MegaTTS 3: Sparse Alignment Enhanced Latent Diffusion Transformer for Zero-Shot Speech Synthesis

Ziyue Jiang[1,2]*, Yi Ren[2], Ruiqi Li[1,2], Shengpeng Ji[1], Boyang Zhang[1], Zhenhui Ye[1], Chen Zhang[2], Bai Jionghao[1], Xiaoda Yang[1], Jialong Zuo[1], Yu Zhang[1], Rui Liu[3], Xiang Yin[2], Zhou Zhao[1]

[1]Zhejiang University, [2]ByteDance, [3]Inner Mongolia University

ziyuejiang341@gmail.com, zhaozhou@zju.edu.cn

2025-04-14

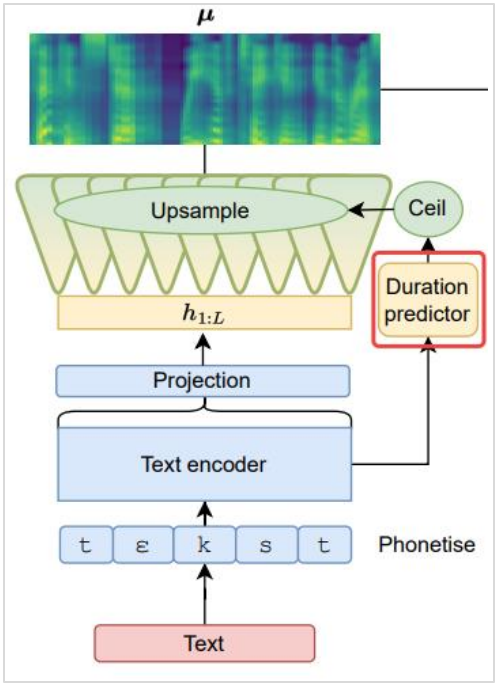
Table of Contents

Background: Explicit v.s. Implicit Alignment Modeling

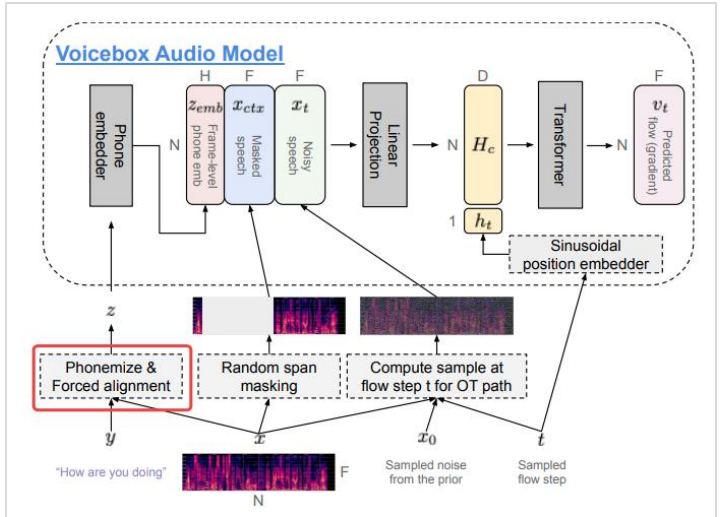
TTS 难点：发音稳健性 v.s. 韵律表现力

显式对齐建模 (duration) ⟸ Matcha-TTS, VoiceBox, NaturalSpeech2, VoiceFlow

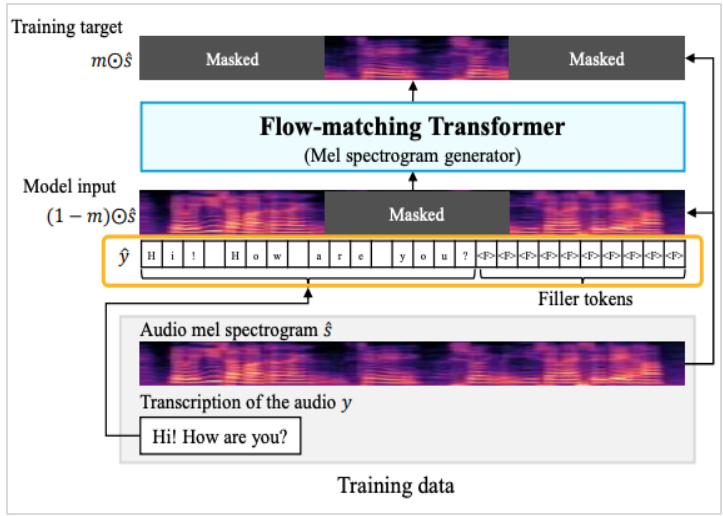
E2-TTS, F5-TTS, $\text{Seed} - \text{TTS}_{\text{DiT}}$, DiTTo-TTS, (LLM Based Models) ⟹ **隐式**对齐建模 (duration)



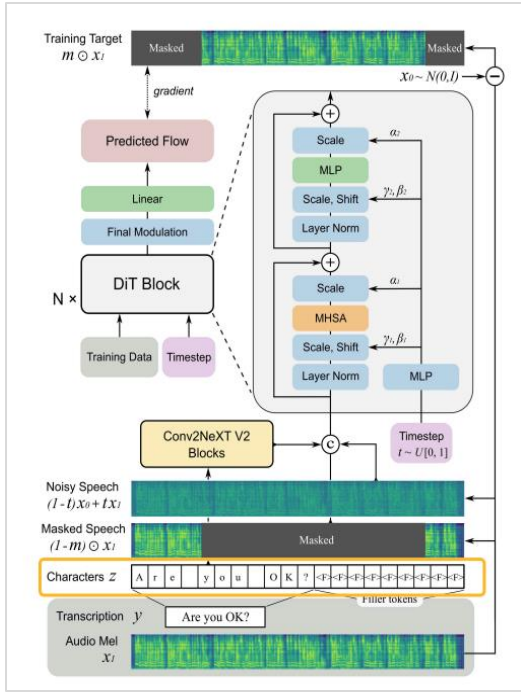
Matcha-TTS



VoiceBox



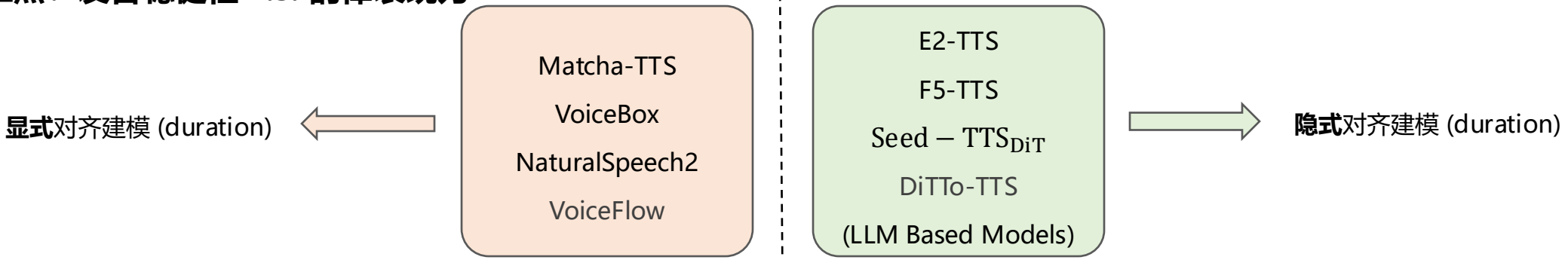
E2-TTS



F5-TTS

Background: Explicit v.s. Implicit Alignment Modeling

TTS 难点：发音稳健性 v.s. 韵律表现力



优点：

- 直接提供文本与语音间的**帧级别映射**信息，降低模型的学习难度
- 合成语音时，严格使用预测的对齐结果，通常发音更稳定（丢字/漏字少）

不足：

- 限制了模型的建模搜索空间，影响合成语音的自然程度
- 对齐结果**不完全可靠**，作为帧级别的绝对学习目标，和真实语音存在偏差

优点：

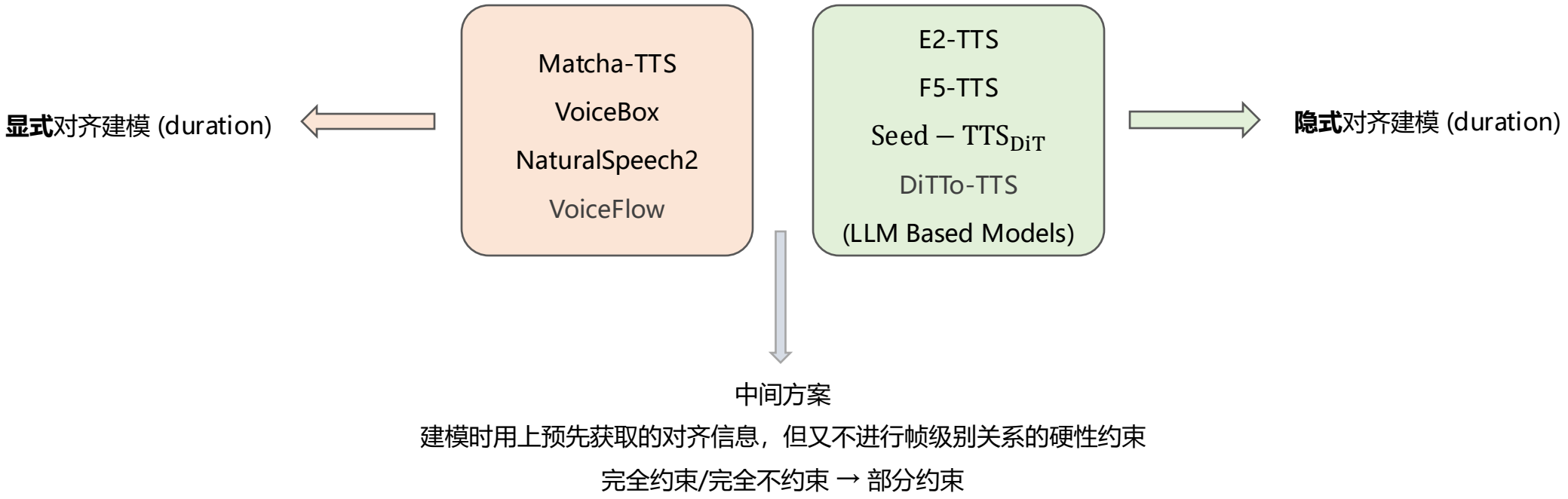
- 没有对模型搜索空间进行先验约束，模型效果的理论上限更高
- 通常具有**更强的韵律表现**，合成结果的自然度/多样性更高

不足：

- 模型自发隐式学习对齐(attention)，任务难度大，难度大的文本**合成准确性差**
- 不能控制字级别/个别发音的长度，只能调整整个语音的长度

Background: Explicit v.s. Implicit Alignment Modeling

TTS 难点：发音稳健性 v.s. 韵律表现力



Overview: Model Architecture

WaveVAE + DiT + Rectified-Flow

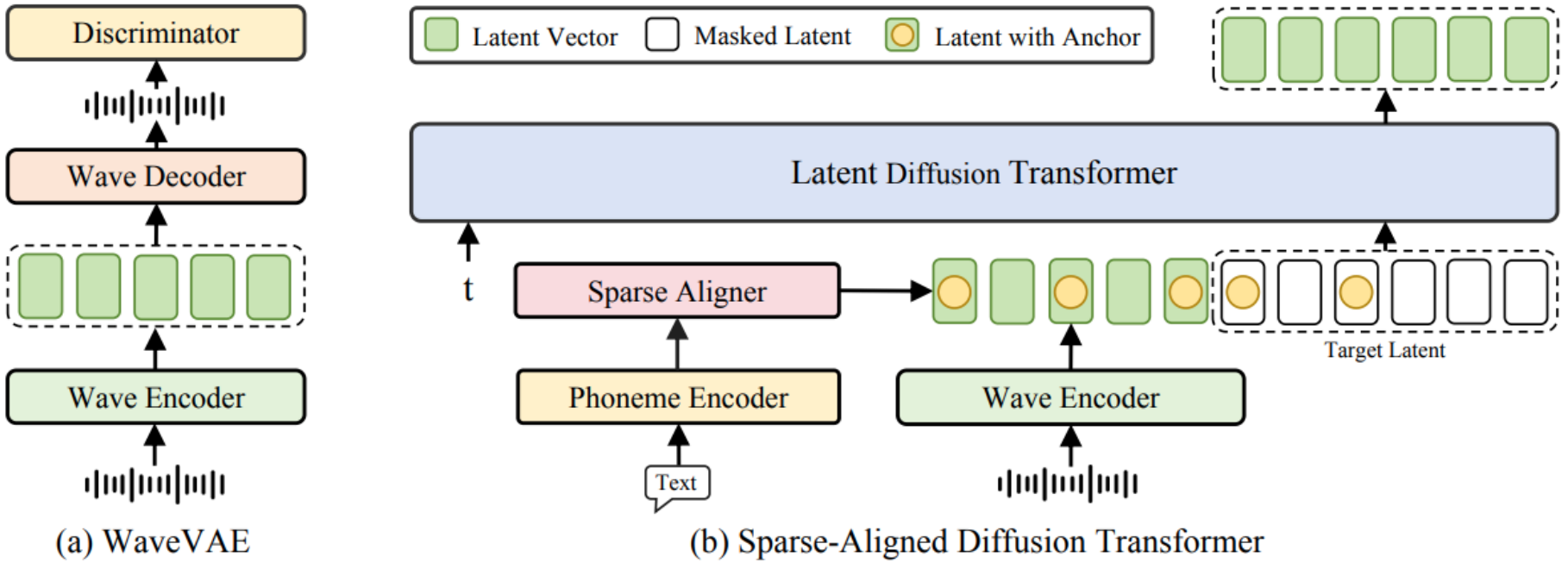


Figure 1: (a) The WaveVAE model; (b) Overview of our model. We insert the sparse alignment anchors into the latent vector sequence to provide coarse alignment information. The transformer blocks in MegaTTS 3 will automatically build fine-grained alignment paths.

WaveVAE: Continuous Speech Tokens

核心思路

- Encoder 将语音编码为帧率更低的 Latent 表征（**25Hz**）
- Decoder 将 Latent 表征直接还原到波形（不是梅尔特征）

$$\hat{s} = D(z) = D(E(s))$$

模型结构

- Encoder 下采样：16 kHz 下采样 640 倍，降低到 25Hz（Encodec/WaveTokenizer）
- Decoder 上采样：25Hz 上采样到 16kHz（HiFiGAN 结构）

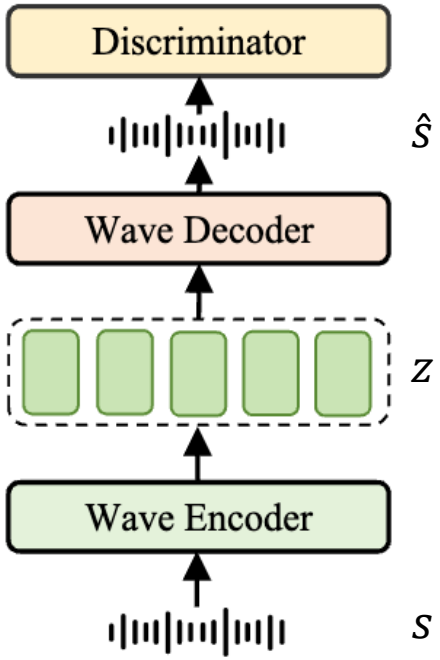
判别器

- Encoder+Decoder 一起作为 Generator，增加 MPD + MSD + MRD 判别器

训练目标

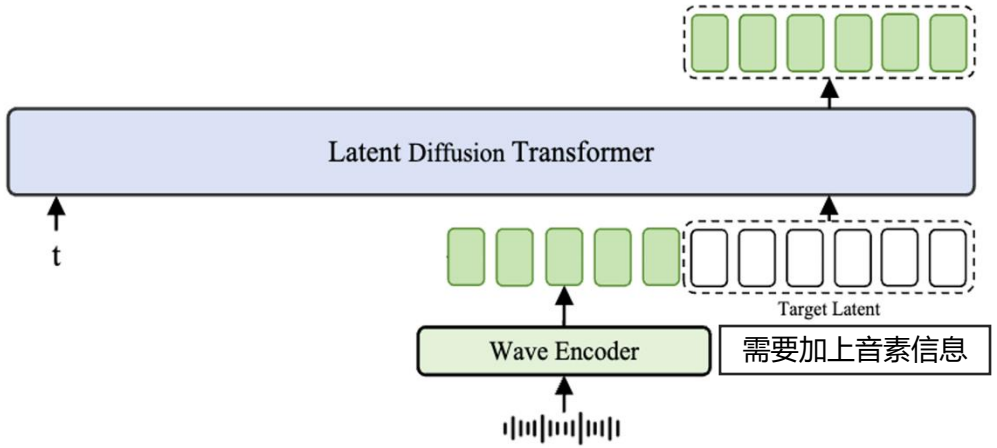
$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{Adv}}$$

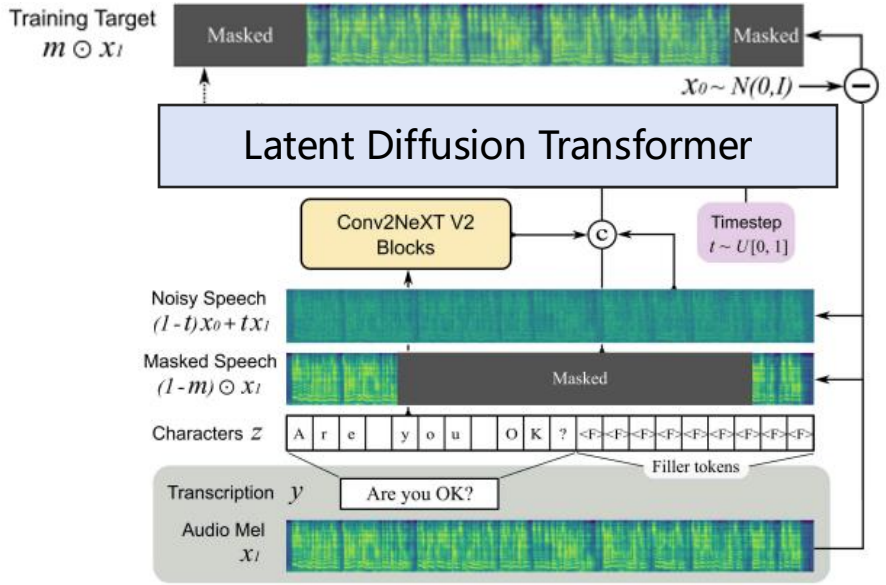
- 重建 Loss：实际是梅尔重建 L2 Loss + SSIM Loss
- 正则 Loss：Latens z 和标准高斯分布之间的 KL 散度 Loss（KL loss 系数很小: 1e-3）
- Adv Loss：LSGAN Loss



(a) WaveVAE

DiT: Latent Diffusion Transformer with Masked Speech Modeling





训练策略

- Latent 序列随机拆分：prompt (10%~90%) 和 target
- 完整音素信息 + prompt 部分 → 预测被 mask 的 target
  - 音素信息：学习平均发音
  - prompt：学习音色/口音/韵律其他信息

模型结构

- Llama + RoPE：参数量 339M（非自回归不用因果 Mask）
- 不使用额外的 Phoneme Encoder

Rectified Flow: Flow Straight and Fast

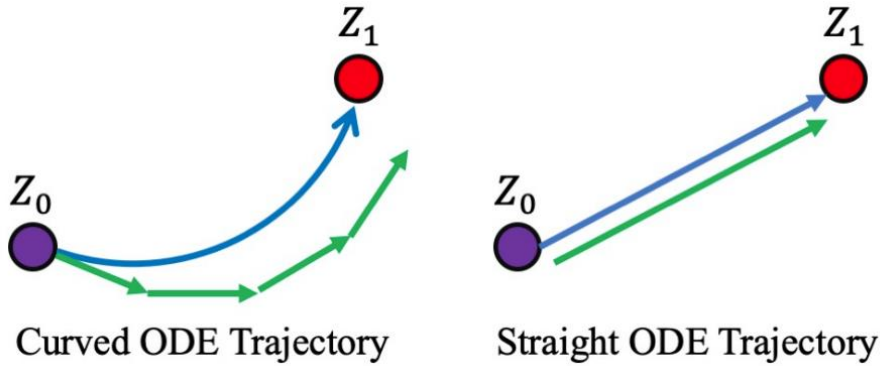
$\boldsymbol{X}_0 \sim \pi_0 \quad \mathcal{N}(\mathbf{0}, \mathbf{I})$ 初始分布：高斯分布，采样简单

$\boldsymbol{X}_1 \sim \pi_1$ 目标数据分布

$\boldsymbol{X}_t = t\boldsymbol{X}_1 + (1-t)\boldsymbol{X}_0$ 加噪条件：初始分布和目标分布之间线性插值

$$\frac{d}{dt}\boldsymbol{Z}_t = \boldsymbol{v}(\boldsymbol{Z}_t, t), \quad \forall t \in [0,1], \quad \boldsymbol{Z}_0 = \boldsymbol{X}_0$$ velocity field

$$\min_v \int_0^1 \mathbb{E}\left[\left\|(X_1 - X_0) - v(X_t,\ t)\right\|^2\right] \mathrm{d}t$$ 训练目标



Algorithm 1 Rectified Flow: Main Algorithm

Procedure: $\boldsymbol{Z} = \texttt{RectFlow}((X_0, X_1))$:

NN

Inputs: Draws from a coupling (X_0, X_1) of π_0 and π_1; velocity model v_θ: $\mathbb{R}^d \to \mathbb{R}^d$ with parameter θ.

Training: $\hat{\theta} = \arg\min_\theta \mathbb{E}\left[\|X_1 - X_0 - v(tX_1 + (1-t)X_0,\ t)\|^2\right]$, with $t \sim \text{Uniform}([0,1])$.

Sampling: Draw (Z_0, Z_1) following $\mathrm{d}Z_t = v_{\hat{\theta}}(Z_t, t)\mathrm{d}t$ starting from $Z_0 \sim \pi_0$ (or backwardly $Z_1 \sim \pi_1$).

Return: $\boldsymbol{Z} = \{Z_t : t \in [0,1]\}$.

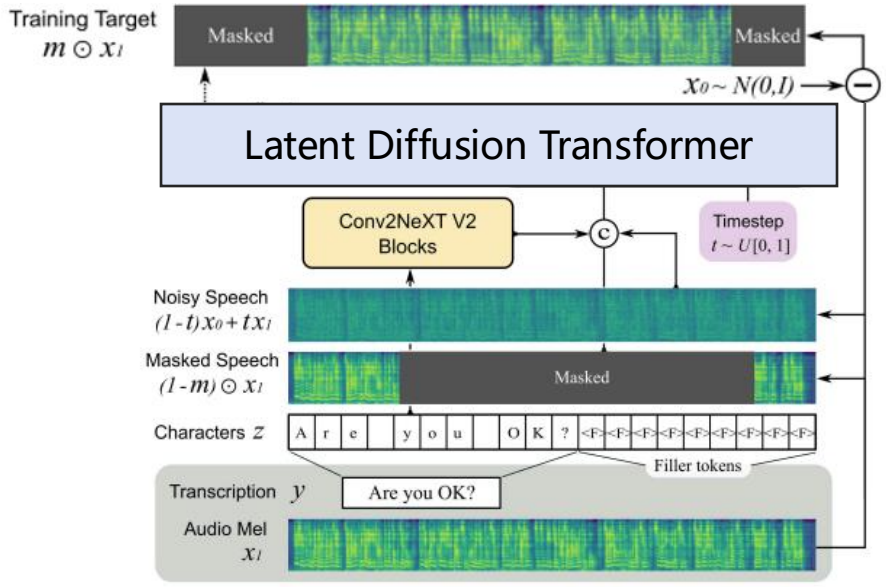
Reflow (optional): $\boldsymbol{Z}^{k+1} = \texttt{RectFlow}((Z_0^k, Z_1^k))$, starting from $(Z_0^0, Z_1^0) = (X_0, X_1)$.

Distill (optional): Learn a neural network $\hat{T}$ to distill the k-rectified flow, such that $Z_1^k \approx \hat{T}(Z_0^k)$.

1-rectified
↓
k-rectified
↓
最后一步到位

Rectified Flow: Flow Straight and Fast

MegaTTS3: Rectified Flow



$$\boldsymbol{X}_t = t\boldsymbol{X}_1 + (1-t)\boldsymbol{X}_0$$

$$\min_v \int_0^1 \mathbb{E}\left[\left\|(X_1 - X_0) - v(X_t,\ t)\right\|^2\right] \mathrm{d}t$$

```python

# Here, x is x1 in CFM
x0 = torch.randn_like(x)# 随机采样
x = inputs['lat'] # 目标 VAE latents.
t, x_noisy, target = self,flow_matcher.sample_location_and_conditional_flow(x0, x)

def sample_location_and_conditional_flow(self, x0, x1):
    # Compute the sample xt(drawn from N(t *x1 +(1- t)* x0, sigma))
    # and the conditional vector field ut(x1|x0)= x1 - x0, see Eq.(15)[1].
    bs = x0.shape[0]
    t = torch.distributions.LogisticNormal,sample([bs])[..., 0].type_as(x0)
    xt = t * x1 + (1 - t) * x0
    ut = x1 - x0
    return t, xt, ut

# speaker prompt + mask
ctx_feature = inputs['lat_ctx']
ctx_mask_emb = self.ctx_mask_proj(inputs['ctx_mask'])
local_cond = torch.cat([ctx_feature, ctx_mask_emb], dim=-1)
local_cond = self.local_cond_project(local_cond)

# define noisy input and target
x_ling = self.forward_ling_encoder(inputs["phone"], inputs["tone"])
x_noisy = self.x_prenet(x_noisy) + self.prenet(local_cond) + x_ling
encoder_out = self.encoder(x_noisy, self.f5_time_embed(t),
                           attn_mask=inputs["text_mel_mask"])
pred = self.postnet(encoder_out)

return pred, target

```

Rectified Flow: Flow Straight and Fast

MegaTTS3: Rectified Flow

$$\boldsymbol{X}_t = t\boldsymbol{X}_1 + (1-t)\boldsymbol{X}_0$$

$$\min_v \int_0^1 \mathbb{E}\left[\left\|(X_1 - X_0) - v(X_t,\, t)\right\|^2\right] \mathrm{d}t$$

CosyVoice: OT(Optimal Transport)-CFM

$$\psi_t(x) = (1 - (1 - \sigma_{\min})t)x + tx_1$$

$$\psi_t(x_0) = (1 - (1 - \sigma_{min})t)x_0 + tx_1$$

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)}\left\|v_t(\psi_t(x_0)) - \left(x_1 - (1-\sigma_{\min})x_0\right)\right\|^2$$

$$v_t(x_t, t)$$

Rectified Flow: Flow Straight and Fast

CosyVoice: OT(Optimal Transport)-CFM

$$\psi_t(x) = (1 - (1 - \sigma_{\min})t)x + tx_1$$

$$\psi_t(x_0) = (1 - (1 - \sigma_{min})t)x_0 + tx_1$$

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)} \left\| v_t(\psi_t(x_0)) - \left(x_1 - (1 - \sigma_{\min})x_0\right) \right\|^2$$

$v_t(x_t, t)$

```python
def compute_loss(self, x1, mask, mu, spks=None, cond=None):
    b, _, t = mu.shape

    # random timestep
    t = torch.rand([b, 1, 1], device=mu.device, dtype=mu.dtype)
    if self.t_scheduler == 'cosine':
        t = 1 - torch.cos(t * 0.5 * torch.pi)
    # sample noise p(x_0)
    z = torch.randn_like(x1)

    y = (1 - (1 - self.sigma_min) * t) * z + t * x1
    u = x1 - (1 - self.sigma_min) * z

    # during training, we randomly drop condition to
    # trade off mode coverage and sample fidelity
    if self.training_cfg_rate > 0:
        cfg_mask = torch.rand(b) > self.training_cfg_rate
        mu = mu * cfg_mask.view(-1, 1, 1)
        spks = spks * cfg_mask.view(-1, 1)
        cond = cond * cfg_mask.view(-1, 1, 1)

    pred = self.estimator(y, mask, mu, t.squeeze(), spks, cond)
    loss = F.mse_loss(pred*mask, u*mask, reduction="sum") / \
                    (torch.sum(mask) * u.shape[1])
    return loss, y
```

Rectified Flow: Flow Straight and Fast

Euler 采样（ODE 常微分方程求解）

$$\tilde{\boldsymbol{Z}}_{t_{k+1}} = \tilde{\boldsymbol{Z}}_{t_k} + (t_{k+1} - t_k)\boldsymbol{v}(\tilde{\boldsymbol{Z}}_{t_k}, t_k), \quad \tilde{\boldsymbol{Z}}_0 \sim \pi_0$$

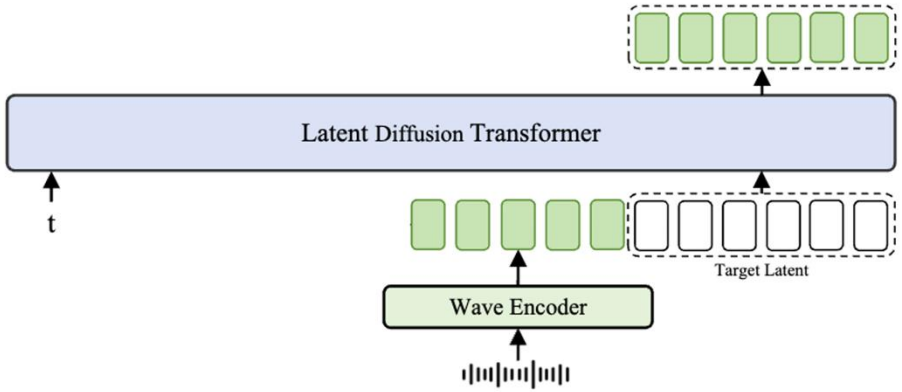
$$t_0 = 0 < t_1 < \cdots < t_N = 1$$

```
z = torch.randn_like(mu) * temperature
t_span = torch.linspace(0, 1, n_timesteps + 1, device=mu.device, dtype=mu.dtype)
if self.t_scheduler == 'cosine':
    t_span = 1 - torch.cos(t_span * 0.5 * torch.pi)
return self.solve_euler(z, t_span=t_span, mu=mu, mask=mask, spks=spks, cond=cond)

def solve_euler(self, x, t_span, mu, mask, spks, cond):
    t, _, dt = t_span[0], t_span[-1], t_span[1] - t_span[0]
    t = t.unsqueeze(dim=0)

    sol = []

    for step in range(1, len(t_span)):
        dphi_dt = self.forward_estimator(x, mask, mu, t, spks, cond)
        # Classifier-Free Guidance inference introduced in VoiceBox
        if self.inference_cfg_rate > 0:
            cfg_dphi_dt = self.forward_estimator(
                x, mask,
                torch.zeros_like(mu), t,
                torch.zeros_like(spks) if spks is not None else None,
                torch.zeros_like(cond)
            )
            dphi_dt = ((1.0 + self.inference_cfg_rate) * dphi_dt -
                       self.inference_cfg_rate * cfg_dphi_dt)
        x = x + dt * dphi_dt
        t = t + dt
        sol.append(x)
        if step < len(t_span) - 1:
            dt = t_span[step + 1] - t

    return sol[-1]
```

Modification 1: Sparse Alignment Strategy (**Robustness**)

模型训练

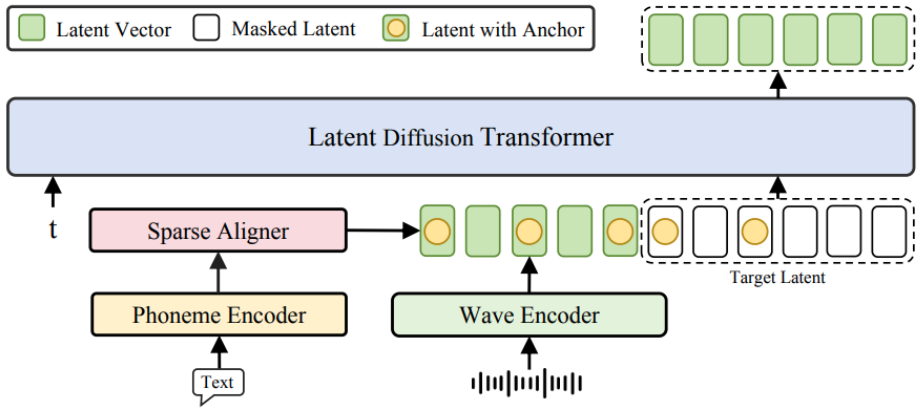
- MFA 强制对齐，得到训练集每个音素的 duration，即时间范围（帧的范围）
- 每个音素的时间范围内随机只选择 1 帧，加入 phoneme embedding，其他位置不加
  - 只给出关键帧（锚点 anchor）的音素对应关系，自主学习其他帧的对齐关系
- Masked Latent: latent 被 mask，但仍会输入高斯 noise



模型推理

- Prompt：文本 + 语音获取 duration
- Target：基于 AR Duration Predictor 预测文本的 duration

} 类似 megatts2



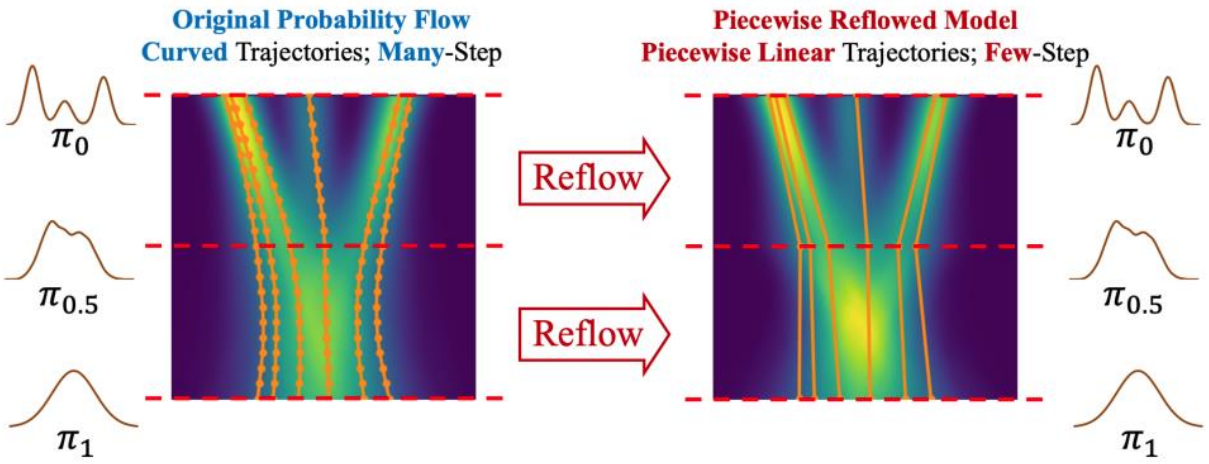
设计细节

- 25Hz 帧率：音素的时间粒度有点粗（部分音素 duration 可能较小，比如 20-30ms）
- 100Hz 帧率的 phoneme 序列，卷积下采样后，再与 vae latents 进行 concat

Modification 2: Piecewise Rectified Flow (**Acceleration**)

PeRFlow (ByteDance)

- 如何进一步降低 Rectified Flow 的采样次数?
- 采样过程在时间 [0, 1] 上拆分为 K 个时间窗口（K 数值较小，时间划分粒度更粗）
- 随机选择其中一个时间窗进行 Reflow
- 模型训练配置
  - Rectified Flow 训练 800k steps
  - PeRFlow 训练 200k steps



Algorithm 1: Piecewise Rectified Flow

Input: Training dataset $\mathcal{D}$, ϵ- or v-prediction teacher model f_ϕ, Noise schedule $\sigma(t)$, ODE solver $\Phi(z_t, t, s, f_\phi)$, Number of windows K, student model ϵ_θ or v_θ,

Create K time windows $\{(t_{k-1}, t_k]\}_{k=1}^K$ with $t_K = 1$ and $t_0 = 0$;
Initialize $\theta = \phi$;
repeat
  Sample $z_0 \sim \mathcal{D}$;
  Sample k from $\{1, \cdots, K\}$ uniformly, then randomly sample time $t \in (t_{k-1}, t_k]$;
  Sample random noise $\epsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$;
  Get $\boldsymbol{z}_{t_k} = \sqrt{1-\sigma^2(t_k)}\boldsymbol{z}_0 + \sigma(t_k)\epsilon$;
  Solve the endpoint of the time window $z_{t_{k-1}} = \Phi(\boldsymbol{z}_{t_k}, t_k, t_{k-1})$;
  Get $\boldsymbol{z}_t = \boldsymbol{z}_{t_k} + \frac{\boldsymbol{z}_{t_k} - \boldsymbol{z}_{t_{k-1}}}{t_k - t_{k-1}}(t - t_k)$;
  if *ϵ-prediction* **then**
    Compute loss $\ell = \left\| \epsilon_\theta(\boldsymbol{z}_t, t) - \frac{\boldsymbol{z}_{t_{k-1}} - \lambda_k \boldsymbol{z}_{t_k}}{\eta_k} \right\|^2$;
  else
    Compute loss $\ell = \left\| \boldsymbol{v}_\theta(\boldsymbol{z}_t, t) - \frac{\boldsymbol{z}_{t_k} - \boldsymbol{z}_{t_{k-1}}}{t_k - t_{k-1}} \right\|^2$;
  end
  Update θ with gradient-based optimizer using $\nabla_\theta \ell$.
until *convergence*;
$\Delta W = \theta - \phi$.
Return: Fast PeRFlow f_θ and ΔW.

$\tilde{\boldsymbol{Z}}_{t_{k+1}} = \tilde{\boldsymbol{Z}}_{t_k} + (t_{k+1} - t_k)\boldsymbol{v}(\tilde{\boldsymbol{Z}}_{t_k}, t_k), \quad \tilde{\boldsymbol{Z}}_0 \sim \pi_0$

Modification 3: Multi-condition Classifier-Free Guidance (CFG)

常规 CFG

$$\hat{g}_\theta(z_t, c) = g_\theta(z_t, \varnothing) + \alpha \cdot [g_\theta(z_t, c) - g_\theta(z_t, \varnothing)]$$

Multi-Condition CFG

- 包含多个 condition 的模型
  - 文本（phoneme embedding） → text guidance
  - 音色（speaker prompt） → speaker guidance
- 训练策略
  - 10% 随机去除 speaker prompt
    - 其中 50% 再随机去除 phoneme 的 condition
  - 相当于：
    - 90% 使用全部 condition
    - 5% 只有 phoneme condition
    - 5% 完全没有 condition

$$\begin{aligned}\hat{g}_\theta(z_t, p, z_{prompt}) =& \alpha_{spk} \left[g_\theta(z_t, p, z_{prompt}) - g_\theta(z_t, p, \varnothing)\right] \\ &+ \alpha_{txt} \left[g_\theta(z_t, p, \varnothing) - g_\theta(z_t, \varnothing, \varnothing)\right] \\ &+ g_\theta(z_t, \varnothing, \varnothing)\end{aligned}$$

Experiments: WaveVAE Evaluation

训练配置

- Libri-Light 6万小时数据
- 训练 2M steps

Models	Tokens/s	Latent Layer	Type	PESQ↑	STOI↑	ViSQOL↑	MCD↓	UTMOS↑
Encodec	600	8	Discrete	3.16	0.94	4.31	1.63	3.07
DAC	450	9	Discrete	**4.13**	**0.97**	<u>4.68</u>	<u>1.05</u>	4.01
WavTokenizer	75	1	Discrete	2.55	0.88	3.83	1.99	4.07
X-codec2	50	1	Discrete	3.03	0.91	4.12	1.72	**4.13**
WaveVAE	25	1	Continuous	<u>3.84</u>	<u>0.96</u>	**4.71**	**1.03**	<u>4.10</u>

Table 5: Comparison of the reconstruction quality. The sampling rate are set to 16 kHz. **Bold** and <u>Underline</u> values indicate the best and second best results. "Tokens/s" means how many tokens a one-second speech will be compressed into.

- 不公平之处：Encodec/DAC 开源的模型，训练数据不匹配
- Encodec/DAC 的 VQ，也可以是一种正则（替代 KL Loss）

- Stable Diffusion 中，除了 KL 正则化，还尝试了 VQ 正则化，使用 VQ 前一层的 embedding 作为 latent 表征（VQ 被吸收为 Decoder 的一部分）
- KL Loss 系数小 ←→ VQ embedding 维度大
- 参考：https://arxiv.org/pdf/2112.10752

Setting	SIM-O↑	WER↓
Ours	**0.71**	**1.82%**
w/ Encodec	0.56	2.24%
w/ DAC	0.64	1.93%

Table 6: Comparison of zero-shot TTS performance of MegaTTS 3 using different speech compression models on the LibriSpeech test-clean set.

Experiments: Zero-Shot TTS Evaluation

Zero-Shot TTS 评测标准

- Sim-O: 合成音频与真实 prompt 之间的音色相似度 + SMOS
- Sim-R: 合成音频与 prompt 重建音频 之间的音色相似度 + SMOS
- WER：Hubert-Large LibriSpeech finetune

$$\begin{aligned}\hat{g}_\theta(z_t, p, z_{prompt}) =& \alpha_{spk}\left[g_\theta(z_t, p, z_{prompt}) - g_\theta(z_t, p, \varnothing)\right] \\ &+ \alpha_{txt}\left[g_\theta(z_t, p, \varnothing) - g_\theta(z_t, \varnothing, \varnothing)\right] \\ &+ g_\theta(z_t, \varnothing, \varnothing)\end{aligned}$$

Multi Condition 配置：text guidance 2.5, speaker guidance 3.5

Model	#Params	Training Data	SIM-O↑	SIM-R↑	WER↓	CMOS↑	SMOS↑	RTF↓
GT	-	-	0.68	-	1.94%	+0.12	3.92	-
VALL-E 2*	0.4B	LibriHeavy	0.64	0.68	2.44%	-	-	-
VoiceBox†	0.4B	Collected (60kh)	0.64	0.67	2.03%	-0.20	3.81	0.340
DiTTo-TTS*	0.7B	Collected (55kh)	0.62	0.65	2.56%	-	-	-
NaturalSpeech 3†	0.5B	LibriLight	0.67	0.76	**1.81%**	-0.10	3.95	0.296
CosyVoice	0.4B	Collected (172kh)	0.62	-	2.24%	-0.18	3.93	1.375
MaskGCT	1.0B	Emilia (100kh)	0.69	-	2.63%	-	-	-
F5-TTS	0.3B	Emilia (100kh)	0.66	-	1.96%	-0.12	3.96	0.307
MegaTTS 3	0.3B	LibriLight	**0.71**	**0.78**	**1.82%**	**0.00**	**3.98**	0.188
MegaTTS 3-accelerated	0.3B	LibriLight	0.70	**0.78**	1.86%	-0.03	3.96	**0.124**

Table 1: Zero-shot TTS results on the LibriSpeech test-clean set following NaturalSpeech 3 (Ju et al., 2024). * means the results are obtained from the paper. † means the results are obtained from the authors. #Params denotes the number of parameters. RTF denotes the real-time factor.

Experiments: Zero-Shot Prosodic Naturalness

韵律维度-客观评测标准

- MCD, SSIM, STOI, GPE, VDE, and FFE（参考 InstructTTS）

方案对比

- 没有和 DiT 方案（F5-TTS）对比，是否对韵律会带来负面影响
- 论文描述：适合有声书/AI 对话等对于合成稳定性要求高的场景

Method	MCD↓	GPE↓	VDE↓	FFE↓
NaturalSpeech 3	4.45	0.44	0.33	0.37
Ours w/ F.A.	4.48	0.44	0.35	0.40
Ours w/ S.A.	**4.42**	**0.31**	**0.29**	**0.34**

Table 4: Comparisons about prosodic naturalness metrics on LibriSpeech test-clean set. "F.A." denotes forced alignment and "S.A." denotes sparse alignment.

- w/ FA: 预先获取的对齐: VoiceBox
- w/ Standard AR Duration: AR Duration Predictor

Method	MCD↓	SSIM↑	STOI↑	GPE↓	VDE↓	FFE↓
Ours w/ Sparse Alignment	**4.56**	**0.52**	**0.62**	**0.34**	**0.30**	**0.35**
Ours w/ Forced Alignment	4.62	0.45	**0.62**	0.42	0.34	0.40
Ours w/ Standard CFG	4.59	0.51	0.61	0.36	0.32	0.37
Ours w/ Standard AR Duration	4.58	0.50	**0.62**	0.36	0.31	0.36

Table 9: Comparisons about "expressiveness" metrics on the LibriSpeech test-clean set.

Experiments: Accented TTS

口音 TTS 评测

评测标准

- MCD / pitch 分布的标准差(σ)、偏度(skewness γ)、峰度(kurtosis κ) + ASMOS 口音相似度
- 偏度反映对称性、峰度反映中心尖锐程度

CFG 配置

- 带口音英语：text guidance 1.5, speaker guidance 6.5
- 标准英语： text guidance 5.0, speaker guidance 2.0
- 预先实验：Text guidance
  - 1.0 – 1.5：发音错误+失真
  - 1.5 - 2.5：与 prompt 一致性很好
  - > 4.0: 数值越大，口音越少
- CTA-TTS 方案: 获取数据的口音强度，作为模型的控制条件

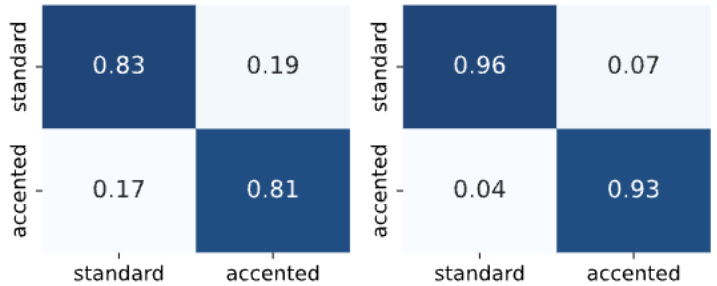


Figure 2: The confusion matrices between the perceived and intended accent categories of synthesized speech. The X-axis and Y-axis represent the intended and perceived categories, respectively.

Model	MCD (dB) ↓	σ ↑	γ ↓	κ ↓	ASMOS ↑	CMOS ↑	SMOS ↑
GT	-	45.1	0.591	0.783	4.03	+0.09	3.95
CTA-TTS	5.98	41.1	0.602	0.799	3.72	-0.60	3.64
MegaTTS 3	**5.69**	**42.3**	**0.601**	**0.790**	**3.84**	**+0.00**	**3.89**

Table 3: The objective and subjective experimental results for accented TTS. MCD (dB) denotes the Mel Cepstral Distortion at the dB level. σ, γ, and κ are the standard deviation, skewness, and kurtosis of the pitch distribution.

Experiments: Duration Controllability

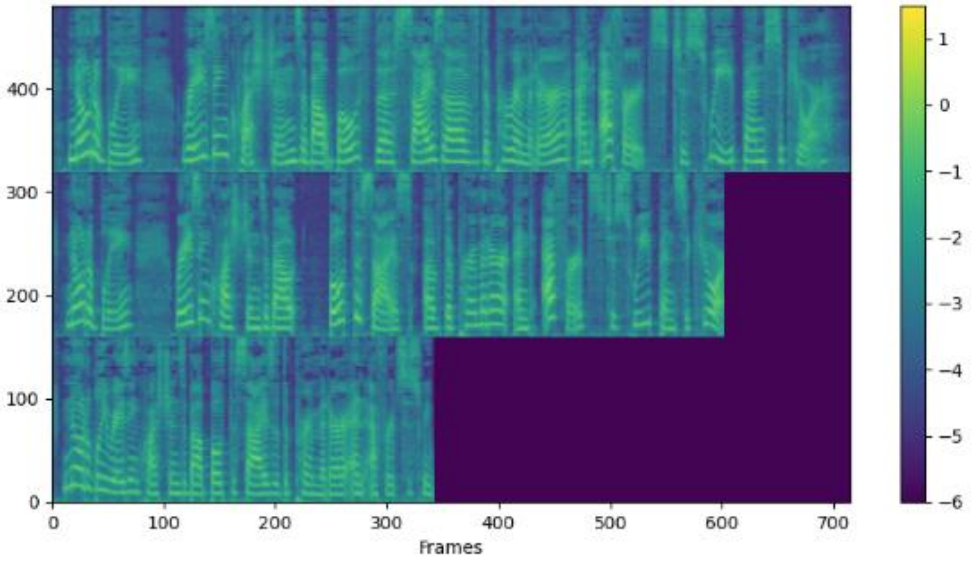


Figure 4: Sentence-level duration control.

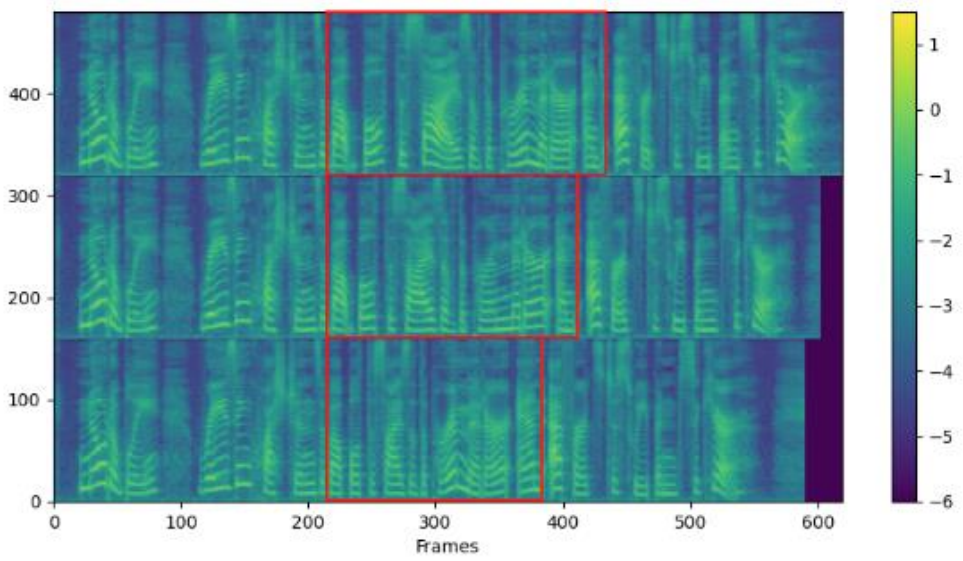


Figure 5: Phoneme-level duration control.

Experiments: Others

不同数据量和参数量的实验

- 更大数据量：Youtube + Podcast + Libri-Light + Wenet/Giga-Speech
- 60 万小时多语种数据（随机 1 万小时训练 MFA模型）
- 各实验中，WaveVAE 模型保持不变
- 更多场景的测试集
  - Noisy: Common-Voice
  - EmotionalTTS: RAVDESS
  - Wild: videos/movies/animations
- 但在这些测试集上，没有和其他论文/开源模型进行对比

Setting	SIM-O↑	WER↓
2kh	0.52	4.27%
40kh	0.63	2.98%
200kh	0.65	2.34%
600kh	**0.66**	**2.10%**
0.5B	0.66	2.10%
1.5B	0.72	1.98%
7.0B	**0.74**	**1.90%**

Table 8: Results of data and model scaling experiments.

50 字以上长文本 & ELLA-V Hard 测试集

Model	WER↓	Substitution↓	Deletion↓	Insertion↓
E2-TTS	8.49%	3.65%	4.75%	0.09%
F5-TTS	4.28%	**1.78%**	2.28%	0.22%
MegaTTS 3	**3.95%**	1.80%	**2.07%**	**0.08%**

Table 11: Comparisons with hard sentences. The results of the baselines are infered from offical checkpoints.

Model - with Longer Texts	WER↓	SIM-O↑
VoiceCraft	12.81%	0.62
CosyVoice	5.52%	0.68
MegaTTS 3	**2.39%**	**0.70**
Model - with Short Texts	**WER↓**	**SIM-O↑**
VoiceCraft	4.07%	0.58
CosyVoice	2.24%	0.62
MegaTTS 3	**1.82%**	**0.71**

Details From Open-Sourced Code

开源链接

- https://github.com/bytedance/MegaTTS3
- https://sditdemo.github.io/sditdemo

1. AR Duration Predictor

- phone + bert_emb 的自回归预测模型
- 理论上具有 In-Context Learning 的能力

2. 推理阶段的 Aligner

- 基于 MFA 的对齐结果，训练了一个 NN 的对齐模型

3. g2p 文本转音素

- Qwen2.5-0.5B，针对 g2p 任务进行 finetune

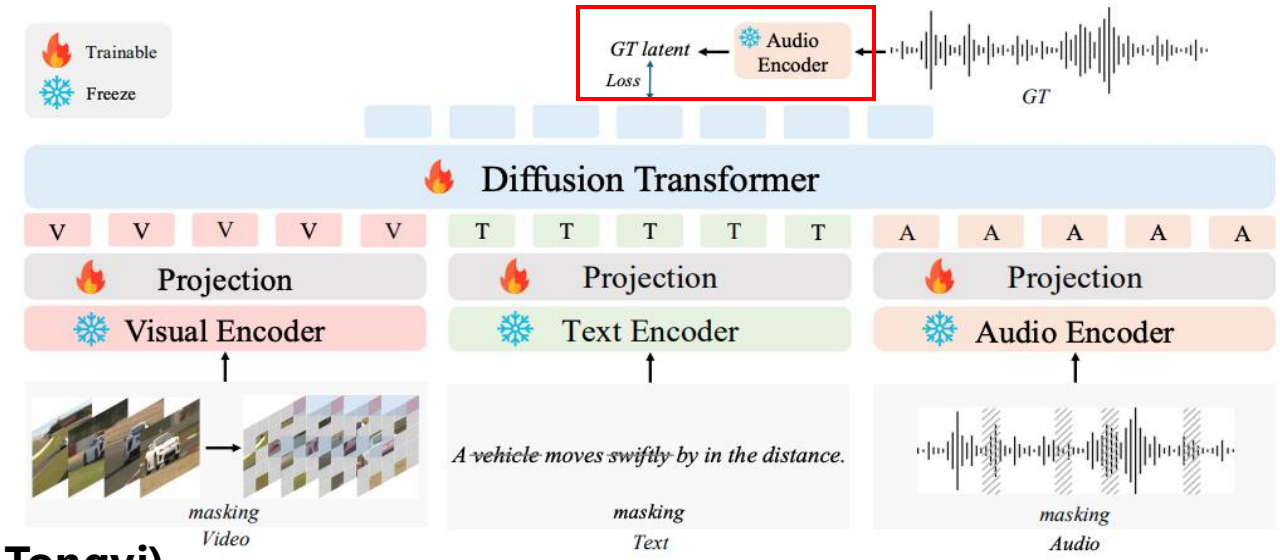
4. Seed-TTS 评测集对比

Model	*test-zh*		*test-en*	
	CER (%) ↓	SIM-O ↑	WER (%) ↓	SIM-O ↑
Human	1.26	0.755	2.14	0.734
FireRedTTS	1.51	0.635	3.82	0.460
MaskGCT	2.27	<u>0.774</u>	2.62	<u>0.714</u>
F5-TTS (32 NFE)	1.56	0.741	<u>1.83</u>	0.647
Llasa-8B	1.59	0.684	2.97	0.574
Spark-TTS	**1.20**	0.672	1.98	0.584
CosyVoice 2	1.45	0.748	2.57	0.652
MegaTTS 3	<u>1.36</u>	**0.790**	**1.82**	**0.773**

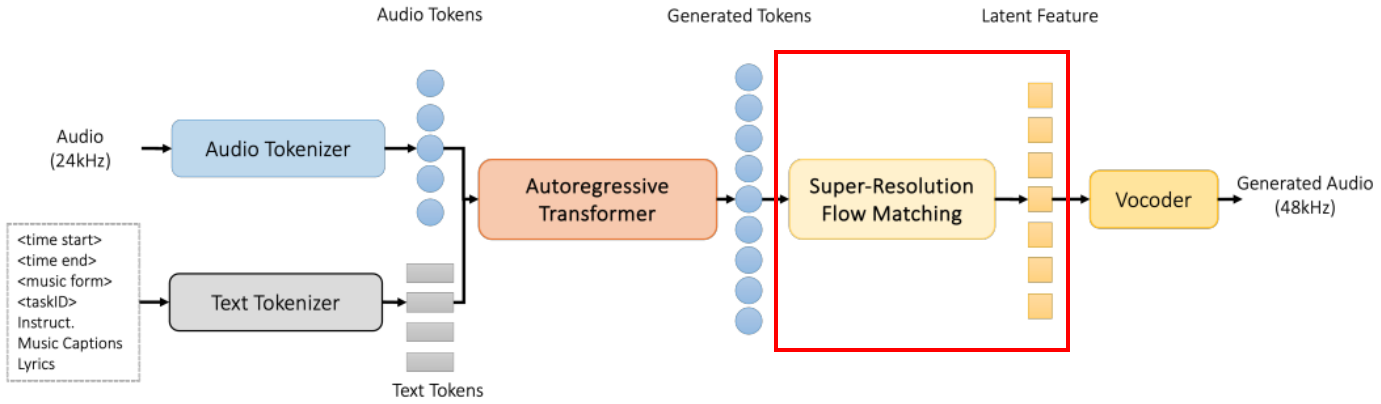
Table 12: Results of MegaTTS 3 and recent open-sourced TTS models on the SEED test sets.

Takeaways – 1: WaveVAE + DiT 范式

AudioX (MoonShot)

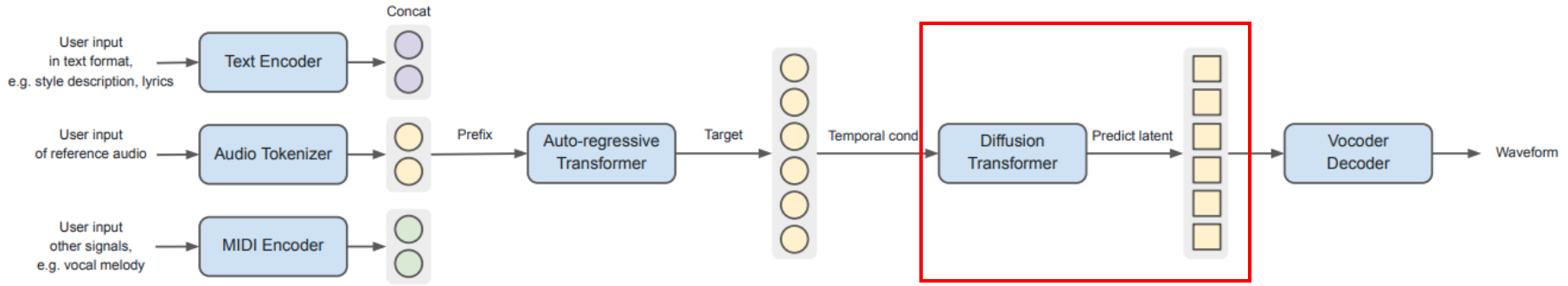


InspireMusic (Alibaba Tongyi)

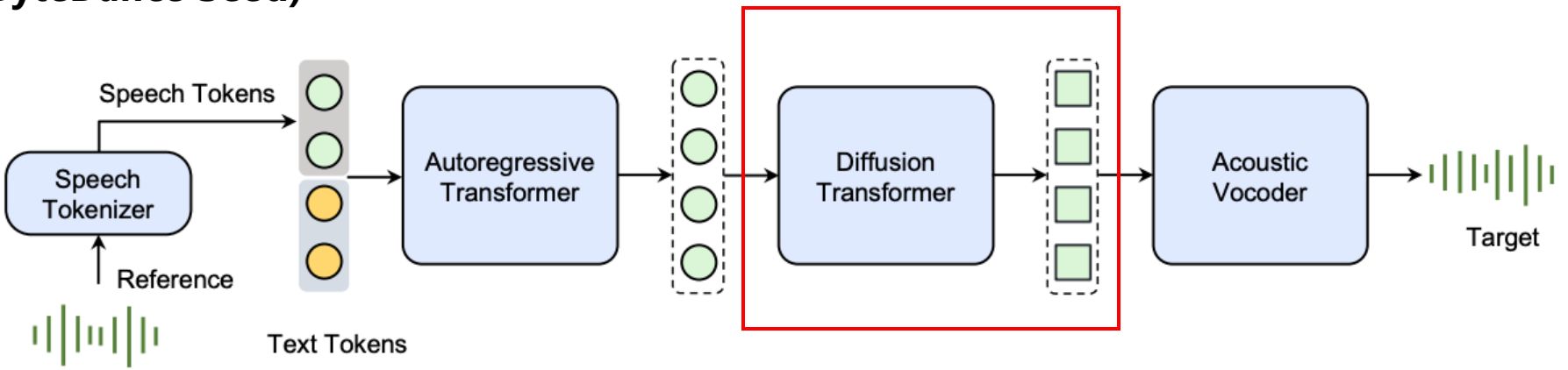


Takeaways – 1: WaveVAE + DiT 范式

Seed-Music (ByteDance Seed)



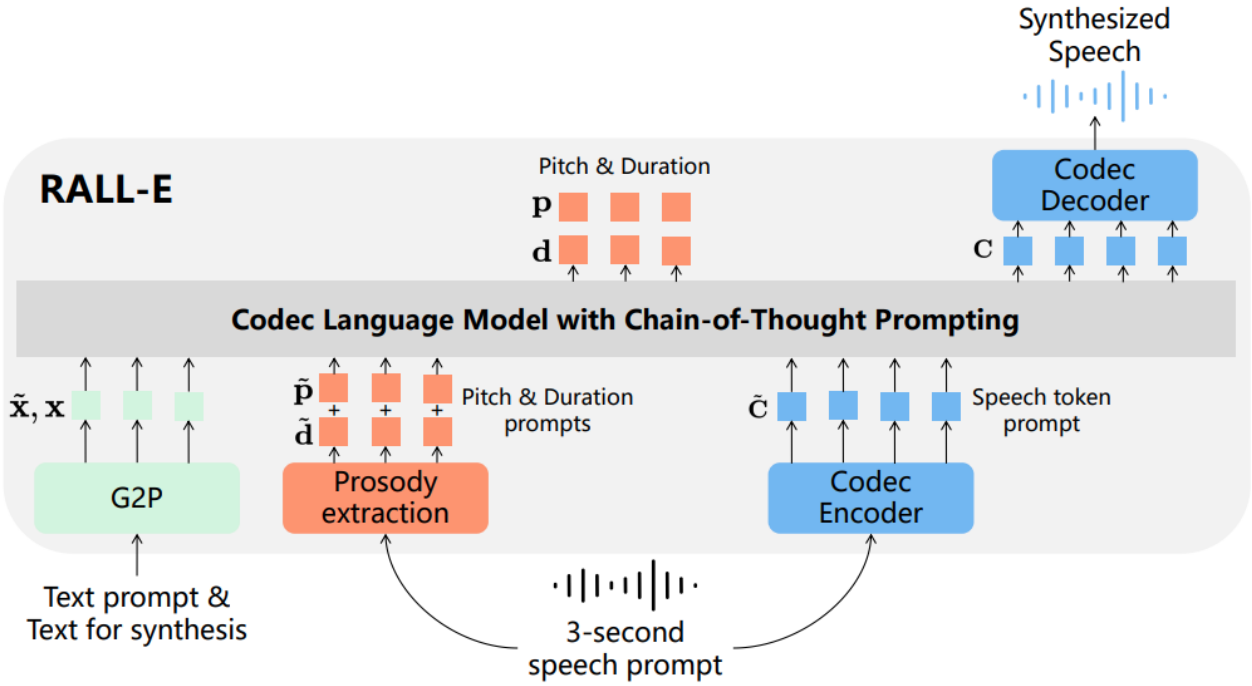
Seed-TTS (ByteDance Seed)



Takeaways – 2: LLM + Explicit Alignment Modeling

LLM Based TTS with Explicit Alignment Modeling

- (Microsoft) RALL-E: Robust Codec Language Modeling with Chain-of-Thought Prompting for Text-to-Speech Synthesis

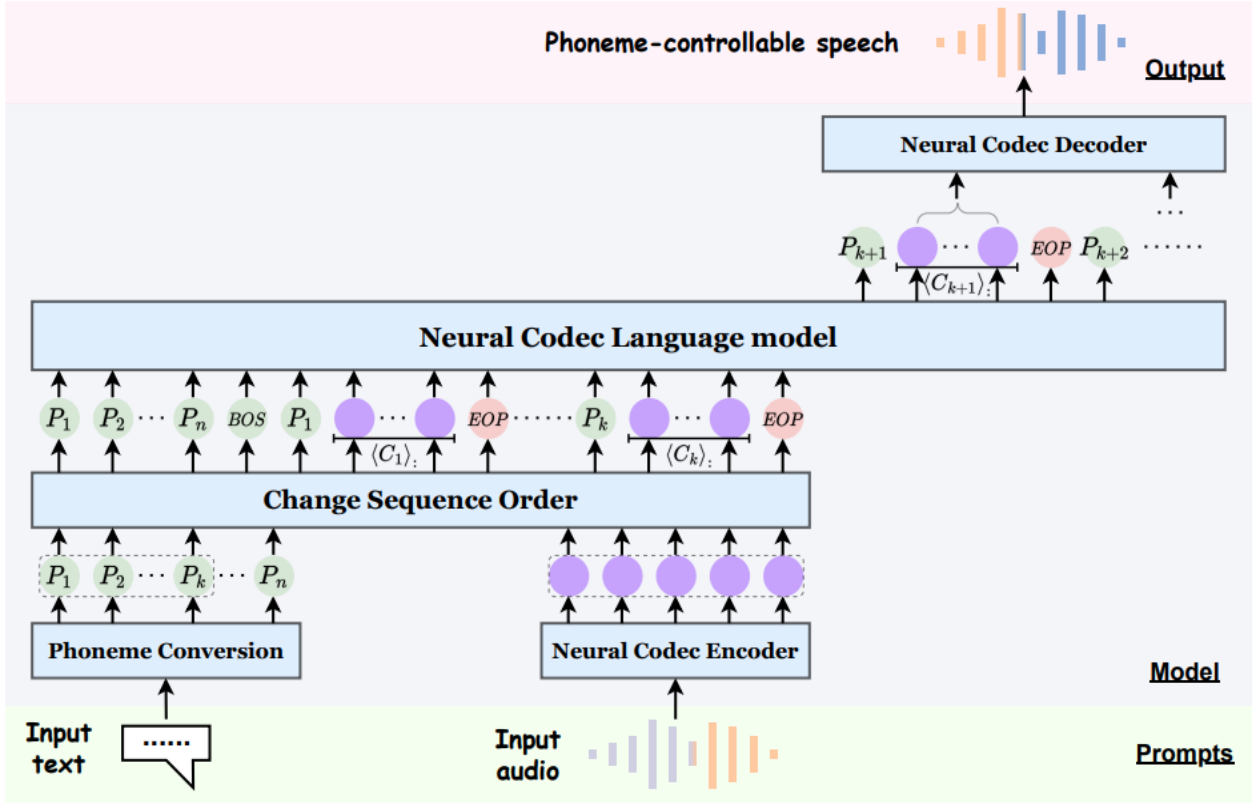


TTS task is a monotonic sequence to sequence task, but the decoder-only transformer structure in LLM only captures the monotonic association between phoneme sequences and audio through self-attention mechanism, which can easily lead to potential attention degradation problems when facing complex or long sequences, resulting in inaccurate generation results.

Takeaways – 2: LLM + Explicit Alignment Modeling

LLM Based TTS with Explicit Alignment Modeling

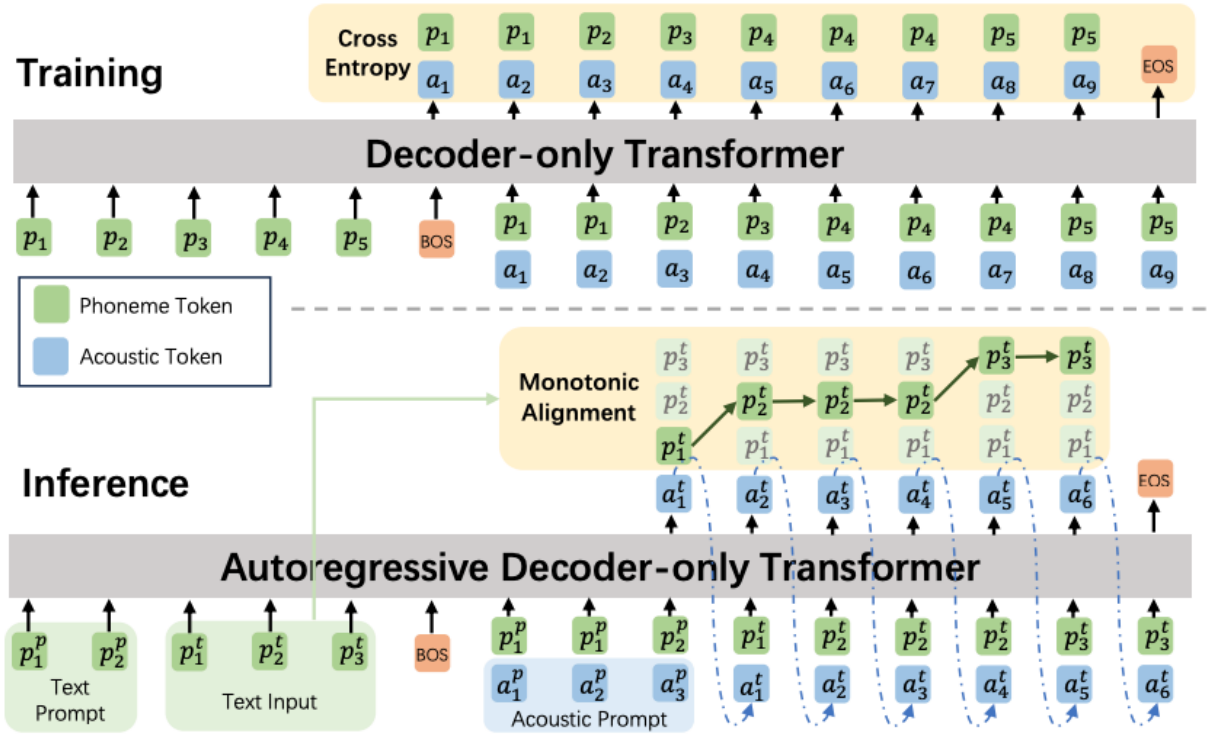
- (Micosoft) ELLA-V: Stable Neural Codec Language Modeling with Alignment-guided Sequence Reordering



Takeaways – 2: LLM + Explicit Alignment Modeling

LLM Based TTS with Explicit Alignment Modeling

- (Microsoft) VALL-E R: Robust and Efficient Zero-Shot Text-to-Speech Synthesis via Monotonic Alignment



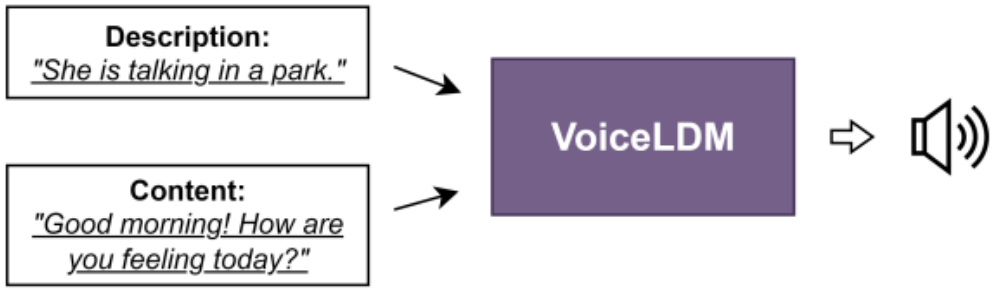
Takeaways – 3: Multi-Condition Trade-Off

DualSpeech (Supertone + ElevenLabs)

$$\begin{aligned}\tilde{\epsilon}_\theta(z_t, t, c_{\text{spk}}, c_{\text{text}}) &= \epsilon_\theta(z_t, t, c_{\text{spk}}, c_{\text{text}}) \\ &+ \omega_{\text{spk}}\big(\epsilon_\theta(z_t, t, c_{\text{spk}}, \emptyset) - \epsilon_\theta(z_t, t, \emptyset, \emptyset)\big) \\ &+ \omega_{\text{text}}\big(\epsilon_\theta\big(z_t, t, \emptyset, c_{\text{text}}\big) - \epsilon_\theta\big(z_t, t, \emptyset, \emptyset\big)\big),\end{aligned}$$

VoiceLDM (KAIST)

- desc: 代表对于环境的描述文本
- cont: 代表待合成的文本
- 不同 condition 之间的 trade-off
  - Instruct / Text / Speaker
- 推广：Semantic Token / Prompt / speaker



$$\begin{aligned}\tilde{\boldsymbol{\epsilon}}_\theta(z_t, \mathbf{c}_{desc}, \mathbf{c}_{cont}) &= \boldsymbol{\epsilon}_\theta(z_t, \mathbf{c}_{desc}, \mathbf{c}_{cont}) \\ &+ w_{desc}\Big(\boldsymbol{\epsilon}_\theta(z_t, \mathbf{c}_{desc}, \emptyset_{cont}) - \boldsymbol{\epsilon}_\theta(z_t, \emptyset_{desc}, \emptyset_{cont})\Big) \\ &+ w_{cont}\Big(\boldsymbol{\epsilon}_\theta(z_t, \emptyset_{desc}, \mathbf{c}_{cont}) - \boldsymbol{\epsilon}_\theta(z_t, \emptyset_{desc}, \emptyset_{cont})\Big)\end{aligned}$$

Takeaways – 3: Multi-Condition Trade-Off

LLM with CFG? (EleutherAI)

- Reference: Stay on topic with Classifier-Free Guidance（增强 prompt 对于 response 的约束）
- 不需要训练去除 conditional 训练，因为 LLM 是 teacher forcing 训练的

$$\log \widehat{\mathrm{P}}_\theta(\epsilon_t|x_{t+1}, c, \bar{c}) = \log \mathrm{P}_\theta(\epsilon_t|x_{t+1}, \bar{c}) + \gamma\big(\log \mathrm{P}_\theta(\epsilon_t|x_{t+1}, c) - \log \mathrm{P}_\theta(\epsilon_t|x_{t+1}, \bar{c})\big)$$

Negative Prompting

$$\log \widehat{\mathrm{P}}_\theta(w_i|w_{j<i}, c) = \log \mathrm{P}_\theta(w_i|w_{j<i}) + \gamma\big(\log \mathrm{P}_\theta(w_i|w_{i<j}, c) - \log \mathrm{P}_\theta(w_i|w_{j<i})\big)$$

	ARC-c	ARC-e	BoolQ	HellaSwag
GPT2-small	22.7 / **23.0**	39.5 / **42.1**	48.7 / **57.0**	31.1 / **31.9**
GPT2-medium	**25.0** / 23.9	43.6 / **47.6**	58.6 / **60.1**	39.4 / **40.9**
GPT2-large	**25.1** / 24.7	46.6 / **51.0**	60.5 / **62.1**	45.3 / **47.1**
GPT2-xl	28.5 / **30.0**	51.1 / **56.5**	61.8 / **62.6**	50.9 / **52.4**
Pythia-160M	**23.5** / 23.0	39.5 / **42.2**	55.0 / **58.3**	30.1 / **31.2**
Pythia-410M	**24.1** / 23.8	45.7 / **50.3**	60.6 / **61.2**	40.6 / **41.6**
Pythia-1B	27.0 / **28.0**	49.0 / **54.9**	60.7 / **61.8**	47.1 / **48.9**
Pythia-1.4B	28.6 / **29.6**	53.8 / **59.6**	63.0 / **63.8**	52.1 / **54.3**
Pythia-2.8B	33.1 / **34.5**	58.8 / **65.4**	64.7 / **64.7**	59.3 / **61.9**
Pythia-6.9B	35.2 / **36.1**	61.3 / **67.4**	63.7 / **64.6**	64.0 / **66.5**
Pythia-12B	36.9 / **38.7**	64.1 / **72.6**	67.6 / **67.8**	67.3 / **69.6**
LLaMA-7B	41.5 / **43.9**	52.5 / **58.9**	**73.1** / 71.8	73.0 / **76.9**
LLaMA-13B	47.8 / **54.2**	74.8 / **79.1**	**78.0** / 75.8	79.1 / **82.1**
LLaMA-30B	52.9 / **57.4**	78.9 / **83.2**	**82.7** / 80.0	82.6 / **85.3**
LLaMA-65B	55.6 / **59.0**	79.7 / **84.2**	**84.8** / 83.0	84.1 / **86.3**

γ=1 / γ=1.5