. Speaker Extraction

- #2) k-means init：随机初始化，不用聚类中心初始化
  - **对相似性影响不大，对音质影响大**
- #3) 不用基向量的相似度 Loss
- #4) 不用基向量的方案，使用普通的Speaker Embedding

2. Speaker Condition

- #5) Phoneme Encoder 不加 CLN
- #6) Encoder 和 Decoder 都不加 CLN

3. Speaker Supervision：对音质影响不大，对说话人相似性影响大

- #7) 不增加 Distribution Loss 关于说话人的限制
- #8) 替换成 cosine 相似度 Loss
- #9) Distribution 和 Cosine Loss 都加上

实验分析：基向量个数的选择

Table 2: *The SMOS and CMOS scores with 95% confidence on LibriTTS for ablation study.*

Ablation Modules	ID	Settings	SMOS (↑)	CMOS (↑)
	#1	AdaSpeech 4	**3.88 ± 0.11**	**0**
Speaker Extraction	#2	#1 − k-means init	3.84 ± 0.11	− 0.36
	#3	#1 − $\mathcal{L}_{reg}$	3.67 ± 0.11	− 0.46
	#4	#1 − basis vectors	3.62 ± 0.12	− 0.17
Speaker Condition	#5	#1 − encoder CLN	3.72 ± 0.12	− 0.21
	#6	#5 − decoder CLN	3.70 ± 0.11	− 0.36
Speaker Supervision	#7	#1 − $\mathcal{L}_{dist}$	3.64 ± 0.12	− 0.06
	#8	#7 + $\mathcal{L}_{cos}$	3.69 ± 0.13	− 0.08
	#9	#1 + $\mathcal{L}_{cos}$	3.71 ± 0.12	− 0.06

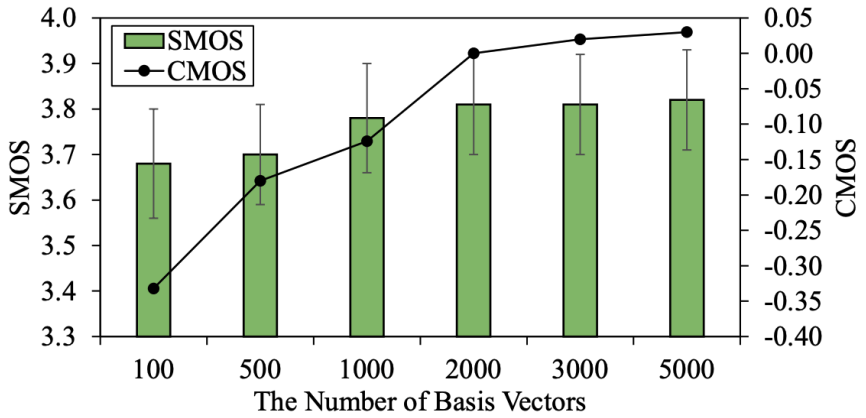


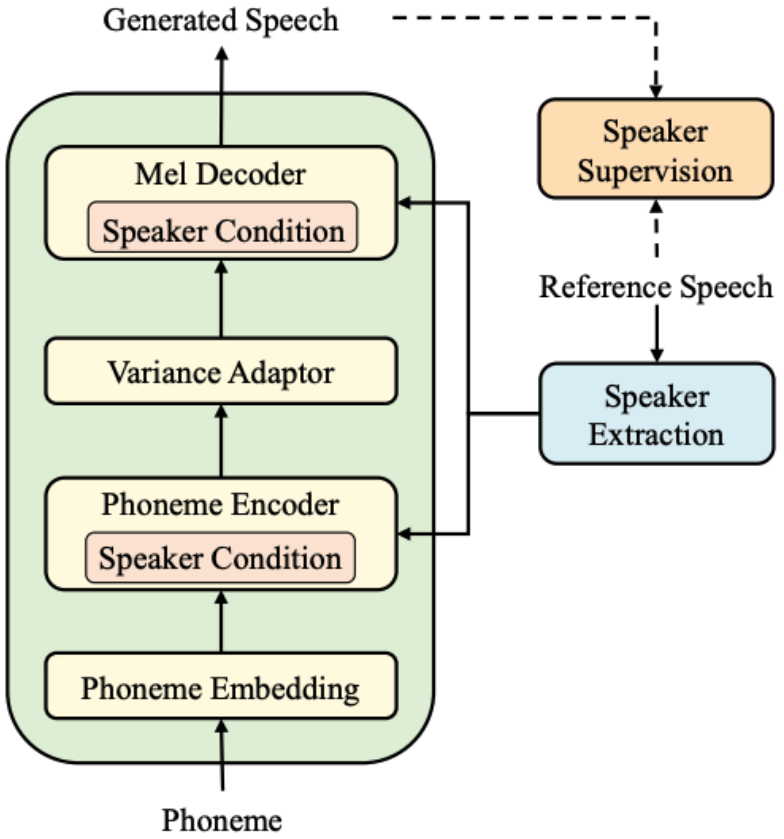
Figure 2: *The SMOS and CMOS of different number of basis vectors on LibriTTS test set.*

AdaSpeech 4

总结

- **如何提取泛化能力更强的说话人特征表示?**
  - GST 的思想学习说话人表征
    - 训练基向量，利用基向量的加权组合
    - Attention Weight 作为基向量的权重
  - Distribution Loss，提高合成语音的 Speaker Embedding 相似度

- **如何让说话人特征更好地融入到 TTS 模型中?**
  - 相比于 AdaSpeech 更增强了模型中说话人信息的注入
    - Mel Decoder CLN+ Phoneme Encoder CLN



Conclusion

Adaptive TTS

目标说话人数据标注	数据量	方案
有标注	充分	Multi-Speaker FastSpeech 2
有标注	较少	AdaSpeech (现有风格的 Adaptation) AdaSpeech 3 (跨风格的 Adaptation)
无标注	充分	有 ASR 模型：AdaSpeech 2 – PPG Encoder 无 ASR 模型：AdaSpeech 2 – Mel Encoder
无标注	1 条	AdaSpeech 4

模型设计的启发

- 训练策略上：多阶段 (训练 + Adapt) 组合
- 网络结构上：CLN / MoE (ASR) / GST
- 损失函数上：L2 Alignment Loss / Distribution Loss

Thanks

Q & A

References

FastSpeech	Ren, Yi, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. "Fastspeech: Fast, robust and controllable text to speech." Advances in Neural Information Processing Systems 32 (2019).
FastSpeech 2	Ren, Yi, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. "Fastspeech 2: Fast and high-quality end-to-end text to speech." arXiv preprint arXiv:2006.04558 (2020).
AdaSpeech	Chen, Mingjian, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, and Tie-Yan Liu. "AdaSpeech: Adaptive Text to Speech for Custom Voice." In International Conference on Learning Representations. 2020.
AdaSpeech 2	Yan, Yuzi, Xu Tan, Bohan Li, Tao Qin, Sheng Zhao, Yuan Shen, and Tie-Yan Liu. "Adaspeech 2: Adaptive text to speech with untranscribed data." In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6613-6617. IEEE, 2021.
AdaSpeech 3	Yan, Yuzi, Xu Tan, Bohan Li, Guangyan Zhang, Tao Qin, Sheng Zhao, Yuan Shen, Wei-Qiang Zhang, and Tie-Yan Liu. "Adaspeech 3: Adaptive text to speech for spontaneous style." arXiv preprint arXiv:2107.02530 (2021).
AdaSpeech 4	Wu, Yihan, Xu Tan, Bohan Li, Lei He, Sheng Zhao, Ruihua Song, Tao Qin, and Tie-Yan Liu. "AdaSpeech 4: Adaptive Text to Speech in Zero-Shot Scenarios." arXiv preprint arXiv:2204.00436 (2022).
NaturalSpeech	Tan, Xu, Jiawei Chen, Haohe Liu, Jian Cong, Chen Zhang, Yanqing Liu, Xi Wang et al. "NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality." arXiv preprint arXiv:2205.04421 (2022).
GST	Wang, Yuxuan, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A. Saurous. "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis." In International Conference on Machine Learning, pp. 5180-5189. PMLR, 2018.
3M-ASR	You, Zhao, Shulin Feng, Dan Su, and Dong Yu. "3M: Multi-loss, Multi-path and Multi-level Neural Networks for speech recognition." arXiv preprint arXiv:2204.03178 (2022).