arXiv:2601.02776v1 [cs.SD] 6 Jan 2026

UniSRCodec：采用子带重建的统一低比特率单码本编解码器

Zhisheng Zhang[1*], Xiang Li[1], Yixuan Zhou[1],Jing Peng[1],Shengbo Cai[1],Guoyang Zeng[2], Zhiyong Wu[1†]1清华大学深圳国际研究生院，中国深圳2ModelBest Inc.，中国北京

†通讯作者

摘要—神经音频编解码器（NACs）通过紧凑压缩和重建来减少传输开销，同时旨在弥合连续信号与离散信号之间的差距。现有神经音频编解码器可分为两类：多码本编解码器和单码本编解码器。多码本编解码器面临结构复杂、难以适配下游任务等挑战；而单码本编解码器虽然结构更简单，却存在保真度低、统一音频建模效果不佳以及无法支持高频音频建模等问题。我们提出了 UniSRCodec，这是一种支持高采样率、低带宽、高保真度和统一音频的单码本编解码器。我们分析了基于波形的压缩效率低下的问题，并提出一种基于梅尔频谱图的时频压缩方法，同时结合声码器恢复原始音频的相位信息。此外，我们提出一种子带重建技术，以实现低频和高频频带上的高质量压缩。主观和客观实验结果表明，UniSRCodec 的令牌率仅为 40，却在跨域单码本编解码器中达到了最先进水平，其重建质量可媲美某些多码本方法。演示页面请访问 https://wxzyd123.github.io/unisrcodec。

索引词—统一音频编解码器、低比特率、高保真度

I. 引言

神经音频编解码器 [1], [2] 是一种压缩与恢复技术，可将连续的语音信号转换为离散令牌，从而降低音频传输成本。近年来，大型音频语言模型（音频语言模型） [3], [4] 凭借出色的对话能力，受到广泛关注。语音分词器部分会将输入音频转换为令牌，并将其输入大语言模型。此外，每秒令牌的信息熵越大，大语言模型提取信息的能力就越强。

现有的神经音频编解码器（NACs）根据所使用码本的数量，大致可分为多码本编解码器和单码本编解码器。多码本编解码器一直是先前研究的主流。通过分层利用多个码本，例如残差向量量化（RVQ），这类编解码器能够实现高保真音频重建。然而，这类编解码器会生成多级令牌序列，从而增加了将其适配到音频语言模型（ALMs）或文本转语音系统中的复杂性。近年来，研究者开始探索单码本编解码器，原因在于其架构简单，例如BigCodec [5],WavTokenizer [6],和 UniCodec [7]。然而，以往的单码本编解码器存在两个主要局限：（1）**通用性**。它们对通用音频的建模能力较弱，例如BigCodec。（2）**高频建模**。它们通常以较低的采样率运行，同时需要较高的带宽或大量计算资源。较低的采样率（如 24kHz）可能足以处理语音内容，但对于音乐或通用音频而言则表现不足，其感知质量低于高采样率。此外，较高的采样率也会给统一音频表示和建模带来重大挑战。

为解决上述局限性，我们提出一种名为**统一子带重建编解码器**（UniS-RCodec）的神经音频编解码器。它是一种高采样率、低比特率、高保真度且统一单码本的音频编解码器，并且训练开销低。单码本设计旨在更好地适配下游任务。高频建模能够带来更丰富、更自然的音频质量和感知保真度，并使编解码器能够有效建模通用音频类型。低比特率要求每个令牌尽可能承载更多信息，体现出更高的信息密度。高保真度可确保离散化过程中信息损失最小，这对压缩至关重要。统一能力要求编解码器能够建模跨域音频。此外，训练开销低意味着该编解码器训练时资源消耗较少。

基于波形的技术 [6], [7]压缩时域时，往往会保留冗余信息，且对频域的建模效果较差。为缓解这一问题，我们旨在提出一种压缩策略，利用梅尔频谱图作为重建表示，以提高每个令牌的信息密度。压缩过程中，我们有意省略相位信息，从而将带宽更高效地分配给感知上至关重要的成分。随后，在音频重建过程中，通过神经声码器恢复舍弃的相位信息；该声码器能够根据梅尔表示合成高质量波形。采用这种方法，我们仅以 40 的令牌率和超低的 0.52 千比特每秒比特率，即可实现高保真音频压缩与重建。此外，与 UniCodec [7],相比，该方法的训练过程计算开销较低，仅需使用 8 张 NVIDIA RTX 4090 GPU 训练约 12 小时。实验结果表明，我们的方法在跨领域音频建模方面达到了单码本编解码器中的最先进水平（SOTA）——

*本研究工作是在第一作者于 ModelBest 实习期间完成的。

同时实现了可与多码本方法相媲美的重建保真度。

我们的主要贡献总结如下：

- 我们提出了 UniSRCodec，这是一种高采样率、超低比特率的统一单码本音频编解码器，通过利用时域和频域表示，实现高质量音频建模。
- 我们提出了一种子带重建方法，以更好地建模低频和高频信息。
- 我们在多种跨域数据集上评估了 UniSRCodec，这些数据集涵盖语音、音乐和通用声音，并证明仅使用每秒 40 个令牌时，其性能在单码本编解码器中达到最先进水平。

II. 相关工作

A. 单码本编解码器

单码本编解码器结构简单，易于适配下游任务，也可用作大语言模型的输入语音分词器。BigCodec [5] 是一种低比特率语音编解码器，通过扩大模型参数规模来增强压缩能力。WavTokenizer [6]率先将多级码本集成到单一码本中。通过使用 K 均值聚类初始化码本，并在解码器中引入注意力机制，WavTokenizer 能够以较低比特率实现高保真语音重建。在 WavTokenizer 的基础上，UniCodec [7]提出采用混合专家（MoE）架构，并结合一个大小为 16384 的大型领域自适应码本来建模统一音频。然而，这些单码本编解码器存在领域限制，例如 BigCodec，或采样率受限，即16kHz 或 24kHz，这限制了它们忠实重建高保真音频的能力。此外，它们在超低带宽条件下对统一音频的重建质量仍有提升空间。

B. 基于频谱的神经音频编解码器

此前已有研究探索了频域压缩。APCodec [8]提出了一种频域压缩方法，对从波形的短时傅里叶变换（STFT）中获得的幅度谱和相位谱分别应用一维卷积。然而，该方法仅在频域进行压缩，缺乏对时间信息的显式建模。FunCodec [9]在此基础上采用二维卷积，对时间和频率维度进行联合压缩。但其多码本结构限制了实际应用。MelCap [10]与我们的工作密切相关，与 FunCodec 相似，但将残差向量量化（RVQ）替换为单码本。值得注意的是，MelCap 利用图像领域预训练的 VGG 权重来构建梅尔频谱图，缺乏针对音频领域的专门设计考量。此外，该方法运行在相对较高的带宽下，令牌率为 260，比特率为 3.4 kbps。因此，我们旨在开发一种面向音频领域的高保真单码本编解码器，使其在更低的带宽条件下仍能有效运行。

III. UniSRCodec 设计

本节将解释我们为何选择梅尔频谱图作为输入表示，随后详细介绍 UniSRCodec 架构。

A. 为什么将梅尔频谱图作为输入？

本节将说明我们为何使用梅尔频谱图作为输入特征。

信息密度。 考虑一段长度为 65536、采样率为 44.1 kHz 的音频。经过跳跃大小为 512 的短时傅里叶变换后，梅尔频谱图的维度为 128×128。将其展平后，序列长度为 16384，约为原始波形长度的 25%。尽管梅尔频谱图舍弃了相位信息，但先进的声码器已证明，仅凭梅尔频谱图也能准确重建相位 [11]。因此，与原始波形相比，使用梅尔频谱图作为输入能够实现更高效、信息更丰富的表示。

二次压缩效率。与一维的原始波形不同，梅尔频谱图本质上是二维的。将波形压缩 n 倍，相当于仅降低时间维度上的分辨率。相比之下，将梅尔频谱图压缩 n 倍，会同时降低时间和频率维度上的分辨率，从而实现 $n\times n = n^2$ 的整体压缩比。这种二次压缩效率的提升，是实现超低令牌率 40 的关键。

时间域与频率域压缩。压缩仅作用于时间信息，这可能会导致频谱能量细节丢失。为实现高保真音频重建，必须保留时间域和频率域中的信息。

B. 架构

UniSRCodec从输入音频中提取梅尔频谱图，并通过编码器–量化器–解码器架构对其进行重建。随后，将重建后的梅尔频谱图输入预训练的 BigVGAN-v2 [11] ，以恢复可听波形。工作流程如图 1 所示。

编码器。 我们的编码器和解码器架构改编自 Open-MagViT2 [12]。编码器采用由多个残差块（ ResBlocks）组成的全卷积二维架构。每个 ResBlock 包含两个带激活函数的 Group-Norm 层和两个卷积层。梅尔频谱图的输入通道数为 1。在整个编码器中，通道数逐步增加至 $[128, 256, 512]$,，同时通过步幅卷积降低空间分辨率。具体而言，时间维度下采样 $[2, 2, 4]$ 倍，频率维度下采样 $[2, 2, 4]$, 倍，最终实现 16×16 的总体压缩率。训练过程中， 128×128 梅尔频谱图被压缩为 8×8 潜在表示。

量化器。由于编码器输出的潜在表示是二维的，因此必须将其展平成一维向量，以便进行量化。我们考虑两种展平策略：按频带和按帧。按频带

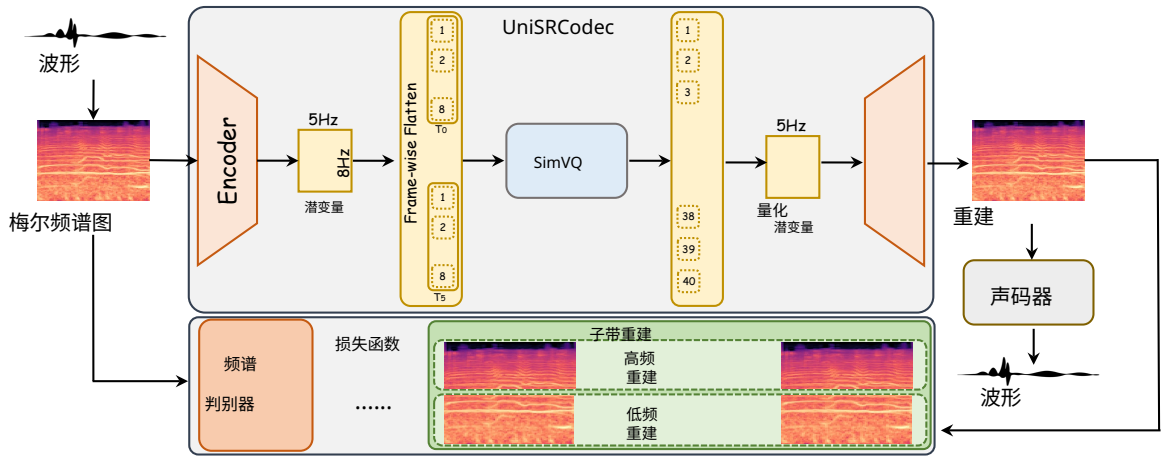

图 1：UniSRCodec 的架构和训练流程。

展平通过拼接每个梅尔频率箱对应的时间序列来保留全部时间信息，而按帧展平则会拼接每个时间帧内的所有频率箱。在 UniSRCodec 中，我们采用按帧展平策略，以保留随时间变化的连贯频率动态，确保每一帧都作为一个完整的频谱单元演变。这两种展平策略的性能将在第 V-D 节中展示。

解码器。解码器采用与编码器对称的架构。从量化后的码本向量出发，解码器重建梅尔频谱图。解码器以 512 个输入通道为起点，并逐步将通道数减少至 $[256, 128, 1]$。同时，它对空间维度进行上采样：时间轴扩展 $[4, 2, 2]$ 倍，频率轴扩展 $[4, 2, 2]$，倍，最终将潜在表示重建为原始128×128 梅尔频谱图大小。

IV. 训练流程

本节提出 UniSRCodec 的子带重建策略、训练流程和目标函数。UniSRCodec 的训练主要涉及编码器、量化器和解码器。

A. 子带重建

由于梅尔谱图的建模涉及不同频带的信息，我们首先重建整个梅尔谱图，并观察到尽管高频信号得到了很好的建模，但低频信号在一定程度上有所退化。此外，梅尔谱图低频区域的信息更为精细，因此模型应给予其更多关注并加以学习。基于这一观察，我们提出了一种子带重建方法。假设梅尔谱图有 m 个梅尔频点，短时傅里叶变换后的时间帧数为 t，则输入数据表示为向量 $x \in \mathbb{R}^{m \times t}$。我们将频率轴分为两半：前半部分对应低频信号 $x_{\text{low}} \in \mathbb{R}^{\frac{m}{2} \times t}$，后半部分对应高频信号 $x_{\text{high}} \in \mathbb{R}^{\frac{m}{2} \times t}$。我们分别计算重建后的低频信号 $\hat{x}_{\text{low}}$ 和高频信号 $\hat{x}_{\text{high}}$ 的 L1损失，并将其加权平均作为整体重建损失，如式（1）所示。我们将在第 V-D 节验证这一方法。

$$\mathcal{L}_{\text{sr}} = \frac{(\alpha_{\text{low}}|x_{\text{low}} - \hat{x}_{\text{low}}|_1 + \alpha_{\text{high}}|x_{\text{high}} - \hat{x}_{\text{high}}|_1)}{\alpha_{\text{low}} + \alpha_{\text{high}}}. \quad (1)$$

B. 梅尔谱图重建训练

在此过程中，我们训练编码器、量化器和解码器，以生成高质量的梅尔频谱图。此外，引入判别器也有助于梅尔频谱图的训练。否则，仅训练模型会导致第 V-D 节中所述的过度平滑伪影。

重建损失。我们采用所提出的子带重建 $\mathcal{L}_{\text{sr}}$，其中包含 $\alpha_{\text{low}} = 2$ 和 $\alpha_{\text{high}} = 1$。

判别器损失。引入判别器有助于生成器学习细粒度的频谱细节。我们采用 DAC 中的多频带多尺度短时傅里叶变换判别器架构 [2]，但移除了多频带和多尺度组件，因为梅尔频谱图输入已具有固定的频率分辨率和尺度。该损失记为 $\mathcal{L}_{\text{disc}}$。

对抗损失 遵循 DAC 的做法，我们使用基于频谱判别器的对抗损失 $\mathcal{L}_{\text{adv}}$，并结合频域中的特征匹配损失，记为 $\mathcal{L}_{\text{fm}}$。

码本损失。我们采用 SimVQ [13] 作为单一码本量化器，并使用承诺损失 $\mathcal{L}_{\text{cm}}$ 优化码本向量，以提高利用率。

训练目标。我们对上述损失项进行优化，从而得到式（2）中的总体训练目标。

$$\mathcal{L} = \lambda_{\text{sr}}\mathcal{L}_{\text{sr}} + \lambda_{\text{disc}}\mathcal{L}_{\text{disc}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{fm}}\mathcal{L}_{\text{fm}} + \lambda_{\text{cm}}\mathcal{L}_{\text{cm}}, \quad (2)$$

其中，系数 λ_{sr}，λ_{disc}，λ_{adv}，λ_{fm}，λ_{cm} 分别设为 15、1、1、1、1。这些超参数的选择基本遵循了以往研究中的成熟做法 [14], [2],，仅做了少量调整以实现最佳性能。

表 I：通用、音乐和语音领域测试数据集上的重建性能客观评估。“Mel-44”和“Mel-16”表示高频和低频上的 Mel 距离，“STFT-44”和“STFT-16”表示短时傅里叶变换距离。**粗体**表示单码本 NAC 中的最佳性能，下划线表示次佳性能。

模型	归因				AudioSet-Eval				MusicDB 测试集				LibriTTS 测试集	
	统一	TPS	千比特每秒/Nq	SR	Mel-44(↓)	STFT-44(↓)	Mel-16(↓)	STFT-16(↓)	Mel-44(↓)	STFT-44(↓)	Mel-16(↓)	STFT-16(↓)	STOI(↑)	PESQ(↑)
声码器使用真实梅尔频谱图进行重建														
BigVGAN [11]	✔	-	-	44.1	0.417	1.713	0.363	1.683	0.380	1.334	0.350	1.263	0.993	4.186
> 100 令牌率														
DAC [2]	✔	900	9/9q	44.1	**0.654**	**1.958**	**0.625**	**1.842**	**0.651**	1.634	0.665	**1.462**	**0.972**	**3.900**
Encodec [1]	✔	600	6/8q	24	1.315	5.030	0.889	2.271	1.372	4.705	1.086	2.020	0.943	2.819
Encodec [1]	✔	300	3/4q	24	1.413	5.134	1.017	2.448	1.463	4.769	1.203	2.134	0.904	2.116
SNAC [15]	✔	240	2.88/4q	44.1	0.828	2.145	0.863	2.234	0.788	1.673	0.845	1.645	0.925	2.561
MelCap [10]	✔	260	3.4/1q	44.1	0.817	2.229	0.873	2.409	0.796	1.813	0.896	1.870	0.888	1.802
UniSRCodec-L	✔	176	2.29/1q	44.1	0.729	2.049	0.692	2.093	0.656	**1.543**	**0.638**	1.529	0.941	2.727
≤ 100 令牌率														
DAC [2]	✔	100	1/1q	44.1	1.187	2.588	1.282	2.752	1.276	2.195	1.474	2.270	0.763	1.308
BigCodec [5]	✗	80	1.04/1q	16	2.250	7.336	1.366	3.024	1.958	6.639	1.031	2.003	**0.943**	2.700
TAAE [16]	✗	50	0.7/1q	16	2.999	7.896	2.385	4.314	2.490	7.006	1.746	2.927	0.890	1.787
WT-Speech [6]	✗	75	0.9/1q	24	1.393	5.179	1.026	2.572	1.341	4.700	0.997	1.923	0.922	2.566
WT-MA [6]	✗	75	0.9/1q	24	1.396	5.132	0.985	2.453	1.390	4.689	1.044	1.977	0.857	1.747
WT-Unified [6]	✔	40	0.48/1q	24	1.505	5.242	1.130	2.634	1.558	4.770	1.255	2.110	0.875	1.912
UniCodec [7]	✔	75	1.3/1q	24	1.376	5.169	0.903	2.401	1.352	4.713	0.943	1.858	0.940	**2.870**
UniSRCodec-B	✔	40	0.52/1q	44.1	**0.904**	**2.250**	**0.900**	**2.330**	**0.882**	**1.747**	**0.893**	**1.768**	0.875	1.836

(1) **WT-Speech**：WavTokenizer [6] 语音 领域。(2) **WT-MA**：适用于音乐和音频领域的 WavTokenizer。(3) **WT-Unified**：统一的 WavTokenizer。(4) **Nq**：量化器的数量。

五、实验与分析

A. 实验设置

数据集。训练集涵盖近 10000 小时的跨域数据。对于语音领域，我们采用 VCTK [17],、LibriTTS [18], 和 Common Voice [19]。对于音乐类别，我们使用 MusicDB [20] 和 Jamendo [21]。对于通用音频，我们使用 AudioSet [22]。测试集则使用 LibriTTS test-clean、MUSDBtest 和 AudioSet eval，每个领域各包含 1000 个样本 [2], [7]。

指标。对于跨域数据，我们采用最佳指标。对于音乐和通用声音，按照 DAC [2], 的做法，分别计算高频（44kHz）和低频（16kHz）分量中原始音频与重建音频的梅尔频谱图和线性频谱之间的 L1损失，以此求得梅尔频谱距离和 STFT 距离。对于语音评估，我们遵循 UniCodec [7],，选用包括 STOI 和 PESQ 在内的语音相关指标，以评估重建语音的生成质量。

对于量化指标，我们使用每秒令牌数（TPS） [7] 和带宽（kbps）。TPS 表示对一秒音频进行建模所需的令牌数量。带宽以千比特每秒（kbps）为单位，表示传输量化音频令牌所需的数据速率，反映编解码器在传输或存储成本方面的效率。对于主观评估，我们在第五节 E 小节进行了受 MUSHRA 启发的听音测试 [2] 。

基线。我们考虑最先进的神经音频编解码器，包括采用多码本的 DAC [2],Encodec [1],SNAC [15]和 Big- Codec [5], MelCap [10], TAAE [16], WavTokenizer [6],，以及采用单码本的 UniCodec [7] 。

训练详情。我们使用 AdamW 优化器，在 8 台 NVIDIA 4090 上训练 UniSRCodec 共 100000(×8) 步，初始学习率为 1×10^{-4}，批量大小为 20。

B. 高频数据训练

要实现高频建模，我们需要大量高分辨率数据。对低采样率音频进行上采样可能会导致高频成分丢失。因此，我们先对训练数据进行预处理筛选。首先，计算每个梅尔频带的平均能量，然后从最高频带开始向下搜索。如果某个频带的平均能量超过预设阈值，我们就认为其下方所有频带都包含有效的能量信息，并将对应于该频带的采样率视为音频的原生采样率。我们根据经验将能量阈值设为 -60dB，并参照 DAC [2], ，选取所有真实带宽超过奈奎斯特频率（22.05kHz）的音频数据作为训练数据。这有助于提升模型对高频信号的建模能力 [2]。

C. 跨领域数据集上的评估

本节评估 UniSRCodec 在语音、音乐和通用声音数据集上的性能。

表 I 展示了我们的方法和基线方法在跨域数据集上的评估结果。我们提供了 UniSRCodec 的两个变体：“UniSRCodec-B”表示基础版本，令牌率极低，仅为 40；“UniSRCodec-L”则是比特率稍高的变体，旨在与多码本神经音频编解码器相匹配。我们观察到，在音乐和通用音频领域，UniSRCodec-B 在单码本方法中取得了最先进性能，并能忠实地重建高频和低频信号。与 UniCodec [7], 相比，我们的方法以更低的比特率和带宽运行，同时仍能实现高保真高频成分重建。根据“Mel-44”和“STFT-44”这两项指标，我们提出的 UniSRCodec 能更好地建模高频信号，表明其在建模高保真音乐和通用音频方面表现更佳。此外，UniSRCodec-L 的表现优于多码本方法，

表 II：在 AudioSet 上进行的消融研究。“w/o”表示训练时不使用该组件，“w”表示使用该组件。

方法	AudioSet			
	Mel-44(↓)	STFT-44(↓)	Mel-16(↓)	STFT-16(↓)
UniSRCodec	**0.904**	**2.250**	**0.900**	**2.330**
不使用判别器	1.261	2.493	1.278	2.645
不使用子带	0.909	2.247	0.922	2.347
使用调度器	0.929	2.287	0.927	2.364
不使用逐帧展平	0.930	2.275	0.932	2.363

例如，SNAC 和 Encodec，而且比特率更低；其性能可与 DAC 持平甚至超越 DAC，在音乐领域的两项指标“STFT-44”和“Mel-16”上均胜过 DAC。

在语音领域，与 UniCodec 相比，UniSRCodec-B 的建模能力略低。这主要是因为 UniCodec 的训练数据绝大部分由 ~700000 小时的语音组成，而且 UniCodec 引入了语义学习，提升语义保真度的同时也增加了训练复杂度。从 UniSRCodec-L 的角度来看，当模型的采样率设为 24kHz、令牌率达到 90 时，其语音建模性能与 UniCodec 相当，而通用音频建模性能则显著优于 UniCodec。

这项实验不仅证明了 UniSRCodec 在令牌率较低（为 40 ）时对跨域数据的建模能力更强，尤其是在音乐和通用声音方面；还验证了其可扩展性：即使令牌率为 176，UniSRCodec 的表现仍优于多码本编解码器 SNAC。

D. 消融研究

本节将探讨各个组件的作用。

判别器。 在 UniSRCodec 中，我们设计了一种轻量级判别器，以增强编解码器对梅尔频谱图细粒度细节的学习能力。具体而言，我们对输入进行时间和频谱下采样，并基于各下采样阶段提取的特征计算特征匹配损失。如表 II 所示，移除判别器及其相关损失函数，仅保留重建损失和码本学习损失，会导致性能显著下降，合成音频中还会出现可听见的电子伪影。

子带重建。子带重建旨在提高低频信号分量的权重，从而增强模型学习低频特征的能力。当我们用传统的重建损失替代所提出的策略时，即，对

整个梅尔频谱图的低频范围内，重建质量有所下降。具体而言，“Mel-16”指标从原始的 0.901 增至 0.922，验证了我们的方法在低频建模方面的有效性。该策略专为我们时频压缩方法的独特特性而设计，可对不同频带进行独立处理。

调度器。 训练期间，由于模型具备相对较快的特征学习能力，学习率固定为1×10^{-4}。我们测试了添加指数调度器的情况，其

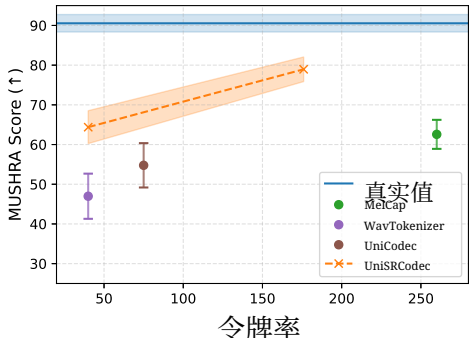

图 2：MUSHRA 主观评估得分与令牌率的关系，误差区间为 95% 置信区间。

衰减率为 0。999976974，在 100000 步后将学习率从 1×10^{-4} 降至 1×10^{-5}。表 II 显示，加入该调度器后性能有所下降，例如，“Mel- 44”指标从 0.903 降至 0.929，这表明恒定学习率可能更适合 UniSRCodec，以实现最佳学习效果。

逐帧展平与逐频带展平。由于输入是梅尔频谱图，编码器的输出仍是二维向量。为了便于查找码本，需要先将该向量展平为一维向量，再输入码本。展平策略有两种：（1）逐帧展平，将每个时间帧内的频率信息拼接起来；或（2）逐频带展平，将不同频带上的时间点拼接起来。我们测试了逐频带展平策略，结果如表 II 所示，该策略在 AudioSet 数据集的所有指标上均导致性能下降。这是因为训练期间沿时间步展平可能会导致推理时出现不一致，因为音频长度可变可能会降低模型性能。相比之下，逐帧展平确保了训练与推理的一致性，因为梅尔频点维度（即频率轴）保持固定。基于这项分析，我们在 UniSRCodec 中采用逐帧展平。

E. 主观评估

为评估 UniSRCodec 的感知性能，我们开展了 MUSHRA 听音测试。从三个领域中各随机选取三个音频样本，共计九个音频片段，并由十位专家根据重建质量进行主观评分。图 2 展示了三个领域的平均结果，其中 UniSRCodec-B 和 UniSRCodec-L 的主观重建评分分别位居第二和第一。

在音频和音乐领域，UniSRCodec 超越了 UniCodec [7]，这一最先进的统一单码本模型。具体而言，在音频领域，UniSRCodec-B 和 UniSRCodec-L 的得分分别为 53.433 和 72.667，优于 UniCodec 的 46.900。在音乐领域，提升更为显著：UniSRCodec-B 和 UniSRCodec-L 的平均 MUSHRA 得分分别为 62.833 和 80.967，优于 MelCap（59.900）和 UniCodec（33.933）。

表 III：跨领域下游分类任务。

模型	TPS	声音		音乐	语音
		UrbanSound-8k	ESC-50	GTZAN	CREMA-D
连续表示					
WavLM [23]	-	0.53	0.32	0.48	0.45
离散表示					
WavTokenizer [6]	40	0.33	0.17	0.40	0.39
UniSRCodec	40	**0.40**	**0.19**	**0.40**	**0.42**

在语音方面，UniSRCodec-B 和 UniSRCodec-L 的平均 MUSHRA 得分分别为 76.867 和 83.233，表现与 UniCodec 的 83.467 相当。此外，我们发现，基础版本在令牌率为 40 时，性能已优于令牌率为 260 的 MelCap，在单码本模型中达到最先进性能，这归功于我们的设计和提出的损失函数。此外，UniSRCodec 优于 UniCodec，源于其高频建模能力更强。

F. 下游理解任务

本节评估 UniS-RCodec 在下游理解任务中的性能。编解码器的编码器和量化器通常用作音频语言模型（ ALMs）的离散化策略。因此，神经音频编解码器理解音频，尤其是通用音频的能力，也值得关注。我们采用 xares 基准测试 [24],，将编解码器量化器输出的嵌入输入 xares 提供的多层感知机，以适应各种下游任务。不同任务的性能分数经过标准化处理，分数越高表示性能越好。我们选取了四项各异且跨领域的任务：UrbanSound-8k 用于城市环境声音分类，ESC-50 用于各种环境声音分类，GTZAN Genre 用于音乐流派分类，CREMA-D 用于情感识别。对于连续表示，我们选择了在大规模数据上预训练的 WavLM [23], 。对于离散表示，我们选择了 WavTokenizer，其令牌率与 UniSRCodec 相同。对于 UniSRCodec，我们采用逐帧策略展平嵌入，以匹配多层感知机层的输入。

表 III 展示了下游任务的性能。所提出的 UniSRCodec 在全部四项任务上均优于 WavTokenizer，并且在某些 CREMA-D 任务上取得了与连续表征模型 WavLM 相当的性能。该实验还验证了 UniSRCodec 在下游任务中进行通用音频理解的有效性。

VI. 结论

本文提出了一种统一的低比特率单码本 UniSRCodec。针对高频和低频建模，我们引入了子带重建技术。实验结果表明，与仅具有 40 令牌率的单码本编码相比，我们的 UniSRCodec 在统一音频建模方面取得了最先进性能。

参考文献

[1] Alexandre D´efossez、Jade Copet、Gabriel Synnaeve 和 Yossi Adi，《高保真神经音频压缩》，arXiv，2022年。[2] Rithesh Kumar、Prem Seetharaman、Alejandro Luebs、Ishaan Kumar 和 Kundan Kumar，《采用改进 RVQGAN 的高保真音频压缩》，载于《神经信息处理系统进展》，2023年。[3] Yunfei Chu、Jin Xu、Qian Yang、Haojie Wei、Xipin Wei、Zhifang Guo、Yichong Leng、Yuanjun Lv、Jinzheng He、Junyang Lin 等，《Qwen2-音频技术报告》，arXiv，2024年。[4] Boyong Wu、Chao Yan、Chen Hu、Cheng Yi、Chengli Feng、Fei Tian、Feiyu Shen、Gang Yu、Haoyang Zhang、Jingbei Li 等，《Step-audio 2 技术报告》，arXiv，2025年。[5] Detai Xin 、 Xu Tan 、 Shinnosuke Takamichi 等，《 BigCodec：突破低比特率神经语音编解码器的极限》，arXiv，2024年。[6] 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》，载于国际学习表征会议，2025年。[7] Yidi Jiang、Qian Chen、Shengpeng Ji、Yu Xi、Wen Wang、Chong Zhang、Xianghu Yue、ShiLiang Zhang 和 Haizhou Li，《UniCodec：采用单领域自适应码本的统一音频编解码器》，载于计算语言学协会年会，2025年。[8] Yang Ai 、 Xiao-Hang Jiang 、 Ye-Xin Lu 、 Hui-Peng Du 和 Zhen-Hua Ling，《APCodec：一种采用并行幅度与相位谱编码和解码的神经音频编解码器》，IEEE/ACM 音频、语音与语言处理汇刊，2024年。[9] Zhihao Du、Shiliang Zhang、Kai Hu 和 Siqi Zheng，《FunCodec：用于神经语音编解码器的基础性、可复现且可集成的开源工具包》，载于 IEEE 国际声学、语音与信号处理会议，2024年。[10] Jingyi Li、Zhiyuan Zhao、Yunfei Liu、Lijian Lin、Ye Zhu、Jiahao Wu、Qiuqiang Kong 和 Yu Li，《MelCap：用于高保真音频压缩的统一单码本神经编解码器》，arXiv，2025年。[11] Sang-gil Lee、Wei Ping、Boris Ginsburg、Bryan Catanzaro 和 Sungroh Yoon，《BigVGAN：采用大规模训练的通用神经声码器》，载于国际学习表征会议，2023年。[12] Zhuoyan Luo、Fengyuan Shi、Yixiao Ge、Yujiu Yang、Limin Wang 和 Ying Shan，《Open-MagViT2：旨在推动自回归视觉生成民主化的开源项目》，arXiv，2024年。[13] Yongxin Zhu、Bocheng Li、Yifei Xin、Zhihua Xia 和 Linli Xu，《采用单个线性层解决向量量化模型中的表示坍塌问题》，载于国际计算机视觉会议，2025年。 [14] Zhen Ye、Xinfa Zhu、Chi-Min Chan 等，《Llasa：扩展基于 Llama 的语音合成的训练时与推理时计算》，arXiv，2025年。[15] Hubert Siuzdak、Florian Gr¨otschla 和 Luca A Lanzend¨orfer，《SNAC：多尺度神经音频编解码器》，载于《Audio Imagination：神经信息处理系统大会 2024年工作坊——人工智能驱动的语音、音乐与声音生成》，2024年。[16] Julian D Parker、Anton Smirnov、Jordi Pons、CJ Carr、Zack Zukowski、Zach Evans 和 Xubo Liu，《扩展 Transformer 以实现低比特率高质量语音编码》，arXiv，2024年。[17] Junichi Yamagishi、Christophe Veaux 和 Kirsten MacDonald，《CSTR VCTK 语料库：用于 CSTR 声音克隆工具包的英语多说话人语料库（版本 0.92）》，2019年。[18] Heiga Zen、Viet Dang、Rob Clark、Yu Zhang、Ron J Weiss、Ye Jia、Zhifeng Chen 和 Yonghui Wu，《LibriTTS：从 LibriSpeech 派生的文本转语音语料库》，arXiv，2019年。[19] Rosana Ardila、Megan Branson、Kelly Davis、Michael Kohler、Josh Meyer、Michael Henretty、Reuben Morais、Lindsay Saunders、Francis Tyers 和 Gregor Weber，《 Common Voice：大规模多语言语音语料库》，载于《第十二届语言资源与评估会议论文集》，2020年，第 4218–4222 页。[20] Zafar Rafii、Antoine Liutkus、Fabian-Robert St¨oter、Stylianos Ioannis Mimilakis 和 Rachel Bittner，《用于音乐分离的 MUSDB18 语料库》，2017年。[21] Dmitry Bogdanov、Minz Won、Philip Tovstogan 等，《用于自动音乐标注的 MTG-Jamendo 数据集》，国际机器学习会议，2019年。[22] Jort F Gemmeke、Daniel PW Ellis、Dylan Freedman 等，《AudioSet：用于音频事件的本体与人工标注数据集》，载于 IEEE 国际声学、语音与信号处理会议，2017年。[23] Sanyuan Chen、Chengyi Wang 和 Zhengyang others Chen，《WavLM：面向全栈语音处理的大规模自监督预训练》，IEEE 信号处理专题期刊，2022年。[24] Junbo Zhang、Heinrich Dinkel、Yadong Niu、Chenyu Liu、Si Cheng、Anbei Zhao 和 Jian Luan，《X-ARES：用于评估音频编码器性能的综合框架》，载于国际语音通信协会年会，2025年。