arXiv:2304.02160v1 [cs.SD] 4 Apr 2023

Pa-HUBERT：通过原始听觉聚类和隐藏单元 BERT 实现自监督音乐源分离

Ke Chen[1], *Gordon Wichern*[2], *Francois G.Germain*[2], *Jonathan Le Roux*[2]

[1]三菱电机研究实验室（MERL），美国马萨诸塞州剑桥[2]加利福尼亚大学圣迭戈分校（UCSD），美国加利福尼亚州拉霍亚

摘要

尽管音乐源分离研究不断取得进展，但公开可用的干净源数据数量仍然很少，持续制约着性能提升。因此，自监督学习的最新进展为利用未标注音乐数据改进分离模型带来了一个尚未充分探索的机会。本文提出一种受 HuBERT 语音表征模型启发的音乐源分离自监督学习框架。我们首先探究原始HuBERT模型可能带来的影响：将其一个改编版本嵌入广为人知的 Demucs V2 时域分离模型架构中。随后，我们提出时频域自监督模型 Pa -HuBERT（读作 Pac-HuBERT，即原始听觉聚类 HuBERT），并将其与 Res-U-Net 解码器结合用于源分离。Pa -HuBERT 使用音乐的原始听觉特征作为无监督聚类标签，并利用自由音乐档案馆（FMA）数据集初始化自监督预训练过程。与原始 Demucs V2 和 Res-U-Net 模型相比，该框架在 MusDB18 测试集上取得了更好的信号失真比（SDR）表现。我们还进一步证明，该框架能够利用少量监督数据提升性能。总之，所提出的框架有效解决了音乐源分离中干净源数据有限这一难题。

索引词— 音乐源分离、原始听觉原理、自监督学习、BERT

1. 引言

音乐源分离旨在从音乐曲目中分离出一个或多个声源（例如人声、鼓、低音和其他乐器）。这项任务在多个领域都有广泛应用，包括人声与伴奏分离以及音乐混音。目前，该任务的最先进方法依赖于深度学习技术 [1]。然而，与语音研究不同 [2],，音乐源分离的数据仍然有限。该任务的基准数据集 MusDB18[3], 仅包含 100 首训练歌曲和 50 首测试歌曲。数据匮乏不仅是因为收集源数据具有挑战性，也因为音乐素材存在版权问题。

自监督学习（SSL）是一种快速发展的深度学习方法，它利用未标注数据来改进模型训练，从而提升泛化能力和任务表现 [4]。SSL 已在众多音频处理任务中取得了令人瞩目的成果，例如语音识别 [5], 、语音质量预测 [6,7], 或音频分类 [8–10],，甚至展现出跨任务泛化的能力 [11,12]。近期，也有多种用于音乐表征学习的自监督学习方法相继提出 [13–15]。然而，这些方法都侧重于学习用于内容标注任务的表示，而源分离则需要一种能够分离音频混合信号中各个独立成分的表示。

自监督学习已用于源分离的一个方向，是使用预训练的 WavLM [16] 模型进行语音增强 [17,18]。WavLM 基于 HuBERT [5],，后者是最有效的语音自监督学习模型之一。HuBERT 采用基于 Transformer 的 BERT 架构 [19]，经过训练后可在每个时间步生成一个离散词元，并利用掩码机制，使 HuBERT 学会根据输入数据及未掩码词元预测被掩码的词元。这些词元是聚类索引，最初通过对无标签语音数据的梅尔频率倒谱系数特征运行 K 均值聚类来确定；后续 HuBERT 迭代则使用对早期迭代中间网络层输出进行聚类所得的词元。

本文探讨了如何利用现成的无标签音乐混合信号（即不包含独立声源或分轨音频），采用 HuBERT 式方法提升音乐源分离模型的质量。然而，原始 HuBERT 的采样率较低（16 kHz），可能无法直接用于音乐分离；此外，用于生成 HuBERT 训练目标的梅尔频率倒谱系数特征，可能无法提供足够的时频（TF）分辨率，因而难以形成适用于分离音乐声源的有效表征。因此，我们探讨在 HuBERT 的初始步骤中，以专为音乐分离设计的时频特征替代梅尔频率倒谱系数特征，例如共同命运（2DFT） [20], 重复结构（REPET） [21], 谐波-打击乐源分离（HPSS） [22],以及旋律轮廓（Melodia） [23]。这些原始听觉特征在某种程度上模拟了人脑对听觉场景进行初步分割的方式 [24], ；此前，它们已被用于为无监督音乐源分离生成伪标签 [25],，但从未用于 HuBERT 式训练范式。

自监督学习和源分离模型的一项基本设计决策是：它们处理的是时域波形，还是频谱图等时频表示。HuBERT 和 wav2vec [26] 等时域音频表示模型会将原始音频信号编码为潜在表示；时域分离模型则包含类似的编码器，同时还包含解码器，以获取分离后的信号，方法可以是直接合成（例如 Wave-U-Net [27]），也可以是对特征进行掩码处理（例如 TasNet [28]）。这类模型广泛应用于语音分离，在实时场景中能够实现较高的分离质量和较低的延迟。然而，音乐和一般声音等非语音源通常以更高的采样率运行，并包含种类繁多、复杂的谐波模式，因此，时频域模型（或混合模型）历来在音频分类 [8, 29] 和音乐源分离 [30–34] 任务中表现更佳。

我们提出了一种新颖的框架，可在时域或时频域中利用自监督学习。我们的主要贡

Th这项工作是在 Ke Chen 于 ME 实习期间完成的 RL.

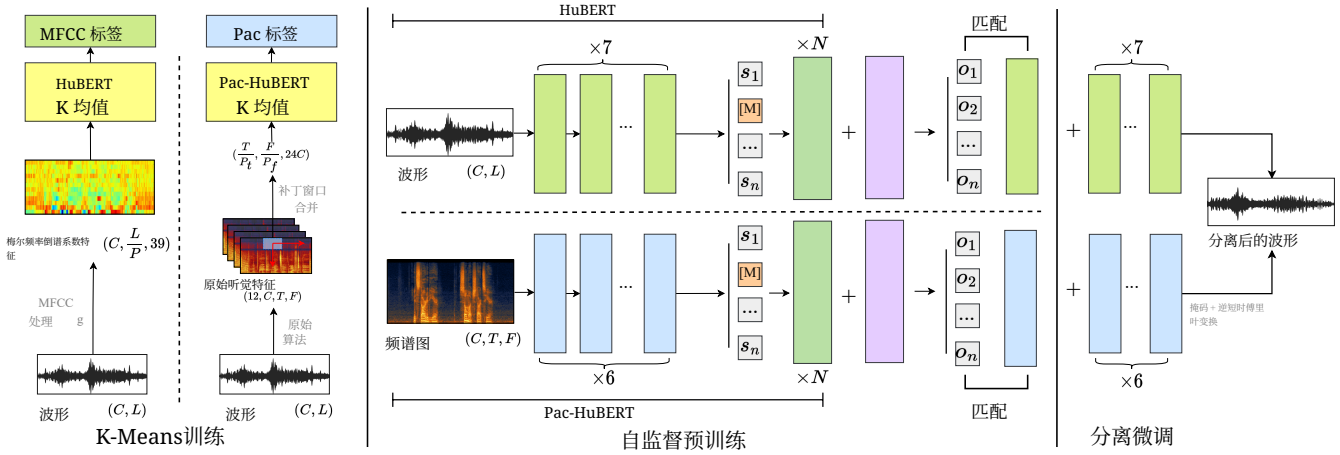

图 1：所提出的自监督音乐源分离框架的完整流程。左侧：用于标签创建的 K 均值模型；中间：自监督预训练（HuBERT 和我们提出的 Pac-HuBERT ）；右侧：针对分离任务进行微调。

献如下。首先，我们将 HuBERT 改造为时域音乐源分离模型，为后续时频域模型设计提供支持。其次，我们提出 Pa -HuBERT（读作 Pac-HuBERT，即原始听觉聚类 HuBERT），这是首个用于时频域音乐源分离的自监督框架。我们的框架利用音乐数据的原始听觉特征启动自监督预训练过程，并将由此得到的预训练层嵌入 Res-U-Net [31] 分离模型中；该架构在这项任务上已表现出很强的性能。最终，我们发现，采用我们在时域和时频域预训练层的模型，其分离性能均优于相应的原始模型。此外，我们还评估了训练数据更少时的性能，结果表明预训练模型表征具有稳健性：即使只有少量数据可用于监督微调，模型仍能保持可靠的性能。

2. 模型架构

我们提出的自监督音乐源分离框架（见图 1）由编码器（卷积块）、瓶颈模型（Transformer）和解码器（反卷积块）组成。学习流程包括 K-Means训练、自监督预训练和分离微调。

2.1. 编码器模型

如图 1 中部所示，编码器充当音频波形或频谱图的特征提取器。对于顶部的时域模型，编码器由 7 个下采样一维卷积层块（卷积神经网络）组成。输入音频波形记为 $x \in \mathbb{R}^{C \times L}$，其中 C 表示通道数， L 表示采样长度。该波形输入编码器后，被转换为瓶颈特征 $s \in \mathbb{R}^{C_b \times \frac{L}{P}}$，其中 C_b 表示通道数， P 表示下采样率。我们遵循 HuBERT 的设计构建一维卷积神经网络块，每个块包含一个卷积神经网络层和一个 GeLU 函数。请注意，这与 Demucs V 2 [35] 中的编码器块也类似（见图 2）。在分离实验中，为了使用 HuBERT 的预训练模型，同时遵循 Demucs V2 的训练方案，每个输入都是一个以 16 kHz 采样的 3 秒单声道音频信号（$C = 1$, $L = 48\,000$），并设置 $C_b = 1024$ 和 $P = 320$。

对于底部的时频域模型，编码器由 6 个下采样二维卷积神经网络块组成。输入音频信号首先转换为短时傅里叶变换频谱图，记为 $x \in \mathbb{R}^{C \times T \times F}$，其中 C 表示通道数， T 表示帧数， F 表示频率数。随后，编码器将 x 转换为瓶颈特征 $s \in \mathbb{R}^{C_b \times \frac{T}{P_t} \times \frac{F}{P_f}}$，其中 P_t 和 P_f 分别表示时间轴和频率轴上的下采样率。我们遵循 Res-U-Net [31] 分离模型的编码器块设计（详情见图 2 ）。由于我们构建自己的时频域自监督学习模型，因此可以使其与传统音乐源分离流程保持一致。因此，每个输入都是一个以 44.1 kHz 采样的 3 秒立体声音频（$C = 2$， $L = 132\,300$）。短时傅里叶变换窗长为 2048，步长为 441；向右零填充 20 帧后得到 $T = 320$ 帧，去除奈奎斯特频率后得到 $F = 1024$ 个频率 bin。模型参数设置为 $C_b = 384$、 $P_t = 32$和 $P_f = 64$。

2.2. 瓶颈模型

瓶颈模型由 N 个 Transformer 编码器块组成，每个编码器块包含一个具有 d [19] 个头、隐藏维度为 h的多头自注意力层，以及一个内部隐藏维度为 $4h$的前馈层。对于时域模型，我们使用 HuBERT-LARGE 模型（$N = 24$、 $d = 16$、 $h = 1024$）。对于时频域模型，为限制 GPU 内存消耗，且由于没有现成的预训练模型，我们设置 $N = 12$、 $d = 8$和 $h = 384$。编码器 s的输出会重塑为时域情况下的（$\frac{L}{P}, C_b$），以及时频域情况下的（$\frac{T}{P_t} \times \frac{F}{P_f}, C_b$），然后输入瓶颈模型。其输出是一个潜在特征 s'，形状与 s相同，并将其送入解码器以执行分离任务（第 2.3 节），或送入投影模块以进行自监督预训练（第 2.4 节）。我们将编码器与瓶颈模型的组合称为时域模型中的 HuBERT，称为时频域模型中的 Pa -HuBERT。

2.3. 解码器模型

对于时域和时频域模型，解码器通过将编码器中的所有 CNN 层替换为反卷积层（DCNN）来设计（见图 1 右侧）。上采样率等于下采样率，以使音频输入与分离输出（或掩码）具有相同的形状。此外，我们在每个编码器块与相应的解码器块之间设置了跳跃连接（图 1 中未显示）。

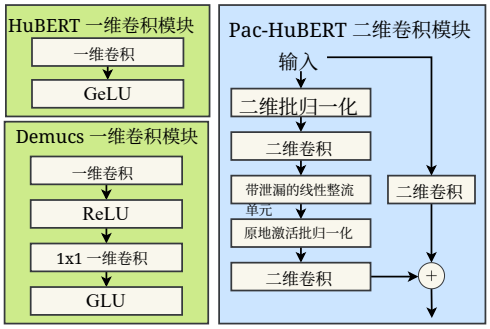

图 2：Demucs V [35], [5], [31] HuBERT 与 Pac-HuBERT 的卷积神经网络模块设计对比（与 Res-U-Net 相同）。

2.4. 训练与推理方案

K-Means训练： HuBERT通过对梅尔频率倒谱系数特征进行 K-Means聚类来生成自监督学习的初始标签（见图 1 左侧），具体如下：1）从 LibriSpeech 和 Libri-Light 音频样本中提取帧长为 20 ms 的 39 维梅尔频率倒谱系数特征 [36, 37],；2）对这些梅尔频率倒谱系数特征拟合一个含 500 个簇的 K-Means模型，并将分配给每个时间帧的簇索引用作标签。

对于 Pa -HuBERT，我们将每个时间帧的梅尔频率倒谱系数特征替换为基于频谱图块计算的原始听觉特征（见图 1 左侧）。形式上，给定一个混合频谱图$x \in \mathbb{R}^{C\times T\times F}$，这里使用的每种原始算法都会生成前景和背景估计值 e_f， $e_b \in \mathbb{R}^{C\times T\times F}$。我们使用的六种原始算法[1]是 HPSS [22,39],、REPET [21],、REPET-SIM [40],、FT2D-M、FT2D-R [20], 和 Melodia [23]。结合每种原始算法得到的前景和背景线索后，我们为每个时频单元获得一个 12 维特征。考虑到 Transformer 的内存需求，我们遵循音频频谱图 Transformer [8], 的方法，将特征组合成形状为（P_t， P_f）个时频单元的非重叠频谱图块（与时频域编码器中的下采样比例相匹配）。我们先将每个块划分为一个包含较低 $P_f/2$ 个时频单元的低频子块，以及一个包含较高 $P_f/2$ 个时频单元的高频子块，再对每个子块中的时频单元取平均，得到两个 12 维特征，并将它们拼接起来，从而为每个块获得一个 24 维特征。对于立体声（$C = 2$）音乐数据，我们为每个时频块获得一个 48 维特征向量，其维度与原始HuBERT模型中使用的 39 维梅尔频率倒谱系数相近。我们使用从 FMA-Large 数据集 [41]（106 574 首音乐曲目，约 890 小时）中提取的所有 48 维特征，拟合一个 960 聚类的 K-Means模型。

通过掩码单元预测进行自监督预训练： 在自监督学习过程中，我们使用 K 均值标签，通过掩码预测 [42] 训练 HuBERT 和 Pa -HuBERT。对于 HuBERT，我们利用训练好的 K-Means模型为每条音轨生成一串标签 $m \in \{1, 2, \ldots, 500\}^{\frac{L}{P}}$。对于 Pa -HuBERT，每个音频频谱图会生成（$\frac{T}{P_t} \times \frac{F}{P_f}$）个标签，共有 960 个类别。我们将瓶颈模型连接到一个投影模块（见图 1 中部），将输出特征 s' 映射为类别特征$o \in \mathbb{R}^{\frac{T}{P_t}\times\frac{F}{P_f}\times E}$，其中 E 为投影维度。损失函数定义如下：

$$L = \frac{1}{|M|}\sum_{t\in M}\frac{\exp\left(\tau\left(o_t \cdot e_c\right)/\left(||o_t||\,||e_c||\right)\right)}{\sum_{c'=1}^{D}\exp\left(\tau\left(o_t \cdot e_{c'}\right)/\left(||o_t||\,||e_{c'}||\right)\right)}, \quad (1)$$

[1]实现来自 https://github.com/nussl/nussl [38]

其中， o_t 是第 t 帧（对于 Pa - HuBERT，则为时频块）的类别特征标记， e_c 是一个学习得到的嵌入，表示该帧 t 被分配到的 K 均值聚类 c，而 $\tau = 10$ 用于缩放 logit。我们采用 HuBERT 使用的跨度掩码机制，随机选择 p% 的帧作为起始索引，并从这些索引开始对 l 帧进行掩码。对于 Pa -HuBERT， $p = 40$% 和 $l = 5$。损失仅在被掩码的帧上计算，这些帧在图 1 中以标签 $[M]$ 标示。**源分离微调：** 使用 LibriSpeech 和 Libri-Light 或 FMA- Large 对编码器和瓶颈模型进行预训练后，我们将瓶颈模型连接到解码器，并使用分离数据集针对分离任务对整个模型进行微调。损失函数是最终分离信号与输入混合信号之间的 L1 损失，与 Demucs V2 和 Res-U-Net 中使用的损失相同。

3. 实验

3.1. 超参数与训练细节

在预训练阶段，对于 HuBERT，我们直接使用了 torchaudio[2] 库中的预训练模型 HuBERT-LARGE。HuBERT-LARGE 在 Libri-Light 语音数据集上进行预训练。其训练过程的详细信息见 [5]。对于 Pa -HuBERT，我们在自由音乐档案馆数据集的“FMA-Large”子集上进行预训练，该子集包含 890 小时的音乐曲目。我们使用 AdamW 优化器，进行 32 000 步的预热 [43]，并在 150 000 步后采用衰减后的学习率调度器。基础学习率为 5×10^{-4}。我们在 8 块 NVIDIA A40 GPU 上以 96 的批量大小训练模型，直到验证集上的掩码准确率不再提升（总计 250 000 步）。

在微调阶段，我们使用 MusDB18 [3] 数据集训练分离模型。我们按照 [35] 将包含 100 首歌曲的训练集划分为 84 首训练歌曲和 16 首验证歌曲。确定验证集上的最佳模型后，我们在包含 50 首歌曲的测试集上对其进行评估。对于时域框架，我们将解码器与预训练模型 HuBERT-LARGE 一同训练。对于时频域框架，我们将解码器与预训练模型 Pa -HuBERT 一同训练。数据格式和音频处理设置见第 2.1 节。我们使用 Adam 优化器，学习率为 3×10^{-4}（时域）或 10^{-3}（时频域）。我们采用 3000 步预热和衰减调度器，每 15 000 步将学习率缩放至 $\alpha = 0.9$。我们在 8 块 NVIDIA A40 GPU 上训练模型，批量大小为 128（时域）或 96（时频域），共训练 200 000 步；此时所有模型在验证集表现方面均已收敛。

3.2. 分离结果

自监督学习的有效性：表 1 展示了在 MusDB18 上的源失真比（信号失真比，SDR）表现。我们遵循 MusDB18 基准，采用 mus eval[3] 实现的 SiSEC2018 [46] 版本 SDR 指标（BSS Eval v4 逐帧 SDR）。我们报告了 MusDB18 测试集中全部 50 首歌曲的 SDR 中位数。

我们将模型记为 *HuBERT-SEP*和 *Pa -HuBERT-SEP*，并用上标表示未进行预训练（N）或进行了 SSL 预训练（SSL）。与 HuBERT 类似，我们对第一个预训练模型中第 6 个 Transformer 块的潜在特征进行聚类，并将新的 Pa -HuBERT 预训练为第二轮预训练，记为 $2SSL$。我们对原始 Res-U-Net 的所有实验

[2]https://github.com/pytorch/audio [3] https://github.com/sigsep/sigsep-mus-eval

表 1：MusDB18 测试集上的源失真比（信号失真比）表现。灰色标注的结果由于使用了额外的训练数据或未使用验证集，无法直接比较。除非另有说明，所有结果均为 44.1 千赫。

模型	预训练	MusDB18 测试集上的信号失真比（dB）			
		人声	鼓	Bass	其他
Demucs V2 [35] - 16 kHz	✗	5.02	6.02	5.40	3.41
HuBERT-SEP^{N} - 16 千赫兹	✗	5.14	5.59	4.96	2.82
HuBERT-SEP^{SSL} - 16 kHz	✓	**5.58**	**6.63**	**6.03**	**3.65**
字节跳动 Res-U-Net [31]	✗	7.83	5.47	5.21	4.90
Pa-HuBERT-SEP^{N}	✗	8.07	5.78	5.21	5.29
Pa-HuBERT-SEP^{SSL}	✓	8.32	5.86	**6.01**	**5.38**
Pa-HuBERT-SEP^{2SSL}	✓	**8.52**	**6.20**	5.76	5.18
Open Unmix [44]	✗	6.32	5.73	5.23	4.02
Spleeter [34]	✗	6.86	6.71	5.51	4.55
D3Net [45]	✗	7.24	6.68	5.25	4.53
MDX-Net [30]	✗	9.00	7.33	7.86	5.95
Band-Split RNN [32]	✗	10.01	9.01	7.22	6.70
HT Demucs [33]	✗	7.93	7.94	8.48	5.72

表 2：不同训练数据量下的分离性能。下标表示自监督学习预训练带来的提升。

模型	Data 比例	MusDB18 测试集上的信号失真比（dB）			
		人声	鼓	Bass	其他
Pa-HuBERT-SEP^{N}	25%	5.72	4.42	4.00	3.38
	50%	6.76	4.50	4.14	4.13
	100%	8.07	5.78	5.21	5.29
Pa-HuBERT-SEP^{SSL}	25%	$6.75_{+1.03}$	$4.57_{+0.15}$	$4.11_{+0.11}$	$3.90_{+0.52}$
	50%	$7.32_{+0.56}$	$5.16_{+0.66}$	$5.07_{+0.93}$	$4.50_{+0.37}$
	100%	$8.32_{+0.25}$	$5.86_{+0.08}$	$6.01_{+0.80}$	$5.38_{+0.09}$

和 Demucs V2 均由我们重新复现，以确保比较公平。时域模型以 16 kHz 进行训练，以利用预训练的 HuBERT 模型，因此基线 Demucs V2 的结果与 [35] 不同。此外，我们还提供了若干最先进模型的已报告性能，尽管其中一些（以灰色标注）使用了额外的训练数据，或是为 MDX 挑战赛开发的 [1], ，该挑战赛直接将 MusDB18 测试集用作模型选择的验证集。

通过比较原始 Demucs V2、Res-U-Net、HuBERT-SEP^{N} 和 Pa-HuBERT ^{N} 模型（即未经预训练的模型），我们观察到 Pa-HuBERT ^{N} 的整体表现优于 Res-U-Net。可能的原因是，与 Res-U-Net 中的 CNN 瓶颈模型相比，Transformer 瓶颈层能够更有效地处理编码器特征。然而，我们发现 Demucs V2 的表现优于 HuBERT-SEP^{N}，这可能是因为 Demucs V2 的编码器和解码器设计更为复杂； [35]中也报告了这一点。

在分离任务上对预训练的 HuBERT 和 Pa-HuBERT 模型进行微调时，我们观察到，与未经预训练的模型相比，四种声源的信号失真比均有所提升。这表明 HuBERT 除了有助于语音任务外，也能有效支持音频分离任务；同时也说明，我们提出的 Pa-HuBERT 模型使用音乐数据中的原始听觉特征进行预训练后，能够提升时频域分离模型的性能。比较 Pa-HuBERT-SEP^{SSL}和 Pa-HuBERT-SEP^{2SSL}时，我们观察到， $2SSL$在人声和鼓的分离上表现更好，但在

表 3：预训练模型在预训练和分离任务中不同训练步数下的性能。

指标	训练步数			
	30 000	60 000	120 000	180 000
掩码准确率（验证集）	0.71	0.78	0.79	0.81
人声 SDR（25% 数据）	6.02	6.75	6.50	6.04

低音和其他声源的分离上表现有所下降。我们认为，迭代式预训练的设计仍可进一步改进，例如探索使用不同层的潜在特征进行预训练，并针对不同分离声源训练数量不同的 K 均值模型。这将是我们未来工作的重点。具体而言，HuBERT 的性能提升幅度大于 Pa-HuBERT，这可能归因于预训练数据集的规模差异（ HuBERT 使用 60000 小时数据，而 Pa-HuBERT 使用 890 小时数据）。仅使用语音进行预训练却能提升音乐分离性能，这也为未来工作提供了另一方向：探索如何利用这些数据，构建适用于全带宽时频域音乐分离的模型。

训练数据有限时的有效性：我们进一步从 MusDB18 的 84 首训练歌曲中提取了两个子集：包含 21 首歌曲的 25% 数据子集，以及包含 42 首歌曲的 50% 数据子集。随后，我们在这些子集上微调 Pa-HuBERT-SEP 模型，并在表 2 中报告其在测试集上的分离性能。结果表明，在这两个子集上，预训练模型的表现都显著优于未进行预训练的模型。值得注意的是，仅使用 25% 的监督数据时，预训练模型就能使人声和其他声源类别的分离性能提升 0.5–1.0 dB。这证明了使用原始听觉特征进行预训练的有效性，因为前景和背景特征与人声及其他声源类别高度相关。然而，对于低音和鼓，仅使用 25% 的监督数据时，预训练几乎没有带来益处，这表明预训练对这些类别的作用较小，因此应探索不同的听觉原语。

预训练模型选择：我们还研究了 Pa-HuBERT 预训练步数变化的影响，并在表 3 中评估了使用 MusDB18 训练数据的 25% 进行微调时的人声分离性能。我们发现，模型的分离性能与其预训练性能（即掩码令牌预测准确率）并不一致。 [9] 中关于音频分类的研究也报告了类似趋势。预训练期间的高准确率可能表明模型过拟合，从而降低所学表征的质量，并对分离任务的性能产生负面影响。

4. 结论

本文研究了自监督学习方法在音乐源分离中的有效性。首先，我们将 HuBERT 改造为时域源分离模型，并取得了显著提升。随后，我们提出了 Pa-HuBERT，这是一种时频域自监督源分离模型，可在预训练期间利用未标注音乐数据中的原始听觉特征。

Pa-HuBERT 显著提升了时频域音乐源分离模型的性能，并且在不同监督数据占比下均稳定提升。我们认为，Pa-HuBERT 是一种有效利用未标注音乐数据进行音乐源分离的方案。未来，我们计划将 Pa-HuBERT 应用于 Res-U-Net 以外的其他分离模型。此外，我们还计划探索 Pa-HuBERT 在少样本设置下分离更多乐器时的泛化能力

5. 参考文献

[1] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.-R. St¨oter, A. D´efossez, M. Kim, W. Choi, C.-Y. Yu, and K.-W. Cheuk, 《2021年音乐解混挑战赛》，《信号处理前沿》，第1卷，2022年。[2] R. Haeb-Umbach, S. Watanabe, T. Nakatani, M. Bacchiani, B. Hoffmeister, M. L. Seltzer, H. Zen, and M. Souden, 《数字家庭助理的语音处理：结合信号处理与深度学习技术》，《IEEE信号处理杂志》，第36卷，第6期，第111–124页，2019年。[3] Z. Rafii, A. Liutkus, F.-R. St¨oter, S. I. Mimilakis, and R. Bittner, 《用于音乐分离的MUSDB18语料库》，2017年。[4] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, 《自监督表征学习：简介、进展与挑战》，《IEEE信号处理杂志》，第39卷，第3期，第42–62页，2022年。[5] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and 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。[6] W.-C. Tseng, C.-Y. Huang, W.-T. Kao, Y. Y. Lin, and H.-Y. Lee, 《利用自监督表征进行平均意见分预测》，载于 Interspeech 论文集，2021年，第2781–2785页。[7] H. Becerra, A. Ragano, and A. Hines, 《探索微调数据对用于盲语音质量预测的 wav2vec 2.0 模型的影响》，载于 Interspeech 论文集，2022年，第4088–4092页。[8] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, 《SSAST：自监督音频频谱图Transformer》，载于 AAAI 论文集，第36卷，第10期，2022年，第10 699–10 709页。[9] P.-Y. Huang, H. Xu, J. B. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, 《能够聆听的掩码自编码器》，载于 NeurIPS 论文集，2022年。[10] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, 《BEATs：使用声学分词器进行音频预训练》，arXiv预印本 arXiv:2212.09058，2022年。[11] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Stein-metz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally 等, 《HEAR：音频表征的全面评估》，载于 2021年竞赛演示专题（NeurIPS），2022年，第125–145页。[12] H.-H. Wu, P. Seetharaman, K. Kumar, and J. P. Bello, 《Wav2clip：从 CLIP 学习鲁棒的音频表征》，载于 ICASSP 论文集，2022年，第4563–4567页。[13] H. Zhu, Y. Niu, D. Fu, and H. Wang, 《MusicBERT：音乐表征的自监督学习》，载于 ACM多媒体论文集，2021年，第3955–3963页。[14] H.-H. Wu, C.-C. Kao, Q. Tang, M. Sun, B. McFee, J. P. Bello, and C. Wang, 《面向音乐分类的多任务自监督预训练》，载于 ICASSP 论文集，2021年，第556–560页。[15] A. N. Carr, Q. Berthet, M. Blondel, O. Teboul, and N. Zeghidour, 《通过可微排序从排列中自监督学习音频表征》，《IEEE信号处理快报》，第28卷，第708–712页，2021年。[16] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao 等, 《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》，第16卷，第6期，第1505–1518页，2022年。[17] H. Song, S. Chen, Z. Chen, Y. Wu, T. Yoshioka, M. Tang, J. W. Shin, and S. Liu, 《探索 WavLM 在语音增强中的应用》，载于 SLT 论文集，2022年，第451–457页。[18] X.-Y. Zhao, Q.-S. Zhu, and J. Zhang, 《使用自监督预训练模型和向量量化进行语音增强》，载于 APSIPA ASC 论文集，2022年，第330–334页。[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, 《注意力机制就是你所需要的一切》，载于 NeurIPS 论文集，2017年。[20] P. Seetharaman, F. Pishdadian, and B. Pardo, 《使用二维傅里叶变换进行音乐/语音分离》，载于 WASPAA 论文集，2017年，第36–40页。[21] Z. Rafii and B. Pardo, 《重复模式提取技术（REPET）：一种简单的音乐/语音分离方法》，《IEEE语音与音频处理汇刊》，第21卷，第1期，第71–82页，2013年。[22] J. Driedger、M. M¨uller 和 S. Disch, 《扩展音频信号的谐波-打击乐分离》，载于 ISMIR 论文集，2014年，第611–616页。[23] J. Salamon 和 E. G´omez, 《利用音高轮廓特征从复音音乐信号中提取旋律》，《IEEE语音与音频处理汇刊》，第20卷第6期，第1759–1770页，2012年。[24] J. H. McDermott、D. P. W. Ellis 和 E. P. Simoncelli, 《从自然声音统计中经验推导声学分组线索》，载于耳鼻喉科研究协会年会论文集，2011年。[25] P. Seetharaman、G. Wichern、J. Le Roux 和 B. Pardo, 《基于原始听觉分组原理自举无监督深度音乐分离》，载于自监督音频与语音研讨会论文集，2020年。[26] S. Schneider、A. Baevski、R. Collobert 和 M. Auli, 《wav2vec：用于语音识别的无监督预训练》，载于 Interspeech 论文集，2019年，第3465–3469页。[27] D. Stoller、S. Ewert 和 S. Dixon, 《Wave-U-Net：用于端到端音频源分离的多尺度神经网络》，载于 ISMIR 论文集，2018年，第334–340页。[28] Y. Luo 和 N. Mesgarani, 《TaSNet：用于实时单通道语音分离的时域音频分离网络》，载于 ICASSP 论文集，2018年，第696–700页。[29] K. Chen、X. Du、B. Zhu、Z. Ma、T. Berg-Kirkpatrick 和 S. Dubnov, 《HTS-AT：用于声音分类与检测的分层令牌语义音频变换器》，载于 ICASSP 论文集，2022年，第646–650页。[30] M. Kim、W. Choi、J. Chung、D. Lee 和 S. Jung, 《KUIELab-MDX-Net：用于音乐解混的双流神经网络》，载于 MDX研讨会论文集，2021年，第1–7页。[31] Q. Kong、Y. Cao、H. Liu、K. Choi 和 Y. Wang, 《利用深度 ResUNet 解耦音乐源分离中的幅度估计与相位估计》，载于 ISMIR 论文集，2021年，第342–349页。[32] Y. Luo 和 J. Yu, 《基于 Band-Split RNN 的音乐源分离》，CoRR，第abs/2209.15174卷，2022年。[33] S. Rouard、F. Massa 和 A. D´efossez, 《用于音乐源分离的混合变换器》，arXiv预印本 arXiv:2211.08553，2022年。[34] R. Hennequin、A. Khlif、F. Voituret 和 M. Moussallam, 《Spleeter：快速高效的音乐源分离工具及其预训练模型》，《开放源代码软件期刊》，第5卷第50期，文章编号2154，2020年。[35] A. D´efossez、N. Usunier、L. Bottou 和 F. R. Bach, 《波形域中的音乐源分离》，CoRR，第abs/1911.13254卷，2019年。[36] V. Panayotov、G. Chen、D. Povey 和 S. Khudanpur, 《LibriSpeech：基于公有领域有声读物的自动语音识别语料库》，载于 ICASSP 论文集，2015年，第5206–5210页。[37] J. Kahn、M. Rivi`ere、W. Zheng、E. Kharitonov、Q. Xu、P. Mazar´e、J. Karadayi、V. Liptchinsky、R. Collobert、C. Fuegen 等, 《Libri-Light：面向有限监督或无监督自动语音识别的基准》，载于 ICASSP 论文集，2020年，第7669–7673页。[38] E. Manilow、P. Seetharaman 和 B. Pardo, 《西北大学源分离库》，载于 ISMIR 论文集，2018年。[39] N. Ono、K. Miyamoto、J. Le Roux、H. Kameoka 和 S. Sagayama, 《通过频谱图上的互补扩散将单声道音频信号分离为谐波/打击乐成分》，载于 EUSIPCO 论文集，2008年。[40] Z. Rafii 和 B. Pardo, 《利用相似度矩阵进行音乐/语音分离》，载于 ISMIR 论文集，2012年，第583–588页。[41] M. Defferrard、K. Benzi、P. Vandergheynst 和 X. Bresson, 《FMA：音乐分析数据集》，载于 ISMIR 论文集，2017年，第316–323页。[42] J. Devlin、M. Chang、K. Lee 和 K. Toutanova, 《BERT：深度双向Transformer语言理解预训练》，载于 NAACL-HLT 2019 论文集，第4171–4186页。[43] P. Goyal、P. Doll´ar、R. Girshick、P. Noordhuis、L. Wesolowski、A. Kyrola、A. Tulloch、Y. Jia 和 K. He, 《准确的大批量随机梯度下降：1小时内训练 ImageNet》，arXiv预印本 arXiv:1706.02677，2017年。[44] F. St¨oter、S. Uhlich、A. Liutkus 和 Y. Mitsufuji, 《Open-Unmix：音乐源分离的参考实现》，《开放源代码软件期刊》，第4卷第41期，文章编号1667，2019年。[45] N. Takahashi 和 Y. Mitsufuji, 《D3Net：用于音乐源分离的密集连接多膨胀率 DenseNet》，arXiv预印本 arXiv:2211.08553，2021年。[46] F.-R. St¨oter、A. Liutkus 和 N. Ito, 《2018年信号分离评测活动》，载于 LVA/ICA 论文集，2018 年，第 293–305 页。