arXiv:2611.30723v2 [eess.AS] 24 Nov 2026

BiSAT：面向语音语言模型的双向对齐语义声学标记化

Yixian Lu *，Lijun Wang*，*IEEE*研究生会员，以及 Shogo Okada†，*IEEE*会员

摘要——**语音语言模型（LM）将语音生成为离散标记序列，因此需要能够以较低比特率同时保留声学细节和语言内容的分词器。语义增强型分词器可能会丢弃声学细节，而面向声学的分词器则会保留妨碍语言建模的声学变化。统一分词器将两者结合起来，但在单一特征空间中对其进行编码可能会导致相互干扰，而且现有方法通常会将声学特征对齐到冻结语义教师模型提供的固定目标。我们提出双向对齐语义声学标记化（BiSAT），这是一种统一框架，采用独立的声学分支和语义分支，并通过共享索引机制将二者耦合。训练期间，每个分支都有各自的编码器、码本和解码器，并在每一帧上通过共享索引最小化两个分支的合并量化误差。因此，BiSAT在彼此独立的连续空间中学习声学和语义表示，同时生成单一离散码流，并在不依赖固定语义教师模型的情况下联合优化两个编码器。基于BiSAT构建的BiSAT-TTS，将自回归文本语音LM与条件流匹配声学解码器相结合，用于零样本文本转语音。重建、冻结码理解和诊断分析表明，BiSAT能以低比特率的单一码流保留易于利用的语言和声学结构，对真实场景和域外语音具有很强的泛化能力，并能学习互补的配对码本。BiSAT-TTS使用更少的训练数据，达到或超越最先进的基于语音LM的TTS基线方法，这表明有利于自回归语音建模的不只是重建质量，还有易于利用的编码结构和模式。音频演示请见https://seekmind-lab.github.io/bisat-demo。**

索引术语—**语音分词、零样本文本转语音合成、文本-语音语言模型、条件流匹配。**

I. 引言

SPEECH 语言模型（LM）将语音生成为离散标记序列，使文本转语音（TTS）系统能够采用大语言模型（LLM）架构和训练方案，以实现可扩展的序列建模 [1]–[3]。具体而言，自回归（AR）语音语言模型能够灵活确定输出长度，而无需显式时长建模，这与许多非自回归（NAR）方法 [1]形成对比。在基于标记的语言模型能够对语音建模之前，必须将连续语音波形离散化。这可以通过量化来实现，例如向量量化（VQ） [4], 、残差向量量化（RVQ） [5], 、有限标量量化（FSQ） [6], 和二元球面量化（BSQ） [7]。VQ、FSQ和BSQ每帧给出一个编码，而RVQ则用进一步的码本对每个阶段的残差进行量化，并产生若干代码流。按照语音标记化领域的常见用法[8],，我们使用“语义”来指代语音的语言内容，包括语音和词汇信息。训练目标会影响编码所保留的信息 [4]。语音重建促使编码保留韵律、音高、音色和说话人身份等声学信息[5], [9]–[11],，而语音语言模型还要求编码保留语义内容。在紧凑的单流瓶颈中同时保留这两类信息，是为语音语言模型设计分词器时需要权衡的核心问题。所保留的信息决定了语言模型必须建模的内容，以及声学解码器能够恢复并重新合成波形的内容。

一般而言，基于语言模型的文本转语音使用三种语音标记器，它们在如何权衡取舍方面有所不同。语义增强 标记器（SET）将量化器嵌入预训练语音理解模型中 [2], [12]–[15]。声学导向标记器（AOT）主要通过波形或梅尔频谱图重建来学习编码 [5], [9], [16], [17]。统一标记器（UT）将声学重建与语义监督相结合 [3], [18]–[22]。语义监督能够捕捉语言内容，但可能会丢弃所需的细粒度声学细节

重建。相反，通过重建学习的编码能够很好地保留声学细节，但也会保留音色、韵律和录音条件方面的变化，这可能削弱与文本的对齐，并导致自回归语言模型重复或跳过词语[18], [23], [24]。近期的 UT 将声学监督与语义监督结合在同一个分词器中，方法包括用语义特征引导第一个 RVQ层 [18], [19], [22] ，或在量化前融合两种表示 [3], [20], [21]。然而，在单一共享或融合的表示空间中编码声学细节和语言内容，可能导致两个目标相互干扰。此外，这些方法从预训练或冻结的编码器中获取语义监督，因此在分词器训练期间，语义目标保持固定。最后，大多数 AOT 和 UT 使用对抗训练（AT）重建波形，因为仅使用回归损失会在低比特率下产生过度平滑的语音 [5], [9]。由于对抗判别器评估的是自然度，而非编码的忠实度，联合训练的波形解码器可能合成编码中不存在的细节，从而使重建质量与结构化语义模式脱钩 [3], [9], [17], [18], ，如第II-A节所述。

这些局限促使我们设计一种低比特率、低帧率的单一代码流分词器，使其在独立的连续空间中学习声学和语义表示，同时在没有固定语义教师模型的情况下联合优化两个分支。TokenFlow [25]在图像分词中表明，由共享索引连接的独立语义码本和像素级码本能够同时支持理解与生成。受这一共享索引原则的启发，我们提出了双向对齐语义-声学分词（BiSAT）。在训练过程中，

作者来自日本先进科学技术研究所（日本石川县能美市，邮编 923-1292）（电子邮箱：yixian.lu@jaist.ac.jp；lijun.wang@jaist.ac.jp；okada-s@jaist.ac.jp）。本研究部分由 JSPS KAKENHI（编号：26K03016）、JST CREST（编号：JPMJCR2563）、JST CRONOS（编号：JPMJCS24K7）和 AMED（资助编号：JP256f0137001）资助。

* 贡献相同；作者顺序通过抛硬币随机确定。 † 通讯作者。

BiSAT采用独立的声学分支和语义分支，每个分支都拥有自己的编码器、码本和解码器。声学分支在不进行对抗训练的情况下重建梅尔频谱图，语义分支则采用连接主义时间分类（CTC）进行训练。在TTS过程中，波形合成被交由独立的声学解码器和声码器完成，因此分词器编码仅由梅尔频谱图重建和CTC监督塑造。在每一帧中，共享索引向量量化器通过最小化两个分支上平方欧氏距离的加权和来选择单个索引。这一设计具有两个主要优点。首先，在有限离散瓶颈 [26], [27], 下，将编码器、码本和训练解码器保留在独立的连续空间中，旨在减少声学重建与语义识别之间可能产生的干扰，同时使每个分支都能学习任务特定表示。其次，与将声学特征对齐至固定语义教师目标的方法不同，BiSAT会联合优化两个编码器。由于共享索引取决于两个分支，任一编码器的更新都会影响两个分支的量化分配。我们将这种耦合优化机制称为双向对齐。我们评估由此得到的单一代码流是否保留了对语言和声学下游任务均有用的信息。

我们通过从编码重建语音，以及在 SUPERB 和 ML-SUPERB 理解探测任务中的评估 [28][29]（涵盖语言学、说话人和副语言任务），验证 BiSAT 在为语音语言模型保留易于获取的语义内容和声学细节方面的有效性。我们还分析了配对码本，并使用 BiSAT-TTS 评估零样本文本转语音；BiSAT-TTS 结合了自回归文本-语音语言模型、参考条件流匹配声学解码器和预训练声码器。

我们的主要贡献总结如下：

1) 我们提出了 BiSAT，这是一种统一的语音标记化框架，能够学习彼此独立的连续声学表示和语义表示，同时耦合配对码本间的索引分配，从而生成单一的低比特率代码流。2) 我们开发了 BiSAT-TTS，这是一种零样本文本转语音系统，它将自回归文本-语音语言模型与使用 BiSAT 令牌的条件流匹配声学解码器相结合。3) 我们对 BiSAT 在语音重建、下游理解、诊断分析和零样本文本转语音方面进行了全面评估，结果显示其在多个基准测试中表现具有竞争力或更优，并验证了整体框架的有效性。

本文其余部分安排如下：第二节回顾相关工作。第三节介绍 BiSAT 和 BiSAT-TTS。第四节详述实验。第五节总结本文，第六节讨论局限性与未来工作。

II. 相关工作

A. 面向语音语言建模的语音标记化

1）语义增强型标记器：SET 将量化模块引入预训练语音理解模型中。CosyVoice 系列 [2], [12], [13] 对理解模型的中间表示进行量化，从采用自动语音识别（ASR）监督的 VQ，逐步发展到 FSQ，并在 CosyVoice 3 中采用包含 ASR、语言识别、情感识别、音频事件检测和说话人分析的多任务监督。Spark-TTS [14]将帧级语义令牌与用于表示话语级属性的定长全局令牌相结合，而 Qwen3-TTS [15] 则使用注重语义内容的 25Hz 单码本分词器。CosyVoice 分词器仅使用理解任务进行训练，因此其令牌所携带的声学细节很少，无法满足重建所需。因而，需要采用基于流的 [2], [13] 或生成对抗网络（GAN）解码器 [14] ，通过额外的条件信息来恢复这些细节，而这可能需要更多数据和更大的解码器容量。

2) 面向声学的分词器：AOT 通过重建来学习编码。EnCodec [9] 等 RVQ 编解码器会增加残差量化层，以恢复更丰富的声学细节，但将多个代码流串行化为一个序列，会显著增加自回归建模的序列长度。因此，VALL-E 系列 [1], [23] 使用自回归语言模型预测第一个代码流，并使用并行非自回归模型预测残差代码流。VoiceCraft 和 VoiceStar [30], [31]通过延迟码本模式，在单个自回归语言模型中预测所有代码流；CLaM-TTS[32]则对概率 RVQ 潜变量进行建模。WavTokenizer [17]和 FocalCodec [16] 通过以低比特率生成单一代码流，减轻语言建模的序列负担。我们也采用低帧率的单一低比特率代码流，在不预测额外残差代码流的情况下实现高效的自回归语言建模。

这些 AOT 没有接受明确的语义监督，因此语言模型必须从具有声学基础的令牌中推断语言结构，这可能导致重复、漏词和无法终止的序列 [23], [24], [33]。另一个问题来自分词与波形重建的耦合。大多数这类编解码器使用自回归变换器重建波形，因为单靠回归损失会在低比特率下产生过度平滑的语音[5], [9]。判别器判断的是输出听起来是否自然，而不是它是否与输入匹配。因此，解码器可能合成编码中并未包含的细节，而且对抗目标不会直接促使编码形成结构化语义模式。这使语言建模变得更加困难，也需要更多数据、更大的模型容量和更多计算资源。这些观察促使我们将分词与波形重建解耦，并直接使用语义内容监督分词器。

3) 统一分词器： UT 将语义监督与声学重建相结合。SpeechTokenizer [18] 和 Mimi[19]利用来自 HuBERT 或 WavLM 的自监督学习（SSL）特征引导第一层 RVQ层，而后续层则编码残余声学信息。LM-SPT [22]从其语义编码中重建语音，并匹配由冻结语音编码器提取的原始语音与重建语音的特征。在这些 RVQ 设计中，第一层必须表示语言内容，同时提供由后续层细化的声学近似，这使得两个目标难以平衡 [18], [19], [22], [34]。XY-Tokenizer [20] 和 X-Codec 系列 [3], [21]则在量化之前融合声学编码器和语义编码器的输出。X-Codec2 使用单个 FSQ 码本量化融合后的特征，使 LLaSA [3] 能够使用一个自回归语言模型。在融合设计中，对声学细节和语言内容进行编码

于单一融合表示中，可能导致训练期间两个目标相互干扰 [20]。这些方法还从预训练或冻结的语音编码器中获取语义监督，即教师特征 [3], [18], [19], [21]或冻结编码器 [20], [22], ，从而使语义目标在分词器训练期间保持固定。最后，UT 将分词与对抗性波形重建相结合，如同 AOT 一样，使重建能够依赖联合训练解码器，而无需确保具有可泛化能力的语义编码模式。

在图像分词中，TokenFlow [25] 使用独立的语义码本和像素级码本，并通过共享索引将两者关联起来，以支持理解和生成。受此设计启发，BiSAT联合更新声学编码器和语义编码器，使用独立码本，并根据两种表示确定它们的共享离散索引。

B. 零样本文本转语音系统

零样本文本转语音系统包括非自回归、自回归、连续自回归和离散扩散模型。非自回归系统并行生成所有帧，包括采用单调对齐搜索或显式时长建模的流匹配模型和基于VAE的模型[35]–[37], ，潜在扩散模型 [10], [38], 以及无需监督即可学习对齐的流匹配模型 [39], [40]。连续自回归模型直接预测连续声学特征，可避免量化误差，但可能出现过度平滑和序列生成不稳定的问题 [41], [42]。离散扩散模型则通过迭代去噪生成标记序列 [43]。离散自回归文本-语音语言模型按序生成语音标记，无需显式时长控制。基于RVQ的系统需要采用多阶段或并行预测方案 [1], [23], [32], ，而单流向量量化或有限标量量化分词器则允许语言模型 [2], [3], [12], [14]。BiSAT-TTS采用这种单流自回归方法。

C. 声学语义表示对齐

在语音领域，研究者通过自监督预训练 [44]–[46], 联合语音-文本预训练[47], [48],以及生成式潜在空间的语义正则化 [49], 探索声学信号与语言内容之间的表示级对齐。这些方法在单一连续空间中对齐声学信息与语言信息。BiSAT则从共享自监督语音骨干网络初始化双编码器 [50], ，并在各自独立的声学监督和语义监督下对其进行专门化，再通过共享索引量化将两个连续空间关联起来，形成单一代码流。

III. 方法

概述。 图1展示了BiSAT分词器和BiSAT-TTS流程。该分词器（见第III-B1节）通过共享索引量化，以双向方式对齐声学表示与语义表示，将语音波形离散化为单一代码序列。对于文本转语音，文本语音语言模型（见第III-B2节）根据文本和参考语音生成编码。流匹配声学解码器（见第III-B3节）将这些编码转换为梅尔频谱图。随后，预训练声码器合成波形。

A. 问题表述

我们通过自回归文本-语音语言模型 [1], [2]生成语音。令 $(x, \boldsymbol{y})$ 表示一个语音-文本对，其中 x 是波形， $\boldsymbol{y}:=(y_1, \ldots\ldots, y_U)$ 是其分词转录文本。CTC监督和文本-语音语言模型训练使用不同的文本分词器，其中 $\boldsymbol{y}$ 和 U 分别表示在各自设置下的标记序列及其长度。令 $\boldsymbol{m} \in \mathbb{R}^{F\times T_{\text{tok}}}$ 表示由分词器编码并重建的 F 频对数梅尔频谱图，令 $\boldsymbol{s} \in \mathbb{R}^{F\times T_{\text{mel}}}$ 表示由TTS声学解码器生成的梅尔频谱图。两个频谱图的采样率和帧率不同，详情见第IV-A2节。BiSAT分词器将x 映射为单个代码序列 $\boldsymbol{z}_{1:T} =(z_1, \ldots\ldots, z_T)$，其中每个编码 $z_t \in \{1, \ldots, K\}$ 对应一个共享索引，用于从两个配对码本$\boldsymbol{c}_a$, $\boldsymbol{c}_s \in \mathbb{R}^{K\times D_{\text{codebook}}}$中各选取一个条目。在全文中，下标 a 和 s 分别表示声学分支和语义分支； $[\cdot;\ \cdot]^{(\cdot)}$表示沿指定轴进行拼接。

*1) BiSAT*分词化：图 1 中的分词器通过共享离散编码学习对齐的声学和语义表示。两个编码器 $\mathcal{E}_a$ 和 $\mathcal{E}_s$，分别由 ψ_a 和 ψ_s参数化，将 $\boldsymbol{m}$ 编码为时间对齐的声学和语义潜变量 $\boldsymbol{e}_{a;1:T}$ 和 $\boldsymbol{e}_{s;1:T}$。两个编码器使用相同的特征维度$D_a = D_s = D_{\text{codebook}}$。共享索引量化器将这些表示转换为成对的量化序列 $\hat{}\,\boldsymbol{z}_{a;1:T}$ 和 $\hat{}\,\boldsymbol{z}_{s;1:T}$。梅尔重建解码器 $\mathcal{D}_a$由 γ_a参数化，根据 $\hat{}\,\boldsymbol{z}_{a;1:T}$重建梅尔频谱图 $\hat{}\boldsymbol{m} \in \mathbb{R}^{F\times T_{\text{tok}}}$。语义解码器 $\mathcal{D}_s$由 γ_s参数化，根据 $\hat{}\,\boldsymbol{z}_{s;1:T}$ 预测语义标签，并通过 CTC监督 [51]进行训练。

梅尔频谱图 $\boldsymbol{m}$ 和文本序列 $\boldsymbol{y}$ 为共享编码$\boldsymbol{z}_{1:T}$提供声学和语义监督。根据 VQ-VAE公式 [4], ，我们定义音频条件近似后验 $q_\psi(\boldsymbol{z}_{1:T} \mid \boldsymbol{m})$，其中$\psi = \{\psi_a, \psi_s\}$ 是确定性的最近邻分配。在每帧的 K 码本索引上采用均匀先验时，$\log p(\boldsymbol{m},\ \boldsymbol{y})$ 的证据下界（ELBO）中的 Kullback-Leibler（KL）散度项等于常数$T \log K$ ，且不依赖于模型参数 $\Theta =\{\psi_a, \psi_s, \gamma_a, \gamma_s, \boldsymbol{c}_a, \boldsymbol{c}_s\}$。假设给定 $\boldsymbol{z}_{1:T}$时， $\boldsymbol{m}$ 和 $\boldsymbol{y}$ 条件独立，最大化 ELBO 可简化为：

$$\begin{aligned}\operatorname*{argmax}_{\Theta} \text{ELBO}_{\text{tok}} = \operatorname*{argmax}_{\Theta} \mathbb{E}_{q_\psi(\boldsymbol{z}_{1:T}|\boldsymbol{m})}\big[&\log p_{\gamma_a}(\boldsymbol{m} \mid \boldsymbol{z}_{1:T}) \\ &+ \log p_{\gamma_s}(\boldsymbol{y} \mid \boldsymbol{z}_{1:T})\big].\end{aligned} \tag{1}$$

前者可简化为在固定尺度因子化拉普拉斯似然下，用于声学监督的 ℓ_1 梅尔频谱图重建损失。对于后者，采用 CTC 对帧级标签路径求和 π ，这些路径会折叠为 $\boldsymbol{y}$ ，用于语义监督。具体而言，重建损失和 CTC 损失 $\mathcal{L}_{\text{recons}}$ 和 $\mathcal{L}_{\text{ctc}}$ 定义如下：

$$\begin{aligned}\mathcal{L}_{\text{recons}} &= \frac{1}{T_{\text{tok}}F}\,\|\boldsymbol{m} - \hat{\boldsymbol{m}}\|_1, \\ \mathcal{L}_{\text{ctc}} &= -\frac{1}{U}\,\log \sum_{\pi\in\mathcal{B}^{-1}(\boldsymbol{y})} \prod_{t=1}^{T} p_{\gamma_s}(\pi_t \mid \hat{\boldsymbol{z}}_{s;1:T}),\end{aligned} \tag{2}$$

其中， $\mathcal{B}^{-1}(\boldsymbol{y})$ 是所有可折叠为$\boldsymbol{y}$的可能路径的集合，而 $p_{\gamma_s}(\pi_t \mid \hat{}\,\boldsymbol{z}_{s;1:T})$ 是标签 π_t在帧 t处的预测概率。由于 CTC 会对代码帧与文本之间的单调对齐进行边缘化处理，因此无需强制对齐。

中间CTC监督。遵循 OWSM-CTC

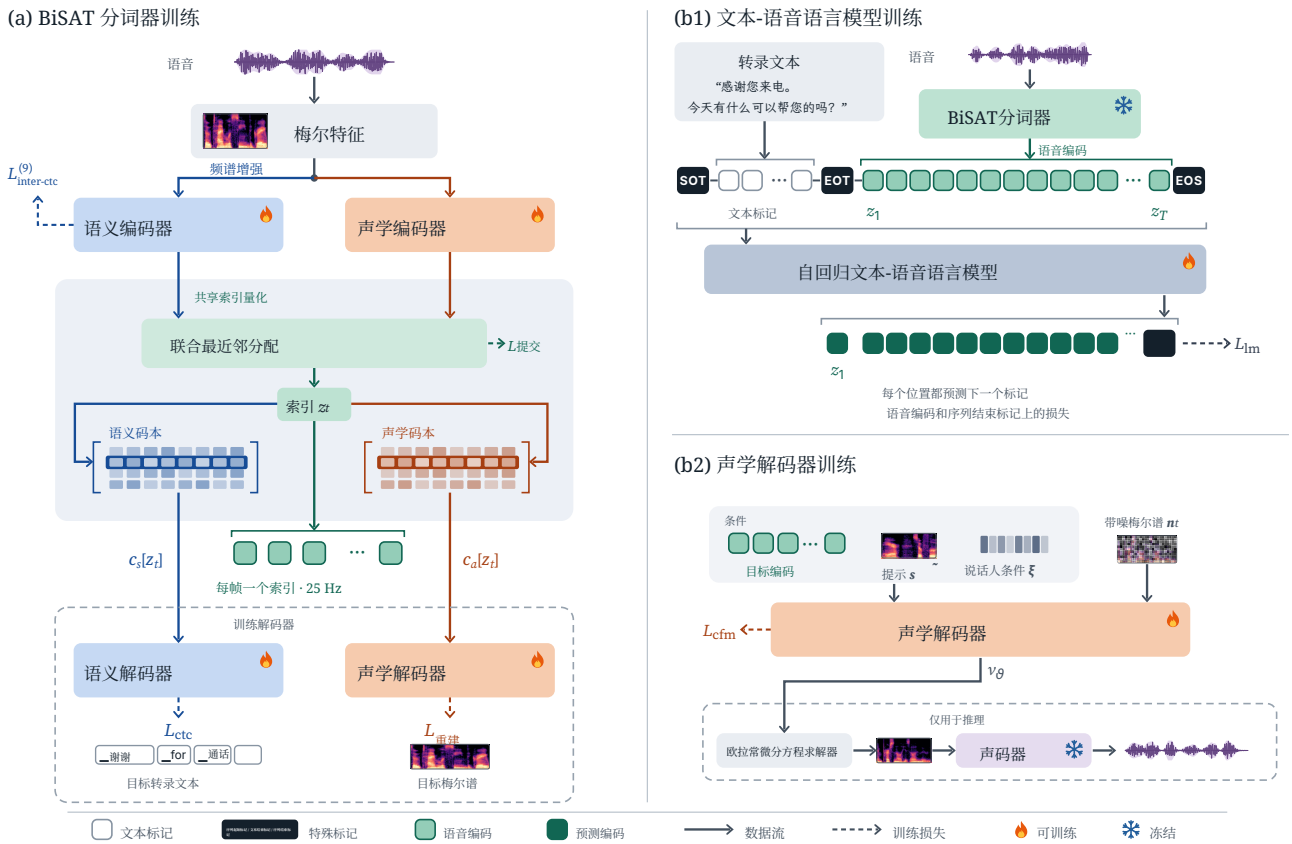

图1。BiSAT分词器和BiSAT-TTS的训练。(a) 声学分支和语义分支分别通过梅尔频谱图重建和CTC进行联合训练。共享索引向量化器从每个码本中为每一帧选择一个条目，以创建单一代码流。使用承诺损失，使两个编码器输出都接近这些条目。(b1) 文本语音语言模型根据文本标记和前置编码，预测每个语音编码以及最终的序列结束标记。(b2) 声学解码器根据噪声插值变量预测流匹配速度，并以编码、提示梅尔频谱图和说话人条件为条件。在推理过程中，欧拉常微分方程求解器生成梅尔频谱图，冻结的声码器将其转换为波形。

[52], 我们将辅助CTC头应用于第九个语义编码器层的输出 $\boldsymbol{h}_s^{(9)}$。损失函数定义如下：

$$\mathcal{L}_{\text{inter-ctc}} = -\frac{1}{U}\log \sum_{\pi\in\mathcal{B}^{-1}(\boldsymbol{y})}\prod_{t=1}^{T} p_{\gamma_{\text{int}}}\left(\pi_t \mid \boldsymbol{h}_s^{(9)}\right), \tag{3}$$

其中，$p_{\gamma_{\text{int}}}$ 是由 γ_{int}参数化的辅助CTC头的标签分布。

共享索引量化。两个编码器输出均按帧进行 ℓ_2 归一化，即对于 $b \in \{a, s\}$，$\bar{\boldsymbol{e}}_{b,t} = \boldsymbol{e}_{b,t}/\max(\|\boldsymbol{e}_{b,t}\|_2, \delta)$，并使用一个较小的常数 $\delta > 0$ 来确保数值稳定性。码本条目 $\{\boldsymbol{c}_{k,a}, \boldsymbol{c}_{k,s}\}_{k=1}^{K}$ 也经过单位归一化。条目 $\boldsymbol{c}_{k,a}$ 和 $\boldsymbol{c}_{k,s}$ 在每个索引 k 处组成一对。对于每一帧，量化器通过最小化联合距离来分配一个索引：

$$z_t = \arg\min_{k} \left[\|\bar{\boldsymbol{e}}_{a,t} - \boldsymbol{c}_{k,a}\|_2^2 + \omega\|\bar{\boldsymbol{e}}_{s,t} - \boldsymbol{c}_{k,s}\|_2^2\right]. \tag{4}$$

两个分支都检索对应的行，分别为 $\hat{\boldsymbol{z}}_{a,t} = \boldsymbol{c}_{z_t,a}$ 和 $\hat{\boldsymbol{z}}_{s,t} = \boldsymbol{c}_{z_t,s}$。对于单位归一化的码本条目，式 (4) 等价于最大化 $\langle \bar{\boldsymbol{e}}_{a,t}, \boldsymbol{c}_{k,a}\rangle + \omega\langle \bar{\boldsymbol{e}}_{s,t}, \boldsymbol{c}_{k,s}\rangle$，因此令 $\omega = 1$ 可对两个分支赋予相同权重。每一帧都由 K 个码对中的一个索引表示。例如，频率为 25 Hz 且 $K = 32768$ 的代码流，其标称比特率为 $25 \log_2 32768 = 375$ bps，与 D_{codebook} 无关。

我们使用旋转技巧 [53] 将解码器梯度通过量化进行反向传播。对于分支 b 和帧 t，令 $\mathbf{R}_{b,t}$ 表示将编码器输出旋转 $\bar{\boldsymbol{e}}_{b,t}$ 朝向选定的码本条目。解码器输入表示为：

$$\boldsymbol{q}_{b,t} = \text{sg}\left(\frac{\|\hat{\boldsymbol{z}}_{b,t}\|_2}{\|\bar{\boldsymbol{e}}_{b,t}\|_2}\mathbf{R}_{b,t}\right)\bar{\boldsymbol{e}}_{b,t}, \tag{5}$$

其中，$\text{sg}(\cdot)$ 表示停止梯度算子，因此在前向传播过程中，$\boldsymbol{q}_{b,t} = \hat{\boldsymbol{z}}_{b,t}$。由于编码器输出和码本条目均经过单位归一化，范数比为 1，因此在反向传播到编码器的过程中，解码器梯度会乘以 $\mathbf{R}_{b,t}^{\top}$。

我们通过采用承诺损失，使编码器输出保持接近其所分配的码本条目：

$$\mathcal{L}_{\text{commit}} = \frac{1}{T}\sum_{t=1}^{T}\left[\|\bar{\boldsymbol{e}}_{a,t} - \text{sg}(\hat{\boldsymbol{z}}_{a,t})\|_2^2 + \|\bar{\boldsymbol{e}}_{s,t} - \text{sg}(\hat{\boldsymbol{z}}_{s,t})\|_2^2\right]. \tag{6}$$

码本使用分配给每一行的特征的指数移动平均（指数移动平均）进行更新（见第 III-B1 节）。分词器目标函数总体表示为：

$$\mathcal{L}_{\text{tok}} = \lambda_a\mathcal{L}_{\text{recons}} + \lambda_s\mathcal{L}_{\text{ctc}} + \lambda_{\text{int}}\mathcal{L}_{\text{inter-ctc}} + \beta\mathcal{L}_{\text{commit}}, \tag{7}$$

其中，λ_a、λ_s、λ_{int} 和 β 为损失加权系数。

2) BiSAT-TTS 生成：**BiSAT-TTS 文本语音语言模型**。文本语音语言模型自回归地生成编码。文本标记和语音编码按如下方式连接成单个序列，作为语言模型的输入：

$$\boldsymbol{x}_{\text{lm}} = \left[\langle\text{SOT}\rangle; y_{1:U}; \langle\text{EOT}\rangle; z_{1:T}; \langle\text{EOS}\rangle\right]^{(\text{time})}, \tag{8}$$

其中，⟨SOT⟩ 和 ⟨EOT⟩ 分别标记文本的开始和结束，⟨EOS⟩ 标记语音的结束。令 z_{T+1} 表示 ⟨EOS⟩。

文本语音语言模型以 θ 为参数化形式，在每一步预测 K 编码和序列结束标记的分布。我们可以将给定文本提示 $\boldsymbol{y}$ 时代码序列$\boldsymbol{z}_{1:T+1}$ 的条件似然分解为：

$$p_\theta(\boldsymbol{z}_{1:T+1} \mid \boldsymbol{y}) = \prod_{t=1}^{T+1} p_\theta\big(z_t \mid \boldsymbol{z}_{<t}, \boldsymbol{y}\big). \tag{9}$$

通过最小化代码序列 $\boldsymbol{z}_{1:T+1}$ 的负对数似然，可以优化参数 θ：

$$\mathcal{L}_{\text{lm}}(\theta) = -\sum_{t=1}^{T+1} \log p_\theta\big(z_t \mid \boldsymbol{z}_{<t}, \boldsymbol{y}\big). \tag{10}$$

在推理时，前缀 $\boldsymbol{x}_{\text{pre}}$ 由提示转录文本 $\tilde{\boldsymbol{y}}$、目标文本 $\boldsymbol{y}$ 和提示编码 $\tilde{\boldsymbol{z}}_{1:T'}$ 组成。文本语音语言模型根据此前缀自回归地生成目标编码，并在 ⟨序列结束标记⟩处停止：

$$\begin{aligned} \boldsymbol{x}_{\text{pre}} &= \big[\langle\text{SOT}\rangle;\, \tilde{\boldsymbol{y}};\, \boldsymbol{y};\, \langle\text{EOT}\rangle;\, \tilde{\boldsymbol{z}}_{1:T'}\big]^{(\text{time})}, \\ z_t &\sim q_t, \qquad t = 1, 2, \ldots, \end{aligned} \tag{11}$$

其中，q_t 是根据第III-B2节中详述的采样过程从语言模型分布中获得的。

BiSAT-TTS 声学解码器。声学解码器使用条件流匹配（CFM）[54]，对给定语音编码时梅尔频谱图的条件分布进行建模。我们采用高斯噪声 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 与目标梅尔频谱图 $\boldsymbol{s}$ [55] 之间的线性插值路径，

$$\boldsymbol{n}_t = t\boldsymbol{s} + (1-t)\boldsymbol{\epsilon}, \qquad t \in [0, 1]. \tag{12}$$

根据 [12]，，我们将训练时间步采样为 $t \sim \mathcal{U}(0,\ 1)$。对插值函数关于 t 求导，可得到条件速度场 $v(t)$：

$$v(t) = \frac{d\boldsymbol{n}_t}{dt} = \boldsymbol{s} - \boldsymbol{\epsilon}. \tag{13}$$

由 ϑ 参数化的神经网络会根据 t、$\boldsymbol{n}_t$、语音编码 $\boldsymbol{z}_{1:T}$、从参考语音中提取的提示梅尔频谱图 $\tilde{\boldsymbol{s}} \in \mathbb{R}^{F \times T_{\text{prompt}}}$，以及由冻结的说话人编码器提取的说话人条件 $\boldsymbol{\xi} \in \mathbb{R}^{192}$（例如 CAM++ [56]），估计条件均值速度。我们通过最小化训练分布 $\mathcal{D}$ 上预测速度与真实速度之间的期望平方误差来优化 CFM：

$$\mathcal{L}_{\text{cfm}}(\vartheta) = \mathop{\mathbb{E}}_{\mathcal{D},\, t,\, \boldsymbol{\epsilon}} \Big\| v_\vartheta\big(t, \boldsymbol{n}_t \mid \boldsymbol{z}_{1:T}, \tilde{\boldsymbol{s}}, \boldsymbol{\xi}\big) - v(t) \Big\|_2^2, \tag{14}$$

其中 $(\boldsymbol{s},\ \boldsymbol{z}_{1:T},\ \tilde{\boldsymbol{s}},\ \boldsymbol{\xi}) \sim \mathcal{D}$。

在推理时，我们使用一阶欧拉求解器求解以下常微分方程（ODE），以生成梅尔频谱图：

$$\frac{d\boldsymbol{s}_t}{dt} = v_\vartheta(t, \boldsymbol{s}_t \mid \boldsymbol{z}_{1:T}, \tilde{\boldsymbol{s}}, \boldsymbol{\xi}), \qquad \boldsymbol{s}_0 = \kappa\boldsymbol{\epsilon}, \tag{15}$$

从 $t = 0$ 到 $t = 1$。κ 是用于缩放噪声的温度。为求解公式 (15)，我们将在时间点 t_i 处对时间区间 $[0, 1]$ 进行离散化；这些时间点通过对均匀网格$u_i = i/J$（$i = 0, \ldots, J$）应用调度得到。J是积分步数。我们考虑余弦调度 $t_i = 1 - \cos(\pi u_i/2)$ [2], [13] 以及 PolyShift 调度 $t_i = u_i^\zeta / \big(u_i^\zeta + \sigma(1-u_i^\zeta)\big)$[57]，，其中更大的 ζ 和 σ 会将更多步数置于 $t = 0$附近。

B. 架构

1) ***编码器 T 双编码器***。声学编码器$\mathcal{E}_a$ 和语义编码器 $\mathcal{E}_s$ 各由 17 个 Conformer 模块组成（维度为 512，包含 8 个注意力头）[58]。Conformer 将卷积用于捕捉局部声学结构，并利用注意力机制处理较长范围的上下文。在训练期间，我们使用 NeMo 的频谱增强 [50], [59]对语义输入中的频带和时间片段进行遮蔽，以增强上下文建模能力。声学编码器接收未经增强的梅尔频谱图特征。令 $\mathcal{A}$ 表示遮蔽操作。编码器输出如下：

$$\begin{aligned} \boldsymbol{e}_a &= f_{\mathcal{E}_{\psi_a}}(\boldsymbol{m}), \\ \boldsymbol{e}_s &= f_{\mathcal{E}_{\psi_s}}\big(\mathcal{A}(\boldsymbol{m})\big), \end{aligned} \tag{16}$$

与 $\boldsymbol{e}_a \in \mathbb{R}^{D_a \times T}$ 和 $\boldsymbol{e}_s \in \mathbb{R}^{D_s \times T}$。随后对编码器输出进行 ℓ_2 归一化和量化。

共享索引向量量化器（SVQ）。受 TokenFlow [25], 共享索引原则的启发，我们设计了一种共享索引量化方案，使用两个大小为$K \times D_{\text{codebook}}$的码本，每个分支各一个，且 $D_{\text{codebook}} = 512$。根据式（4），量化器每帧选择一个共享索引 $z_t \in \{1, \ldots, K\}$，因此帧级离散瓶颈仍为 $\log_2 K$ 位。该设计有两个关键特性。

（*i*）独立的连续表示空间。训练期间，每个共享索引都会检索成对的声学和语义条目 $\boldsymbol{c}_{z_t,\ a}$ 和 $\boldsymbol{c}_{z_t,\ s}$，并将它们传递给各自的解码器。分别学习这两个连续空间和码本，使每个分支能够形成特定于任务的表示，而无需将声学细节和语言单元编码到同一个连续特征向量中。同时，式（4）中的联合距离使得在一个分支中相似、但在另一个分支中不同的帧能够分配到不同的码本索引，从而根据声学和语义结构对 K 成对条目进行划分。结合球面 k 均值初始化、EMA 更新和死码替换，BiSAT-32k 和 BiSAT-64k 在训练期间的码本利用率分别达到 100% 和 99.8%，如表 II 所示。

（*ii*）双向对齐。公式（4）中的共享索引 z_t 同时取决于 $\bar{\boldsymbol{e}}_{a,\ t}$ 和 $\bar{\boldsymbol{e}}_{s,\ t}$，依据联合声学距离和语义距离对帧进行分组。梅尔频谱图重建更新声学编码器，CTC 更新语义编码器，而公式（17）中的 EMA 更新则利用相同的已分配帧调整两个码本。通过共享索引分配，任一编码器中的更新都会影响两个分支的量化分配，从而使二者的优化相互耦合，而无需依赖固定的语义教师特征 [3], [18]，[19], [21], [22]。下游模型（如文本语音语言模型、声学解码器和理解探测任务）仅从分词器接收离散代码序列 $\boldsymbol{z}_{1:T}$，并在无法访问 $\boldsymbol{c}_a$ 或 $\boldsymbol{c}_s$的情况下学习各自的嵌入表；然而，它们仍能直接受益于代码流所捕获的声学结构和语言结构，相关效果将在第 IV 节中评估。

码本初始化与更新。我们从每种训练语言中采样第 1 阶段编码器特征（见第 IV-A 节），并使用公式 (4) 中的联合距离进行球面k均值聚类，以初始化码本。我们使用指数移动平均 [60] 对分配特征进行更新。由于两个分支在每个帧上选择相同的索引，因此它们共享一个指数移动平均计数

C_k 个被分配到索引 k 的帧。每个分支都维护各自的指数移动平均特征和 $\boldsymbol{S}_{k,\,b}$。对于每个索引 k 和分支$b \in \{a, s\}$，码本更新如下：

$$\begin{aligned} C_k &\leftarrow \eta C_k + (1-\eta)\, n_k, \\ \boldsymbol{S}_{k,b} &\leftarrow \eta \boldsymbol{S}_{k,b} + (1-\eta) \sum_{t:\, z_t = k} \bar{\boldsymbol{e}}_{b,t}, \\ \boldsymbol{c}_{k,b} &\leftarrow (\boldsymbol{S}_{k,b}/C_k) \,/\, \|\boldsymbol{S}_{k,b}/C_k\|_2, \quad (n_k > 0), \end{aligned} \tag{17}$$

其中，n_k 是当前批次中分配给 k 的帧数，η 是指数移动平均衰减因子。我们还使用 $n_k / \sum_{k'} n_{k'}$的指数移动平均来跟踪每个索引的使用情况，即当前批次中分配给索引 k的帧所占比例。每经过 Δ_{dead} 步，若某个编码对的指数移动平均频率低于ν_{dead}/K，就将其视为死码对。随后，我们从近期帧的储存池中抽取成对的编码器特征替换该编码对，并同时重置其嵌入和指数移动平均统计量。

双解码器。标记器解码器仅在训练期间使用，通过梅尔频谱图回归和 CTC 对两个分支进行监督，而非采用对抗式波形合成；波形生成则在 TTS 中单独处理。在每一帧，共享索引 z_t 选择声学和语义条目 $\hat{\boldsymbol{z}}_{a,\,t} = \boldsymbol{c}_{z_t,\,a}$ 以及 $\hat{\boldsymbol{z}}_{s,\,t} = \boldsymbol{c}_{z_t,\,s}$。设 $\boldsymbol{q}_b$ 表示式 (5) 中针对$b \in \{a, s\}$的旋转技巧参数化输出。声学解码器 $\mathcal{D}_a$ 由 9 个 ConvNeXt V2 块组成，[61] 并在第 4 和第 7 个块之后进行 $2\times$ 卷积时间上采样，将帧率从 25 Hz 提高到 100 Hz，遵循 [32], [62]。语义解码器 $\mathcal{D}_s$ 由 9 个 Conformer 块（512 维、8 个注意力头、2048 维前馈网络（FFN））组成，后接一个 CTC 头。两个解码器的输出为：

$$\begin{aligned} \hat{\boldsymbol{m}} &= f_{\mathcal{D}_{\gamma_a}}\big(\sqrt{D_a}\,\boldsymbol{q}_a\big), \\ p_{\gamma_s}(\pi_t \mid \hat{\boldsymbol{z}}_{\boldsymbol{s};1:T}) &= \Big[\operatorname{softmax}\Big(f_{\mathcal{D}_{\gamma_s}}\big(\sqrt{D_s}\,\boldsymbol{q}_s\big)\Big)\Big]_{t,\pi_t}. \end{aligned} \tag{18}$$

代码提取。提取编码时禁用频谱图增强。冻结的分词器同时使用两个编码器和相同的联合最近邻查找来生成 $\boldsymbol{z}_{1:T}$。这些标记作为语言模型目标和声学解码器输入。

2) BiSAT-TTS 文本语音语言模型：我们使用 MiniCPM4-0.5B [63]初始化文本语音语言模型的文本嵌入层和 Transformer 主干网络，并根据公式 (10) 继续使用 BiSAT 编码进行训练。与 MiniCPM4-0.5B 一样，该骨干网络采用分组查询注意力、SwiGLU 激活函数和带 LongRoPE 位置编码的 RMSNorm。标记 ⟨SOT⟩ 复用了 MiniCPM4 预训练的文本序列起始标记。我们随机初始化一个包含 $K+2$ 个条目的音频嵌入表，用于 K 语音编码、⟨EOT⟩和⟨EOS⟩，同时为 K 语音编码和 ⟨EOS⟩的 $K+1$个类别设置一个无偏置输出头。

解码采样。我们采用 top-k 和 top-p 采样，并使用重复惩罚 [15]。令 $\mathcal{H}_t = \{z_1, \ldots, z_{t-1}\}$ 表示目标话语中已生成编码的集合（不包括参考编码）。令 $o_t(v)$ 表示第 t 步候选编码 v 对应的 logit。对于 $v \in \mathcal{H}_t$，重复惩罚的应用方式如下：

$$\tilde{o}_t(v) = \begin{cases} o_t(v)/\rho, & o_t(v) \geq 0, \\ \rho o_t(v), & o_t(v) < 0, \end{cases} \tag{19}$$

其中，$\rho \geq 1$ 是重复惩罚。其他对数几率保持不变。我们通过使用温度 τ 进行缩放并应用softmax函数，从前k个和前p 个对数几率所构成的分布中进行采样。对于 U 目标文本词元和 T' 参考语音码（见公式(11)），令 $L_{\max}$ 和 $L_{\min}$ 分别表示最大和最小生成长度。受 [12],启发，我们设定 $L_{\max} = \min(L_{\text{cap}}, \lfloor r^{(\ell)}_{\max} U \rfloor, C_{\text{audio}} - T')$和$L_{\min} = \min(L_{\max}, \max(0, \lfloor r^{(\ell)}{}_{\min} U \rfloor))$，其中 $C_{\text{audio}} > T'$限制参考语音码与生成编码的总长度，L_{cap} 为生成上限长度，$r^{(\ell)}{}_{\min} r^{(\ell)}_{\max}$ 是语言 ℓ对应的码到文本比率。我们将低于 $L_{\min}$ 的EOS进行屏蔽，并在遇到EOS或达到 $L_{\max}$时停止。推理设置见第IV-A4节。

3) BiSAT-TTS 声学解码器：TTS声学解码器将离散编码映射为梅尔频谱图。声码器和梅尔特征的设置见第 IV-A2 节。

解码器架构。声学解码器由一个学习得到的码嵌入层（512 维）和一个扩散Transformer（DiT）速度解码器 [64] 组成。DiT 遵循 CosyVoice 3 声学解码器的骨干网络配置（1024 维、16 个注意力头和 2048 维前馈网络（FFN））[13]。码嵌入经过 [13] 中使用的预前瞻模块，并从 25 Hz 上采样至 50 Hz 的梅尔帧率。速度解码器堆叠了 22 个带旋转位置编码的 DiT 模块 [65]。根据公式（20），拼接输入的维度为 $D_{\text{feat}} = D_{\text{code}} + 2F + 192 = 864$。

前向传播。每个编码都通过学习得到的嵌入表 Emb_ϑ 映射为一个 D_{code}维向量，该表包含 K 个条目。随后，预前瞻模块 $\mathcal{P}_\vartheta$ 和上采样器 Up(·)生成长度为 T_{mel} 帧的码条件 $\boldsymbol{h}$。将提示梅尔频谱图 $\tilde{\boldsymbol{s}}_{\text{pad}}$ 补零至 T_{mel} 帧，并在各帧上进行广播后，速度解码器的输入如下：

$$\begin{aligned} \boldsymbol{u} &= \text{Emb}_\vartheta(\boldsymbol{z}_{1:T}) \in \mathbb{R}^{D_{\text{code}} \times T}, \\ \boldsymbol{h} &= \text{Up}\left(f_{\mathcal{P}_\vartheta}(\boldsymbol{u})\right) \in \mathbb{R}^{D_{\text{code}} \times T_{\text{mel}}}, \\ \boldsymbol{x}_t &= [\boldsymbol{n}_t;\, \boldsymbol{h};\, \boldsymbol{\xi}\mathbf{1}^\top;\, \tilde{\boldsymbol{s}}_{\text{pad}}]^{(\text{feat})}. \end{aligned} \tag{20}$$

推理时，拼接提示编码和生成的编码$[\tilde{\boldsymbol{z}}_{1:T'};\ \boldsymbol{z}_{1:T}]^{(\text{time})}$ 替换 $\boldsymbol{z}_{1:T}$，并在声码器合成前，从生成的梅尔频谱图中移除提示的初始帧。DiT $\mathcal{V}_\vartheta$ 按如下方式预测速度：

$$v_\vartheta(t, \boldsymbol{n}_t \mid \boldsymbol{z}_{1:T}, \tilde{\boldsymbol{s}}, \boldsymbol{\xi}) = f_{\mathcal{V}_\vartheta}(t, \boldsymbol{x}_t). \tag{21}$$

时间条件信息。时间步 t 的正弦嵌入经过一个带有 Swish 激活函数的两层多层感知机（MLP），并通过 AdaLN-Zero [64]注入每个 DiT 模块。

梅尔频谱图提示与说话人条件信息。我们将解码器的条件设为梅尔提示 $\tilde{\boldsymbol{s}}$，以提供参考韵律、音色和风格信息，同时使用 CAM++ 说话人嵌入 $\boldsymbol{\xi}$ [56]。训练期间，我们从目标话语中随机截取一个前缀作为 $\tilde{\boldsymbol{s}}$，其长度在话语长度的 0% 到 30% 之间均匀抽取；此外，为进行正则化，我们以概率$p_{\text{prompt}} = 0.5$ 丢弃此提示。请注意，CFM损失是在包括前缀在内的完整序列上计算的。推理时，两种条件信息均从参考语音中提取。

无分类器引导。在训练期间，我们以概率 $p_{\text{cfg}} = 0.2$ 同时丢弃$\boldsymbol{h}$、$\boldsymbol{\xi}$和 $\tilde{\boldsymbol{s}}$，该概率独立于提示词丢弃进行采样，以学习无条件速度预测[66]。在推理时，无分类器引导（CFG）将

表 I 多语言训练数据。

语言	小时（千）	来源
英语	51.7	Emilia [67],LibriTTS-R [68],GLOBE V2 [69],VCTK [70]
中文	42.6	Emilia，AISHELL-3 [71]
德语	8.4	Emilia，Emilia-YODAS [72], CML-TTS [73], HUI [74]
法语	8.1	Emilia , Emilia-YODAS , CML-TTS
韩语	7.1	Emilia , Emilia-YODAS
日语	2.5	Emilia、Emilia-YODAS、JVS [75]
荷兰语	0.756	CML-TTS
西班牙语	0.436	CML-TTS
意大利语	0.129	CML-TTS
葡萄牙语	0.077	CML-TTS
波兰语	0.044	CML-TTS
总计	122	–

条件速度预测和无条件速度预测 v_{cond} 与v_{uncond} 按照引导尺度 g 进行组合：

$$v_{\text{cfg}} = v_{\text{cond}} + g\big(v_{\text{cond}} - v_{\text{uncond}}\big). \tag{22}$$

流式支持。 随后， [2], [13], 每个训练批次以概率 $p_{\text{offline}} = 0.5$使用完整的双向注意力，否则使用分块大小为 $M = 50$ 梅尔帧的分块因果注意力。因此，单个检查点既可在离线模式下运行，也可在流式模式下运行：每次解码 $M/2$ 个编码，并采用 3 个编码的前瞻，再将生成的梅尔帧传递给因果声码器。

IV. 实验

A. 实验设置

1) Training Data and 预处理： 我们在涵盖11种语言、约122 k小时语音的数据上训练BiSAT分词器和BiSAT-TTS。表I列出了所有训练数据集及其时长，并按话语语言进行分组。对于码本初始化，我们使用第一阶段编码器，从按语言和时长采样的训练语句中提取特征（每种语言6144条话语，时长区间为2、5、10和20秒）。初始化池包含约130万个帧。随后，在式（4）的联合距离下，球面k均值在该池上运行12轮，每次处理1024个帧；对于空聚类，则从量化误差最大的帧中重新播种。

2）声学特征和声码器： 分词器处理16 kHz语音，使用100 Hz、80个频段的对数梅尔频谱图，并对每条话语、每个频段进行归一化，将其同时作为输入和重建目标。编码器将该表示下采样四倍，生成25 Hz编码。对于TTS，我们使用CosyVoice的预训练因果HiFT声码器 3 [13],，其梅尔配置采用24 kHz语音、80个梅尔频段以及480个采样点的跳跃大小，对应50 Hz的帧率。

3) 训练细节： 我们为每个组件使用八张 NVIDIA RTX 6000 Ada GPU，包括两个标记器训练阶段、文本语音语言模型和 CFM 声学解码器。标记器训练完成后，我们冻结标记器并提取代码序列，以训练语言模型和声学解码器。

BiSAT分词器。 两个编码器均初始化自英语预训练的17层NeMo自监督学习模型 ssl_en_conformer_large [50]。[1] CTC 头使用16 k多语言字节对词表。我们的默认配置使用 $K = 32768$（BiSAT-32k，375 bps），此外还在相同的训练设置下（第IV-B4 节）训练$K = 65536$ （BiSAT-64k，400 bps）。训练分为两个阶段。首先，在各自的重建损失或 CTC损失下，声学分支和语义分支分别进行训练，不使用 VQ，每个分支训练100 k步。其次，我们插入基于第1阶段编码器特征初始化的 SVQ，并根据公式（7）联合训练两个分支200 k步。在公式（4）中，我们设定 $\lambda_a = 1$、 $\beta = 0.5$、 $\omega = 1$，并按照 OWSM-CTC[52]设定 $\lambda_s = 0.7$ 和 $\lambda_{\text{int}} = 0.3$ 。指数移动平均衰减因子为 $\eta = 0.99$，每 $\Delta_{\text{dead}} = 100$ 步应用一次死码替换，阈值为$\nu_{\text{dead}} = 0.01$。两个阶段均使用 AdamW，预热5 k步，采用 BF16 混合精度，并将每个 rank 的批次预算设为312.5秒语音。预热后，第1阶段的学习率保持为10^{-4}；第2阶段则保持为 7.5×10^{-5}直至100 k步，随后在线性衰减至 7.5×10^{-6}，覆盖剩余的100 k步。语义分支上的频谱图增强应用两个最多包含27个频点的频率掩码，以及十个最多覆盖序列长度5%的时间掩码。

文本语音语言模型。 我们以 MiniCPM4-0.5B 为基础，采用 BF16 混合精度微调所有参数，训练 400 k 步；每个 rank 的批次预算为 24 k 个文本和语音标记。训练使用 AdamW，峰值学习率为 10^{-4}，线性预热 10k 步，并采用余弦衰减至 5×10^{-6}。我们为其文本分词器扩展了 2,916 个单字符中文、日语和韩语标记。我们保留 MiniCPM4 的输入、残差分支和输出缩放因子，分别为 12、 $1.4/\sqrt{24}$ 和 $256/1024$。

声学解码器。 我们使用 BF16 混合精度训练 100 万步，每个 rank 的批次预算为 375 秒目标语音，并采用 AdamW [79] ，峰值学习率为 10^{-4}，进行 2 万步线性预热，然后线性衰减至 10^{-6}。

4）推理细节： 对于声学解码器，我们使用欧拉求解器，设置为 $J = 10$、 $\kappa = 1$ 和 CFG 缩放系数 $g = 0.7$ ，并采用余弦调度或 PolyShift 调度，参数为 $\zeta = 2$ 和 $\sigma = 1$。所有重建和零样本文本转语音基准测试均使用离线解码，以便与离线基线方法直接比较。根据文本语音语言模型采样的推理超参数扫描结果，我们为 BiSAT-TTS-32k 设置 top-$p = 1.0$、 $k = 25$、重复惩罚 $\rho = 1.05$ 和温度 $\tau = 0.8$ ；为 BiSAT-TTS-64k 设置 top-$p = 0.9$、 $k = 25$、 $\rho = 1.1$ 和 $\tau = 0.9$。计算长度时，我们排除 $\langle$SOT$\rangle$ 和 $\langle$EOT$\rangle$。我们将英语和普通话的 $r^{(\ell)}{}_{\text{最小}} = 1$ 分别设为 $r_{\max}^{(\ell)} = 30$ 和 25，将英语和普通话的 $r_{\max}^{(\ell)} = 30$ 分别设为 $C_{\text{audio}} = 2048$ 和 $L_{\text{cap}} = 1536$。在所有运行中，BiSAT-TTS 均未生成空输出，且有一条 BiSAT-TTS-32k 话语达到 $L_{\max}$。

分词器基线。 我们将 BiSAT 与第 II-A 节各类别中公开可用的分词器进行比较，测试数据集为 LibriSpeech-PC test-clean [76] 和 Common Voice 27[77], ，使用其官方检查点且不进行重新训练。这些基线包括 AOT：FocalCodec [16] 和 WavTokenizer[17], ；SET：CosyVoice 2 [2] 和 CosyVoice 3 [13],；以及 UT：SpeechTokenizer [18], 、LM-SPT [22], 和 X-Codec2[78]。由于 SpeechTokenizer 和 LM-SPT 依赖多个 RVQ 流来实现完整重建，因此我们将其纳入重建比较，并将单流理解探测任务限制在单码本分词器上。遵循

[1] https://catalog.ngc.nvidia.com/orgs/nvidia/nemo/models/ssl_英语_conformer_large

表 II 语音分词器规格及其在 LibriSpeech-PC test-clean [40], [76] 和 Common Voice 27 [77], 上的重建结果：英语、法语和德语采用 WER，中文、日语和韩语采用 CER。BiSAT-32k/64k 使用 $K = 32768/65536$。UTIL. 指 BiSAT 在训练语料库上的结果，以及基线方法根据原始报告得出的结果。N_q 表示 RVQ 层，cosine/PolyShift 表示声学解码器 ODE 调度方案。**BOLD** 和下划线分别表示最佳和次佳结果。

类别	分词器	规格					LibriSpeech-PC			Common Voice WER/CER↓					
		Rate (Hz)	比特率（bps）	码本 size	利用率 (%)	Data (h)	词错误率↓	SIM↑	UTMOS↑	英语	法语	德语	中文	日语	韩语
–	真实值	–	–	–	–	–	1.89	1.000	3.21	11.01	13.27	6.35	9.93	15.06	6.51
声学	FocalCodec [16]	25	325	8192	99.8	585	2.94	0.960	2.94	25.52	–	–	–	–	–
	WavTokenizer [17]	40	480	4096	100	80k	9.22	0.933	3.00	47.04	–	–	–	–	–
语义	CosyVoice 2 [2]	25	317	6561	100	200k	3.21	0.968	**3.51**	24.96	31.57	20.61	15.02	25.09	15.34
	CosyVoice 3 [13]	25	317	6561	100	530k	2.26	0.969	3.25	17.49	20.36	12.30	13.19	<风格 id='1'>19.33</风格>11.89	
统一	SpeechTokenizer（$N_q = 1$） [18]	50	500	1024	95.9	960	4.59	0.828	1.78	54.51	–	–	–	–	–
	SpeechTokenizer ($N_q = 2$)	50	1000	1024	95.9	960	3.65	0.905	1.54	46.40	–	–	–	–	–
	LM-SPT ($N_q = 2$) [22]	12.5	325	16384/4096	86/98	100k	2.55	0.935	2.67	23.63	27.31	18.79	16.33	25.22	16.47
	X-Codec2 [78]	50	800	65536	99.0	150k	2.27	**0.981**	3.23	19.13	20.74	13.10	13.72	21.09	11.86
Ours	BiSAT-32k（余弦）	25	375	32768	100	122k	2.18	0.976	3.26	16.49	19.35	11.81	13.14	21.10	10.40
	BiSAT-32k（PolyShift）	25	375	32768	100	122k	**2.15**	0.976	3.26	16.33	18.92	11.90	12.93	21.25	10.34
	BiSAT-64k（余弦）	25	400	65536	99.8	122k	2.16	0.977	3.28	**16.17**	18.79	**11.76**	**12.80**	21.47	**10.15**
	BiSAT-64k（PolyShift）	25	400	65536	99.8	122k	2.18	0.976	3.28	16.22	<风格 id='1'>18.62</风格>11.91	<风格 id='3'>12.80</风格>21.42	10.25		

表 III 离散编码上的理解评估。英语、法语和德语的自动语音识别结果以词错误率（WER）报告，中文和日语的结果以字符错误率（CER）报告。自动语音识别探针在 Common Voice 和 FLEURS 上进行零样本评估。LibriSpeech test-clean 上的音素识别（PR）以音素错误率（PER）报告；VoxCeleb1 测试集上的自动说话人验证（ASV）以等错误率（EER）报告；VoxCeleb1 上的说话人识别（SID）以准确率报告；FLEURS 测试集和 VoxLingua107 开发集上的语言识别（LID）以准确率报告。IEMOCAP 上的情感识别（ER）结果为五折加权准确率的平均值。

类别	分词器	ASR↓										PR↓	自动说话人验证↓	说话人识别↑	情感识别↑	语言识别↑
		Common Voice					FLEURS									
		EN	FR	DE	ZH	JA	EN	FR	DE	ZH	JA					
声学	FocalCodec [16]	86.71	–	–	–	–	85.96	–	–	–	–	8.33	18.0	31.3	47.5	–
	WavTokenizer [17]	110.48	–	–	–	–	99.63	–	–	–	–	30.62	28.2	10.1	46.9	–
语义	CosyVoice 2 [2]	81.71	67.07	63.27	39.23	46.17	55.74	66.43	64.59	32.04	29.66	7.71	19.4	24.8	48.6	82.0
	CosyVoice 3 [13]	74.17	57.80	54.34	34.18	**41.37**	47.17	56.04	52.33	28.16	**26.34**	5.68	18.6	24.7	**54.7**	90.6
统一	X-Codec2 [78]	89.67	75.01	80.76	51.43	54.72	87.66	71.20	75.77	46.45	49.08	8.82	23.2	5.6	44.5	23.6
Ours	BiSAT-32k	**37.71**	**52.55**	**43.95**	**32.55**	42.29	**24.42**	**49.21**	**39.54**	**26.68**	27.99	**1.92**	**14.3**	**40.4**	48.8	**95.5**

[9]，令 N_q 表示所使用的 RVQ 层数。对于 SpeechTokenizer，我们按照 X-Codec2[78], 中的设置评估 $N_q \in \{1, 2\}$；对于 LM-SPT，我们使用 $N_q = 2$，使其比特率与 BiSAT 的相当。值得注意的是，CosyVoice 分词器是与 BiSAT 最具可比性的基线，因为它们同样以相近的比特率生成单个 25Hz 代码流，并通过 CFM 声学解码器重建语音。具体而言，BiSAT 的 CFM 声学解码器采用与 CosyVoice 3 相同的 DiT 骨干配置，但使用 512 维编码和 192 维说话人条件，而不是 80 维投影。

TTS基线。我们将 BiSAT-TTS 与使用自回归语言模型生成离散语音令牌的零样本文本转语音系统进行比较，采用各自的官方检查点和默认推理设置。基线方法包括 CosyVoice 1、2，以及 3 [2], [12], [13],Spark-TTS [14], VoiceStar [31], 和基于 X-Codec2 构建的 LLaSA-1B 和 LLaSA- 3B [3] 。对于 VoiceStar，所需的目标时长按照 [31] 中的方法，根据提示的语速进行估计。需要注意的是，CosyVoice 系列和 BiSAT-TTS 的声学解码器以参考梅尔提示和说话人嵌入为条件，而 Spark-TTS、VoiceStar 以及 LLaSA 仅根据代码序列合成波形。

5) 评估指标：**分词器评估**。重建和理解评估编码中保留的声学与语义信息。

重建。感知自然度 通过神经网络平均意见分（MOS）预测器 UTMOSv2 [80] 进行测量。说话人相似度 （SIM）是使用 WavLM-Base+ 说话人验证模型 [46]提取的重合成语音与原始语音的说话人嵌入之间的余弦相似度。在 LibriSpeech-PC 上，我们通过 Whisper-large-v3 [81] 的词错误率（WER）来测量可懂度 。CosyVoice 2、CosyVoice 3 和 BiSAT 无需 mel 提示，使用从目标话语中提取的编码和说话人条件来重建语音。其他分词器则仅根据编码进行解码。所有系统都在没有单独参考话语的情况下，重建相同的输入话语。针对更具挑战性的真实场景设置，我们还使用相同的重建协议，在众包 Common Voice 27 [77]测试集的英语、法语、德语、普通话、日语和韩语上进行评估。英语测试集排除了 Seed-TTS test-en 中使用过的片段和说话人。

对于普通话，我们使用 Paraformer-zh 计算字符错误率（CER）。对于其他语言，我们使用带有 Whisper 规范化器的 Whisper-large- v3，并报告日语和韩语的 CER，以及英语、法语和德语的 WER。

理解。我们将 SUPERB [28] 和 ML-SUPERB [29] 协议调整为使用离散编码，而不是连续自监督学习特征。对于每个分词器，我们在冻结的编码上，以该分词器自身的帧率训练探针。该探针由一个经过学习的代码嵌入表和一个 8 层、512 维的 Transformer 编码器组成；对于识别任务，后接 CTC 头，对于其他任务，则使用注意力池化和分类头。所有分词器针对每项任务均采用相同的探针和训练方案。

英语任务遵循 SUPERB。其他语言的自动语音识别和语言识别（LID）遵循 ML-SUPERB。对于英语自动语音识别，探针在 LibriSpeech train-clean-100 [82]上训练。对于其他语言，我们遵循 ML-SUPERB 的单语自动语音识别设置，在单一语料库的 100 小时数据上为每种语言分别训练一个探针。具体而言，法语和德语使用 MLS [83] 的 100 小时子集，普通话使用 WenetSpeech [84] 的子集 S，日语使用 ReazonSpeech [85] 的小型子集。我们仅在零样本域外设置下评估自动语音识别，对每种语言使用 Common Voice [77] 和 FLEURS [86] 测试集。这些语料库不用于训练自动语音识别探针。我们在 LibriSpeech train-clean- 100 上训练音素识别（PR）探针，并在 test-clean 上测试。对于自动说话人验证（ASV）和说话人识别（SID），我们使用 VoxCeleb1[87]。对于情感识别（ER），我们在 IEMOCAP [88], 、CREMA-D [89], 、RAVDESS [90],和 ESD [91]的英语部分上训练一个共享的七分类头。我们按照 SUPERB 的五折留一会话法协议，在 IEMO- CAP 上进行评估。遵循 ML-SUPERB 的多语言语言识别设置，单个语言识别探针覆盖 12 种语言（普通话、粤语、英语、日语、韩语、法语、德语、荷兰语、意大利语、波兰语、葡萄牙语和西班牙语）。该探针在 FLEURS 和 VoxLingua107 [92]上训练，并在合并后的 FLEURS 测试集和 VoxLingua107 开发集上测试。自动语音识别报告词错误率（WER），普通话和日语则报告字符错误率（CER）；PR 报告音素错误率（PER）；ASV 报告等错误率（EER）；SID 和 LID 报告准确率；ER 报告五折上的加权准确率。

文本转语音评测。我们按照 VALL-E 协议 [1] 和 [40]的评测设置，在 LibriSpeech-PC test-clean [76],上评测跨句零样本文本转语音。提示音频是来自同一说话人的另一段话语，文本语音语言模型接收带标注的提示转录文本，随后接收目标文本。我们还在 Seed-TTS 的 test-zh 和 test-en [93]上评测跨句设置。在 LibriSpeech-PC 上，我们报告 Whisper-large-v3 结合其文本规范化器得到的词错误率、使用 WavLM-Base+ 说话人验证模型 [46], 计算的相对于原始提示音频的说话人相似度（SIM-o）、用于评估感知自然度的 UTMOSv2，以及用于评估整体语音质量的 NISQA [94] 。在 Seed-TTS 上，我们使用官方 WavLM-Large ECAPA-TDNN 模型报告 SIM-o，同时报告 test-en 上 Whisper-large-v3 的词错误率、test-zh 上 Paraformer-zh [95]的字符错误率，以及两组数据上的 UTMOSv2 和 NISQA。我们对三个随机种子下的话语级分数取平均。对于词错误率和字符错误率，我们将空音频的错误率设为 100%，并对非空音频的错误率不设上限，同时保留非空音频的错误率不设上限。对于相似度和质量指标，我们对可评分音频的分数取平均。失败率（失败）是所有评测数据集中的音频片段里，空音频或词错误率、字符错误率至少为 100% 的片段所占的百分比。对于 test-en，为获得更准确的结果，我们使用 Whisper 规范化器，而不是 Seed-TTS 工具包中的规范化器，因为前者会排除仅存在于书写形式、但发音相同的差异，例如数字格式差异。

B. 实验结果

1) 分词器评估：**重建**。表 II 显示，BiSAT 无需对抗式波形训练，即可在低比特率下兼顾可懂度和说话人保真度。在比特率相近的分词器中，BiSAT 在 LibriSpeech-PC 上的可懂度和说话人相似度最高，UTMOSv2 仅落后于 CosyVoice 2。尽管离散编码和目标说话人条件共同影响 BiSAT 与 CosyVoice 分词器的 SIM，但 BiSAT-32k 在表 III 的 SID 和自动说话人验证（ASV）上也表现领先；这两项指标剥离了说话人条件的影响，表明编码本身保留了丰富的说话人信息。此外，尽管训练数据不到 CosyVoice 3 的四分之一，BiSAT 相较于 CosyVoice 3 的优势仍从干净的 LibriSpeech-PC 上的 0.11% 扩大到野外 Common Voice 27 数据集上英语、法语、德语和韩语的 0.4%–1.7%，日语是唯一的例外。这些结果表明，与纯语义分词或特征融合分词相比，BiSAT 的双向对齐生成的编码在不同口音、麦克风和背景噪声下具有更好的泛化能力。

在仅使用编码并通过对抗训练重合成波形的编解码器中，WavTokenizer 的 UTMOSv2 接近真实值，但其 WER 约为 FocalCodec 的三倍。因此，听起来自然的输出并不能确保内容得到保留，正如第 II-A2 节所述。X-Codec2 的重建 WER 接近 BiSAT，但在表 III 中，基于其编码训练的探针在语音和音素识别任务上的表现明显更差，这表明仅有较低的重建误差并不意味着语言信息能够从编码中轻易获取。

理解。表 III 中的冻结编码探针通过 ASR、PR 和 LID 评估语言内容，通过 SID 和 ASV 评估说话人信息，并通过 ER 评估副语言信息。CosyVoice 分词器在大多数内容任务中排名第二和第三，但其说话人识别准确率低于 BiSAT 和 FocalCodec（24.8% 和 24.7%，相比之下为 40.4% 和 31.3%），且自动说话人验证错误率更高。说话人识别依赖频谱和韵律线索 [96], 因此，这些结果表明，仅针对理解任务训练的 SET 所保留的说话人相关声学细节少于 BiSAT。FocalCodec 则呈现相反的模式，其说话人识别准确率高于 SET，但识别错误更多。

如重建结果和探测结果所示，对抗性训练的编解码器在两个测试数据集上都具有较高的识别错误率，即使是采用语义监督的 X-Codec2 也不例外，这表明对抗性波形重建并不能确保编码模式能够跨语料库泛化。BiSAT 在三个任务组上的表现均优于 X-Codec2，而其比特率不到后者的一半。不同于 X-Codec2 在量化前将声学特征和

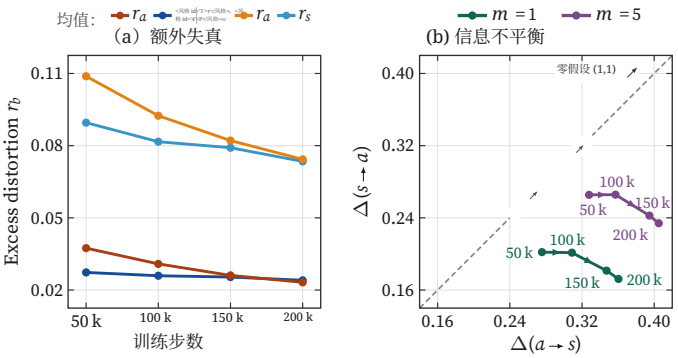

图 2. 联合 SVQ 轨迹：(a) 在 Seed-TTS 的 test-en 和 test-zh 上取平均的额外失真 r_b；(b) 方向性信息不平衡。

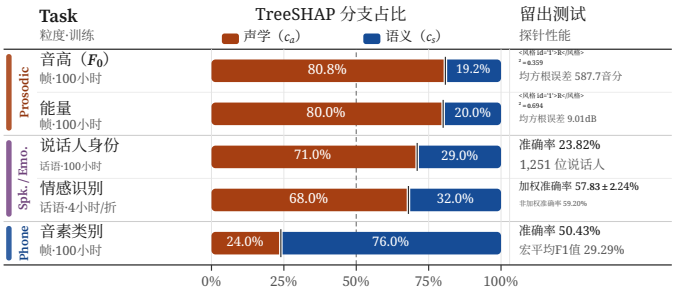

图 3。配对码本 $\boldsymbol{c}_a$ 和 $\boldsymbol{c}_s$ 之间按任务划分的 TreeSHAP 归因份额，以及留出集 XGBoost 探针性能。

语义特征融合为一种表示（见第 II-A3 节），BiSAT 在训练期间通过独立分支学习连续表示，同时在每个帧选择一个共享码本索引，以形成单一代码流。尽管两个系统在编码器架构和训练监督方面也存在差异，因此仅凭跨系统比较无法将这一因素单独剥离出来（见第 IV-B2 节），但这些结果仍表明，BiSAT 的代码流在紧凑的单流框架内实现了很强的表示质量。除日语 ASR外，BiSAT在所有语言和说话人相关任务上也均领先；在日语ASR任务中，CosyVoice 3 在两个语料库上均领先，这可能得益于其规模大得多的日语训练数据 [13]。结合重建结果，这些发现表明，BiSAT 保留了对语言理解和声学重建均有用的信息。

ER 是一个例外：CosyVoice 3 以明显优势领先，而 BiSAT 与 CosyVoice 2 和 AOT 不相上下。一个可能的原因是，CosyVoice 3 的分词器采用包含 ER 在内的多任务监督进行训练，而 BiSAT 未接收任何副语言标签。将此类监督加入语义分支将留待未来研究。

此外，ASR、PR、SID 和 ASV 的探测结果均表现强劲，这表明 BiSAT 编码还可用于强化学习（RL）后训练中的令牌级奖励建模（例如 Token2Text 内容和多任务奖励 [13]）。

2）诊断分析： 为分析 BiSAT 如何组织其配对码本，我们通过额外失真考察分配对齐，通过方向性信息不平衡考察码本几何结构，并通过 TreeSHAP 归因考察分支专门化。

分支级过量量化失真。 根据公式 (4)，两个分支在每个帧共享索引 z_t，因此 z_t 不一定是每个分支自身码本中的最近邻。对于分支 $b \in \{a, s\}$ 和帧 t，令 $d_{b,t}(k) = \|\tilde{e}_{b,t} - \boldsymbol{c}_{k,b}\|_2^2$表示经过 ℓ_2 归一化后的平方欧氏距离。由于在码本 [97], 固定时，最近邻查找会使失真最小化，因此我们将索引共享带来的额外失真定义为 $r_{b,t} = d_{b,t}(z_t) - \min_k d_{b,t}(k) \geq 0$；仅当 z_t 是分支 b自身的最近条目时，该值才为零。我们在 Seed-TTS 的 test-en（1,088）和 test-zh（2,020）测试集上，于 50 k、100 k、150 k 和 200 k 个第二阶段步数进行评估，报告两个数据集平均后的帧加权均值和第 95 百分位数（P_{95}）。如图 2(a) 所示，训练期间，r_a和 r_s 的均值与 P_{95} 均稳步下降：声学分支起始值较高，但下降更快；到 200k 步时，二者都收敛到几乎相同的水平，表明联合训练逐步对齐了它们的量化分配。

方向性信息不平衡。 额外失真反映的是共享 z_t 所产生的帧级成本，而非 $\boldsymbol{c}_a$ 和 $\boldsymbol{c}_s$ 的相对几何结构。为衡量一个码本中的相邻索引在另一个码本中是否仍保持相邻关系，我们计算方向性信息不平衡 [98]。设 $\mathcal{I} \subset \{1, \ldots, K\}$ 为 $V = |\mathcal{I}|$ 个活跃码本索引的固定子集，并在每个码本中经过 ℓ_2 归一化后计算平方欧氏距离。对于每个查询索引 $i \in \mathcal{I}$，令 $\mathcal{N}_m^{(a)}(i) \subset \mathcal{I} \setminus \{i\}$ 表示其在 $\boldsymbol{c}_a$ 中的 m 个最近邻索引，并令 $R_s(i, j) \in \{1, \ldots, V-1\}$ 表示索引 j 相对于 i 在 $\boldsymbol{c}_s$ 中的距离排名。我们将从 $\boldsymbol{c}_a$ 到 $\boldsymbol{c}_s$ 的信息不平衡定义为：

$$\Delta(a \to s) = \frac{2}{V} \cdot \frac{1}{Vm} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{N}_m^{(a)}(i)} R_s(i, j), \qquad (23)$$

并且通过交换 a和 s，可以对称地定义 $\Delta(s \to a)$。在随机索引配对下，预期秩为$V/2$，因此 $2/V$ 将随机基线归一化为 $\mathbb{E}[\Delta] = 1$；而较小的数值意味着，源码本中的 m近邻在目标码本中仍然彼此接近。在相同的四个检查点，我们针对$V = 28$, 中最常出现的 173 个索引（占汇总训练 EMA 使用量的 95%）评估 $m \in \{1, 5\}$，并与 199 次随机置换进行比较。在图 2(b) 中，所有检查点都明显低于 (1，1) ($p = 0.005$)。对于 $m = 1$, $\Delta(s \to a)$ 从 0.20 降至 0.17，而 $\Delta(a \to s)$ 从 0.28 升至 0.36；在 150 k 步之后，变化幅度变小（ $m = 5$ 也类似）。因此，$\boldsymbol{c}_s$ 中的近邻会映射到$\boldsymbol{c}_a$中更接近的秩，而反向映射则有所减弱，这可能是因为相似的声学帧出现在不同的音系语境中。由此可见，两个码本空间在演化过程中保持着非随机的邻域对应关系，同时呈现出不对称性。由于方向性不平衡反映的是几何对应关系，而非因果信息传递或梯度影响，因此这些变化轨迹与耦合表示学习是一致的。

任务级码本归因。 为了衡量每个码本对各项语音属性的贡献，我们在冻结的连续码本向量上训练 XGBoost 探针 [99]，并计算分支级 TreeSHAP 归因 [100]（不同于表 III 中针对离散 ID 的序列探针）。帧级探针直接使用 $[\boldsymbol{c}_{z_t,a};\ \boldsymbol{c}_{z_t,\ s}] \in \mathbb{R}^{1024}$ 而话语级探针则对拼接前每个 512 维分支取均值池化。对于每个测试样本，我们对每个分支的 512 个 TreeSHAP 值求和，取绝对值，并将数据集均值归一化为分支占比。音高（F_0）、能量和音素探针基于

表 IV 在 LibriSpeech-PC 和 Seed-TTS 上使用三个随机种子进行的跨句零样本文本转语音结果。UTMOS 为 UTMOSv2，NISQA 为总体质量，FAIL 表示空音频或 $\geq 100\%$ WER/CER 音频片段所占的比例，Multi. 表示多个数据集，-32K/-64K 表示 $K = 32768/65536$。

系统	参数 (M)	Data （小时）	LibriSpeech-PC				Seed-TTS (test-en)				Seed-TTS (test-zh)				失败↓
			(%) 词错误率↓	SIM-o↑	UTMOS↑	NISQA↑	词错误率↓	SIM-o↑	UTMOS↑	NISQA↑	字符错误率↓	SIM-o↑	UTMOS↑	NISQA↑	
真实值	–	–	1.89	0.944	3.21	4.07	1.47	0.734	2.75	3.63	1.43	0.755	2.94	4.08	–
CosyVoice 1 [12]	300	172k 内部数据	2.87	0.906	3.25	4.26	3.85	0.512	3.10	4.13	3.59	0.643	3.00	**4.35**	0.157
CosyVoice 2 [2]	500	167k 内部数据	1.78	0.947	**3.58**	**4.40**	2.12	0.659	**3.26**	4.28	1.61	0.752	**3.10**	4.34	0.039
CosyVoice 3 [13]	500	1M 内部数据	1.72	**0.951**	3.43	4.28	1.55	**0.698**	3.07	4.15	1.43	**0.779**	2.89	4.22	0.063
Spark-TTS [14]	500	10万 VoxBox	2.69	0.938	3.57	4.07	2.61	0.571	3.16	4.07	2.70	0.658	3.05	4.09	0.756
VoiceStar [31]	840	6.5万多语言	1.94	0.944	3.26	4.34	2.28	0.606	2.92	4.28	–	–	–	–	0.166
LLaSA-1B [3]	1000	25万多语言	2.00	0.930	3.44	4.23	1.91	0.588	3.14	4.10	2.27	0.675	2.99	4.13	0.409
LLaSA-3B	3000	25万多语言	2.17	0.931	3.43	4.22	2.38	0.594	3.11	4.12	1.69	0.682	2.96	4.12	0.512
BiSAT-TTS-32k（余弦）	500	122k 多语言	1.56	0.950	3.43	4.37	1.30	0.669	3.03	4.27	1.39	0.734	2.89	4.28	0.031
BiSAT-TTS-32k（PolyShift）	500	122k 多语言	**1.50**	0.950	3.42	4.38	1.31	0.669	3.04	4.29	1.39	0.735	2.90	4.29	0.024
BiSAT-TTS-64k（余弦）	500	122k 多语言	1.52	0.949	3.43	4.39	1.23	0.667	3.05	4.29	1.29	0.733	2.93	4.30	**0.016**
BiSAT-TTS-64k（PolyShift）	500	122k 多语言	1.51	0.949	3.43	4.39	**1.21**	0.667	3.06	**4.30**	**1.27**	0.734	2.92	4.30	**0.016**

LibriSpeech train-clean-100 进行训练，并在 test-clean 上评估，使用来自 WORLD[2] 的有声 F_0 以及来自 Montreal Forced Aligner 的音素对齐结果。[3] 我们使用 VoxCeleb1 进行 SID，并在 IEMOCAP 上进行五折交叉验证，以完成四分类 ER。由于两个分支共享 z_t，这些占比反映的是探针对$\boldsymbol{c}_a$ 和 $\boldsymbol{c}_s$ 的依赖，而非预测准确率。在图 3 中，尽管 F_0 和能量的 $\boldsymbol{c}_a$ 得分不同（0.359 对 0.694），探针主要依赖 $\boldsymbol{c}_a$（约 80%）；对于音素，则依赖 $\boldsymbol{c}_s$（76%）；而说话人和情感探针则同时利用两个码本。

3) 零样本文本转语音：**可懂度**。如表 IV 所示，BiSAT-TTS 的词错误率、字符错误率和失败率均低于所有基线方法，且显著优于 Spark-TTS 和 LLaSA。与 CosyVoice 3 相比，在相同的 0.5B 语言模型规模、25 Hz 帧率和 DiT 骨干网络条件下，BiSAT-TTS 在每个数据集上的错误率都低得多，而其训练集规模仅约为对方 100 万小时训练集的八分之一 [13], ，将 LibriSpeech-PC 上的词错误率优势从重建任务中的 0.11 个百分点扩大到文本转语音任务中的 0.23 个百分点。对比表 II、表 III 和表 IV，可以看清编码结构如何影响自回归建模，尤其是 X-Codec2。尽管 X-Codec2 在 LibriSpeech-PC 重建任务中的词错误率接近 BiSAT（2.27% 对 2.15%），且比特率超过其两倍，但其自动语音识别、音素识别和说话人识别探测任务表现较弱（表 III），这表明基于AT的重建即便没有结构清晰的编码，也能实现较低的错误率。因此，尽管 LLaSA-1B 和 LLaSA-3B 的语言模型规模最高可达 6× 且训练数据量是 BiSAT-TTS 的两倍，它们在所有数据集上的错误率和失败率仍远高于 BiSAT-TTS。由此可见，强大的重建能力可能掩盖结构较弱的编码，而这类编码会妨碍自回归生成；相比之下，BiSAT 较低的错误率和失败率以及表 III 的结果表明，保留易于利用的语义和声学结构能够产生更适合自回归语音建模的编码表示。

说话人相似度。BiSAT-TTS 在 LibriSpeech-PC 上的表现已接近 CosyVoice 3 的 SIM-o，在 Seed-TTS test-en 上仅次于 CosyVoice 3。在 test-zh 上与 CosyVoice 2 和 3 的差距较大，这可能反映的是它们使用了更多普通话训练数据，而非分词器的瓶颈；这与 BiSAT 在表 III 中领先的说话人识别和自动说话人验证得分一致。

感知自然度和语音质量。与 CosyVoice 3 相比，BiSAT-TTS 在三个数据集上的 UTMOSv2 得分均相当，而 CosyVoice 2 和 Spark-TTS 的得分最高。不同于使用 AT 波形解码器的 LLaSA 和 Spark-TTS， [3], [14], BiSAT-TTS 与 CosyVoice 会根据参考梅尔提示对流匹配解码器进行条件化。由于提示条件流匹配往往会保留提示声学特征和背景噪声， [101], 在真实场景 Seed-TTS 提示上的无参考 UTMOSv2 得分可能低于 AT 解码器。在 NISQA 评估的总体质量指标中，该指标综合考察噪声度、音色失真、不连续性和响度， [94], BiSAT-TTS 的两个变体在三个数据集上仍优于 CosyVoice 3、Spark-TTS 以及两个 LLaSA 模型。CosyVoice 2 在这两项指标上的出色得分，也可能得益于其卷积式 Transformer U-Net 解码器，而非 CosyVoice 3 和 BiSAT 所采用的 DiT [2], [13]。总体而言，在实际应用中，BiSAT 较好地平衡了可懂度、说话人相似度、自然度和语音质量。

4) 码本大小消融：更大的码本能够提升容量，但也会增加 VQ 崩溃的风险，并扩大自回归输出词汇表，使下一词元预测更加困难，通常还需要更大的模型和更多训练数据。对比表 II 和表 IV 中的 BiSAT-32k（$K = 32768$，375 bps）与 BiSAT-64k（$K = 65536$，400 bps）可见，在比特率提高 6.7% 的情况下，64 k 码本仍保持 99.8% 的训练利用率，这表明跨独立连续空间的共享索引分配，结合球面 k 均值初始化、EMA 更新和死码替换，即使面对大型成对词汇表，也能防止 VQ 崩溃。在表 II 中，BiSAT-64k 在 LibriSpeech-PC 上的 UTMOSv2 得分略有提升，同时 WER 和 SIM 相当；除日语外，在大多数 Common Voice 27 数据集上，WER 和 CER 也有所小幅降低。这表明 32 k 已足以处理干净语音，而 64 k 则能提升模型对噪声更大的自然场景语音的鲁棒性。在大多数 BiSAT-32k 重建数据集上，PolyShift 略优于余弦调度；而在 64 k 和 TTS 中，两种调度表现相当。在表 IV 中，BiSAT-TTS-64k 降低了 Seed-TTS 的 WER/CER，提高了 Seed-TTS 的 UTMOSv2，并在全部三个数据集上改善了 NISQA，同时 LibriSpeech-PC 的 WER 相当，SIM-o 几乎没有变化。由于语言模型输出头大小和采样设置也存在差异，这体现了端到端权衡；同时也表明，在相同的语言模型规模和数据量下，BiSAT 能够有效扩展到 64 k 码本，而不会降低自回归建模性能。

[2]https://github.com/mmorise/World [3] https://montreal-forced-aligner.readthedocs.io

V. 结论

我们提出了 BiSAT，这是一种统一语音标记器，通过共享索引分配，在不依赖固定语义教师模型的情况下，将独立的声学分支和语义分支联合优化为单一代码流。在 BiSAT 的基础上，BiSAT-TTS 将自回归文本-语音语言模型与条件流匹配声学解码器结合起来，用于零样本文本转语音。涵盖重建、理解和诊断分析的实验表明，BiSAT 能够保留语言和声学结构，泛化到自然场景语音和域外语音，并学习到互补的配对码本。在零样本文本转语音方面，BiSAT-TTS 以更少的训练数据达到或超越了最先进的自回归基线。最后，这些结果表明，仅凭重建指标无法充分反映可供自回归语言模型利用的结构，因此，冻结代码理解探针是衡量标记质量的重要补充指标。

VI. 局限性与未来工作

由于训练预算有限，我们无法在更大规模上验证 BiSAT，尤其是分词器方面：两个编码器均初始化自英语预训练的 NeMo SSL Conformer，而非更大的多语编码器。再加上少得多的普通话和日语训练数据 [13]，，这可能会限制模型在这些语言上的性能，使其逊于 CosyVoice 3；此外，文本语音语言模型的扩展也仍有待研究。不同于 X-Codec2 等仅从编码直接重建波形的端到端波形编解码器，BiSAT 依赖独立的说话人条件流匹配解码器和声码器。尽管我们进行了诊断分析和码本大小消融，本研究仍将 BiSAT 作为一个整体进行评估，未对其各个架构组件开展受控的全规模消融实验。未来，我们计划单独研究这些组件的贡献，探索直接的端到端波形重建，并将 BiSAT 扩展至基于代码的强化学习后训练和语音到语音语言模型。

参考文献

[1] S. Chen 等，“神经编解码器语言模型是零样本文本到语音合成器”，*IEEE* 音频、语音与语言处理汇刊，第 33 卷，第 705–718 页，2025年。[2] Z. Du 等，“CosyVoice 2：基于大语言模型的可扩展流式语音合成”，*arXiv* 预印本 *arXiv:2412.10117*，2024年。[3] Z. Ye 等，“Llama 语音合成中训练时与推理时计算的扩展”，*arXiv* 预印本 *arXiv:2502.04128*，2025年。[4] A. van den Oord、O. Vinyals 和 K. Kavukcuoglu，“神经离散表示学习”，载于 *NeurIPS* 论文集，第 30 卷，2017年。[5] N. Zeghidour、A. Luebs、A. Omran、J. Skoglund 和 M. Tagliasacchi，“SoundStream：端到端神经音频编解码器”，*IEEE/ACM* 音频、语音与语言处理汇刊，第 30 卷，第 495–507 页，2022年。[6] F. Mentzer、D. Minnen、E. Agustsson 和 M. Tschannen，“有限标量量化：让 VQ-VAE 变得简单”，载于*ICLR* 论文集，2024年。[7] Y. Zhao、Y. Xiong 和 P. Kr¨ahenb¨uhl，“使用二元球面量化进行图像与视频分词”，载于*ICLR* 论文集，2025年，第 90 844–90 868 页。[8] Z. Borsos 等，“AudioLM：一种音频生成的语言建模方法”，*IEEE/ACM* 音频、语音与语言处理汇刊，第 31 卷，第 2523–2533 页，2023年。[9] A. D´efossez、J. Copet、G. Synnaeve 和 Y. Adi，“高保真神经音频压缩”，机器学习研究汇刊，2023年。[10] Z. Ju 等，“NaturalSpeech 3：使用因子化编解码器和扩散模型的零样本语音合成”，载于*ICML* 论文集，第 235 卷，2024年，第 22 605–22 623 页。[11] H. Wu等，“Codec-SUPERB：对声音编解码模型的深入分析”，载于*ACL Findings*，2024年。[12] Z. Du 等，“CosyVoice：一种可扩展的多语言零样本文本到sp语音合成器y基于监督p语义标记”，*arXiv* 预印本 arXiv:2407.05407，2024年。[13] Z. Du 等，《CosyVoice 3：通过扩展规模与后训练实现真实场景语音生成》，arXiv 预印本 arXiv:2505.17589，2025年。[14] X. Wang 等，《Spark-TTS：一种采用单流解耦语音标记的高效基于大语言模型的文本转语音模型》，arXiv 预印本 arXiv:2503.01710，2025年。[15] H. Hu 等，《Qwen3-tts 技术报告》，arXiv 预印本 arXiv:2601.15621，2026年。[16] L. Della Libera、F. Paissan、C. Subakan 和 M. Ravanelli，《FocalCodec：通过焦点调制网络实现低码率语音编码》，载于 NeurIPS 论文集，2025年，第 26 926–26 951 页。[17] S. Ji 等，《WavTokenizer：用于音频语言建模的高效声学离散编解码器标记器》，载于 ICLR 论文集，2025年，第 93 809–93 826 页。[18] X. Zhang、D. Zhang、S. Li、Y. Zhou 和 X. Qiu，《SpeechTokenizer：面向语音语言模型的统一语音标记器》，载于 ICLR 论文集，2024年，第 31 798–31 818 页。[19] A. D´efossez 等，《Moshi：用于实时对话的语音文本基础模型》，arXiv 预印本 arXiv:2410.00037，2024年。[20] Y. Gong 等，《XY-tokenizer：缓解低码率语音编解码器中的语义声学冲突》，载于 ACL 论文集，2026年，第 9350–9369 页。[21] Z. Ye 等，《编解码器确实重要：探索编解码器对音频语言模型的语义缺陷》，载于 AAAI 论文集，2025年。[22] D. Jo、J. Yun、B. Roh 和 S. Kim，《LM-SPT：面向语音标记化的语言模型对齐语义蒸馏》，IEEE《音频、语音与语言处理汇刊》，第 34 卷，第 3714–3727 页，2026年。[23] S. Chen 等，《VALL-E 2：神经编解码器语言模型是达到人类水平的零样本文本转语音合成器》，arXiv 预印本 arXiv:2406.05370，2024年。[24] Y. Song、Z. Chen、X. Wang、Z. Ma 和 X. Chen，《ELLA-V：采用对齐引导的序列重排实现稳定的神经编解码器语言建模》，载于 AAAI 论文集，第 39 卷，第 24 期，2025年，第 25 174–25 182 页。[25] L. Qu 等，《TokenFlow：用于多模态理解与生成的统一图像标记器》，载于 CVPR 论文集，2025年，第 2545–2555 页。[26] N. Tishby、F. C. Pereira 和 W. Bialek，《信息瓶颈方法》，载于 Allerton 通信、控制与计算会议论文集，1999年，第 368–377 页。[27] A. A. Alemi、I. Fischer、J. V. Dillon 和 K. Murphy，《深度变分信息瓶颈》，载于 ICLR 论文集，2017年。[28] S.-w. Yang 等，《SUPERB：语音处理通用性能基准》，载于 Interspeech 论文集，2021年。[29] J. Shi 等，《ML-SUPERB：多语言语音通用性能基准》，载于 Interspeech 论文集，2023年。[30] P. Peng、P.-Y. Huang、S.-W. Li、A. Mohamed 和 D. Harwath，《VoiceCraft：真实场景中的零样本语音编辑与文本转语音》，载于 ACL 论文集，2024年，第 12 442–12 462 页。[31] P. Peng、Z. Zheng、S.-W. Li、A. Mohamed 和 D. Harwath，《Voic- eStar：具备时长控制与外推能力的稳健零样本自回归文本转语音》，载于 ACL 论文集，2026年，第 11 737–11 754 页。[32] J. Kim、K. Lee、S. Chung 和 J. Cho，《CLaM-TTS：改进用于零样本文本转语音的神经编解码器语言模型》，载于 ICLR 论文集，2024年，第 50 278–50 301 页。[33] Y. Song、Z. Chen、X. Wang、Z. Ma、G. Yang 和 X. Chen，《TacoLM：配备门控注意力的编解码器语言模型是高效的零样本文本转语音合成器》，载于 Interspeech 论文集，2024年，第 4433–4437 页。[34] A. Hussein、S. Khurana、G. Wichern、F. G. Germain 和 J. Le Roux，《HASRD：分层声学与语义表征解耦》，载于 Interspeech 论文集，2025年。[35] S. Mehta、R. Tu、J. Beskow、 ´E. Sz´ekely 和 G. E. Henter，《Matcha-TTS：采用条件流匹配的快速文本转语音架构》，载于 ICASSP 论文集，2024年，第 11 341–11 345 页。[36] S.-H. Lee、H.-Y. Choi、S.-B. Kim 和 S.-W. Lee，《HierSpeech++：通过用于零样本语音合成的分层变分推断，弥合语音语义表征与声学表征之间的差距》，IEEE《神经网络与学习系统汇刊》，2025年。[37] D. Yang 等，《SimpleSpeech 2：面向采用基于流的标量潜变量变换器扩散模型的简单高效文本转语音》，IEEE《音频、语音与语言处理汇刊》，2025年。[38] K. Shen 等，《NaturalSpeech 2：潜在扩散模型是自然且支持零样本的语音与歌声合成器》，载于 ICLR 论文集，2024年，第 698–722 页。[39] S. E. Eskimez 等，《E2 TTS：简单易用的完全非自回归零样本文本转语音》，载于 SLT 论文集，2024年，第 682–689 页。[40] Y. Chen 等，《F5-TTS：通过流匹配生成流畅且忠实语音的讲故事者》，载于 ACL 论文集，2025年，第 6255–。[41] L. Meng 等，《无需向量量化的自回归语音合成》，载于 ACL 论文集，2025年，第 1287–1300 页。[42] Y. Zhou 等，《VoxCPM：面向上下文感知语音生成与逼真语音克隆的无标记器文本转语音

speech generation and true-to-life voice cloning》，arXiv 预印本 arXiv:2509.24650，2025年。[43] J. Moon 等，《DELTA-TTS：将自回归模型适配为用于文本转语音的扩散语言模型》，arXiv 预印本 arXiv:2607.04140，2026年。[44] A. Baevski、Y. Zhou、A. Mohamed 和 M. Auli，《wav2vec 2.0：语音表征自监督学习框架》，载于 NeurIPS 论文集，第 33 卷，2020年，第 12 449–12 460 页。[45] W.-N. Hsu、B. Bolte、Y.-H. H. Tsai、K. Lakhotia、R. Salakhutdinov 和 A. Mohamed，《HuBERT：通过隐藏单元掩码预测进行自监督语音表征学习》，IEEE/ACM《音频、语音与语言处理汇刊》，第 29 卷，第 3451–3460 页，2021年。[46] S. Chen 等，《WavLM：面向全栈语音处理的大规模自监督预训练》，IEEE《精选信号处理主题期刊》，第 16 卷，第 6 期，第 1505–1518 页，2022年。[47] J. Ao 等，《SpeechT5：面向口语语言处理的统一模态编码器-解码器预训练》，载于 ACL 论文集，2022年，第 5723–5738 页。[48] Y. Tang 等，《用于语音翻译与识别的统一语音-文本预训练》，载于 ACL 论文集，2022年，第 1488–1499 页。[49] Z. Niu 等，《Semantic-VAE：用于更佳语音合成的语义对齐潜在表征》，载于 Interspeech 论文集，2026年。[50] O. Kuchaiev 等，《NeMo：使用神经模块构建人工智能应用的工具包》，arXiv 预印本 arXiv:1909.09577，2019年。[51] A. Graves、S. Fern´andez、F. Gomez 和 J. Schmidhuber，《连接主义时间分类：利用循环神经网络对未分段序列数据进行标注》，载于 ICML 论文集，2006年。[52] Y. Peng、Y. Sudo、M. Shakeel 和 S. Watanabe，《OWSM-CTC：面向语音识别、翻译和语言识别的开放式仅编码器语音基础模型》，载于 ACL 论文集，2024年，第 10 192–10 209 页。[53] C. Fifty 等，《利用旋转技巧重构向量量化》，载于 ICLR 论文集，2025年，第 19 153–19 188 页。[54] Y. Lipman、R. T. Q. Chen、H. Ben-Hamu、M. Nickel 和 M. Le，《用于生成建模的流匹配》，载于 ICLR 论文集，2023年。[55] X. Liu、C. Gong 和 Q. Liu，《流的直化与加速：利用整流流学习生成和迁移数据》，载于 ICLR 论文集，2023年。[56] H. Wang、S. Zheng、Y. Chen、L. Cheng 和 Q. Chen，《CAM++：利用上下文感知掩码进行说话人验证的快速高效网络》，载于 Interspeech 论文集，2023年，第 5301–5305 页。[57] W. Chen 等，《WavTTS：通过直接原始波形建模实现高质量零样本文本转语音》，arXiv 预印本 arXiv:2606.03455，2026年。[58] A. Gulati 等，《Conformer：用于语音识别的卷积增强型Transformer》，载于 Interspeech 论文集，2020年。[59] D. S. Park 等，《SpecAugment：一种用于自动语音识别的简单数据增强方法》，载于 Interspeech 论文集，2019年，第 2613–2617 页。[60] A. Razavi、A. van den Oord 和 O. Vinyals，《利用 VQ-VAE-2 生成多样的高保真图像》，载于 NeurIPS 论文集，第 32 卷，2019年。[61] S. Woo 等，《ConvNeXt V2：利用掩码自编码器共同设计并扩展卷积网络》，载于 CVPR 论文集，2023年，第 16 133–16 142 页。[62] H. Siuzdak，《Vocos：弥合时域与基于傅里叶变换的神经声码器之间的差距，实现高质量音频合成》，载于 ICLR 论文集，2024年，第 25 719–25 733 页。[63] MiniCPM团队，《MiniCPM4：终端设备上的超高效大语言模型》，https://huggingface.co/openbmb/MiniCPM4-0.5B，2025年，技术报告：arXiv:2506.07900。[64] W. Peebles 和 S. Xie，《采用 Transformer 的可扩展扩散模型》，载于 ICCV 论文集，2023年。[65] J. Su、M. Ahmed、Y. Lu、S. Pan、W. Bo 和 Y. Liu，《RoFormer：带旋转位置嵌入的增强型 Transformer》，《神经计算》，第 568 卷，第 127063 页，2024年。[66] J. Ho 和 T. Salimans，《无分类器扩散引导》，载于 NeurIPS 深度生成模型研讨会论文集，2021年。[67] H. He 等，《Emilia：用于大规模语音生成的广泛、多语言、多样化语音数据集》，载于 SLT 论文集，2024年，第 885–890 页。[68] Y. Koizumi 等，《LibriTTS-R：恢复型多说话人文本转语音语料库》，载于 Interspeech 论文集，2023年，第 5491–5495 页。[69] W. Wang、Y. Song 和 S. Jha，《GLOBE：具有全球口音的高质量英语语料库，用于零样本说话人自适应文本转语音》，载于 Interspeech 论文集，2024年，第 1365–1369 页。[70] J. Yamagishi、C. Veaux 和 K. MacDonald，《CSTR VCTK语料库：用于 CSTR语音克隆工具包的英语多说话人语料库（版本 0.92）》，爱丁堡大学语音技术研究中心，2019年。[71] Y. Shi、H. Bu、X. Xu、S. Zhang 和 M. Li，《AISHELL-3：多说话人普通话文本转语音语料库》，载于 Interspeech 论文集，2021年，第 2756–2760 页。

[72] X. Li、S. Takamichi、T. Saeki、W. Chen、S. Shiota 和 S. Watanabe，“YODAS：面向YouTube的音频与语音数据集”，载于 ASRU 会议论文集，2023年，第 1–8 页。[73] F. S. Oliveira、E. Casanova、A. Cˆandido J´unior、A. S. Soares 和 A. R. Galv˜ao Filho，“CML-TTS：面向低资源语言的多语言语音合成数据集”，载于 TSD 会议论文集，2023年，第 188–199 页。[74] P. Puchtler、J. Wirth 和 R. Peinl，“HUI德语音频语料库：高质量TTS数据集”，载于 KI 会议论文集，2021年，第 204–216 页。[75] S. Takamichi、K. Mitsui、Y. Saito、T. Koriyama、N. Tanji 和 H. Saruwatari，“JVS语料库：免费日语多说话人语音语料库”，arXiv 预印本 arXiv:1908.06248，2019年。[76] A. Meister、M. Novikov、N. Karpov、E. Bakhturina、V. Lavrukhin 和 B. Ginsburg，“LibriSpeech-PC：用于评估端到端ASR模型标点与大小写能力的基准测试”，载于 ASRU 会议论文集，2023年。[77] R. Ardila 等，“Common Voice：大规模多语言语音语料库”，载于 LREC 会议论文集，2020年，第 4218–4222 页。[78] Z. Ye 等，“X-Codec-2 0，2025，0，234 2 2。” https://github.com/zhenye /X-Codec-。模型检查点：https://huggingface.co/HKUSTAudio/xcodec 。[79] I. Loshchilov 和 F. Hutter，“解耦权重衰减正则化”，载于 ICLR 会议论文集，2019年。[80] K. Baba、W. Nakata、Y. Saito 和 H. Saruwatari，“VoiceMOS挑战赛2024的T05系统：从深度图像分类器迁移学习，用于高质量合成语音的自然度MOS预测”，载于 SLT 会议论文集，2024年。[81] A. Radford、J. W. Kim、T. Xu、G. Brockman、C. McLeavey 和 I. Sutskever，“通过大规模弱监督实现鲁棒语音识别”，载于 ICML 会议论文集，2023年，第 28 492–28 518 页。[82] V. Panayotov、G. Chen、D. Povey 和 S. Khudanpur，“LibriSpeech：基于公版有声读物的ASR语料库”，载于 ICASSP 会议论文集，2015年。[83] V. Pratap、Q. Xu、A. Sriram、G. Synnaeve 和 R. Collobert，“MLS：用于语音研究的大规模多语言数据集”，载于 Interspeech 会议论文集，2020年。[84] B. Zhang 等，“WenetSpeech：一个包含 10000+ 小时的多领域普通话语音识别语料库”，载于 ICASSP 会议论文集，2022 年。[85] Y. Yin 、 D. Mori 和 S. Fujimoto，“ReazonSpeech：免费的超大规模日语ASR语料库”，载于日本语言处理学会年会论文集，2023年。[86] A. Conneau 等，“FLEURS：语音通用表征的小样本学习评测”，载于 SLT 会议论文集，2023年。[87] A. Nagrani、J. S. Chung 和 A. Zisserman，“VoxCeleb：大规模说话人识别数据集”，载于 Interspeech 会议论文集，2017年。[88] C. Busso 等，“IEMOCAP：情感双人动作捕捉数据库”，《语言资源与评估》，第 42 卷，第 335–359 页，2008年。[89] H. Cao、D. G. Cooper、M. K. Keutmann、R. C. Gur、A. Nenkova 和 R. Verma，“CREMA-D：众包情感多模态演员数据集”，《IEEE情感计算学报》，第 5 卷，第 4 期，第 377–390 页，2014年。[90] S. R. Livingstone 和 F. A. Russo，“Ryerson情感语音与歌曲视听数据库（RAVDESS）”，《PLoS ONE》，第 13 卷，第 5 期，文章编号 e0196391，2018年。[91] K. Zhou、B. Sisman、R. Liu 和 H. Li，“利用新的情感语音数据集进行已见与未见情感风格迁移的语音转换”，载于 ICASSP 会议论文集，2021年。[92] J. Valk 和 T. Alum¨ae，“VoxLingua107：口语语言识别数据集”，载于 SLT 会议论文集，2021年，第 652–658 页。[93] P. Anastassiou 等，“Seed-TTS：高质量通用语音生成模型系列”，arXiv 预印本 arXiv:2406.02430，2024年。[94] G. Mittag、B. Naderi、A. Chehadi 和 S. M¨oller，“NISQA：利用众包数据集进行多维语音质量预测的深度CNN-自注意力模型”，载于 Interspeech 会议论文集，2021年，第 2127–2131 页。[95] Z. Gao、S. Zhang、I. McLoughlin 和 Z. Yan，“Paraformer：用于非自回归端到端语音识别的快速、准确并行Transformer”，载于 Interspeech 会议论文集，2022年，第 2063–2067 页。[96] S. Kajarekar 等，“利用韵律与词汇特征进行说话人识别”，载于 ASRU 会议论文集，2003年，第 19–24 页。[97] A. Gersho 和 R. M. Gray，《向量量化与信号压缩》。美国马萨诸塞州波士顿：Kluwer Academic Publishers，1992年。[98] A. Glielmo、C. Zeni、B. Cheng、G. Cs´anyi 和 A. Laio，“对距离度量的信息内容进行排序”，《PNAS Nexus》，第 1 卷，第 2 期，文章编号 pgac039，2022年。[99] T. Chen 和 C. Guestrin，“XGBoost：可扩展的树提升系统”，载于 ACM SIGKDD 会议论文集，2016年，第 785–794 页。[100] S. M. Lundberg 等，“从局部解释到利用可解释人工智能理解树模型的全局规律”，《Nature Machine Intelligence》，第 2 卷，第 1 期，第 56–67 页，2020年。[101] X. Wang 等，“基于流匹配的零样本文本转语音的噪声鲁棒性研究”，载于 Interspeech 会议论文集，2024年，第 687–691 页。