利用扩散自编码器扩展语音分词器

Yuancheng Wang[1,2*], **Zhenyu Tang**[1], **Yun Wang**[1], **Arthur Hinsvark**[1], **Yingru Liu**[1], **Yinghao Li**[1], **Kainan Peng**[1], **Junyi Ao**[1,2], **Mingbo Ma**[1], **Mike Seltzer**[1], **Qing He**[1], **Xubo Liu**[1]

[1] Meta 超级智能实验室[2] 香港中文大学（深圳）

arXiv:2602.06602v1 [cs.SD] 6 Feb 2026

摘要

语音分词器是语音语言模型的基础，但现有方法面临两大挑战：（1）如何平衡用于理解的语义编码与用于重建的声学编码之间的权衡；（2）如何实现低比特率和低令牌率。我们提出语音扩散分词器（***SiTok***），这是一种扩散自编码器，通过监督学习联合学习富含语义的表示，并借助扩散实现高保真度音频重建。我们将 SiTok 扩展至 1.6B 个参数，并使用 200 万小时的语音数据进行训练。实验表明，SiTok 在理解、重建和生成任务上的表现优于强基线，同时令牌率极低，仅为 12.5 Hz，比特率为每秒 200 比特。

1 引言

语音分词器（Parker 等人，2024；Guo 等人，2025；Mousavi 等人，2025）定义了连续语音信号与离散语音语言模型（Borsos 等人，2023；Nguyen 等人，2025；D´efossez 等人，2024；Grattafiori 等人，2024；Zeng 等人，2024b;a；Fang 等人，2024；Wang 等人，2024；Huang 等人，2025；Ding 等人，2025；Xu 等人，2025）之间的接口，直接塑造了语音的感知、建模与生成方式。随着语音语言模型不断扩展，并将理解与生成统一起来，其离散语音表示的质量与结构成为关键瓶颈。

理想的语音分词器通常需要平衡三个相互竞争的目标：**（1）极致压缩**，以实现可扩展的语言建模；**（2）高保真重建**，以生成自然语音；以及**（3）富含语义的表示**，以支持下游语音理解。然而，现有语音分词器难以同时满足这些目标，尤其是在低词元速率区间，此时每个离散词元都需要编码大量信息。

大多数既有方法都通过启发式折中来应对这一矛盾，而非寻求有原则的解决方案。(1) 为了减轻低比特率下的重建退化，许多方法采用残差向量量化（RVQ）（Zeghidour 等人，2021；D´efossez, 2022；Kumar 等人，2023）或提高帧率（Xin 等人，2024；Ji 等人，2024；Ju 等人，2024；叶等人，2025b；Wang 等人，2025a），这会直接增加令牌预算，损害语言建模的效率。(2) 相反，主要针对声学保真度进行优化的分词器往往忽视语言结构，导致其表征不适用于语音理解。(3) 一些方法依赖多阶段训练流程（Guo 等人，2024；Anastassiou 等人，2024；Du 等人，2024a；Zeng 等人，2024a；Zhang 等人，2025c；Wang 等人，2025c），其中表征学习与波形重建相互分离，需要在第二阶段单独学习从令牌到波形的映射，因此无法进行端到端联合优化。

在这项工作中，我们旨在探索一种语音分词器范式，使其同时实现极限压缩、高质量重建，并为语音语言模型提供有效表征。关键在于，我们观察到，在传统的声学重建目标下，单纯增加训练数据或模型规模，在低令牌率下会带来递减的收益。这一局限表明

*在 Meta 实习期间完成的工作。电子邮件：yuanchengwang@link.cuhk.edu.cn

向量量化带来了结构性瓶颈：仅使用确定性重建损失进行训练时，激进压缩会迫使离散潜在空间消除不确定性，往往优先保留低层信号细节，而非具有语言学意义的结构。因此，这类分词器难以为下游语音理解任务（如自动语音识别（自动语音识别））提供理想的语义表示（Zhang 等，2023；D´efossez 等人，2024）。要克服这一瓶颈，语音分词器需要采用生成式框架，显式建模激进量化所引入的不确定性，而不是强制进行确定性逆映射。扩散模型（Ho 等人，2020；Song 等人，2021；Lipman 等人，2023）为此提供了自然的解决方案，因为它们能够学习逆转随机扰动过程，并已在许多领域展现出强大的生成能力和可扩展性（Rombach 等人，2022；Liu 等人，2023；Shen 等人，2024；Le 等人，2023；Polyak 等人，2024）。一些先前的研究（Anastassiou 等人，2024；Guo 等人，2024；Du 等人，2024a;b；Zhang 等，2025c;b）探索了基于扩散的语音分词，但通常采用两阶段设计：首先量化语音自监督表征（Baevski 等人，2020；Chung 等人，2021；Hsu 等人，2021；Chen 等人，2022；Chiu 等人，2022），然后单独训练扩散模型来合成波形或梅尔频谱图。这种解耦训练使量化器无法针对重建保真度进行优化，也迫使扩散解码器去适配次优的离散编码。相比之下，我们在一个**扩散自编码器**中联合优化量化与重建（Rey 等人，2019；Preechakul 等人，2022），确保离散编码既具有极高的压缩率，又与语音的生成分布明确对齐。

除重建之外，我们还引入**语义正则化**，以显式塑造离散潜在空间。仅通过重建损失训练的语音分词器往往注重声学保真度，却缺乏与语言信息的对齐，这不利于语音语言建模中的理解和生成任务。先前的研究（Zhang 等，2023；D´efossez 等人，2024；Li 等人，2025a；Della Libera 等人，2025）尝试通过语义……来缓解这一问题
语义蒸馏，通过均方误差或余弦相似度损失，使潜在表示与自监督语音特征对齐。然而，这种间接对齐并未明确约束语言一致性。在本研究中，我们引入一个辅助的连接时序分类（CTC）解码器，并通过 CTC 损失（Graves 等人，2006）对其进行优化，从而直接对量化潜在空间施加语义监督，促使离散令牌保留丰富的语义信息和具有语言学意义的结构。我们在附录 B 中对低比特率语音分词器和基于扩散的语音分词进行了更全面的相关工作综述。

由于扩散模型在推理过程中需要迭代采样，解码效率成为一项关键挑战。我们进一步研究了快捷微调（Frans 等人，2024）及其他加速技术，这些技术能够在保持高重建质量的同时，大幅减少扩散步数（例如，减少至 2 步或 4 步）。

总而言之，我们提出了语音扩散分词器（*SiTok*）（***SiTok***），将其扩展至 1.6B 个参数，并使用 200 万小时的语音数据进行训练。SiTok 在极限压缩设置下，即**令牌速率为 12.5 Hz 、比特率为 0.2 kbps**，表现出色。我们全面评估了 SiTok 在语音重建和多种理解任务上的表现，包括情感识别、关键词检出、说话人验证和自动语音识别。此外，我们证明，学习得到的离散表示可有效用于语音生成，在相同的低令牌速率设置下实现高质量合成。针对码本大小、码本维度和残差向量量化（RVQ）开展的大量消融研究，进一步揭示了可扩展的基于扩散的语音分词器的设计要点。语音重建和生成样本见 https://sitok-demo.github.io/。

2 方法

本节介绍 SiTok。我们首先介绍基于扩散自编码器的语音分词架构。随后，我们将描述我们的关键设计——语义正则化。最后，我们将解释如何通过快捷微调加速解码，并介绍能够进一步提升重建质量的其他技术。图 1 展示了我们模型的概览。

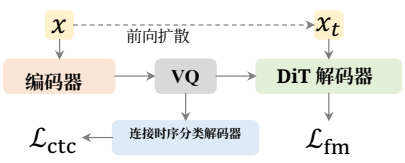


图1：SiTok概述。

2.1 概述

语音分词器通常基于自编码器构建：首先由编码器将原始语音表示映射为潜在特征，然后由量化器将潜在特征编码为一系列离散令牌，最后由解码器根据这些离散令牌重建原始语音表示。一些语音分词器直接使用原始波形信号作为建模目标，并依靠对抗训练来提升感知质量。然而，我们认为这种范式不利于扩展，原因如下：(1) 直接处理波形序列的效率较低，因为其长度过大，需要大量的上采样和下采样，并且往往迫使以往的工作将波形截断为仅几秒的片段；(2) 对抗训练需要复杂的损失设计以及额外的判别器优化，因而往往不稳定。

在这项工作中，我们提出了一种新颖的框架：(1) 使用**梅尔频谱图**作为输入和重建目标，并借助声码器直接合成相应的波形；(2)**以扩散自编码器取代对抗训练**，从而实现更稳定且可扩展的训练。我们认为，这种基于扩散的框架能够实现更优的压缩和重建。通过学习逆转扩散过程，模型经过训练能够有效捕捉底层数据分布，从而能够根据量化表示更可靠地恢复原始信号。形式上，给定输入梅尔频谱图 $\boldsymbol{x}$，训练过程如下：

1. **下采样**：为提高计算效率，降低 $\boldsymbol{x}$ 的时间分辨率。在这项工作中，我们将默认帧率设为 12.5 Hz。2. **编码**：编码器 $\mathcal{E}_\theta$ 将下采样后的频谱图 $\boldsymbol{x}$ 映射为一系列潜在特征：

$$\boldsymbol{z} = \mathcal{E}_\theta(\boldsymbol{x}).$$

3. **量化**：将潜在序列 $\boldsymbol{z}$ 中的每个特征向量映射到离散码本 $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_K\}$中与其最接近的条目，从而生成一系列离散索引 $\boldsymbol{q}$。此向量量化步骤可表示为：

$$\boldsymbol{q} = \mathrm{VQ}(\boldsymbol{z}; \boldsymbol{E}).$$

4. **扩散建模**：训练解码器 $\mathcal{D}_\phi$，使其在量化表示的条件下重建 $\boldsymbol{x}$。首先将离散索引 $\boldsymbol{q}$ 映射回对应的码本嵌入 $\boldsymbol{z}_q$。解码器 $\mathcal{D}_\phi$ 通过流匹配目标学习预测一个速度场，将带噪样本转换回原始数据。该过程定义如下：

$$\boldsymbol{x}_t = t\boldsymbol{x} + (1-t)\,\boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \text{ and } t \sim U(0,1).$$

优化解码器预测的速度 v_ϕ，使其与真实速度（$\boldsymbol{x} - \boldsymbol{\epsilon}$）相匹配：

$$v_\phi(\boldsymbol{x}_t, t, \boldsymbol{z}_q) = \mathcal{D}_\phi(\boldsymbol{x}_t, t, \boldsymbol{z}_q) \to \boldsymbol{x} - \boldsymbol{\epsilon}.$$

2.2 语义正则化

在初步研究中，我们发现，仅依赖重建——无论采用扩散损失还是回归损失——都会导致可懂度较差（词错误率显著更高），并降低下游理解任务的表现。受先前研究的启发，**我们引入一种辅助损失，直接对向量量化后的潜在空间进行监督**。不同于采用表征对齐（Yu 等，2025）或语义蒸馏（Zhang 等，2023；D´efossez 等，2024），将潜在表示与自监督模型提取的特征相匹配的方法，我们通过一个额外的轻量级解码器 $\mathcal{D}_{\phi_{\text{ctc}}}$ 直接预测文本内容，并使用连接时序分类损失对其进行训练。一些先前工作，如 Beichuan-Audio 分词器（Li 等，2025b）和 XY-Tokenizer（Gong 等，2025），也采用基于自动语音识别的监督来丰富语义表征，但仍依赖额外的自动语音识别模型作为语义编码器来提取潜在特征。相比之下，SiTok 直接从原始语音中学习表征。

总损失 $\mathcal{L}_{\text{total}}$ 由三个部分组成：扩散重建损失（$\mathcal{L}_{\text{rec}}$）、语义连接时序分类损失（$\mathcal{L}_{\text{ctc}}$）和矢量量化损失（$\mathcal{L}_{\text{vq}}$）。给定真实文本转录 $\boldsymbol{y}$，优化目标为：

$$\mathcal{L}_{\text{total}} = \underbrace{\mathbb{E}_{t,\boldsymbol{x},\boldsymbol{\epsilon}}\left[\|\mathcal{D}_\phi(\boldsymbol{x}_t, t, \boldsymbol{z}_q) - (\boldsymbol{x} - \boldsymbol{\epsilon})\|\right]}_{\text{Reconstruction Loss}} + \lambda_{\text{ctc}} \underbrace{\text{CTC}(\mathcal{D}_{\phi_{\text{ctc}}}(\boldsymbol{z}_q), \boldsymbol{y})}_{\text{CTC Loss}} + \underbrace{\mathcal{L}_{\text{vq}}}_{\text{VQ Loss}}$$

其中， $\boldsymbol{z}_q$ 是量化嵌入序列， $\mathcal{D}_{\phi_{\text{ctc}}}$ 是辅助 CTC 解码器， λ_{ctc} 是平衡超参数。我们发现， λ_{ctc} 对性能至关重要。

2.3 高效解码

传统扩散模型通常需要多个推理步骤，这可能导致解码的计算效率低下。为此，我们探索了两种能够显著加速扩散解码的策略：**快捷微调** 和**轻量级扩散头**。

快捷微调 我们探索通过微调解码器实现高效的少步解码，所采用的是 Frans 等人（2024）提出的快捷模型目标。在微调阶段，我们冻结编码器和向量量化模块。随后，微调过程仅更新解码器权重。快捷模型的核心思想是训练一个网络，使其不仅以时间步t 为条件，还以期望的步长 d为条件。这样，模型便能在一次前向传播中，学习从带噪输入直接映射到显著去噪后的输出，从而有效地“跳过”标准迭代扩散过程中的多个中间步骤。

The 微调采用组合损失函数，联合优化流匹配目标 e
and 自一致性目标。总损失的形式如下：

$$\mathcal{L}_{\text{S}} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{x}_1,t,d}\Big[\underbrace{\|\boldsymbol{s}_\phi(\boldsymbol{x}_t,t,0) - (\boldsymbol{x}_1-\boldsymbol{x}_0)\|_2^2}_{\text{Flow-Matching Loss}} + \underbrace{\|\boldsymbol{s}_\phi(\boldsymbol{x}_t,t,2d) - \boldsymbol{s}_{\text{target}}\|_2^2}_{\text{Self-Consistency Loss}}\Big]$$

其中， $\boldsymbol{s}_\phi$ 是正在训练的快捷模型（即我们的解码器）。自一致性损失的目标 $\boldsymbol{s}_{\text{target}}$由模型自身通过连续执行两个较小的步长生成，并且梯度已分离：

$$\boldsymbol{s}_{\text{target}} = \text{stopgrad}\left(\frac{1}{2}\boldsymbol{s}_\phi(\boldsymbol{x}_t,t,d) + \frac{1}{2}\boldsymbol{s}_\phi(\boldsymbol{x}_{t+d},t+d,d)\right)$$

以及 $\boldsymbol{x}_{t+d} = \boldsymbol{x}_t + \boldsymbol{s}_\phi(\boldsymbol{x}_t,\ t,\ d)d$。

第一项将模型在无穷小步长（$d=0$）下的行为锚定下来，确保其与真实的数据速度一致。第二项则要求单个大小为 $2d$ 的大步长与连续执行两个大小为 d 的步长得到相同结果。这种自一致性训练使解码器能够沿去噪轨迹准确地进行大幅离散跳跃，从而在保持较高重建质量的同时，大幅减少推理步数。

轻量级扩散头 我们还探索了一种轻量级扩散头，通过将解码器拆分为主体（仅运行一次）和一个在扩散步数间重复使用的小型头部，来降低迭代去噪的成本。此设计降低了每步计算量；详情见附录 C.1。

2.4 重建精炼

为进一步提升重建语音的质量，我们采用了两种不同的精炼策略。第一种是**解码器微调**，即冻结编码器和 VQ 模块，仅继续训练解码器。这一步使解码器专门适配从固定离散表示中进行高保真合成。第二种是引入**令牌无分类器引导（Token CFG）**。为实现这一点，我们训练解码器，使其有条件地依赖离散令牌：以 10% 的概率随机丢弃所有输入令牌，从而迫使解码器同时学习无条件生成目标。在推理期间，我们可以结合有条件（输入令牌）和无条件（丢弃令牌）两种过程的预测，来引导解码过程，从而实现更稳健、更准确的重建。我们的结果（表 1）通过实验证明了这两种可选精炼技术的有效性。

3 实验与结果

3.1 实现细节

数据与预处理 我们使用 200 万条内部语音数据训练模型。该数据集涵盖多种语言，其中绝大多数为英语。我们不会对语音数据进行额外的预处理，例如拆分为较短的片段；而是直接使用原始话语长度及其对应的转录文本进行训练

。我们使用 50 Hz、128 个频带的梅尔频谱图，作为分词器的输入和重建目标；同时先将每四个连续帧堆叠起来，将帧率降至 12.5 Hz，以提高训练效率。对于波形合成，我们采用基于 Vocos（Siuzdak，2024）的声码器，将梅尔频谱图转换回 24 kHz 的音频波形。

模型 我们的模型采用标准的 Llama 风格 Transformer 块构建（Touvron 等人，2023；Grattafiori 等人，2024）。编码器和辅助 CTC 解码器分别由 16 层和 4 层因果 Llama 解码器层组成。除非另有说明，我们将隐藏层大小设为 1536、中间层大小设为 4096，注意力头数量设为 16。对于 VQ 模块，我们采用默认配置：维度为 32，码本包含 65,536 个条目，并使用指数移动平均（EMA）进行更新（van den Oord 等人，2017）。扩散解码器通过将因果 Llama 解码器层改为非因果形式实现，共包含 16 层。我们将 RMSNorm（Zhang 和 Sennrich，2019）替换为自适应 RMSNorm 变体，以融入扩散步嵌入。更多架构细节见附录 A；关于码本维度、码本大小和整体模型规模的消融研究见第 3.4 节。

训练 我们对所有模型训练一个训练轮次，对应约 450K 步。优化时，我们采用 AdamW（Loshchilov 和 Hutter，2019）优化器，使用 $\beta_1 = 0.9$、$\beta_2 = 0.999$、0.01 的权重衰减，以及 8×10^{-5} 的学习率，并预热 32K 步。

3.2 评估

我们从压缩、重建和语音可懂度几个方面评估分词器。我们还通过零样本文本转语音评估 SiTok 的语音生成效果，结果见附录 C.2。

压缩 我们通过令牌速率（TPS：每秒令牌数）、帧率（FPS：每秒帧数）和比特率（BR）来评估分词器的效率。这些指标直接反映了压缩与表征能力之间的权衡。

重建 为评估语音重建质量，我们测量可懂度、说话人相似度和语音质量。可懂度通过词错误率（WER）评估，该指标使用 whisper-large-v3（Radford 等人，2023）计算。说话人相似度（SIM）通过提示语音与生成语音的 WavLM-TDNN 嵌入之间的余弦相似度计算（Chen 等人，2022）。语音质量使用官方 UTMOS 检查点评估（Saeki 等人，2022）。我们在 SeedTTS *test-en*（Anastassiou 等人，2024）评估集上报告这些结果。

理解 我们按照 DASB 基准（Mousavi 等人，2024）的设置，在三项语音理解任务上评估学习到的表征：情感识别（ER）、说话人验证（SV）和关键词检出（KS）。此外，我们遵循 Yang等人（2025）的做法，使用具有 1B 参数大语言模型骨干网络的基于大语言模型的自动语音识别（LLM ASR）模型；该模型以语音分词器生成的离散语音令牌为输入，并自回归地预测相应文本。我们还报告使用分词器的直接 CTC 解码器得到的自动语音识别结果（CTC 自动语音识别）。自动语音识别评估在 LibriSpeech *test-clean* 数据集上进行（Panayotov 等人，2015）。

评估基线 我们还将所提出的方法与一系列开源语音分词器进行比较，基线方法的更多详情见下文各节。

3.3 结果与比较

本节将对我们提出的语音分词器进行全面评估。我们首先在第 3.3.1 节报告语音重建的主要结果，比较其在不同压缩设置下与多种现有分词器的表现。随后，我们在第 3.3.2 节评估下游性能，涵盖多种理解任务。除这些比较之外，我们还在第 3.3.3 节分析语义正则化的有效性，在第 3.3.4 节探讨扩大模型规模的影响，并在第 3.3.5 节研究高效解码策略，以在不牺牲质量的前提下提升推理速度。最后，我们在第 3.4 节开展全面的消融研究，以分离不同组件的贡献，包括损失设计、码本配置和帧率选择，从而深入理解可扩展语音分词器的设计原则。

3.3.1 重建的主要结果

表 1：重建的主要结果。“解码器微调”表示冻结编码器和向量量化（VQ），同时对解码器进行若干步的进一步训练。“Token CFG”表示使用无分类器引导，更多详情见第 3.4 节。“CN”表示码本数量。

模型	帧每秒/令牌每秒	CN	比特率（kbps）	词错误率(↓)	相似度 (↑)	UTMOS (↑)
真实值	-	-	-	2.14	0.730	3.53
SpeechTokenizer（Zhang 等，2023）	50/100	2	1.00	7.98	0.468	2.47
BigCodec（Xin 等人，2024）	80/80	1	1.04	3.25	0.615	3.59
DualCodec（李等人，2025a）	12.5/75	6	0.925	2.63	0.624	3.78
WavTokenizer（Ji 等人，2024）	75/75	1	0.90	6.65	0.483	3.36
Mimi（D´efossez 等人，2024）	12.5/75	6	0.825	4.51	0.527	3.09
X-codec 2（Ye 等人，2025b）	50/50	1	0.80	2.63	0.620	3.68
SemantiCodec（Liu 等人，2024）	25/50	2	0.675	5.11	0.488	2.83
BiCodec（Wang 等人，2025a）	50/50	1	0.65	3.05	0.612	3.68
Vevo 分词器 (Zhang 等人, 2025c)	50/50	1	0.65	3.04	0.534	3.50
StableCodec (Parker 等人, 2024)	25/25	1	0.40	11.1	0.410	3.87
FireRedTTS 分词器 (Guo 等人, 2024)	25/25	1	0.35	3.35	0.597	3.40
CosyVoice 分词器（Du 等人，2024a）	25/25	1	0.30	5.63	0.465	3.65
SiTok(CN = 1)	12.5/12.5	1	0.20	4.06	0.641	3.44
+解码器微调	12.5/12.5	1	0.20	3.79	0.682	3.48
+Token CFG	12.5/12.5	1	0.20	3.34	0.635	3.60
SiTok(CN = 2)	12.5/25	2	0.35	3.17	0.658	3.44
SiTok(CN = 4)	12.5/50	4	0.70	2.80	0.660	3.46

表 1 详细比较了我们的语音分词器与其他基线模型在重建任务上的表现。在极具挑战性的设置下，我们的模型展现出卓越性能。在基础配置（单码本）下，我们的分词器以极低的 0.2 kbps 比特率运行，令牌速率仅为 12.5 Hz，显著低于所有竞争方法。尽管压缩程度如此之高，它仍取得了极具竞争力的结果。我们还证明，通过简单而有效的策略，模型性能可以得到显著提升。解码器微调将说话人相似度提升至令人瞩目的 0.682。应用令牌无分类器引导可将词错误率降至 3.34。此外，通过残差向量量化增加码本数量，可持续改善词错误率和相似度。

3.3.2 下游理解

理解任务 表2显示，我们的分词器在多项语音理解任务上的表现显著优于所有基线模型。具体而言，我们在基于大语言模型的自动语音识别（词错误率为4.95）上取得了显著提升，并且在错误率、说话人验证和主观意见分数方面始终优于所有基线模型。基线模型的基于大语言模型的自动语音识别结果取自Yang等人（2025）。“CN”表示码本数量，“CS”表示码本大小。

表2：理解任务的主要结果。

模型	帧每秒/令牌每秒	码本数量/码本大小	比特率（kbps）	CTC自动语音识别（↓）	自动语音识别（↓）	错误率（↑）	说话人验证（↓）	主观意见分数（↑）
离散音频编解码器 Rombach 等人（2022）	50/150	3/1024	1.5	-	58.4	48.9	17.8	68.8
EnCodec D´efossez 等人（2022）	50/150	3/1024	1.5	-	77.2	47.4	15.5	79.3
Mimi D´efossez 等人（2024）	12.5/100	8/2048	1.1	-	23.1	54.3	19.7	92.2
WavTokenizer 吉等人（2024）	40/40	1/4096	0.48	-	45.6	51.1	19.4	65.3
StableCodec Parker 等人（2024）	25/25	1/46656	0.40	-	28.0	-	-	-
GLM4-Voice Zeng 等人（2024a）	12.5/12.5	1/16384	0.20	-	16.3	-	-	-
SiTok(中文 = 1)	12.5/12.5	1/65536	0.20	9.50	4.95	63.5	13.8	96.9
SiTok(中文 = 4)	12.5/50	4/16384	0.70	8.30	4.49	64.4	8.59	97.7

3.3.3 语义正则化的有效性

表 3 展示了语义正则化对重建和理解任务的影响，这对于提升重建质量和学习有意义的表征至关重要

，以服务于下游理解。没有正则化时，模型的可懂度会严重下降（词错误率从 4.06 升至 33.0），下游性能也较差。相比之下，采用基于 CTC 的正则化能显著降低词错误率、提升相似度和语音质量，并改善所有理解任务。这表明，CTC 监督将量化潜在空间锚定于语言学意义，确保令牌既忠实于声学特征，又能传递语义信息，尤其适用于低码率分词器。

表 3：语义正则化对重建和理解的效果。

CTC 正则化	帧每秒/令牌每秒	重建			理解				
		词错误率（↓）	相似度（↑）	UTMOS（↑）	CTC自动语音识别（↓）	自动语音识别（↓）	错误率（↑）	说话人验证（↓）	主观意见分数（↑）
Yes	12.5/12.5	4.06	0.641	3.44	9.50	4.95	63.5	13.8	96.9
	12.5/25	3.17	0.658	3.44	8.64	4.72	61.7	11.1	97.8
	12.5/50	2.80	0.660	3.46	8.30	4.49	64.4	8.59	97.7
No	12.5/12.5	33.0	0.495	2.68	-	29.4	57.9	18.9	86.1
	12.5/25	10.1	0.598	2.99	-	9.53	55.3	15.5	92.7
	12.5/50	5.17	0.611	2.84	-	7.27	60.4	13.5	92.8

观察： 对量化潜在空间进行语义正则化，对于 b重建和表征学习都至关重要，尤其是在令牌率较低时。

3.3.4 模型扩展的有效性

从 0.63B 的“S”模型扩展到 1.61B 的“XL”模型的实验揭示了重建保真度与下游任务性能之间存在明显的权衡，如表4所示。较大的模型始终能带来更好的重建质量，其中 XL 模型的词错误率、相似度和 UTMOS 表现最佳；但在理解任务上的表现则以 1.12B 的“L”模型为最高，该模型在自动语音识别方面表现更优。最大的模型并未全面胜过较小的模型，甚至在说话人验证等任务中表现有所下降，这表明过大的模型容量可能会优先关注精细的声学细节，而非理解任务所需的抽象判别特征。因此，我们认为“L”模型是最佳配置，能够在高质量合成与稳健泛化之间实现最有效的平衡。架构设计的进一步探索留待未来工作。

表4：模型规模扩展的结果。我们在固定 CTC 层的同时调整编码器层和解码器层的数量，以评估其对重建和理解任务的影响。

Size	Enc.	Dec.	参数（B）	重建			理解				
				词错误率（↓）	相似度（↑）	UTMOS (↑)	CTC自动语音识别 (↓)	自动语音识别 (↓)	错误率 (↑)	说话人验证 (↓)	主观意见分数 (↑)
S	8	8	0.63	4.18	0.608	3.43	11.2	5.24	60.8	13.7	96.9
B	12	12	0.88	4.01	0.634	3.46	9.78	5.19	62.5	13.8	96.9
L	16	16	1.12	4.06	0.641	3.44	9.50	4.95	63.5	13.8	96.9
XL	24	24	1.61	3.84	0.649	3.51	9.62	5.07	63.5	14.7	97.3

3.3.5 高效解码

快捷微调 表2显示，直接减少推理步数会导致可懂度和音频质量明显下降。相比之下，快捷微调显著缓解了这一问题，即使扩散步数很少，也能实现更低的词错误率和更高的说话人相似度。此外，这项改进还显著降低了实时因子（RTF）：模型在**16**、**8**和**4** 个扩散步数下的实时因子分别为**0.041**、**0.024**和**0.013**。这些结果表明，快捷微调能够有效地使模型适应更快的采样计划，同时保持重建保真度并实现高效推理。

观察： 快捷微调能够实现高效的低步数推理，在显著加快解码速度的同时，保持较高的可懂度和相似度。

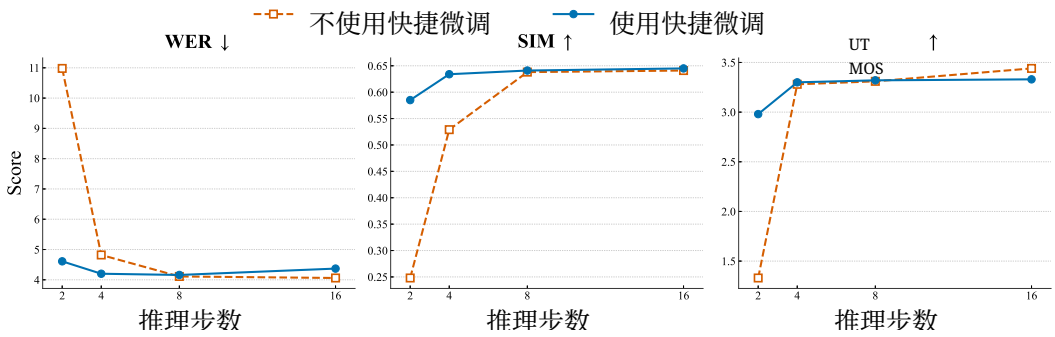


图 2：快捷微调对不同推理步数的影响。我们报告词错误率、相似度和 UTMOS。快捷微调始终能带来更高的可懂度和相似度，尤其是在步数较少时。

轻量级扩散头 我们还探索了使用轻量级扩散头来加速扩散推理。结果见附录 C.1。

3.3.6 SiTok 用于语音生成

除了语音理解之外，语音分词器的一个重要用途是作为语音生成的建模目标。为此，我们进一步在零样本文本转语音（TTS）任务上评估 SiTok，考察其学习到的离散表示能否有效支持高质量语音合成。实验结果表明，SiTok 不仅可有效用于语音理解表征，还能实现高质量、高效率的语音生成。尤其是，SiTok 极低的令牌速率显著缩短了生成所需的序列长度，从而加快推理，同时保持出色的感知质量。这些结果表明，SiTok 提供了一种统一表征，非常适用于语音理解和语音生成。零样本文本转语音的详细实验结果见附录 C.2。

3.4 消融研究

为了更好地理解分词器中不同设计选择的贡献，我们进行了一系列消融研究，结果如表5所示。我们系统地考察了训练目标、正则化策略、细化机制、码本配置和帧率。这些分析不仅验证了我们所提出组件的有效性，还揭示了效率、重建质量与下游性能之间的权衡。

扩散与回归 我们通过将所提出的基于扩散的目标函数（D）与传统的基于回归的目标函数（R）（例如梅尔频谱图上的L1损失）进行比较，研究了重建目标的选择。如表5所示，使用扩散损失训练的基线模型在所有关键指标上均显著优于基于回归的对照模型。具体而言，扩散模型取得了大幅更低的词错误率（4.06 vs. 4.66）、更高的说话人相似度（0.641 vs. 0.587）以及更好的下游自动语音识别性能（4.95 vs. 6.06）。为了进一步探究扩散解码器能否挽救经回归训练的模型，我们进行了一项实验：在回归预训练模型之上，仅使用扩散目标对解码器进行微调（R + D）。因此，按照设计，（R）与（R + D）的理解指标完全相同，而重建指标（词错误率/相似度/UTMOS）会发生变化。尽管这适度提升了说话人相似度，**但它未能达到端到端扩散模型的性能**，甚至使词错误率恶化至5.73。这表明，在扩散目标下学习到的表征在高保真重建和下游任务迁移能力方面本质上更为优越。

> **观察：** 采用基于扩散的目标函数对于学习高质量表征至关重要，在重建保真度和下游任务性能方面均显著优于标准回归目标函数。

表5：消融研究。

								重建			理解				
	Loss	CTC W.	CS	CN	CD	Tok. CFG	FPS	WER (↓)	SIM (↑)	UTMOS (↑)	CTC ASR (↓)	ASR (↓)	ER (↑)	SV (↓)	KS (↑)
	D	0.1	2^{16}	1	32	✗	12.5	4.06	0.641	3.44	9.50	4.95	63.5	13.8	96.9
Loss	R	0.1	2^{16}	1	32	✗	12.5	4.66	0.587	3.28	12.2	6.06	63.3	13.6	95.2
	R + D	0.1	2^{16}	1	32	✗	12.5	5.73	0.634	3.35	12.2	6.06	63.3	13.6	95.2
CTC W.	D	0	2^{16}	1	32	✗	12.5	33.0	0.495	2.68	-	29.4	57.9	18.9	86.1
	D	0.02	2^{16}	1	32	✗	12.5	5.05	0.607	3.44	12.2	7.41	58.3	13.5	97.2
	D	0.5	2^{16}	1	32	✗	12.5	8.81	0.614	3.20	11.0	7.87	64.2	16.6	91.2
	D	1	2^{16}	1	32	✗	12.5	10.1	0.585	3.38	10.5	8.90	62.1	13.9	96.8
CS	D	0.1	2^{13}	1	32	✗	12.5	5.48	0.640	3.39	11.7	5.72	55.7	16.3	95.7
	D	0.1	2^{14}	1	32	✗	12.5	4.30	0.641	3.33	11.5	5.40	58.7	16.0	96.4
	D	0.1	2^{15}	1	32	✗	12.5	4.26	0.648	3.43	10.5	5.33	61.2	15.0	96.4
	D	0.1	2^{17}	1	32	✗	12.5	3.94	0.651	3.39	10.6	5.09	60.6	13.7	97.3
CN	D	0.1	2^{14}	2	32	✗	12.5	3.17	0.658	3.44	8.64	4.72	61.7	11.1	97.8
	D	0.1	2^{14}	4	32	✗	12.5	2.80	0.660	3.46	8.30	4.49	64.4	8.59	97.7
	D	0.1	2^{14}	8	32	✗	12.5	2.50	0.645	3.30	8.42	4.68	60.0	7.53	98.2
CD	D	0.1	2^{14}	1	64	✗	12.5	4.08	0.642	3.46	9.58	5.27	61.7	14.4	97.3
	D	0.1	2^{14}	1	128	✗	12.5	3.85	0.642	3.36	9.14	4.59	59.7	14.3	97.3
	D	0.1	2^{14}	1	256	✗	12.5	5.04	0.641	3.30	10.9	5.54	64.1	16.5	95.9
Tok. CFG	D	0.1	2^{16}	1	32	✓	12.5	3.34	0.635	3.60	9.54	4.89	62.4	14.0	96.8
	D	0.1	2^{14}	4	32	✓	12.5	2.56	0.645	3.58	8.34	4.67	61.7	9.11	98.2
FPS	D	0.1	2^{16}	1	32	✗	6.25	23.0	0.428	3.10	15.8	12.7	52.6	20.7	88.5
	D	0.1	2^{16}	1	32	✗	25	3.05	0.688	3.72	9.19	4.45	63.5	7.28	97.8

CTC 损失权重 我们分析了 CTC 损失权重的影响，该权重用作语义正则项。结果明确表明，这种正则化不可或缺。将权重设为 0 会导致性能灾难性崩溃，词错误率飙升至 33.0，下游自动语音识别错误率则达到 29.4，这证实了没有文本监督时，模型无法学习到任何有意义的表征。相反，权重过高（例如 0.5 或 1.0）也会降低模型在重建任务和理解任务上的性能，这可能是因为模型被迫舍弃过多声学细节，以保留语义内容。我们的实验确定，权重设为 0.1 时，能够在确保语义一致性与保留高保真音频重建之间达到最佳平衡。

> **观察：** CTC 损失权重是一个关键的超参数；数值过低会无法确保语义一致性，而数值过高则会损害重建保真度。

重建精炼 如我们的主要重建结果（表 1）所示，解码器微调和令牌无分类器引导（CFG）都是提升重建质量的有效技术。解码器微调是一种训练阶段策略，专门针对固定表征优化合成模块，从而显著提升可懂度，尤其是说话人相似度。此外，在推理阶段应用令牌 CFG 提供了一种互补方法，可进一步提升保真度。我们的消融研究表明，CFG 能够持续且显著地降低词错误率，在 CD = 4配置下达到最低 2.56，同时还能改善感知质量（ UTMOS）。这些结果表明，训练阶段适配和推理阶段引导都是精炼最终输出的高效策略。

码本大小我们研究了码本大小（CS）的影响，该参数决定了离散令牌的词汇量。我们发现，将码本从 2^{13} 增大到 2^{17} ，重建质量始终有所提升，词错误率也稳步从 5.48 降至 3.94，证实了这一点。下游任务性能，尤其是自动语音识别，也因此受益，并在码本大小为 2^{16}, 时达到最佳，表明这一大小在表征能力与泛化之间实现了最佳平衡。

码本数量我们还使用残差向量量化（RVQ）评估了增加码本数量（CN）的效果，这会直接提高比特率。我们发现，将 CN 从 1 增至 8，可使大多数指标获得显著且持续的提升。重建词错误率从 4.30 大幅降至 2.50，而自动语音识别和说话人验证等任务的下游性能也显著提升。这表明，在我们的框架中，压缩与保真度之间实现了有效权衡。

> **观察：** 向量量化器的设计提供了在质量与复杂度之间灵活权衡的方式。增加码本数量，可以直接提升保真度和下游性能，但代价是比特率增加。

帧率与性能的权衡 我们还研究了两种备选帧率设置：6.25 Hz 和 25 Hz。我们发现，将帧率降至 6.25 Hz 会显著降低重建效果和下游任务性能；而将帧率提升至 25 Hz 虽能改善性能，却会使帧率翻倍。因此，我们采用 12.5 Hz 作为默认设置，以平衡效率与性能。

4 结论

在这项工作中，我们提出了 ***SiTok***，一种基于扩散自编码器的语音分词器，可端到端联合建模重建与量化，从而提升声学保真度。我们进一步引入语义正则化，以学习适用于语音理解的有效且语义丰富的表征，并探索快捷微调技术，以显著加速扩散解码。大量实验表明，SiTok 在语音重建和多种语音理解任务上均表现出色。此外，我们还进行了广泛的消融研究，为关键设计选择提供了见解。

伦理声明

本研究探讨了基于扩散自编码器的语音分词器。我们的模型旨在用于学术研究及语音理解、语音语言建模等下游任务。尽管分词器本身是中性的，我们仍承认其在下游系统中可能遭到滥用（例如，生成用于冒充的合成语音），并倡导以负责任且合乎伦理的方式使用我们的模型。

可复现性声明

为确保可复现性，我们在论文正文和附录中详细描述了模型架构、训练设置和评估方案。

大语言模型的使用

本研究中，大语言模型仅用于辅助目的，例如语法检查和文稿润色。然而，所有技术贡献、模型实现和实验分析均由作者完成。我们在适当之处承认使用了大语言模型，并确保其参与不会损害本工作的原创性。

参考文献

Philip Anastassiou、Jiawei Chen、Jitong Chen、Yuanzhe Chen、Zhuo Chen、Ziyi Chen、Jian Cong、Lelai Deng、Chuang Ding、Lu Gao、Mingqing Gong、Peisong Huang、Qingqing Huang、Zhiying Huang、Yuanyuan Huo、Dongya Jia、Chumin Li、Feiya Li、Hui Li、Jiaxin Li、Xiaoyang Li、Xingxing Li、Lin Liu、Shouda Liu、Sichao Liu、Xudong Liu、Yuchen Liu、Zhengxi Liu、Lu Lu、Junjie Pan、Xin Wang、Yuping Wang、Yuxuan Wang、Zhen Wei、Jian Wu、ChaoYao、Yifeng Yang、Yuanhao Yi、Junteng Zhang、Qidi Zhang、Shuo Zhang、Wenjie Zhang、Yang Zhang、Zilin Zhao、Dejian Zhong 和 Xiaobin Zhuang。Seed-TTS：一系列高质量、多功能的语音生成模型。*arXiv* 预印本，abs/2406.02430，2024。

Alexei Baevski、Yuhao Zhou、Abdelrahman Mohamed 和 Michael Auli。wav2vec 2.0：一种自监督学习语音表征的框架。载于*NeurIPS*，2020。

Xiaoyu Bie、Xubo Liu 和 Ga¨el Richard。学习神经音频编解码器中的源分离。载于*IEEE*国际声学、语音与信号处理会议 *(ICASSP)*，第 1–5 页，2025。

Zal´an Borsos、Rapha¨el Marinier、Damien Vincent、Eugene Kharitonov、Olivier Pietquin、Matthew Sharifi、Dominik Roblek、Olivier Teboul、David Grangier、Marco Tagliasacchi 和 Neil Zeghi-dour。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，31：2523–2533，2023。

Sanyuan Chen、Chengyi Wang、Zhengyang Chen、Yu Wu、Shujie Liu、Zhuo Chen、Jinyu Li、Naoyuki Kanda、Takuya Yoshioka、Xiong Xiao、Jian Wu、Long Zhou、Shuo Ren、Yanmin Qian、Yao Qian、Jian Wu、Michael Zeng、Xiangzhan Yu 和 Furu Wei。Wavlm：面向全栈语音处理的大规模自监督预训练。*IEEE* 信号处理精选专题期刊，16(6):1505–1518，2022。

Chung-Cheng Chiu、James Qin、Yu Zhang、Jiahui Yu 和 Yonghui Wu。采用随机投影量化器进行语音识别的自监督学习。载于国际机器学习会议，第 3915–3924 页。PMLR，2022。

Yu-An Chung、Yu Zhang、Wei Han、Chung-Cheng Chiu、James Qin、Ruoming Pang 和 Yonghui Wu。w2v-bert：结合对比学习和掩码语言建模的自监督语音预训练。载于*ASRU*，第 244–250 页。IEEE，2021。

Alexandre D´efossez、Jade Copet、Gabriel Synnaeve 和 Yossi Adi。高保真神经音频压缩。*arXiv* 预印本，abs/2210.13438，2022。

Alexandre D´efossez、Laurent Mazar´e、Manu Orsini、Am´elie Royer、Patrick P´erez、Herv´e J´egou、Edouard Grave 和 Neil Zeghidour。Moshi：用于实时对话的语音文本基础模型。*arXiv* 预印本 *arXiv:2410.00037*，2024。

LucaDella Libera、Francesco Paissan、Cem Subakan 和 Mirco Ravanelli。Focalcodec：通过焦点调制网络实现低码率语音编码。*arXiv* 预印本 *arXiv:2502.04465*，2025。

Ding Ding、Zeqian Ju、Yichong Leng、Songxiang Liu、Tong Liu、Zeyu Shang、Kai Shen、Wei Song、Xu Tan、Heyi Tang 等。Kimi-audio 技术报告。*arXiv* 预印本 *arXiv:2504.18425*，2025。

Zhihao Du、Qian Chen、Shiliang Zhang、Kai Hu、Heng Lu、Yexin Yang、Hangrui Hu、Siqi Zheng、Yue Gu、Ziyang Ma、Zhifu Gao 和 Zhijie Yan。Cosyvoice：一种基于监督语义标记的可扩展多语言零样本文本转语音合成器。*arXiv* 预印本，abs/2407.05407，2024a。

Zhihao Du、Yuxuan Wang、Qian Chen、Xian Shi、Xiang Lv、Tianyu Zhao、Zhifu Gao、Yexin Yang、Changfeng Gao、Hui Wang、Fan Yu、Huadai Liu、Zhengyan Sheng、Yue Gu、Chong Deng、Wen Wang、Shiliang Zhang、Zhijie Yan 和 Jingren Zhou。Cosyvoice 2：利用大型语言模型实现的可扩展流式语音合成。*arXiv* 预印本，abs/2412.10117，2024b。

Stefan Elfwing、Eiji Uchibe 和 Kenji Doya。用于强化学习中神经网络函数逼近的 Sigmoid加权线性单元。*Neural Networks*，107：3–11，2018年。

Qingkai Fang、Shoutao Guo、Yan Zhou、Zhengrui Ma、Shaolei Zhang 和 Yang Feng。Llama-omni：与大型语言模型实现无缝语音交互。*arXiv* 预印本 *arXiv:2409.06666*，2024。

Kevin Frans、Danijar Hafner、Sergey Levine 和 Pieter Abbeel。通过快捷模型实现一步扩散。*arXiv* 预印本 *arXiv:2410.12557*，2024。

Yitian Gong、Luozhijie Jin、Ruifan Deng、Dong Zhang、Xin Zhang、Qinyuan Cheng、Zhaoye Fei、ShiminLi 和 Xipeng Qiu。XY-Tokenizer：缓解低比特率语音编解码器中的语义-声学冲突。*arXiv* 预印本 *arXiv:2506.23325*，2025。

Aaron Grattafiori、Abhimanyu Dubey、Abhinav Jauhri、Abhinav Pandey、Abhishek Kadian、Ahmad Al-Dahle、Aiesha Letman、Akhil Mathur、Alan Schelten、Alex Vaughan 等。《Llama 3》模型家族。*arXiv* 预印本 *arXiv:2407.21783*，2024。

Alex Graves、Santiago Fern´andez、Faustino Gomez 和 J¨urgen Schmidhuber。连接主义时序分类：使用循环神经网络为未分段序列数据标注。载于第*23*届国际机器学习会议论文集，第369–376页，2006年。

Hao-Han Guo、KunLiu、Fei-Yu Shen、Yi-Chen Wu、Feng-Long Xie、Kun Xie 和 Kai-Tuo Xu。Firer edtts：面向行业级语音生成应用的基础文本转语音框架。*arXiv*预印本，abs/2409.03283，2024年。

Yiwei Guo、Zhihan Li、Hankun Wang、Bohan Li、Chongtian Shao、Hanglei Zhang、Chenpeng Du、Xie Chen、Shujie Liu 和 Kai Yu。离散语音词元的最新进展：综述。*arXiv*预印本 *arXiv:2502.06490*，2025年。

Haorui He、Zengqiang Shang、Chaoren Wang、Xuyuan Li、Yicheng Gu、Hua Hua、Liwei Liu、Chen Yang、Jiaqi Li、Peiyang Shi、Yuancheng Wang、Kai Chen、Pengyuan Zhang 和 Zhizheng Wu。Emilia：一个面向大规模语音生成的广泛、多语言且多样化的语音数据集。载于*SLT*。IEEE，2024。

Jonathan Ho、Ajay Jain 和 Pieter Abbeel。去噪扩散概率模型。神经信息处理系统进展，33：6840–6851，2020。

Wei-Ning Hsu、Benjamin Bolte、Yao-Hung Hubert Tsai、Kushal Lakhotia、Ruslan Salakhutdinov 和 Abdelrahman Mohamed。Hubert：通过对隐藏单元进行掩码预测实现自监督语音表征学习。*IEEE/ACM* 音频、语音与语言处理汇刊，29：3451–3460，2021。

Ailin Huang、Boyong Wu、Bruce Wang、Chao Yan、Chen Hu、Chengli Feng、Fei Tian、Feiyu Shen、Jingbei Li、Mingrui Chen 等。Step-audio：智能语音交互中的统一理解与生成。*arXiv* 预印本 *arXiv:2502.11946*，2025。

Shengpeng Ji、Ziyue Jiang、Wen Wang、Yifu Chen、Minghui Fang、Jialong Zuo、Qian Yang、Xize Cheng、Zehan Wang、Ruiqi Li 等。Wavtokenizer：用于音频语言建模的高效声学离散编解码器分词器。*arXiv* 预印本 *arXiv:2408.16532*，2024。

Yidi Jiang、Qian Chen、Shengpeng Ji、Yu Xi、Wen Wang、Chong Zhang、Xianghu Yue、ShiLiang Zhang 和 Haizhou Li。Unicodec：采用单一域自适应码本的统一音频编解码器。*arXiv* 预印本 *arXiv:2502.20067*，2025。

Zeqian Ju、Yuancheng Wang、Kai Shen、Xu Tan、Detai Xin、Dongchao Yang、Eric Liu、Yichong Leng、Kaitao Song、Siliang Tang、Zhizheng Wu、Tao Qin、Xiangyang Li、Wei Ye、Shikun Zhang、Jiang Bian、Lei He、Jinyu Li 和 Sheng Zhao。Naturalspeech 3：基于因式分解编解码器与扩散模型的零样本语音合成。收录于第四十一届国际机器学习大会，*ICML 2024*，奥地利维也纳，*2024*年*7*月*21*日至*27*日。OpenReview.net，2024。URL https: //openreview.net/forum?id=dVhrnjZJad。

Rithesh Kumar、Prem Seetharaman、Alejandro Luebs、Ishaan Kumar 和 Kundan Kumar。通过改进的 RVQGAN 实现高保真度音频压缩。载于 *NeurIPS*，2023 年。

Matthew Le、Apoorv Vyas、Bowen Shi、Brian Karrer、Leda Sari、Rashel Moritz、Mary Williamson、Vimal Manohar、Yossi Adi、Jay Mahadeokar 和 Wei-Ning Hsu。Voicebox：大规模文本引导的多语言通用语音生成。载于 Alice Oh、Tris- tan Naumann、Amir Globerson、Kate Saenko、Moritz Hardt 和 Sergey Levine（编），神经信息处理系统进展 *36*：神经信息处理系统年度会议 *2023，NeurIPS 2023*，美国路易斯安那州新奥尔良，*2023* 年 *12* 月 *10* 日至 *16* 日，2023 年。URL http://papers.nips.cc/paper_files/paper/2023/hash/2d8911db9ecedf866015091b28946e15-Abstract-Conference.html。

Doyup Lee、Chiheon Kim、Saehoon Kim、Minsu Cho 和 Wook-Shin Han。使用残差量化进行自回归图像生成。载于 *IEEE/CVF*计算机视觉与模式识别会议论文集，第 11523–11532 页，2022。

Jiaqi Li、Xiaolong Lin、Zhekai Li、Shixi Huang、Yuancheng Wang、Chaoren Wang、Zhenpeng Zhan 和 Zhizheng Wu。DualCodec：用于语音生成的低帧率语义增强神经音频编解码器。*arXiv* 预印本 *arXiv:2505.13000*，2025a。

Tianpeng Li、Jun Liu、Tao Zhang、Yuanbo Fang、Da Pan、Mingrui Wang、Zheng Liang、Zehuan Li、Mingan Lin、Guosheng Dong 等。百川音频：端到端语音交互的统一框架。*arXiv* 预印本 *arXiv:2502.17239*，2025b。

Yaron Lipman、Ricky T. Q. Chen、Heli Ben-Hamu、MaximilianNickel 和 Matthew Le。用于生成建模的流匹配。载于第十一届国际学习表征会议，*ICLR 2023*，卢旺达基加利，*2023*年*5*月*1*日至*5*日。OpenReview.net，2023年。网址：https://openreview.net/forum?id=PqvMRDCJT9t。

Haohe Liu、Zehua Chen、Yi Yuan、Xinhao Mei、Xubo Liu、Danilo Mandic、Wenwu Wang 和 Mark D Plumbley。Audioldm：基于潜在扩散模型的文本到音频生成。*arXiv* 预印本 *arXiv:2301.12503*，2023年。

Haohe Liu、Xuenan Xu、Yi Yuan、Mengyue Wu、Wenwu Wang 和 Mark D Plumbley。Se manti- codec：面向通用声音的超低比特率语义音频编解码器。*IEEE*信号处理精选主题期刊，2024年。

Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。发表于 *ICLR*（海报）。Open-Review.net，2019。

Fabian Mentzer、David Minnen、Eirikur Agustsson 和 Michael Tschannen。有限标量量化：让向量量化变分自编码器变得简单。*arXiv* 预印本 *arXiv:2309.15505*，2023。

Pooneh Mousavi、Luca Della Libera、Jarod Duret、Artem Ploujnikov、Cem Subakan 和 Mirco Ravanelli。DASB——离散音频与语音基准。*arXiv* 预印本 *arXiv:2406.14294*，2024。

Pooneh Mousavi、Gallil Maimon、Adel Moumen、Darius Petermann、Jiatong Shi、Haibin Wu、Haici Yang、Anastasia Kuznetsova、Artem Ploujnikov、Ricard Marxer 等。离散音频词元：不止是一篇综述！*arXiv* 预印本 *arXiv:2506.10274*，2025。

Tu Anh Nguyen、Benjamin Muller、Bokai Yu、Marta R Costa-Jussa、Maha Elbayad、Sravya Popuri、Christophe Ropers、Paul-Ambroise Duquenne、Robin Algayres、Ruslan Mavlyutov 等。Spirit-lm：交错口语与书面语言模型。计算语言学协会汇刊，13:30–52，2025。

Vassil Panayotov、Guoguo Chen、Daniel Povey 和 Sanjeev Khudanpur。Librispeech：基于公有领域有声书的自动语音识别语料库。载于*ICASSP*，第 5206–5210 页。IEEE，2015。

Julian D Parker、Anton Smirnov、Jordi Pons、CJ Carr、Zack Zukowski、Zach Evans 和 XuboLiu。面向低比特率、高质量语音编码的 Transformer 扩展。*arXiv* 预印本 *arXiv:2411.19842*，2024。

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang 等。Movie gen：一系列媒体基础模型。*arXiv*预印本 *arXiv:2410.13720*，2024。

Konpat Preechakul、Nattanat Chatthee、Suttisak Wizadwongsa 和 Supasorn Suwajanakorn。扩散自编码器：迈向有意义且可解码的表示。载于 *IEEE/CVF*计算机视觉与模式识别会议论文集，第 10619–10629 页，2022。

Alec Radford、Jong Wook Kim、Tao Xu、Greg Brockman、Christine McLeavey 和 Ilya Sutskever。通过大规模弱监督实现鲁棒语音识别。发表于 *ICML*，第 202 卷，第 28492–28518 页，2023 年。Luis A P´erez Rey、Vlado Menkovski 和 Jacobus W Portegies。扩散变分自编码器。*arXiv* 预印本 *arXiv:1901.08991*，2019 年。Robin Rombach、Andreas Blattmann、Dominik Lorenz、Patrick Esser 和 Bj¨orn Ommer。利用潜在扩散模型合成高分辨率图像。发表于 *IEEE/CVF* 计算机视觉与模式识别会议论文集，第 10684–10695 页，2022 年。Takaaki Saeki、Detai Xin、Wataru Nakata、Tomoki Koriyama、Shinnosuke Takamichi 和 Hi-roshi Saruwatari。UTMOS：面向 Voicemos Challenge 2022 的 Utokyo-sarulab 系统。*arXiv* 预印本 *arXiv:2204.02152*，2022 年。

Kai Shen、Zeqian Ju、Xu Tan、Eric Liu、Yichong Leng、Lei He、Tao Qin、Sheng Zhao 和 Jiang Bian。NaturalSpeech 2：潜在扩散模型是自然且零样本的语音与歌唱合成器。载于第十二届国际学习表征会议，*ICLR 2024*，奥地利维也纳，*2024*年*5*月*7*日至*11*日。OpenReview.net，2024。网址：https://openreview.net/ forum?id=Rc7dAwVL3v。

Hubert Siuzdak。Vocos：弥合时域与基于傅里叶变换的神经声码器在高质量音频合成方面的差距。载于*ICLR*。OpenReview.net，2024。

Yang Song、Jascha Sohl-Dickstein、Diederik P. Kingma、Abhishek Kumar、Stefano Ermon 和 Ben Poole。通过随机微分方程进行基于分数的生成建模。载于第九届国际学习表征会议，*ICLR 2021*，奥地利，线上会议，*2021*年*5*月*3*日至*7*日。OpenReview.net，2021。网址：https://openreview.net/forum?id=PxTIG12RRHS。

Jianlin Su、Murtadha Ahmed、Yu Lu、Shengfeng Pan、Wen Bo 和 Yunfeng Liu。Roformer：采用旋转位置嵌入的增强型 Transformer。*Neurocomputing*，568:127063，2024。Hugo Touvron、Louis Martin、Kevin Stone、Peter Albert、Amjad Almahairi、Yasmine Babaei、Niko- lay Bashlykov、Soumya Batra、Prajjwal Bhargava、Shruti Bhosale 等。Llama 2：开放基础模型和微调聊天模型。*arXiv* 预印本 *arXiv:2307.09288*，2023。A¨aron van den Oord、Oriol Vinyals 和 Koray Kavukcuoglu。神经离散表征学习。载于*NIPS*，第 6306–6315 页，2017。Xinsheng Wang、Mingqi Jiang、Ziyang Ma、Ziyu Zhang、Songxiang Liu、Linqin Li、Zheng Liang、Qixi Zheng、Rui Wang、Xiaoqin Feng 等。Spark-tts：一种采用单流解耦语音令牌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*，2025a。Xiong Wang、Yangze Li、Chaoyou Fu、Yunhang Shen、Lei Xie、Ke Li、Xing Sun 和 Long Ma。Freeze-omni：一种采用冻结大语言模型的智能低延迟语音到语音对话模型。*arXiv* 预印本 *arXiv:2411.00774*，2024。Yuancheng Wang、Dekun Chen、Xueyao Zhang、Junan Zhang、Jiaqi Li 和 Zhizheng Wu。Tadi-codec：用于语音语言建模的文本感知扩散语音分词器。载于神经信息处理系统进展，2025b。预印本见 arXiv:2508.16790。Yuancheng Wang、Haoyue Zhan、Liwei Liu、Ruihong Zeng、Haotian Guo、Jiachen Zheng、Qiang Zhang、Xueyao Zhang、Shunsi Zhang 和 Zhizheng Wu。Maskgct：采用掩码生成式编解码器 Transformer 的零样本文本转语音。载于 *ICLR*。OpenReview.net，2025c。

Simon Welker、Matthew Le、Ricky TQ Chen、Wei-Ning Hsu、Timo Gerkmann、Alexander Richard 和 Yi-Chiao Wu。Flowdec：一种基于流的全频带通用音频编解码器，具有出色的感知质量。*arXiv* 预印本 *arXiv:2503.01485*，2025。

Detai 辛、Xu Tan、ShinnosukeTakamichi 和 HiroshiSaruwatari。Bigcodec：突破低比特率神经语音编解码器的极限。*arXiv* 预印本 *arXiv:2409.05377*，2024。

Jin Xu、Zhifang Guo、Jinzheng He、Hangrui Hu、Ting He、Shuai Bai、Keqin Chen、Jialin Wang、Yang Fan、Kai Dang 等。Qwen2.5-omni 技术报告。*arXiv* 预印本，2025。

An Yang、Baosong Yang、Beichen Zhang、Binyuan Hui、Bo Zheng、Bowen Yu、Chengyuan Li、Dayiheng Liu、Fei Huang、Haoran Wei、Huan Lin、Jian Yang、Jianhong Tu、Jianwei Zhang、Jia nxin Yang、Jiaxi Yang、Jingren Zhou、Junyang Lin、Kai Dang、Keming Lu、Keqin Bao、Kexin Yang、Le Yu、Mei Li、Mingfeng Xue、Pei Zhang、Qin Zhu、Rui Men、Runji Lin、Tianhao Li、Tingyu Xia、Xingzhang Ren、Xuancheng Ren、Yang Fan、Yang Su、Yichang Zhang、Yu Wan、YuqiongLiu、Zeyu Cui、Zhenru Zhang 和 Zihan Qiu。Qwen2.5 技术报告。*arXiv* 预印本，abs/2412.15115，2024a。

Dongchao Yang、Songxiang Liu、Haohan Guo、Jiankun Zhao、Yuanyuan Wang、Helin Wang、Zeqian Ju、Xubo Liu、Xueyuan Chen、XuTan 等。Almtokenizer：一种用于音频语言建模的低比特率、语义丰富的音频编解码器分词器。*arXiv* 预印本 *arXiv:2504.10344*，2025。

Haici Yang、Inseon Jang 和 Minje Kim。通过潜在扩散为神经语音编解码器进行生成式反量化。载于 *ICASSP 2024-2024 IEEE*国际声学、语音与信号处理会议 *(ICASSP)*，第 1251–1255 页。IEEE，2024b。

Zhen Ye、Peiwen Sun、Jiahe Lei、Hongzhan Lin、Xu Tan、Zheqi Dai、Qiuqiang Kong、Jianyi Chen、Jiahao Pan、Qifeng Liu 等。编解码器至关重要：探索编解码器对音频语言模型的语义缺陷。载于*AAAI*人工智能大会论文集，第 39 卷，第 25697–25705 页，2025a。

Zhen Ye、Xinfa Zhu、Chi-Min Chan、Xinsheng Wang、Xu Tan、Jiahe Lei、Yi Peng、Haohe Liu、Yizhu Jin、ZheqiDai 等。Llasa：扩展基于 Llama 的语音合成在训练时和推理时的计算量。*arXiv* 预印本 *arXiv:2502.04128*，2025b。

Sihyun Yu、Sangkyung Kwak、Huiwon Jang、Jongheon Jeong、Jonathan Huang、Jinwoo Shin 和谢赛宁。用于生成的表征对齐：训练扩散 Transformer 比你想象的更简单。载于 *ICLR*。OpenReview.net，2025。

Neil Zeghidour、Alejandro Luebs、Ahmed Omran、Jan Skoglund 和 Marco Tagliasacchi。Sound-stream：一种端到端神经音频编解码器。*IEEE/ACM*音频、语音与语言处理汇刊，30：495–507，2021。

Aohan Zeng、Zhengxiao Du、Mingdao Liu、Kedong Wang、Shengmin Jiang、Lei Zhao、董宇晓和唐杰。GLM-4-Voice：迈向智能且类人化的端到端语音聊天机器人。*arXiv* 预印本 *arXiv:2412.02612*，2024a。

Aohan Zeng、Zhengxiao Du、Mingdao Liu、Lei Zhang、Shengmin Jiang、董宇晓和唐杰。利用合成交错数据扩展语音-文本预训练。发表于*ICLR*。OpenReview.net，2024b。

Biao Zhang 和 Rico Sennrich。均方根层归一化。神经信息处理系统进展，32，2019。

Xin Zhang、Dong Zhang、Shimin Li、周雅倩和 Xipeng Qiu。SpeechTokenizer：面向语音大语言模型的统一语音分词器。*arXiv* 预印本 *arXiv:2308.16692*，2023。

Xueyao Zhang、Yuancheng Wang、Chaoren Wang、Ziniu Li、Zhuo Chen 和 Zhizheng Wu。通过偏好对齐提升跨多样化领域的零样本语音合成可懂度。载于 *ACL*。计算语言学协会，2025a。

Xueyao Zhang、Junan Zhang、Yuancheng Wang、Chaoren Wang、Yuanzhe Chen、Dongya Jia、Zhuo Chen 和 Zhizheng Wu。Vevo2：通过统一韵律学习，弥合可控语音与歌声生成之间的差距。*arXiv* 预印本 *arXiv:2508.16332*，2025b。

Xueyao Zhang、Xiaohui Zhang、Kainan Peng、Zhenyu Tang、Vimal Manohar、Yingru Liu、Jeff Hwang、Dangna Li、Yuhao Wang、Julian Chan、Yuan Huang、Zhizheng Wu 和 Mingbo Ma。Vevo：通过自监督解耦实现可控的零样本语音模仿。发表于*ICLR*。OpenReview.net，2025c。

Yue Zhao、Yuanjun Xiong 和 Philipp Kr¨ahenb¨uhl。采用二进制球面量化进行图像和视频词元化。*arXiv* 预印本 *arXiv:2406.07548*，2024年。

A 模型架构

模型 该分词器的架构由因果编码器、向量量化器（VQ）、扩散解码器和因果辅助 CTC 解码器组成。对于因果编码器和因果辅助 CTC 解码器，我们采用标准的 Llama 风格 Transformer 块（Touvron 等人，2023；Grattafiori，2024），并使用 RoPE 位置编码（Su 等人，2024）和 SiLU（Elfwing 等人，2018）激活函数。编码器具体采用 16 个因果 Llama 解码器层，辅助 CTC 解码器则采用 4 个此类层。除非另有说明，这些组件（以及扩散解码器）均采用一致的配置：隐藏维度为 1536，中间维度为 4096，注意力头数为 16。VQ 模块默认配置为 32 维，码本包含 65,536 个条目，并通过指数移动平均（EMA）进行更新（van den Oord 等人，2017）。扩散解码器由 16 层构成，其结构由因果 Llama 解码器改造为非因果形式。通过将 RMSNorm（Zhang 和 Sennrich，2019）替换为自适应 RMSNorm 变体，将扩散步嵌入纳入模型。为研究模型容量的影响，我们在保持其他架构设置不变的情况下，调整编码器和扩散解码器的层数。我们测试了四种配置：**S**（8 个编码器层 / 8 个解码器层，0.63B 个参数）、**B**（12/12，0.88B）、**L**（16/16，1.12B）和**XL**（24/24，1.61B）。除非另有说明，我们默认采用**L** 配置。

B 相关工作

离散语音分词器 语音分词器是语音语言模型的基础组件。早期方法（Zeghidour 等人，2021；D´efossez 等人，2022；Kumar 等人，2023）主要侧重于音频压缩，依赖残差向量量化（RVQ）（Zeghidour 等人，2021；Lee 等人，2022），并采用较高的帧率和比特率，这并不利于语言建模。近来的研究转向专为语言建模设计的分词器，着重采用较低的帧率（D´efossez 等人，2024；Li 等人，2025a；Della Libera 等人，2025）、语义丰富的表征（Zhang 等，2023；D´efossez 等人，2024；Li 等人，2025a；Wang 等人，2025a；Ye 等人，2025a;b；Liu 等人，2024；Guo 等人，2024；Du 等人，2024b；Zhang 等人，2025c；Jiang 等人，2025；Bie 等人，2025），以及简化的单层码本（Parker 等人，2024；Xin 等人，2024；Ji 等人，2024）。尽管如此，许多此类分词器仍难以实现更高的压缩率，例如使用单一码本达到 12.5 Hz。TaDi-Codec（Wang 等人，2025b）借助扩散自编码器并将文本融入解码器，成功将帧率降至 6.25 Hz；然而，这种文本感知设计固有地限制了其适用性，使其不适用于语音理解任务。本研究旨在设计一种分词器，使其能够同时实现足以支持高效语言建模的压缩率、保持高质量的音频重建，并学习有效且语义丰富的表征，以理解语音。

基于扩散的语音分词器 基于扩散的方法（Ho 等人，2020；Song 等人，2021）已成为一个颇具前景的方向，在低令牌率下展现出出色的可扩展性和鲁棒性。然而，现有的大多数方法仍采用两阶段设计：首先从自监督语音模型中提取令牌（Baevski 等人，2020；Chung 等人，2021；Hsu 等人，2021；Chen 等人，2022；Chiu 等人，2022；Radford 等人，2023），然后才通过扩散重建波形。这种分离限制了联合优化，因为量化器并未与解码器进行端到端训练。近期研究（Welker 等人，2025；Yang 等人，2024b）尝试利用扩散提高反令牌化保真度，但仍受限于相对较高的令牌率，并且仍依赖两阶段建模。因此，如何在紧凑且语言模型友好型框架中，将基于扩散的分词器推进到超低比特率（例如，低于 0.2 kbps 或 20 个令牌/秒），仍是一项悬而未决且至关重要的挑战。在这项工作中，我们通过***SiTok***应对这一挑战。SiTok 是一种基于扩散的语音分词器，在端到端框架中统一了向量量化与重建建模，同时引入语义正则化，以确保学习到的编码既高度压缩，又富含语义信息，适用于语音语言建模。

C 更多实验与结果

C.1 轻量级扩散头

扩散推理效率低下的一个主要原因是，每个去噪步骤都需要对整个模型执行一次前向传播。我们的初步实验表明，对于基于自编码器的分词器，仅使用简单的回归损失（例如 L1 或 L2）也能以可接受的可懂度重建语音，尽管感知质量较差。这表明，解码器的主体能够生成语音的基本结构，而迭代扩散过程主要用于提升其质量并细化细节。

基于这一洞见，我们提出对解码器 $\mathcal{D}_\phi$ 进行划分，将主体结构生成与迭代细化解耦。我们将解码器分为两个组件：规模较大的主体部分 $\mathcal{D}_{\phi_{\text{main}}}$，以及更小的**轻量级扩散头** $\mathcal{D}_{\phi_{\text{head}}}$。主体部分由最初的、更深层的 Transformer 模块组成，而扩散头则由最后几个模块构成。

在解码过程中，量化嵌入序列 $\boldsymbol{z}_q$ 首先仅经过主体部分 $\mathcal{D}_{\phi_{\text{main}}}$ 一次，以生成**基础表示** $\boldsymbol{h}_{\text{base}}$：

$$\boldsymbol{h}_{\text{base}} = \mathcal{D}_{\phi_{\text{main}}}(\boldsymbol{z}_q).$$

随后，该基础表示提供基础条件信息，再由扩散头进一步细化为最终谱图。迭代去噪过程完全由轻量级扩散头 $\mathcal{D}_{\phi_{\text{head}}}$执行，该扩散头以带噪声谱图 $\boldsymbol{x}_t$ 和基础表示 $\boldsymbol{h}_{\text{base}}$ 作为条件信息，预测速度：

$$v_\phi(\boldsymbol{x}_t, t, \boldsymbol{h}_{\text{base}}) = \mathcal{D}_{\phi_{\text{head}}}(\boldsymbol{x}_t, t, \boldsymbol{h}_{\text{base}}).$$

这种架构调整显著降低了每个推理步骤的计算开销，因为 $\mathcal{D}_{\phi_{\text{main}}}$ 解码器的大部分参数只需在一次前向传播中使用。这种方法能够实现快速推理，同时保留扩散模型的高质量合成能力。

在这项工作中，我们将所提出的轻量级扩散头应用于包含 16 个 Transformer 解码器层的基础模型架构。具体而言，解码器的前 12 层（$3/4$ ）构成主体 $\mathcal{D}_{\phi_{\text{main}}}$，最后 4 层则作为扩散头 $\mathcal{D}_{\phi_{\text{head}}}$。推理期间，主体只计算一次基础表示 $\boldsymbol{h}_{\text{base}}$ ，而只有轻量级扩散头会在扩散步骤中迭代运行。默认设置为 16 个扩散步骤，与每个去噪步骤都运行全部 16 层相比，理论上每步计算量可减少 $4\times$ 。

如表 6 所示，这种架构调整带来的重建性能几乎与完整模型相同。轻量级扩散头的词错误率和感知指标（SIM 和 UTMOS）与完整模型相当，同时显著降低了计算成本。这表明，内容和结构生成的大部分繁重工作由主体完成，而轻量级扩散头足以在扩散推理期间细化声学细节。

表 6：轻量级扩散头的消融研究。

模型	**重建**		
	词错误率	**相似度**	**UTMOS**
Base	4.06	0.641	3.44
使用轻量级扩散头	3.97	0.610	3.46

C.2 使用 SITOK 进行零样本合成

我们还在下游语音生成任务**零样本文本转语音（TTS）**中评估 SiTok，以进一步证明 SiTok 在语音语言建模中的有效性与效率。该实验验证了 SiTok 学习到的离散表示不仅适用于重建和理解，还能作为语音语言模型的强大生成接口。

表 7：SiTok-AR-TTS 与具有代表性的自回归文本转语音系统的零样本文本转语音结果对比。帧每秒/令牌每秒的定义见第 3.2 节。实时率是在单张 A100 GPU 上测得的。

模型	帧每秒/令牌每秒	词错误率 (↓)	相似度 (↑)	实时率 (↓)
CosyVoice 2（Du 等人，2024b）	25/25	2.89	0.66	0.455
SparkTTS（Wang 等人，2025a）	12.5/12.5	2.50	0.57	0.601
Llasa（Ye 等人，2025b）	50/50	3.94	0.58	0.422
SiTok-AR-TTS	12.5/12.5	2.46	0.64	0.234

我们构建了一个 0.5B 参数的基于大语言模型的文本转语音模型（记为*SiTok-AR-TTS*），由 Qwen2.5-0.5B（Yang 等人，2024a）初始化，以自回归方式根据文本预测 SiTok 令牌。该模型在 Emilia（He 等人，2024）数据集的 10 万小时数据上，按照标准自回归文本转语音训练方案进行训练。在推理过程中，预测得到的离散令牌序列由我们的扩散解码器解码以获得梅尔频谱图；本实验中我们使用默认的 16 步解码。

我们遵循第 3.2 节中的评估协议，并在 SeedTTS 测试英语集上报告 WER 和 SIM。为进一步评估实际效率，我们还报告在单个 NVIDIA A100 GPU 上测得的**实时率（RTF）**，该数据取自合成一段 10 秒语句的 10 次运行的平均值。结果汇总于表 7。我们采用了一些强大的基于自回归的文本转语音模型作为基线。

结果表明，与强基线相比，SiTok-AR-TTS 的可懂度和说话人相似度具有竞争力，甚至更胜一筹，同时推理成本显著更低。有趣的是，SiTok-AR-TTS 的 WER 甚至低于 SiTok 本身的重建 WER。这一趋势与近期的一些零样本文本转语音系统一致，例如 Du 等人（2024a）、Guo 等人（2024）和 Zhang 等人（2025a），这些系统生成的语音令牌直接以文本为条件。

另一项关键发现是效率显著提升。由于 SiTok 的运行频率仅为 12.5 Hz，自回归文本到令牌的解码所处理的序列长度只有传统神经编解码器（运行频率为 25 至 50 Hz）的一半。这一缩减直接加快了语音生成的推理速度，使实时率（RTF）显著降至 0.234，因此 SiTok 对大规模文本转语音或语音生成系统尤具吸引力。总体而言，这些发现表明，SiTok 是一种高效的语音生成接口：其离散表示不仅支持高质量重建和出色的下游理解，还能在统一的语音令牌化框架中实现高效、高保真的文本转语音。

观察： SiTok 在零样本文本转语音方面表现出色，兼具高可懂度和高相似度；其极低的令牌速率还使推理速度比现有的基于自回归的文本转语音系统快得多。

C.3 与替代量化方法的比较

SiTok 并不依赖于某种特定的量化设计。原则上，任何量化方法，例如有限标量量化（FSQ）（Mentzer 等，2023）和二元球面量化（BSQ）（Zhao 等，2024），都可以集成到我们的扩散自编码器中。本节将我们的标准 VQ 模块与一种有代表性的替代方案——固定标量量化（FSQ）进行比较。我们采用了常用的 FSQ 配置，各维度的基数为$[2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2]$,，其笛卡尔积得到的码本大小为 2^{16} =65536，与我们 VQ 设置所用的码本大小完全相同。

表 8 总结了实验结果。尽管码本大小相同，标准向量量化在重建质量和下游理解任务方面均表现出更强的性能。我们将这一性能差距归因于大规模训练方案的两个特性。首先，向量量化中可学习的嵌入向量相比 FSQ 固定的标量分区具有更强的表示灵活性，能够更丰富地建模细粒度的声学和语义结构。其次，在大批次大小、指数移动平均更新和基于扩散的优化下，SiTok 在整个训练过程中都表现出稳定的码本利用率，超过 95%，这意味着 FSQ 通常具有的优势——更高的

量化稳定性——在我们的设置中带来的收益有限。因此，向量量化成为表达能力更强且经验上更有效的选择，不过我们的结果证实，SiTok 仍然兼容广泛的量化技术。

表 8：SiTok 中向量量化与 FSQ 的比较。

模型	词错误率（↓）	相似度（↑）	UTMOS（↑）	CTC自动语音识别 (↓)	自动语音识别 (↓)	错误率 (↑)	说话人验证 (↑)	主观意见分数 (↑)
SiTok (向量量化)	4.06	0.641	3.44	9.50	4.95	63.5	13.8	96.9
SiTok (FSQ)	5.23	0.629	3.44	10.02	5.33	62.0	14.1	96.9

D 可复现性声明

为支持可复现性并促进未来研究，我们在本附录中提供 SiTok 的全面实现细节。具体而言，我们包括：(1) 详细的架构规范（附录 A）以及 SiTok 模型的伪代码（附录 D.1）；(2) 概述扩散自编码器核心端到端训练循环的伪代码（附录 D.2）；以及 (3) 有关训练超参数、数据预处理和其他实现考量的补充信息（附录 D.3）。

我们还确认，将于发表时向研究界发布完整的推理代码和预训练模型检查点（基于公开的、仅供研究使用的数据集），使研究人员能够复现我们的结果，并在 SiTok 的基础上进一步开展研究。

D.1 SiTok 伪代码

列表 1：SiTok 的伪代码。

```
类 SiTok：def __init__( self, 输入_维度=128, 隐藏_大小=1536, 中间层_大小=4096, 编码器_层数=16, 解码器_层数=16, CTC_解码器_层数=4, 头数_, 向量量化_嵌入_维度=16, 下采样_因子=4, 词汇表_大小=32100, ): # 时间堆叠（将帧率降低至 12.5 Hz） self.stack_in =StackIn(downsample_factor) self.stack_out = StackOut(downsample_factor) # Transformer 编码器（Llama 风格的因果模型） self.encoder = LlamaModel( 隐藏_大小, 中间层_大小, 编码器_层数, 头数_, 输入_维度 * 下采样_因子) # 向量量化器（二元球面量化） self.vq_in =Linear(隐藏_大小,向量量化_嵌入_维度) self.vq = BinarySphericalQuantizer(向量量化_嵌入_维度) self.vq_out = Linear( 向量量化_嵌入_维度, 隐藏_大小) # 扩散解码器（DiT 风格的 Transformer） self.decoder = DiT(
```

```
            隐藏_大小, 中间层_大小, 解码器_层数, 头数_,
            use_cond=True, use_diff_step=True)

        # CTC 语义解码器（Llama 风格的因果模型）self.ctc_解码器
         = LlamaModel(hidden_大小, intermediate_大小, ctc_解码
        器_层数, num_头数, vocab_大小)


# ---------------------------------------------------------- # forward：训练时用于计算损失的输
出 # ----------------------------------------------------------def forward(self, x, x_
mask):"""SiTok 的前向传播，用于计算训练损失。"""


        # 1) 堆叠 + 编码为连续潜变量h = self.stack_
        in(x)h = self.encoder(h, x_mask)

        # 2) 向量量化为离散语音词元 z =self.vq_in(h) z_q, vq_info =
        self.vq(z) cond = self.vq_out(z_q) # 3) 前向扩散（流匹配） t =
        sample_uniform() # t ~U(0, 1) eps = randn_like(x) # eps ~
        N(0,I) x_t = self.forward_diffuse(x, eps, t) # 含噪梅尔谱 x_t =
        self.stack_in(x_t) # 4) 扩散解码器预测流/速度 flow_pred =
        self.decoder(x_t, t, cond)




        # 5）CTC 语义逻辑值（用于语义正则化）ctc_逻辑值 = self.ctc_
        decoder(cond)

        return {"x":x,                                  # GT 梅尔"noise":eps,
        # 扩散噪声目标 "flow_pred":flow_pred, # 用于流匹配损失 "ctc_
        logits":ctc_logits,       # 用于 CTC 损失"vq_loss":vq_info["
        commit"], # 向量量化承诺损失}


# ---------------------------------------------------------- # 前向扩散（流匹配目标） #


def forward_diffuse(self, x, eps, t):"""应用前向扩散，得到带噪样本 x_t。
"""#x_t =(1-alpha(t)) * eps +alpha(t) * x # 实际上，alpha(t) 实现了流匹配调
度。x_t =(1-t) * eps +t * xreturn x t_


# ---------------------------------------------------------- # 推理辅助函数 # ----------------------------------------------------------

定义 encode(self, x, mask):"""将梅尔编码为量化后的向量量化嵌入 / 索引。
"""h = self.stack_in(x)h = self.encoder(h, mask)z =
self.vq_in(h)
```

```
    z_q, 索引  = self.vq(z)返回 z_q, 索引定义 decode(self, z_q,
prompt=None, steps=N):"""通过反向扩散从 VQ 嵌入生成梅尔频谱。"""cond
 = self.vq_out(z_q) 梅尔频谱 =扩散_
reverse(self.decoder,cond,prompt,steps)返回 self.stack_
out(mel)
```

D.2 训练循环伪代码

清单 2：SiTok 训练循环的伪代码。

```
对于 批次 中的 数据加载器:

    # ------------------------------------------------------------
    # 1) 准备梅尔特征和掩码
    # ------------------------------------------------------------
    x          = 梅尔_提取器(batch.语音)                    # [B,  T, d]
    x 掩码_    = batch.语音_掩码
    文本 ID_       = batch.文本 ID_                          # 语义监督
    text_mask  = 批次.text      _mask

    # ------------------------------------------------------------
    # 2) 通过 SiTok 进行前向传播
    # ------------------------------------------------------------
    out  = sitok.forward(x    ,  x_掩码)
    x_gt           = 输出["x"]
    噪声           = 输出["噪声"]
    流_预测        = 输出["流_预测"]
    连接时序分类_logits      = 输出["连接时序分类_logits"]
    向量量化_损失          = 输出["向量量化_损失"]

    # ------------------------------------------------------------
    # 3）扩散（流匹配）损失
    # ------------------------------------------------------------
    # 目标速度 v*  =x - eps
    flow_gt  = x_gt - 噪声
    diff_损失  =L1(flow_pred,          flow_gt)

    # ------------------------------------------------------------
    # 4) CTC语义损失
    # ------------------------------------------------------------
    ctc_损失  = CTC_损失(ctc_logits, text_ids, text_mask)

    # ------------------------------------------------------------
    # 5) 总损失
    # ------------------------------------------------------------
    总_损失  =扩散_损失  + 向量量化_损失  + lambda_连接时序分类 * 连接时序分类_损失

    # ------------------------------------------------------------
    # 6) 优化
    # ------------------------------------------------------------
    优化器.zero_grad()
    总_损失.backward()
    clip_梯度(sitok.parameters(), max_norm=0.5)
    优化器.step()
```

D.3 更多实现细节

数据与预处理 我们使用 200 万小时的内部数据来训练模型。该数据集涵盖多种语言，其中绝大多数为英语。我们不对语音数据进行额外的预处理，例如将其拆分为更短的片段；相反，我们直接使用原始的话语长度及其配套的转录文本进行训练。我们使用以 24K Hz 采样率提取的 50 Hz、128 个梅尔频带的梅尔频谱图，帧移为 480 个采样点（20 ms），窗口大小为 1920 个采样点（80 ms）。STFT 使用 $n_{\text{fft}} = 1920$ 汉宁窗计算，梅尔滤波器覆盖 $[f_{\min}, f_{\max}] = [0, 12{,}000]$ Hz。最后，我们使用预先计算的统计数据（均值 -4.92，方差 8.14）对梅尔特征应用全局均值-方差归一化，并将其同时作为分词器的输入和重建目标；首先将每四个连续帧进行堆叠，以将帧率降至 12.5 Hz，从而提高训练效率。在波形合成方面，我们采用基于 Vocos 的（Siuzdak，2024）声码器，将梅尔频谱图还原为 24K Hz 的音频波形。

训练 我们对所有模型训练一个训练轮次，相当于约 450K 个步骤。在优化方面，我们采用 AdamW（Loshchilov 和 Hutter，2019）优化器，使用 $\beta_1 = 0.9$、$\beta_2 = 0.999$，权重衰减为 0.01，学习率为 8×10^{-5}，并进行 32K 步预热。为了最大化 GPU利用率并稳定不同话语长度下的训练，我们采用<策略：动态批次大小： 在每个 GPU 上，我们不断打包话语，直到总时长达到约 300 秒语音，对应于在 12.5 Hz 令牌速率下约 3750 个令牌。这样可以确保每个批次保持一致的计算开销，同时保留完整话语训练，而无需进行分段。

E 局限性

尽管 SiTok 在语音重建和下游语音理解任务上都展现出强劲性能，但其表现仍不及连续特征表示。未来的工作将致力于缩小这一性能差距。此外，我们基于扩散的解码器给流式生成带来了挑战。目前，我们正在研究微调策略，例如分块自回归扩散，以实现低延迟或流式输出。