机器感知实验 **(Due: 2020/6/8)**

# 声源分离算法探究与应用

导师：曲天书、陈婧 姓名：蔡辉宇，学号：1600011742

## 1 引言

声源分离（Music Source Separation）是指从含有多种声源（如人声、钢琴、吉他、贝斯、架子鼓等）的混制音频（mixed audio）中提取或还原出单一声源（又称单轨）或部分所需声源的技术，如图1。这一技术在音乐分析和音乐制作上有着广泛的应用，如：自动乐谱识别、基于人声–伴奏分离的自动伴奏生成和歌词识别、再混音（remixing）和部分声部采样（sampling）等。作为我校阿卡贝拉清唱社的一员，笔者常常需要将流行歌曲改编为纯人声的阿卡贝拉编曲，而这首先需要笔者对歌曲中各个乐器演奏的内容有一个清晰、准确的把握。因此，笔者对这一技术及其应用有着浓厚的兴趣。

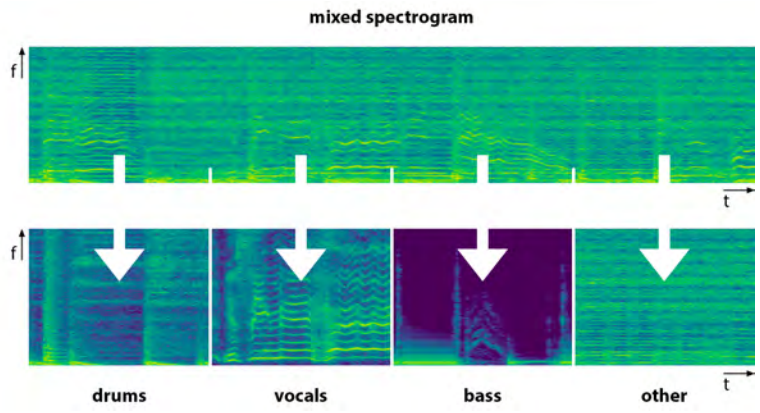



图 1: 声源分离示意图

尽管应用前景广阔，但长期以来声源分离一直是一个比较困难的问题。首先，人声和乐器在发声时都有泛音列（harmonic series），即在以人耳感知的某个频率（即基频）发声时，基频整数倍的频率也会有能量的存在。这使得从频谱图中完全分离出某种乐器格外困难。而如果只提取出某个声源的低频部分，而高频部分没有完全提取干净的话，一方面被提取的声源会显得闷而混浊（因为缺少高频成分），另一方面去除了该声源的音频又会有刺耳的高频噪音。打击乐由于其频谱范围极宽（从大鼓的数十赫兹到镲的数千赫兹），也同样难以从混制音频中完整地提取出来。其次，在音乐制作过程中，会对声音进行各式各样的处理，如增益（Gain）、均衡器（Equalizer）、声像（PAN）、混响（Reverb）、延迟（Delay）、扭曲（Distortion）、压缩（Compression）等，其中许多处理是非线性的。同时，在乐曲的不同部分，为了呈现出不同的听觉效果，制作者往往会采取不同的处理。这给声源分离算法的鲁棒性提出了很大挑战。

音乐的结构性给声源分离提供了一些线索。不同乐器各个泛音成分的比例不同（或者说，它们傅里叶变换后各频率成分的系数不同），这给我们分离它们提供了一个可能的抓手。音乐中常由架子鼓和贝斯提供节奏感，它们常常以小节为单位重复（贝斯可能有音高的变化，但节奏常常是大体相似的）。流行乐在结构上常分为前奏、主歌、副歌、桥段、尾奏等，而不同段落之间常有联系，如前奏和主歌或副歌的伴奏部分是相似的。这些结构特点也给我们设计声源分离算法提供了宝贵的提示。

图 2: 一种常见的混音时声源位置示意图。主唱、贝斯和鼓被放置在中间。[2]

基于这些线索，科学家设计了各种声源分离的算法，其中一些算法已经在工业级产品中有所应用。例如，常见的音频处理软件都有基于中置声道提取（参见2.1）的人声消除功能。在 iZotope RX 7[1]中更是有基于深度学习的音乐再平衡（Music Rebalance）功能，能够将人声、鼓、贝斯和其他乐器分离开来。在本项目中，笔者将探索近年来有影响力的声源分离算法的原理，并利用前沿算法进行声源分离实践应用。

## 2 实验原理

本节笔者将综述数种声源分离算法。其中中置声道分离（2.1节）为基于声像的朴素算法；旋律–打击乐分离技术（2.2节）、REPET-SIM 技术（2.3）为基于统计的算法；MMDenseLSTM（2.4节）、Demucs（2.5节）、Meta-TasNet（2.6）为基于深度学习的算法。后三种算法均在Rafii u. a. (2017) 发布的 MusDB18 数据集上进行训练，因而可以进行直接的性能比较（见节）。而其他算法和后三种方法分离出的声源种类有所不同，无法采用相同的机制进行评价，故笔者采取了人工比较的方法（见节）。

### 2.1 中置声道提取：经典的声源分离算法

中置声道提取（Center Channel Extraction, CCE）是简单而有效的一种算法，在目前主流的音频处理软件如 Adobe Audition[3]、FL Studio[4]中都有实现。算法的输入须为双声道音频。这一算法利用了流行音乐混音技术中常常把主唱、贝斯和鼓放在中间的特点（见图2）。中置声道在双声道音频中两个声道的分量是完全相同的，因而算法通过将左声道和右声道的音频信号相减（在时域和频域作差均可），就得到消音后的单声道音频。

如此消音必然将其他中置的声源，如鼓、贝斯等也一并消除。而人为加入的非中置人声，如混响、和声伴唱和非中置主唱等，是这一算法无能为力的。为了减缓算法对鼓和贝斯的削减，可以采用低通滤波的算法，如图3。将信号的低通部分（如 200 Hz 以下的部分，对应大鼓和贝斯的频段）泄露到相减后的音频中，可以一定程度上保留这些乐器。

### 2.2 旋律–打击乐分离技术：中值滤波的妙用

为更精准地分离声源，我们需要比声像更准确的声源特征。观察流行乐的声谱图（图4）我们发现，整个图呈现明显的横纵相间的图案。横线是在一个较窄的音区内演奏的钢琴和人声的泛音列，而竖线是频谱较宽但时域上只是偶然出现（在节奏点才会出现）的打击乐器。基于此，Ono u. a. (2008) 提出了基于互补

[1] https://www.izotope.com/en/products/rx.html
[2] 图标由 icongeek26, freepik 绘制，从 www.flaticon.com 取得。
[3] https://www.adobe.com/cn/products/audition.html
[4] https://www.image-line.com/flstudio/

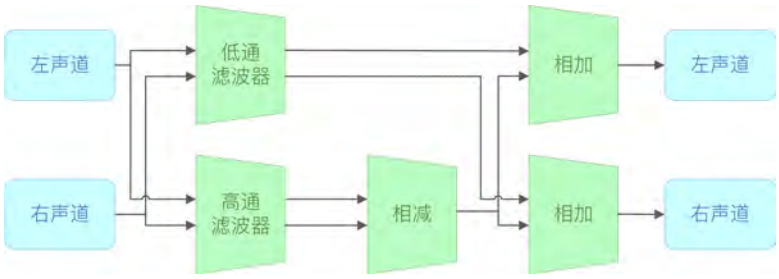



图 3: 一种中置声道提取算法示意图

扩散（Complementary Diffusion）的分离算法，其损失函数为

$$J(\boldsymbol{H},\boldsymbol{P}) = \frac{1}{2\sigma_H^2}\sum_{h,i}(H_{h,i-1}-H_{h,i})^2 + \frac{1}{2\sigma_P^2}\sum_{h,i}(P_{h-1,i}-P_{h,i})^2 \tag{2.1}$$

$$\text{s.t. } H_{h,i}+P_{h,i}=W_{h,i} \tag{2.2}$$

$$H_{h,i}\geq 0, P_{h,i}\geq 0 \tag{2.3}$$

其中 $h$ 表示频率桶， $i$ 表示时间桶。这一函数通过幅度谱（magnitude spectogram） $\boldsymbol{W}$ 上两个方向的梯度将其分解为两个矩阵 $\boldsymbol{H}$ 和 $\boldsymbol{P}$ 之和，有一定的合理性。但是基于此进行优化会导致 $H_{h,i-1}-H_{h,i}$ 和 $P_{h-1,i}-P_{h,i}$ 趋向高斯分布，与事实不符。故作者引入压缩因子 $\gamma\in(0,1]$ 作为幂指数将原声谱图 $\boldsymbol{W}$ “压缩”。但这同样具有浓厚的经验主义色彩，缺乏理论依据。

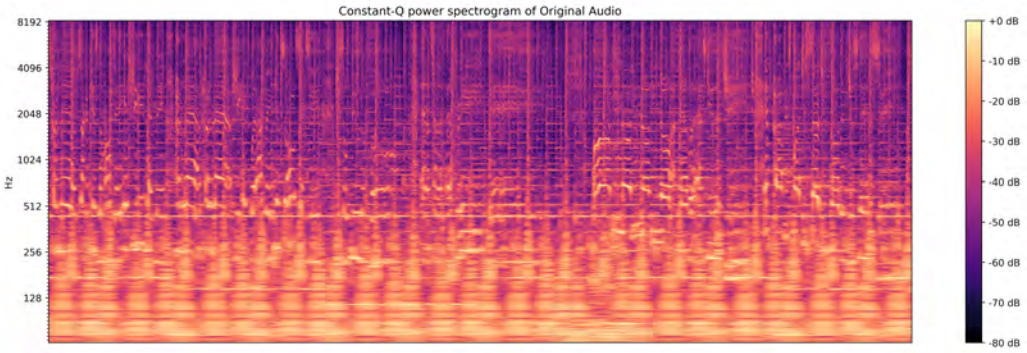



图 4: Just the way you are 选段的声谱图。
横竖相间的图案为旋律–打击乐分离算法提供了灵感。

Fitzgerald (2010) 等设计了一个巧妙的旋律–打击乐分离（Harmonic-Percussive Source Separation, HPSS）算法。它们将旋律视作声谱图频率轴方向的异常值，将打击乐视作声谱图时间轴方向的异常值。类比图像增强中处理椒盐噪声的方法，他们用中值滤波器分别去除这两个方向的异常值，得到旋律和打击乐对应的幅度矩阵 $\boldsymbol{H}$ 和 $\boldsymbol{P}$。初步实验结果（图5）表明，中值滤波有效地去除了旋律乐器的共振峰，得到了清晰干净的打击乐。但算法分离出的旋律却不能令人满意，因为对时域的中值滤波难免会让旋律音符的时值的顺序改变。

中值滤波得到的 $\boldsymbol{H}$ 和 $\boldsymbol{P}$ 矩阵之和并不是原始矩阵。为了尽量保留原始音频中的信息，作者采用软遮罩（soft-masking）策略：由 $\boldsymbol{H}$ 和 $\boldsymbol{P}$ 矩阵先得到两个和 $\boldsymbol{W}$ 大小相同的遮罩 $\boldsymbol{M}_H$ 和 $\boldsymbol{M}_P$（它们的和为全 1 的矩阵），再用遮罩逐元素乘以 $\boldsymbol{W}$ 得到最终的旋律和打击乐幅度矩阵。Driedger u. a. (2014) 对此作了进一步改进，引入乘性因子 $\rho_H$ 和 $\rho_P$（作者称为边距 margin，默认为 1）控制软遮罩中目标声源幅度和被分离声源幅度的比例。如此，当被分离声源的信号被过多引入目标声源时，我们可以调高目标声源的乘性因子以加强分离效果。

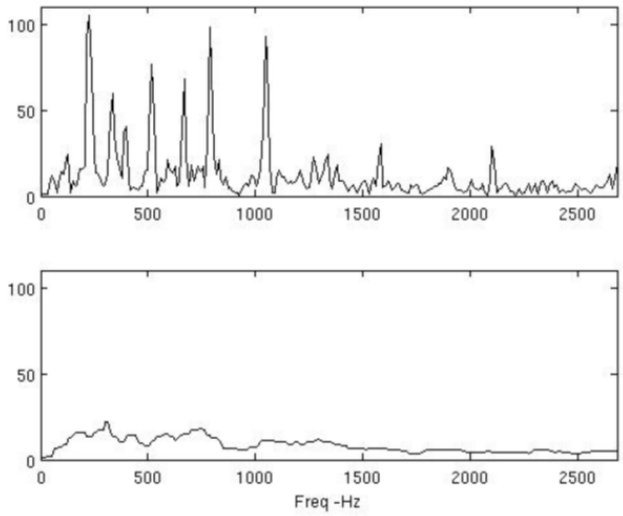



图 5: 钢琴加小鼓的混合音频在中值滤波前（上）和滤波后（下）的幅频图

$$\boldsymbol{H} = \text{Median}\left(\boldsymbol{W}, (1, l_H)\right), \quad \boldsymbol{P} = \text{Median}\left(\boldsymbol{W}, (l_P, 1)\right) \tag{2.4}$$

$$\boldsymbol{M}_H = \frac{\boldsymbol{H}^p}{\boldsymbol{H}^p + (\rho_H \boldsymbol{P})^p}, \quad \boldsymbol{M}_P = \frac{\boldsymbol{H}^p}{(\rho_P \boldsymbol{H})^p + \boldsymbol{P}^p} \tag{2.5}$$

$$\boldsymbol{H} = \boldsymbol{M}_H \odot \boldsymbol{W}, \quad \boldsymbol{P} = \boldsymbol{M}_P \odot \boldsymbol{W} \tag{2.6}$$

其中 $l_H$ 和 $l_P$ 为中值滤波器的长度； $p$ 为按元素幂的指数，一般设为 2； $\odot$ 表示按元素相乘。

## 2.3 REPET-SIM：利用音乐的重复结构

Rafii und Pardo (2012) 观察到，流行音乐中伴奏常常具有重复性，如前奏和主歌/副歌部分的伴奏往往具有相似性，不同小节的伴奏也具有一定的相似性；而人声则在一个更大的尺度上重复，少有局域的重复性，例如一首歌曲中的三遍副歌是一样的，但它们至少相隔八小节以上。基于此，他们提出了利用音乐重复结构的 REPET-SIM 算法。在将音频信号转换到频域以后，作者将 $f \times t$ 的幅度谱 $\boldsymbol{W}$ 分解为时间帧 $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_t$，并计算它们两两之间的余弦相似度，得到相似度矩阵

$$S_{i,j} = \frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{\|\boldsymbol{w}_i\| \|\boldsymbol{w}_j\|}.$$

随后对相似度矩阵逐行排序，得到和第 $i$ 帧最相似的帧的集合 $\mathcal{T}_i$。在这一步可以做一些筛选操作，如设定余弦相似度的最小值（低于此值不纳入 $\mathcal{T}_i$）、 $|\mathcal{T}_i|$ 的最大值（降低计算量）、 $\mathcal{T}_i$ 中的帧和第 $i$ 帧的最小时间间隔（避免邻域帧的干扰）等。

得到 $\mathcal{T}_i$ 后，求其中的中值，作为第 $i$ 帧中背景音的估计。因为背景音量不得超过混制后的音量，所以对背景音估计中比第 $i$ 帧幅值更小的部分，以第 $i$ 帧中的幅值填充。

$$\boldsymbol{b}_i = \min\left\{\text{Median}\left(\mathcal{T}_i, l\right), \boldsymbol{w}_i\right\}$$

第 $i$ 帧的前景音 $\boldsymbol{f}_i$ 则定义为第 $i$ 帧中除背景音外的剩余部分。随后用此背景音和前景音用和2.2节中相同的方法制作软遮罩，与原幅度谱矩阵按元素相乘，再加入原相位谱信息做 ISTFT，即得分离后的音频信

号。

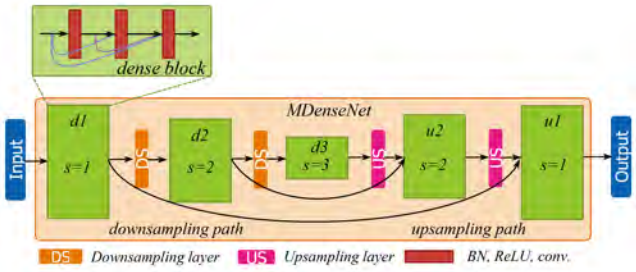
FitzGerald (2012) 中利用高斯径向基函数作为软遮罩，也取得了不错的效果。

$$M_B = \exp\left(-\frac{(\log B - \log F)^2}{2\lambda^2}\right)$$

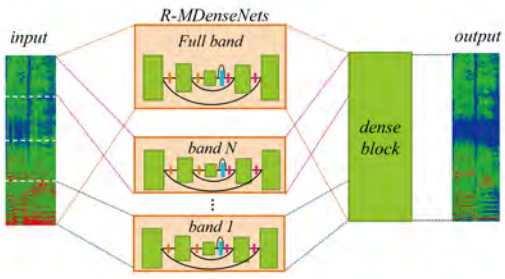
其中 $M_B$ 为背景音遮罩， $B$ 和 $F$ 分别为每一帧的背景音和前景音组成的幅度谱矩阵。

## 2.4 MMDenseLSTM：端到端学习频域变换

Huang u. a. (2016) 提出的 DenseNet 作为计算机视觉中优秀的网络架构，得到了广泛应用。Takahashi und Mitsufuji (2017) 借鉴其中跳跃连接（skip connection），缩短梯度传导路径的思想，提出了 MMDenseNet，即多尺度多频段稠密网络（Multi-scale multi-band DenseNet，图6a）。该网络用稠密卷积模块（由多个跳跃连接的卷积模块组成，每个卷积模块由批量归一化层、ReLU 激活层、卷积层构成）、下采样和上采样层构成。在编码阶段，幅度谱矩阵经 $s-1$ 个稠密卷积模块和下采样层后，再经过 $s = s_{\max}$ 的卷积模块得到隐层表示。在解码阶段，隐层表示经 $s-1$ 个上采样层和稠密层后得到某一声源的幅度谱。在 $s$ 值相等的下采样层前和上采样层后作者引入了跳跃连接来弥补下采样对高频信号的损失。针对对幅度谱各频段使用相同的卷积核带来的性能问题，作者训练了多个网络，对幅度谱不同频段使用不同的网络参数，再在最后用一个稠密模块对不同频段的信息进行综合。这种端到端的学习方法展现出强大的性能，在 SiSEC 2016（Liutkus u. a. (2017)）中拔得头筹。




(a) $s_{\max} = 3$ 的 MDenseNet 网络结构。模块上标有尺度（scale）值。




(b) $s_{\max} = 2$ 的 MMDenseLSTM 网络结构。

如何进一步提高网络性能呢？Takahashi u. a. (2018) 希望引入 LSTM 增强提取全局特征的能力。他们经过实验，选择将 LSTM 模块插入到 $s > 1$ 的所有稠密模块后面（ $s = 1$ 时时间步过多，LSTM 网络性能不佳），得到的 MMDenseLSTM 网络在 MusDB18 竞赛（Rafii u. a. (2017)）中再度夺魁。

## 2.5 Demucs：基于卷积和循环神经网络的时域模型

近年来，基于时域的语音生成模型成为研究热点。频域模型由于信息易于提取，用于信号处理起点较高，但由于完全忽略了相位信息的变换，因而“天花板”可能较低。Defossez u. a. (2019) 基于 Wave-U-Net（Stoller u. a. (2018)）和 SING（Défossez u. a. (2018)）的模型提出了 Demucs 模型，是目前时域声源分离模型中的佼佼者。

如图7所示，Demucs 模型由卷积编码器–LSTM–卷积解码器架构组成。其中卷积编码器有六层。每个卷积核大小为 8，步长为 4，输入频道数为 $C_{i-1}$，输出频道数为 $C_i$。卷积后经 ReLU 激活、 $1 \times 1$ 卷积和 GLU 激活（Dauphin u. a. (2017)）。此处 GLU 的两个输入由 $1 \times 1$ 卷积后得到的矩阵拆分而成。 $i > 1$ 时，$C_i = 2C_{i-1}$。较高的步长给了模型指数提高频道数的能力，最后一层编码器的频道数为 1536。LSTM 模块由两层双向 LSTM 构成，每层 LSTM 隐藏层大小为 $C_L$，输出大小为 $2C_L$，经过 $1 \times 1$ 卷积后为 $C_L$。解

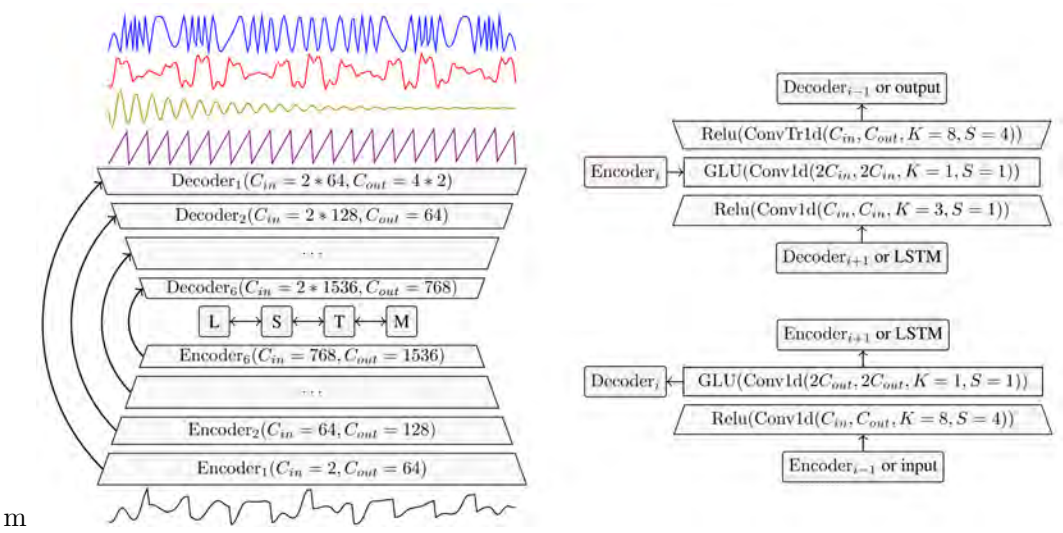
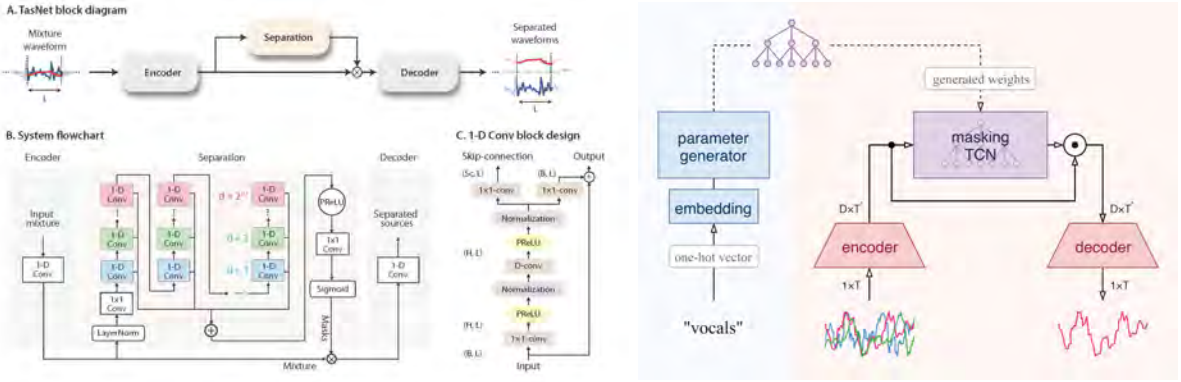



图 7: Demucs 模型结构图（左）。编码器细节图（右下）。解码器细节图（右上）。

码器也有六层，信号首先经过卷积核大小为 3、步长为 1、输入输出频道数均为 $C_i$ 的 $1\times1$ 卷积和 ReLU 激活。激活后的信号与编码器对应层数的输出拼接（这就是 U 形连接）后，经过 $1\times1$ 卷积和 GLU 激活，再经卷积核大小为 8、步长为 4 的 1 转置卷积和 ReLU 激活得到 $C_{i-1}$ 个频道的输出。最后一层解码器的输出频道数为 $4C_0=8$（假定输入双声道、待分离声源有四个）。模型的损失函数为各已知的声源音频与模型生成音频的 $\updownarrow1$ 距离之和。

Demucs 使用了权重伸缩（weight scaling）来减缓权重初始化算法带来的不同频道数网络层之间梯度差距过大的问题。值得一提的是，作者还利用了声源分离领域非常丰富的无标签数据（即只有混音后的音频而没有混音前的单轨的数据）。他们先训练了一个能识别声源是否存在的模型，然后用这个模型在无标签数据中寻找某种声源 $i$ 不存在的片段 $\boldsymbol{m}$，找到以后从 MusDB 数据集中随机抽取一段这一声源的单轨音频 $\boldsymbol{s}_i$，与 $\boldsymbol{m}$ 相加得到伪造的“混音”数据。用模型分离此音频后，对 $\hat{\boldsymbol{s}}_i$ 和 $\boldsymbol{s}_i$ 之间的 $\updownarrow1$ 损失加较大的权重、对 $\sum_{j\neq i}\hat{\boldsymbol{s}}_j$ 和 $\boldsymbol{m}$ 之间的 $\updownarrow1$ 损失加较小的权重作为损失函数进行训练。实验表明这样的无标签数据能显著提高模型性能。

## 2.6 Meta-TasNet：元学习能带来什么？




(a) Conv-TasNet 模型结构图　　(b) Meta-TasNet 模型结构图

图 8: Conv-TasNet 和 Meta-TasNet 模型结构图

Luo und Mesgarani (2018) 借鉴频域模型 STFT–频域变换–ISTFT 的范式开发出了编码器–分离器–解

码器的 TasNet 模型，后经Luo und Mesgarani (2019) 改进为全卷积版本的 Conv-TasNet（图8a）。模型首先将输入切成有重叠的 $L$ 长片段，随后将每一段音频 $\boldsymbol{x}$ 用式 $\boldsymbol{w} = \boldsymbol{x}U$ 转换为“变换域”向量表示 $\boldsymbol{w}$，其中 $U$ 为 $N \times L$ 矩阵，可看做 $N$ 个 $L$ 长基向量组成的一组基。分离器学习一个乘性遮罩 $\boldsymbol{m}_i \in \mathbb{R}^{1\times N}$，遮罩作用于 $\boldsymbol{w}$（即与其按元素乘）后，得到各个声源的“变换域”向量表示 $\boldsymbol{d}_i = \boldsymbol{w} \odot \boldsymbol{m}_i$。解码器用 $\hat{\boldsymbol{s}}_i = \boldsymbol{d}_i V$ 将分离器得到的“变换域”表示还原回时域，再对各片段求和，得到声源分离后的音频。

这个乘性遮罩 $\boldsymbol{m}_i$ 如何得到呢？时间卷积网络（Temporal Convolutional Network, TCN）模块在其中起到关键作用。TCN 是由 $X$ 个层叠的一维空洞卷积（dilated convolution）组成的，其空洞系数分别为 $1, 2, \ldots, 2^{X-1}$。编码后的表示 $\boldsymbol{w}$ 经过层归一化、$1\times1$ 卷积导入 TCN 模块，其输出经过加总、PReLU 激活、$1\times1$ 卷积和 Sigmoid 激活得到遮罩 $\boldsymbol{m}_i$。每个一维空洞卷积模块（如图8aC）都含有跳跃连接，为了压缩参数数目，作者采用深度可分离卷积（depthwise separable convolution）的技巧，将频道数和卷积核大小的维度分离，先进行逐深度卷积（depthwise convolution），再进行逐点卷积（pointwise convolution），将参数个数由普通卷积的输入通道数 × 输出通道数 × 卷积核大小变为 输入通道数×(输出通道数＋卷积核大小)，使得所需参数个数压缩为原来的 $\frac{1}{\text{卷积核大小}}$。

Samuel u. a. (2020) 采用元学习来进一步缩减 Conv-TasNet 的参数空间。作者用 $\boldsymbol{e}_i$ 代表声源 $i$ 的词向量，在每一个卷积层或组归一化（group normalization）层，其对应声源 $i$ 的参数 $\boldsymbol{\theta}_{k,i}$ 由 $\boldsymbol{\theta}_{k,i} = W_{k,i}P_{k,i}e_i$ 生成，$W_{k,i}$ 和 $P_{k,i}$ 为线性层，且 $W_{k,i}$ 输出比输入小，起到筛选信息的瓶颈作用。作者还对 Conv-TasNet 做了三项改进，包括能同时接收时域和频域信息的编码器、元学习相关的额外损失函数、多层解码，即依次用三个 Conv-TasNet 生成 8 kHz、16 kHz、32 kHz 的音频，其中低采样率的 TCN 输出会与高采样率的编码器输出拼接，一起作为高采样率分离器的输入。以上措施的综合应用取得了良好的结果。

## 3 仪器设备

- PC 机（i5 6300HQ / 4 Cores @ 2.3GHz, 16 GB DDR4 2133 RAM, 128 GB SSD + 1 TB HDD, GTX 960M 4GB）
- Python 3.6
  - Python 库：pytorch>=1.4.0、librosa>=0.7、lameenc>=1.2.2、musdb==0.3.1、museval==0.3.0、tqdm>=4.36、scipy>=1.3.1、matplotlib>=3.1.0、numpy>=1.17、simpleaudio>=1.0.4
  - 其他依赖项：ffmpeg>=4.2
- 商业软件
  - iZotope RX 7
  - Adobe Audition CC 2018
- 歌曲音频
  - Bruno Mars - Just the way you are 选段（含钢琴、鼓、人声）
  - 田馥甄 - 小幸运选段（含钢琴、鼓、贝斯、人声）
  - Scott LaFaro, Bill Evans, Paul Motian - Waltz for Debby（含钢琴、鼓、贝斯）
  - 五月天 - 仓颉（含钢琴、鼓、贝斯、人声）
  - 五月天 - 天使（含钢琴、鼓、贝斯、人声）

# 4 实验内容和步骤

1. 读入和保存音频，并绘制和保存频谱图。这些函数都在 `data_utils.py` 中，且都通过了测试（测试文件为 `test/data_utils_test.py`。
2. 设计程序架构。本项目中所有音频处理都用 `models.BaseAudioProcessor` 子类的实例完成。类似 sci-kit learn 中机器学习模型的架构，`BaseAudioProcessor` 中定义了 `__init__`、`__str__`、`process` 三个方法和 `result_key` 这个静态变量。音频处理器在处理（`process`）音频后，可以通过类似字典访问的 `processor['Drums']` 语句访问其中的声源，而字典的合法键保存在 `result_key` 元组中。
3. 认真研读论文，了解算法原理，有公开代码的基于公开代码实现，无公开代码的自己编程实现。
4. 编写 `main.py`，调用以上提到的所有功能，并完善异常处理，使得用户运行一次就能得到所有模型分离结果的音频和频谱图。

# 5 实验结果与分析

## 5.1 定量结果

由于训练深度学习模型成本过高（如 Demucs 需要在数张 32 GB 显存的 V100 GPU 上训练数百个 epoch），本项目中没有进行深度学习模型的训练和调优。表1中显示了 PapersWithCode 网站上公布的不同模型在 MusDB18 测试集上的评测结果。评测指标为信号–扭曲比（Signal-Distortion Rate, SDR，见Vincent u. a. (2006)）。可见，经过额外训练的 Demucs 和 Conv-TasNet 性能最优。根据Samuel u. a. (2020)，Meta-TasNet 在验证集中超过了 Conv-TasNet，但测试集中没能超过，可能是由于元学习限制了模型的表达能力，或者是因为它们没有采用相同的超参数。

| Model | Vocal | Drums | Bass | Other | Average |
|---|---|---|---|---|---|
| Demucs (extra) | 7.05 | 7.08 | 6.70 | 4.47 | 6.32 |
| Conv-TasNet (extra) | 6.74 | 7.11 | 7.00 | 4.44 | 6.32 |
| Conv-TasNet | 6.81 | 6.08 | 5.66 | 4.37 | 5.73 |
| Demucs | 6.29 | 6.08 | 5.83 | 4.12 | 5.58 |
| Meta-TasNet | 6.40 | 5.91 | 5.58 | 4.19 | 5.52 |

表 1: 不同模型在 MusDB18 测试集上的 SDR [5]
extra 表示训练时用到了2.5中描述的 2000 首无标签歌曲

## 5.2 案例研究

接下来笔者根据各模型分离实际流行歌曲的效果，评判各模型的优劣并分析背后的原因。附录A, B, C中有各模型分离 *Just the way you are*、《小幸运》和 *Waltz for Debby* 得到的实验结果的频谱图，对应的音频也已随本报告提交[6]。笔者还将各模型和商业软件分离的流行歌曲交由 10 位阿卡贝拉清唱社的社员进行悦耳程度（主要指目标声源声音清晰，其他杂音小）的评分，得到了如表2的结果。可见，Demucs 模型得分最高，Meta-TasNet 次之，但 2018 年推出的商业软件 iZotope RX 7 也实力强劲。下面进行各模型的具体分析。

[5] 数据来自 https://paperswithcode.com/sota/music-source-separation-on-musdb18
[6] 出于文件大小的考虑，提交的音频为 mp3 格式，由模型生成的 wav 文件转换得到

| 模型 | CCE | HPSS | REPET-SIM | Demucs | Meta-TasNet | iZotope |
|---|---|---|---|---|---|---|
| 人声 | 3.0 | | 3.5 | 3.7 | 4.0 | 3.9 |
| 鼓 | | 2.9 | | 4.6 | 4.3 | 4.2 |
| 贝斯 | | | | 4.2 | 4.2 | 3.7 |
| 其他 | 3.5 | 2.3 | 1.1 | 3.6 | 3.4 | 3.5 |
| 伴奏 | 3.5 | | 1.1 | 4.3 | 3.9 | 4.0 |

表 2: 十位阿卡贝拉清唱社社员对各方法分离所得音频的平均听感得分

#### 5.2.1 中置声道提取

中置声道提取仍然是目前主流且热门的人声消除技术，根据笔者的使用经历，唱吧、全民 K 歌上大多数伴奏都是用此法制作而成。该方法最大的特点便是计算量小、效果较好。虽然如2.1节理论分析的那样，鼓和贝斯等因也在中置声道所以损失严重，人声的混响因为常常会方位不在中间而无法消除，而且得到的基本是单声道音频，但总体而言，效果勉强可接受，且没有杂音（artifact）。经过低频泄露，贝斯和大鼓获得了较好的补偿，但小鼓和镲等乐器依然损耗严重。以 *Just the way you are* 为例（如图9），三种方法的人声都有能听见的残留，而低频泄露后的版本低频明显丰富了许多。笔者自己实现的低频泄露的版本分离效果与 Adobe Audition 分离的效果基本相同，证明笔者正确掌握了中置声道提取的相关原理。

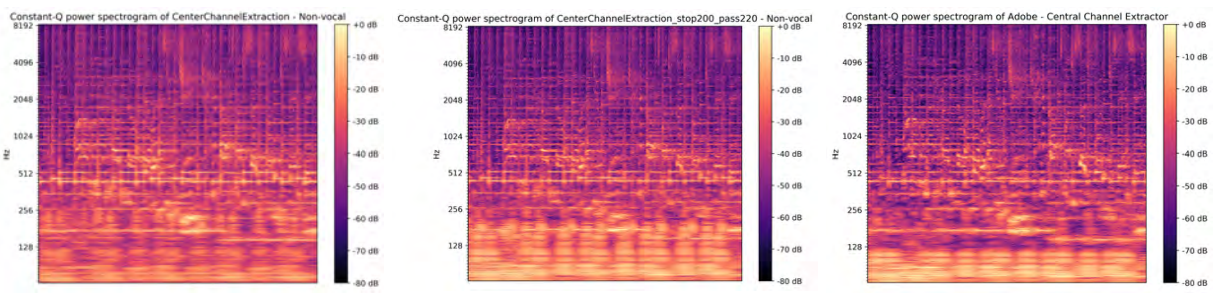
另外必须注意，中置声道提取本身没有任何声源分离的功能，完全依赖于录音时各乐器的位置和混音时对声像的处理。例如，对 *Waltz for Debby* 这一三重奏现场录音进行中置声道提取，就只能消除中间的钢琴声部，得到在两边的贝斯和鼓。

#### 5.2.2 旋律–打击乐分离技术（HPSS）

HPSS 通过中值滤波器，巧妙地实现了旋律和打击乐的分离，令笔者叹为观止。但作为一种通过先验分离声源的算法，其难免有设计者考虑未及之处。实验表明，HPSS 在分离 *Just the way you are* 时，由于其中的节奏明快、鼓点较为密集，因此鼓点也被部分纳入了“旋律”的部分（见附图3b）。而且人声的声母/辅音部分和部分器乐的“音头”（如贝斯的拨弦声）由于频谱范围较宽且出现突然，会被纳入“打击乐”部分，而“旋律”部分听起来则只有人声的元音和乐器的音尾。这提示我们专家系统的不可靠性和数据驱动模型的重要性。

#### 5.2.3 REPET-SIM

REPET-SIM 通过在 $k$ 个相似的帧中取中位数来估计背景噪声。实验表明，其分离出的人声部分较为清晰，说明其对背景噪声的估计较为准确。但“伴奏”的估计则与事实相去千里，如 *Just the way you are* 用 REPET-SIM 过滤得到的伴奏（附图4a）几乎每个十六分音符都有鼓点，《小幸运》间奏部分的伴奏完




(a) 无低频泄露 (b) 参数：200 Hz 低频泄露 (c) Adobe Audition 的提取结果

图 9: 不同方法使用中置声道提取进行人声消除的结果比较

全走样，如同噪声，小提琴独奏被归入人声（和 HPSS 一样，这是基于先验算法的通病）。这是因为算法得到的伴奏的每一帧并不是对当前帧的某种过滤，而是一个和当前帧较接近的帧。因此，如果一帧和很多帧都较为接近，它完全可能被过度利用，在伴奏音频的各处重复出现。

#### 5.2.4 Demucs

Demucs 的总体表现良好，在鼓的分离上尤为出色。分离出的人声有一些电子杂音，大约是因为泄露了一些伴奏的频率进来。贝斯的分离也不尽人意，一些比较高音的贝斯和亮眼的贝斯独奏没有被纳入贝斯的部分（见5），这可能是 MusDB 的训练数据中贝斯大都以低音铺底出现导致的过拟合现象。其他乐器上整体比较模糊，感觉音头缺失较明显，频率有损失也有泄露，但相比其他模型来说还是相当出色的。值得一提的是，将 Demucs 分离出的贝斯、鼓、其他三个声源加在一起，可以得到质量可接受的卡拉 OK 伴奏。另外，用额外无标签数据训练过的 Demucs 也的确比只用 MusDB 训练的 Demucs 模型分离效果更好，这提示我们利用额外数据提高深度学习模型性能的有效性。

#### 5.2.5 Conv-TasNet 和 Meta-TasNet

Conv-TasNet 总体表现良好。和 Demucs 相比，美中不足的偶尔会有一些明显的频谱泄露或损失。如《小幸运》中钢琴的高音部分损失严重，*Just the way you are* 中的鼓有提前停止的现象（即鼓此时应还在震动，但分离出的鼓却已戛然而止）。Meta-TasNet 的声源分离做得不错，但生成的音频音质整体较低，听感较闷，笔者猜测与其三级解码、逐级提高采样率的结构还有待优化有关。由于上述原因，这两个模型生成的卡拉 OK 伴奏质量也稍逊于 Demucs。

#### 5.2.6 iZotope RX 7

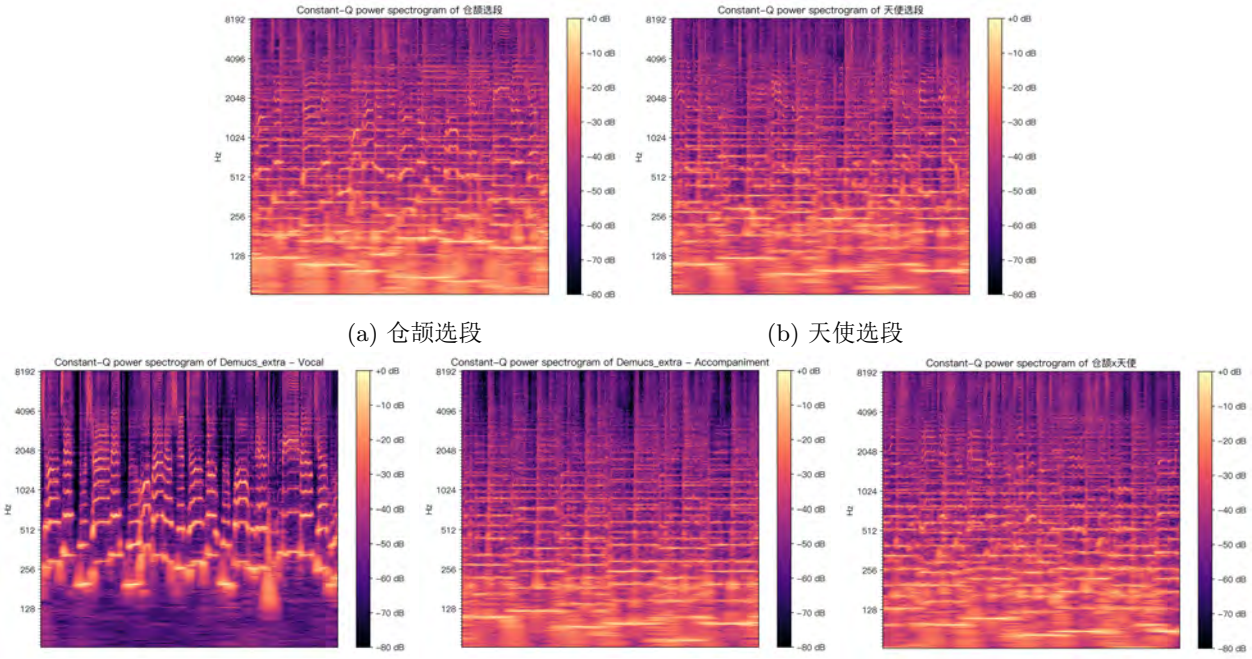
作为一个商业软件，要在有限的算力、内存、时间内完成有效的声源分离殊为不易。但 iZotope RX 7 似乎很好地完成了任务。虽然贝斯有多处未识别正确（如《小幸运》间奏中的贝斯）、常有比 Demucs 更大的频谱泄漏，但鼓和人声的处理都相当自然——即使有杂音也不刺耳，体现了商业软件公司性能优化的深厚功底。

### 5.3 模型应用

声源分离模型给扒谱、自动伴奏生成、重混等带来了无限可能。笔者利用最新的声源分离技术，发挥想象力，制作了一段两首歌曲的重混（remix）。《仓颉》和《天使》两首歌曲速度相近、和弦进行基本相同、调性差一个全音，笔者用 Demucs 模型将《仓颉》的人声提取出来，和降一个全音、长度稍作压缩（这两个操作由 iZotope RX 7 完成）后的《天使》的伴奏混在一起，得到仓颉 × 天使这个重混片段，如图10。

## 6 总结与讨论

声源分离是一项有着广阔应用前景、近几年来进展迅速的技术。本文回顾了近十年来声源分离技术重要的模型，剖析了算法背后的直觉和原理，并结合实验分析了它们的效果、比较了它们的优劣，并探究了背后的原因。传统的声源分离算法多基于人们对流行音乐频谱图的经验和观察，因而产生的音频质量较低，且泛化能力有限；目前最优秀的声源分离算法（如 Demucs）是基于深度学习的、尤其是在时域直接进行端到端变换的算法，它们已能对一些简单的流行歌曲产生令人满意的伴奏，并且已经开始落地（iZotope RX 7 等工业级音频处理软件中已开始应用深度学习的算法来分离声部）。




(a) 仓颉选段 (b) 天使选段

(c) 仓颉选段 - 人声部分 (d) 天使选段 - 伴奏部分 (e) 仓颉 × 天使

图 10: 用最优秀的声源分离技术进行歌曲重混

但同时，现有声源分离算法产生的音频往往带有明显的杂音（artifact），尤其是单轨的人声和其他（去除了人声、贝斯、鼓后的部分）仍然不尽人意。模型方面，空洞卷积、跳跃连接对提升模型性能大有裨益。但最近在序列生成领域大放异彩的 Transformer 模型还未崭露头角。这一模型可能是未来值得在声源分离领域尝试的。笔者的实验（5.2.4节）证明，丰富的无标注数据能为声源分离模型带来可观的性能提升。因此，除了模型的改进，也许收集大量无监督数据也能帮助性能的提高。一些后处理技巧可能也值得关注。例如，提取音频后利用降噪和声音增强技术对音频进行优化，可能有助于清除频域泄露、增强音质。同时，已分离的声部也许可以作为信息引导其他声部的分离。如附图5所示，“其他”中未消除干净的人声和“人声”是高度重叠的，如能设法将这样的音频“完璧归赵”，转入“人声”部分，就可以提高声源分离算法的性能。笔者相信通过后续研究者的努力，声源分离技术能真正实用化并落地，那无疑是音乐工作者和像笔者这样的音乐爱好者的福音。

## 参考文献

[Dauphin u. a. 2017] Dauphin, Yann N. ; Fan, Angela ; Auli, Michael ; Grangier, David: Language Modeling with Gated Convolutional Networks. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, JMLR.org, 2017 (ICML’17), S. 933–941

[Defossez u. a. 2019] Defossez, Alexandre ; Usunier, Nicolas ; Bottou, Léon ; Bach, Francis: *Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed*. 09 2019

[Défossez u. a. 2018] Défossez, Alexandre ; Zeghidour, Neil ; Usunier, Nicolas ; Bottou, Léon ; Bach, Francis: SING: Symbol-to-Instrument Neural Generator. In: *ArXiv* abs/1810.09785 (2018)

[Driedger u. a. 2014] Driedger, Jonathan ; Müller, Meinard ; Disch, Sascha: Extending Harmonic-Percussive Separation of Audio Signals, 01 2014

[FitzGerald 2012] FitzGerald, D.: Vocal separation using nearest neighbours and median filtering. In: *IET Irish Signals and Systems Conference (ISSC 2012)*, 2012, S. 1–5

[Fitzgerald 2010] Fitzgerald, Derry: Harmonic/Percussive Separation using Median Filtering. In: *13th International Conference on Digital Audio Effects (DAFx-10)* (2010), 01

[Huang u. a. 2016] Huang, Gao ; Liu, Zhuang ; Weinberger, Kilian: Densely Connected Convolutional Networks. (2016), 08, S. 12

[Liutkus u. a. 2017] Liutkus, Antoine ; Stöter, Fabian-Robert ; Rafii, Zafar ; Kitamura, Daichi ; Rivet, Bertrand ; Ito, Nobutaka ; Ono, Nobutaka ; Fontecave, Julie: The 2016 Signal Separation Evaluation Campaign, 02 2017, S. 323–332. – ISBN 978-3-319-53546-3

[Luo und Mesgarani 2018] Luo, Yi ; Mesgarani, Nima: *TasNet: Surpassing Ideal Time-Frequency Masking for Speech Separation.* 09 2018

[Luo und Mesgarani 2019] Luo, Yi ; Mesgarani, Nima: Conv-TasNet: Surpassing Ideal Time – Frequency Magnitude Masking for Speech Separation. In: *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 27 (2019), August, Nr. 8, S. 1256–1266. – URL https://doi.org/10.1109/TASLP.2019.2915167. – ISSN 2329-9290

[Ono u. a. 2008] Ono, Nobutaka ; Miyamoto, Kenichi ; Le Roux, Jonathan ; Kameoka, Hirokazu ; Sagayama, Shigeki: Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram. (2008), 01

[Rafii u. a. 2017] Rafii, Zafar ; Liutkus, Antoine ; Stöter, Fabian-Robert ; Mimilakis, Stylianos I. ; Bittner, Rachel: *The MUSDB18 corpus for music separation.* Dezember 2017. – URL https://doi.org/10.5281/zenodo.1117372

[Rafii und Pardo 2012] Rafii, Zafar ; Pardo, Bryan: Music/Voice Separation Using the Similarity Matrix. In: *ISMIR*, 2012

[Samuel u. a. 2020] Samuel, David ; Ganeshan, Aditya ; Naradowsky, Jason: Meta-Learning Extractors for Music Source Separation, 05 2020, S. 816–820

[Stoller u. a. 2018] Stoller, Daniel ; Ewert, Sebastian ; Dixon, Simon: *Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation.* 06 2018

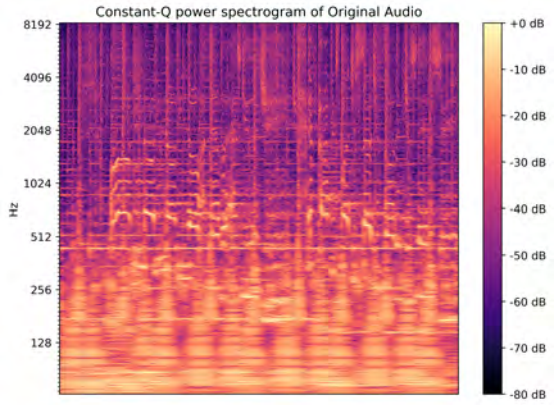
[Takahashi u. a. 2018] Takahashi, N. ; Goswami, N. ; Mitsufuji, Y.: Mmdenselstm: An Efficient Combination of Convolutional and Recurrent Neural Networks for Audio Source Separation. In: *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2018, S. 106–110

[Takahashi und Mitsufuji 2017] Takahashi, Naoya ; Mitsufuji, Yuki: Multi-Scale multi-band densenets for audio source separation, 10 2017, S. 21–25

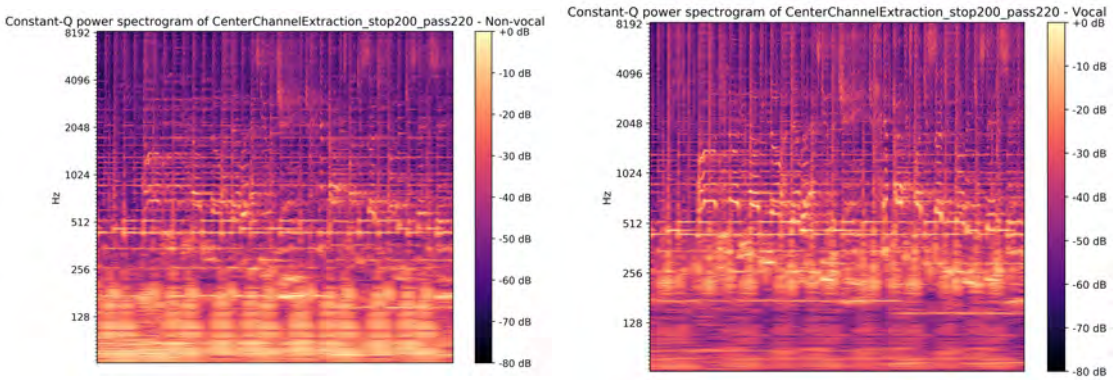
[Vincent u. a. 2006] Vincent, Emmanuel ; Gribonval, Rémi ; Févotte, Cédric: Performance measurement in blind audio source separation. In: *Audio, Speech, and Language Processing, IEEE Transactions on* 14 (2006), 08, S. 1462 – 1469

# A 附录：用各模型分离 *Just the way you are* 的实验结果

本附录中所有图片均经常数 Q 变换（Constant-Q Transform）绘制。

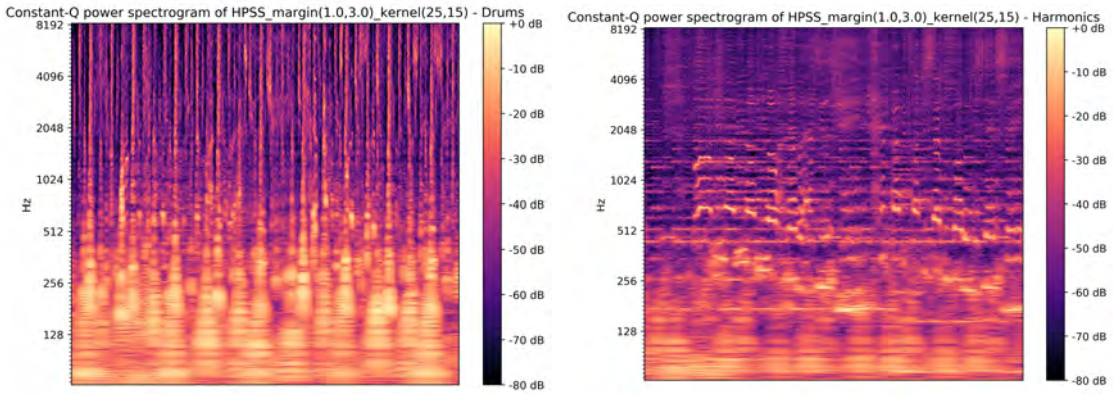



附图 1: *Just the way you are* 原始音频的频谱图



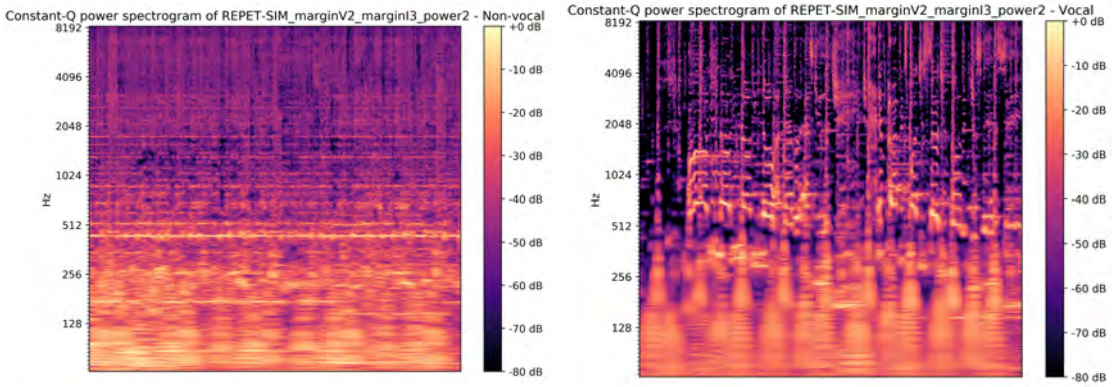

(a) 两侧声道（背景音） (b) 中置声道（前景音）

附图 2: 中置声道提取法分离 *Just the way you are* 结果的频谱图
参数：低频泄露 200 Hz



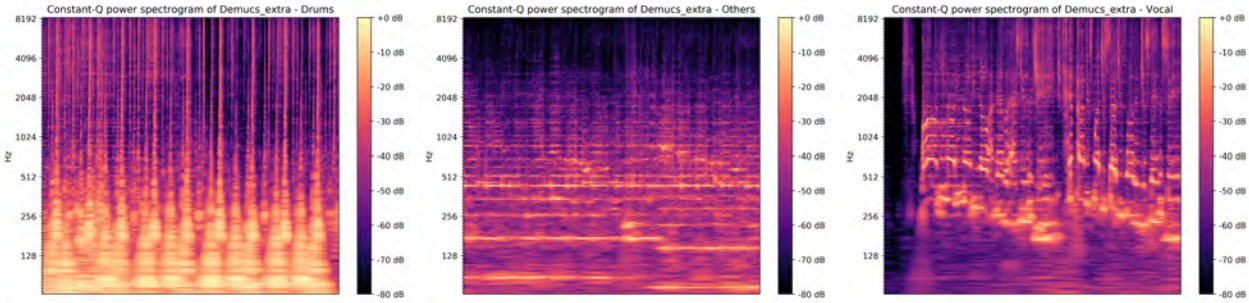

(a) 打击乐 (b) 旋律音

附图 3: HPSS 分离 *Just the way you are* 结果的频谱图
参数：边距 (旋律1, 节奏3)，中值滤波器大小 (旋律25, 节奏15)



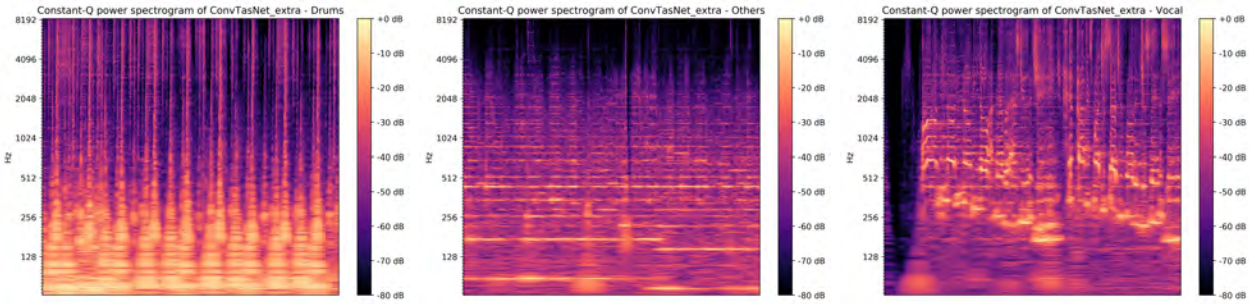

(a) 伴奏 (b) 人声

附图 4: REPET-SIM 分离 *Just the way you are* 结果的频谱图
参数：边距 (人声2, 伴奏3)，中值滤波器大小 (旋律25, 节奏15)



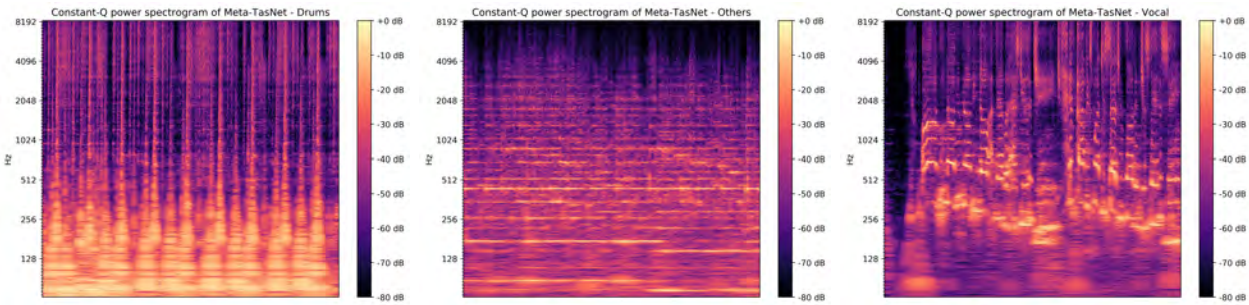

(a) 鼓 (b) 其他伴奏 (c) 人声

附图 5: Demucs (extra) 分离 *Just the way you are* 结果的频谱图



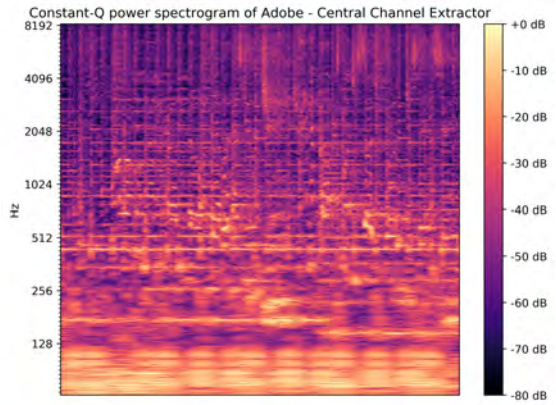

(a) 鼓 (b) 其他伴奏 (c) 人声

附图 6: Conv-TasNet (extra) 分离 *Just the way you are* 结果的频谱图

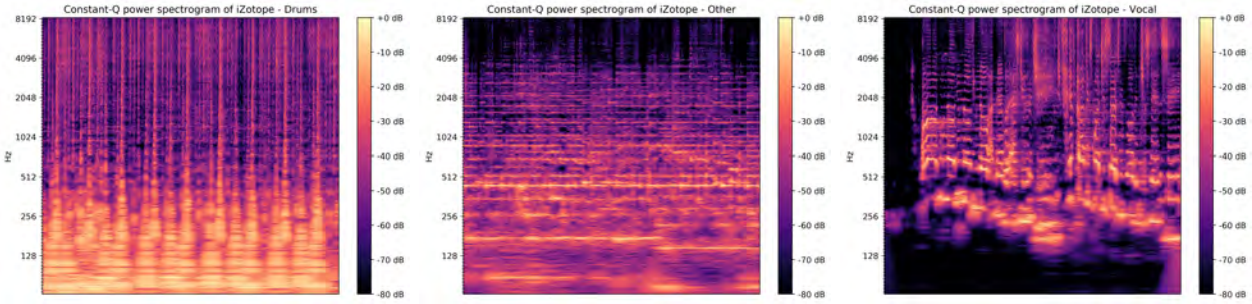



(a) 鼓 (b) 其他伴奏 (c) 人声

附图 7: Meta-TasNet 分离 *Just the way you are* 结果的频谱图

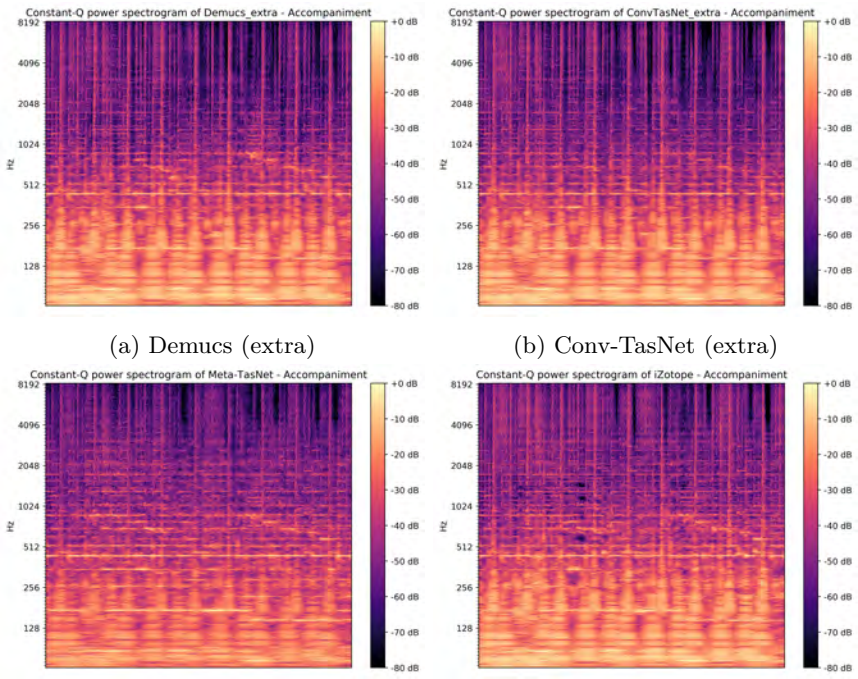



附图 8: 用 Adobe Audition 的中置声道提取功能分离 *Just the way you are* 结果的频谱图




(a) 鼓 (b) 其他伴奏 (c) 人声

附图 9: iZotope RX 7 分离 *Just the way you are* 结果的频谱图



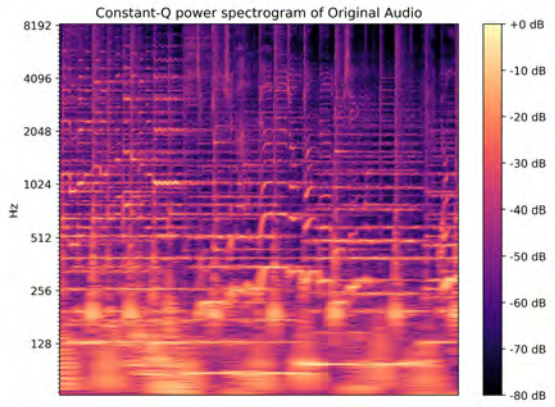

(a) Demucs (extra) (b) Conv-TasNet (extra)

(c) Meta-TasNet (d) iZotope RX 7

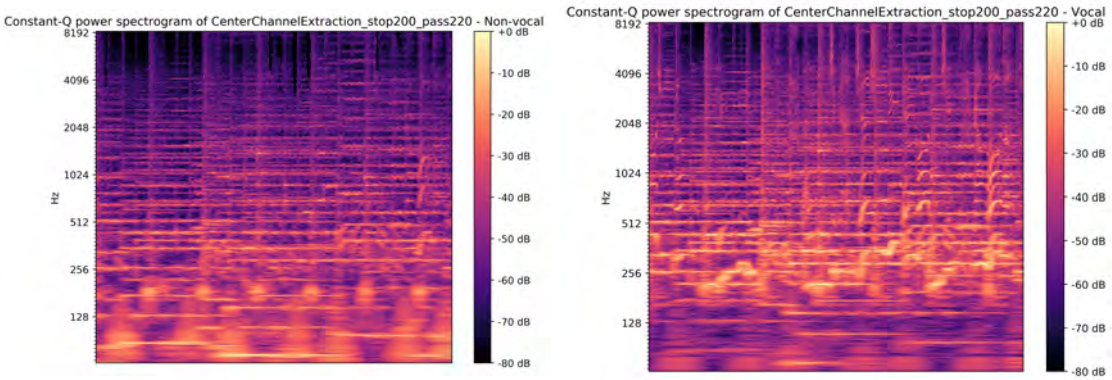
附图 10: 各四声源分离模型得到的 *Just the way you are* 伴奏带比较

# B 附录：用各模型分离《小幸运》的实验结果

本附录中所有图片均经常数 Q 变换（Constant-Q Transform）绘制。

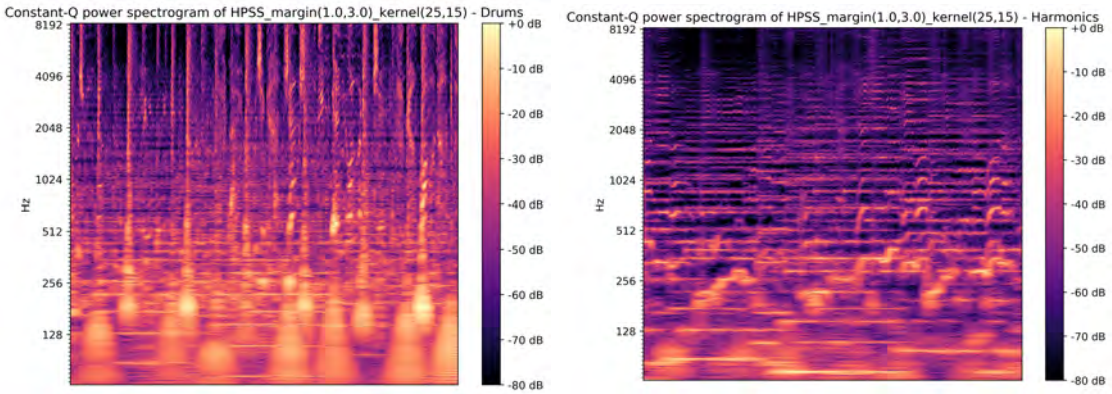



附图 1: 《小幸运》原始音频的频谱图



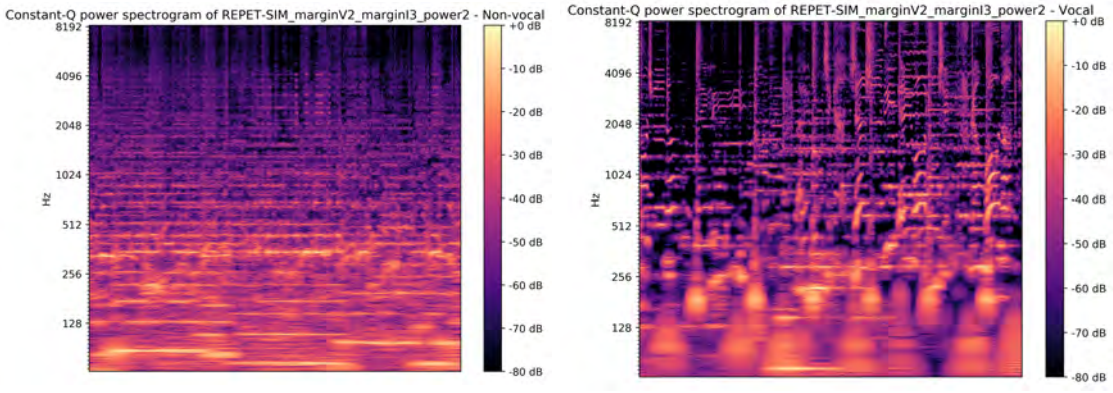

(a) 两侧声道（背景音） (b) 中置声道（前景音）

附图 2: 中置声道提取法分离《小幸运》结果的频谱图
参数：低频泄露 200 Hz



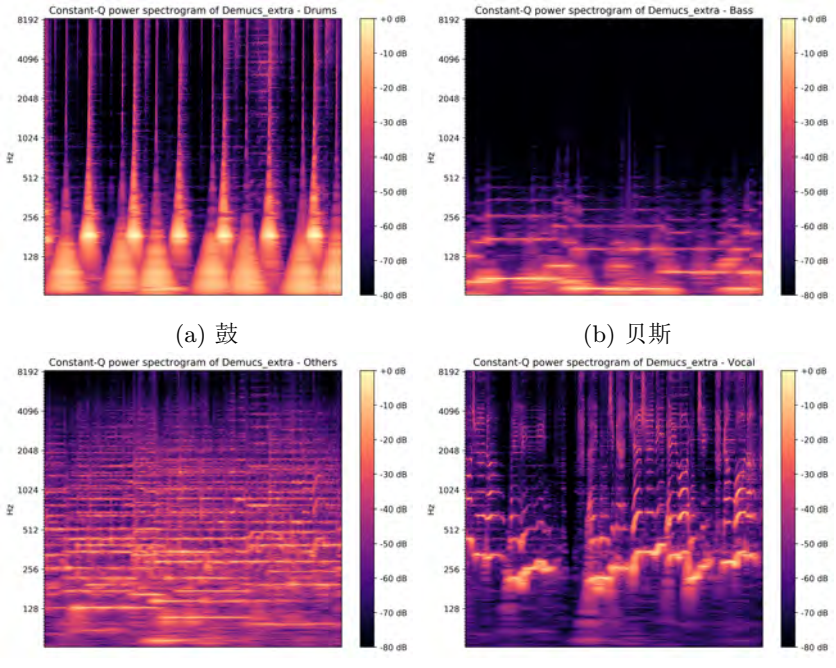

(a) 打击乐 (b) 旋律音

附图 3: HPSS 分离《小幸运》结果的频谱图
参数：边距 (旋律1, 节奏3)，中值滤波器大小 (旋律25, 节奏15)




(a) 伴奏 (b) 人声

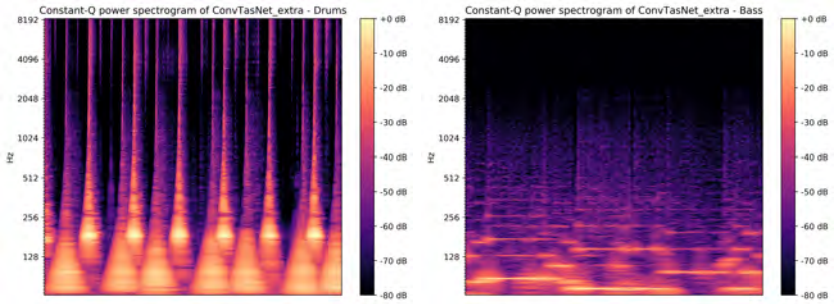
附图 4: REPET-SIM 分离《小幸运》结果的频谱图
参数：边距 (人声2, 伴奏3)，中值滤波器大小 (旋律25, 节奏15)



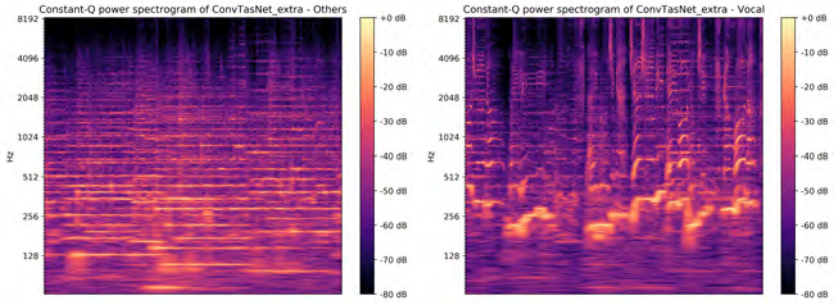

(a) 鼓 (b) 贝斯

(c) 其他伴奏 (d) 人声

附图 5: Demucs (extra) 分离《小幸运》结果的频谱图

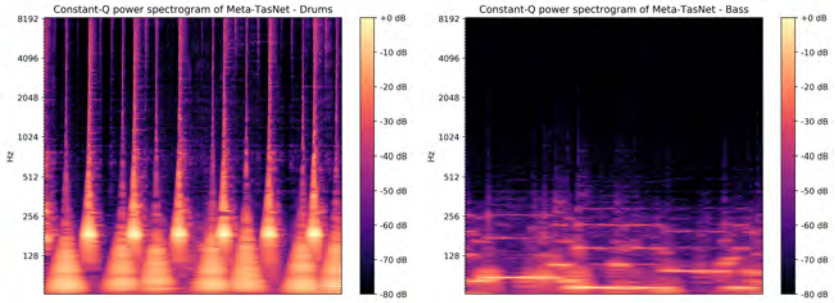



(a) 鼓 (b) 贝斯

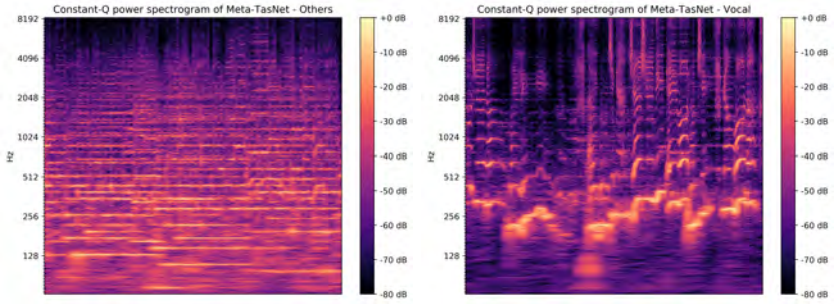



(c) 其他伴奏 (d) 人声

附图 6: Conv-TasNet (extra) 分离《小幸运》结果的频谱图

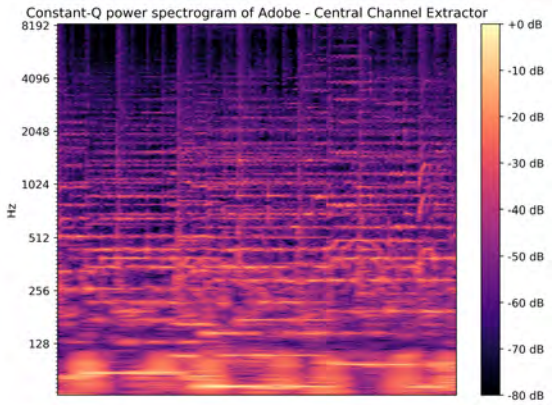



(a) 鼓 (b) 贝斯

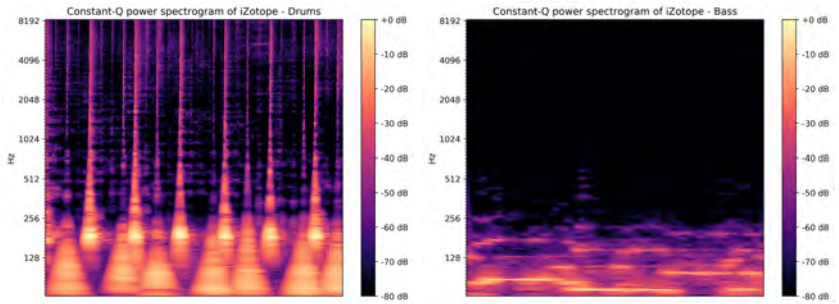



(c) 其他伴奏 (d) 人声

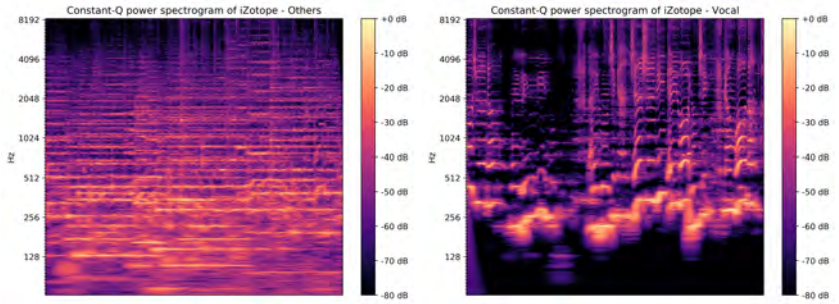
附图 7: Meta-TasNet 分离《小幸运》结果的频谱图




附图 8: 用 Adobe Audition 的中置声道提取功能分离《小幸运》结果的频谱图

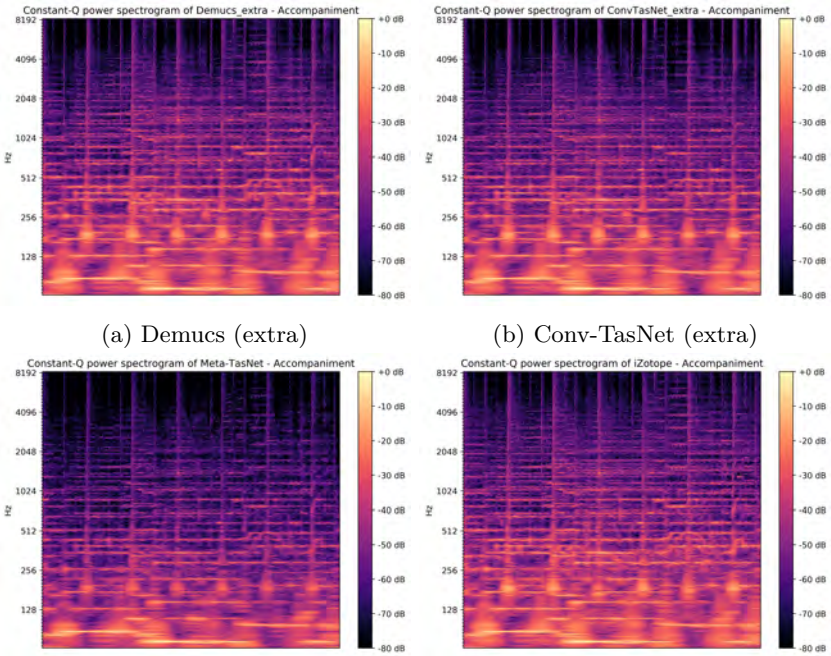



(a) 鼓 (b) 贝斯




(c) 其他伴奏 (d) 人声

附图 9: iZotope RX 7 分离《小幸运》结果的频谱图




(a) Demucs (extra)

(b) Conv-TasNet (extra)

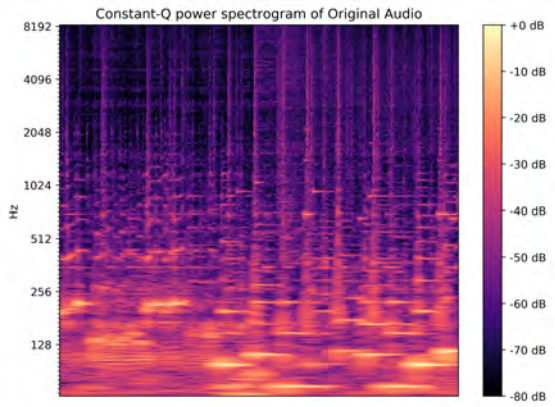
(c) Meta-TasNet

(d) iZotope RX 7

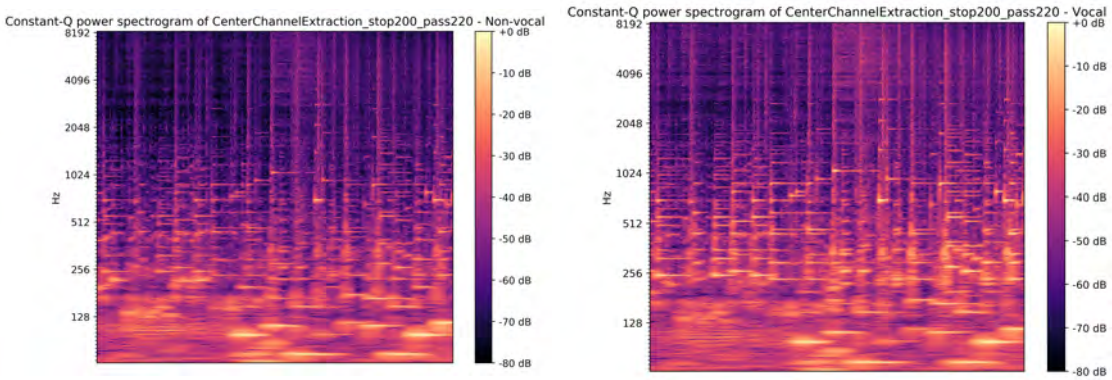
附图 10: 各四声源分离模型得到的《小幸运》伴奏带比较

# C 附录：用各模型分离 *Waltz for Debby* 的实验结果

本附录中所有图片均经常数 Q 变换（Constant-Q Transform）绘制。

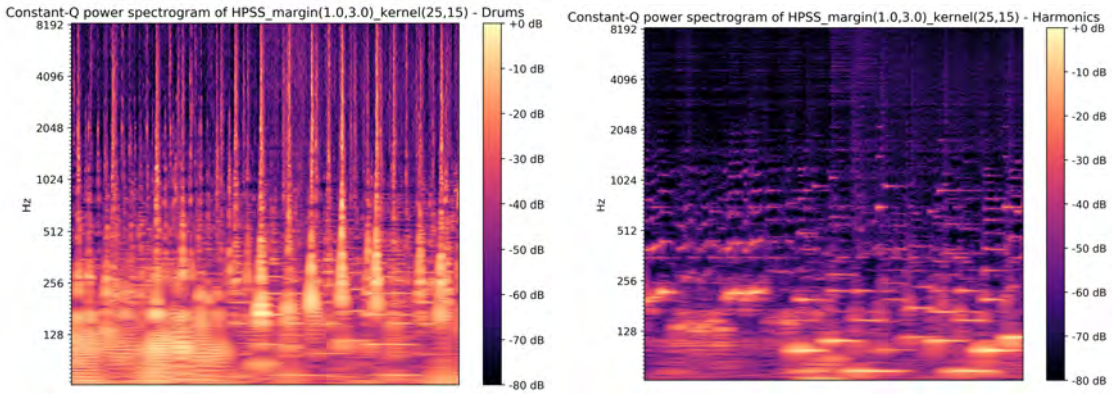



附图 1: *Waltz for Debby* 原始音频的频谱图



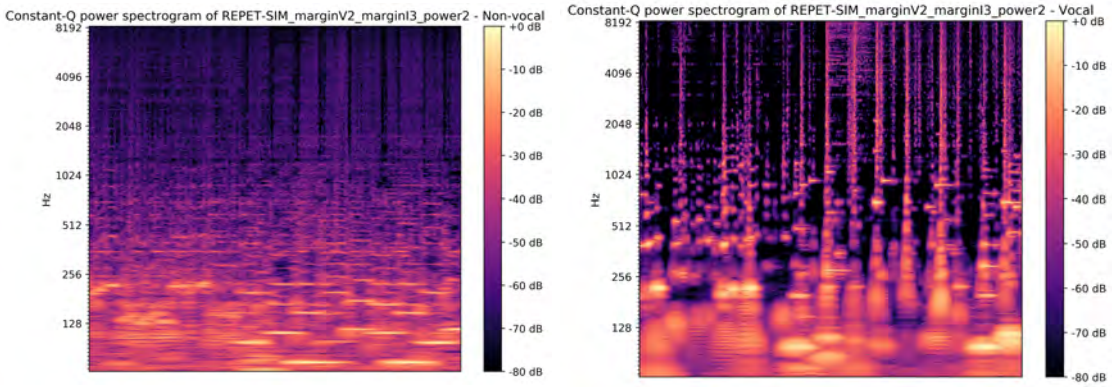

(a) 两侧声道（背景音） (b) 中置声道（前景音）

附图 2: 中置声道提取法分离 *Waltz for Debby* 结果的频谱图
参数：低频泄露 200 Hz



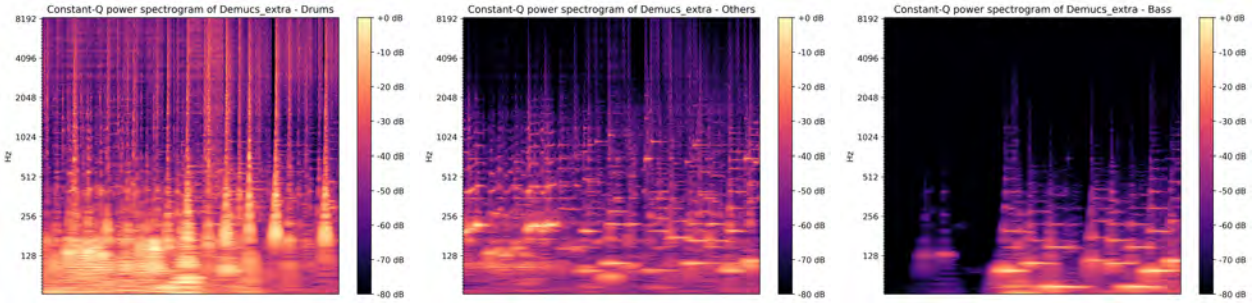

(a) 打击乐 (b) 旋律音

附图 3: HPSS 分离 *Waltz for Debby* 结果的频谱图
参数：边距 (旋律1, 节奏3)，中值滤波器大小 (旋律25, 节奏15)



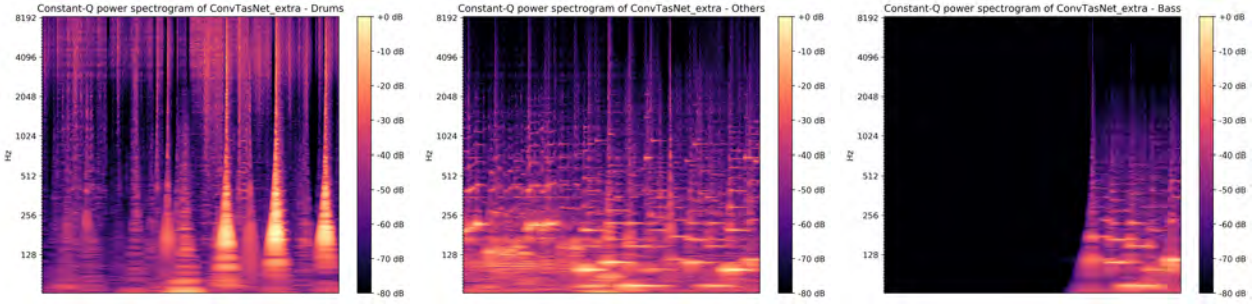

(a) 伴奏 (b) 前景音

附图 4: REPET-SIM 分离 *Waltz for Debby* 结果的频谱图
参数：边距 (人声2, 伴奏3)，中值滤波器大小 (旋律25, 节奏15)



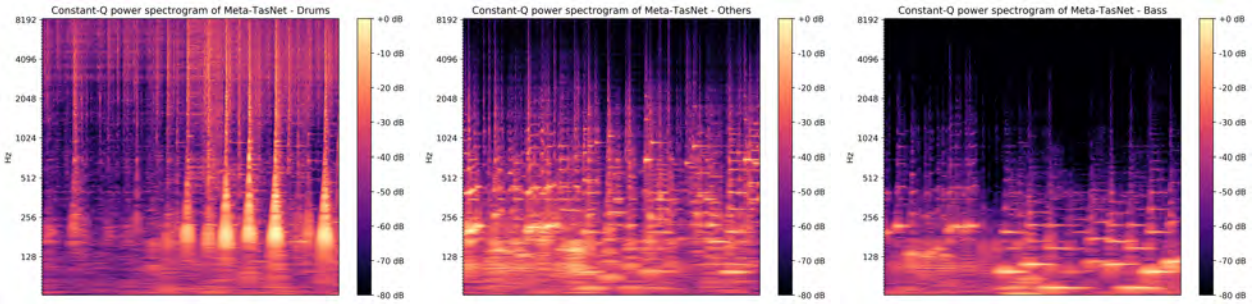

(a) 鼓 (b) 其他伴奏 (c) 贝斯

附图 5: Demucs (extra) 分离 *Waltz for Debby* 结果的频谱图




(a) 鼓 (b) 其他伴奏 (c) 贝斯

附图 6: Conv-TasNet (extra) 分离 *Waltz for Debby* 结果的频谱图




(a) 鼓 (b) 其他伴奏 (c) 贝斯

附图 7: Meta-TasNet 分离/*Waltz for Debby* 结果的频谱图