# 声枢：一种基于LSTM-CNN轻量化深度学习的多头声学特征分类模型——以声乐技术识别为例

**刘麓宽***

**通讯作者**：刘麓宽；邮箱：1253213827@qq.com；电话：13924638713
**指导老师**：李枫
**单位**：深圳中学

**摘要：** 随着歌唱语音合成（SVS）与语音语言模型（SLM）的快速发展，高质量、多维度的声学特征标注数据集成为技术突破的关键支撑，而传统声乐学习中存在优质指导资源稀缺、技巧界定缺乏客观量化标准的核心痛点，人工标注复杂声学特征不仅耗时耗力，还难以满足大规模SVS模型训练需求。针对这一问题，本文提出一种轻量化多头声乐技术识别模型“声枢（VocalPivot v4）”，基于LSTM-CNN融合架构，整合CNN的频域局部特征提取优势与LSTM的时序动态建模能力，通过1D卷积核优化、梅尔频谱裁剪、噪声注入等轻量化策略，实现对真声、假声、混声等7类声乐技术标签的精准同步识别。

实验基于GTSinger数据集完成，结果表明，该模型参数量轻量版本可压缩在0.13M以内，增强版本7类标签分类平均准确率94.99%，轻量化程度优于STARS等主流框架。该模型及附带开发的交互界面软件“知音”，不仅可为声乐学习者提供实时反馈，更能作为高效音频标注工具，为SVS模型训练提供高质量多维度标注数据，缓解ASR-LM-TTS技术链路的连接误差。

相关代码可查看：https://github.com/bgArray/VocalPivot；https://github.com/bgArray/ZhiYin。

**关键词：**自动歌唱标注（ASA）；声乐技术识别（Vocal Technique Identification）；LSTM-CNN融合框架；轻量化优化

# 1 引言

## 1.1 研究背景、意义及现状

### 1.1.1 声乐学习的存在痛点

声乐学习作为一门高度依赖实践与反馈的学科，长期面临两大瓶颈：一是优质教学资源分布不均，多数学习者难以获得专业教师对真声、混声、假声等核心技巧的实时指导与量化评估；二是声乐技术的界定缺乏客观标准，传统依赖人工听觉判断的方式主观性强，难以精准区分相似技巧（如混声与假声）的特征差异，导致学习者难以把握技巧习得的关键节点。这些痛点严重制约了声乐学习的普及与效率提升。

### 1.1.2 SVS与语音语言模型（SLM）的发展需求

声乐方面，近年来歌唱语音合成（SVS）技术取得显著突破：从早期DiffSinger[3]基于浅层扩散机制解决过平滑问题，到TCSinger[20]实现零样本歌唱语音合成与多维度风格控制，SVS模型对输入数据的精细化程度要求日益提高。

同时，语音语言模型（SLM）的兴起推动了端到端语音交互的发展，但“ASR+LM+TTS”的传统链路存在信息损失、误差累积与传递等问题，亟需通过高质量音频标注数据构建更精准的语音tokenize框架。[16]

然而，当前SVS与SLM发展面临共同瓶颈：缺乏大规模、多维度的复杂声学特征标注数据集。现有数据集要么标注维度单一（仅涵盖音符、音素等基础信息），要么依赖人工标注导致规模有限，难以支撑模型对声乐技术、情感基调等复杂特征的学习。自动歌唱标注（ASA）技术虽能缓解这一问题，但现有标注模型存在参数量大、部署困难（如STARS[6]框架）等问题。因此，开发高精度、轻量化的多标签声乐技术识别模型，为SVS与SLM提供高效标注工具，成为推动相关领域发展的关键。

### 1.1.3 研究价值与应用场景

本研究提出的VocalPivot系列模型，以声乐技术识别为切入点，实现了轻量化与高精度的平衡：

从应用价值看，模型可集成于移动应用等本地端，为声乐学习者提供“AI助教”式实时反馈，降低专业技巧学习门槛；

从技术价值看，模型可作为高效音频标注工具，支撑大规模复杂声学特征标注数据集构建，为SVS模型（如TCSinger）与SLM的优化提供数据支撑；

从产业价值看，轻量化特性使其可嵌入音乐制作、虚拟歌手等场景，推动音频处理技术的平民化应用。

## 1.2 研究目标与主要贡献

本研究核心目标是构建一套高精度、轻量化、支持多标签分类的声乐技术识别模型（VocalPivot系列），实现对7类声乐技术的精准识别，为声乐学习辅助与SVS/SLM数据标注提供支撑。主要贡献如下： (a) 提出三级模型演进方案，从单一CNN到LSTM-CNN 3标签分类，最终形成多头分类LSTM-CNN架构，系统验证了融合架构在声乐技术识别中的优势；

(b) 优化轻量化技术策略，通过1D卷积核设计、梅尔频谱裁剪、噪声注入等方法，在保证92.70%准确率的前提下，将模型参数量控制在0.13M以内；（VocalPivot v3）

(c) 实现7类声乐技术的同步识别，为SVS与SLM提供多维度标注工具，支撑复杂声学特征数据集构建；

(d) 拓展跨语言适配能力，基于 v3 架构升级为 Enhanced Multi-label CNN-LSTM 架构（VocalPivot v4），通过中英文混合数据联合训练，优化双层 LSTM 时序建模与多维度特征融合策略，在保持轻量化特性的同时，提升模型跨语言泛化性能，实现中英文声乐技术的统一精准识别（准确率94.99%）。

# 2 已有相关研究综述

## 2.1 自动歌唱标注（ASA）与SVS技术

自动歌唱标注（ASA）是SVS的核心支撑技术，旨在从歌唱音频中提取音素对齐、音符转录、声乐技术等多维度信息。早期ASA方法多采用分段处理策略，如ROSVOT[10]通过多尺度架构实现音符转录，但缺乏对声乐技术的识别能力；STARS框架首次实现音素对齐、音符转录、技术识别与风格标注的统一，但其复杂架构导致轻量化部署困难。

SVS技术的发展与ASA技术紧密相关：DiffSinger通过扩散模型提升了合成自然度，但依赖高质量标注数据；TCSinger实现零样本风格迁移，需要丰富的风格与技术特征标注作为支撑。现有SVS模型的性能瓶颈日益凸显为数据驱动的限制，亟需高效的ASA工具提供大规模复杂特征标注数据。

## 2.2 声乐技术识别模型研究

### 2.2.1 模型1：XGB-DE差分进化优化模型

该模型由Boratto等人提出，通过差分进化（DE）算法优化XGBoost的超参数，提取14项时域特征实现chest、mixed、head三类声乐寄存器的分类。其核心优势是在小样本（单一歌手、单一元音，350条涵盖3种声乐技术的数据）训练下能取得较高准确率，但存在明显局限：依赖人工设计特征，泛化能力差；小样本训练导致真实场景下性能波动剧烈，难以适配大规模标注需求。不过该实验仅使用14个时域声学特征进行提取，方案具有参考价值。[1]

### 2.2.2 模型2：GTSinger自带ROSVOT改编模型

该模型基于ROSVOT的多尺度架构改编，是GTSinger数据集的配套标注工具，支持混声、假声等6类声乐技术的识别。模型通过融合Conformer[2]与U-Net架构提升特征提取能力，但仅支持固定类别标签识别，且参数量较大（2.51M），缺乏多标签扩展能力与轻量化特性。

### 2.2.3 模型3：STARS统一标注框架

STARS是浙江大学提出的统一歌唱标注框架，采用“CMU Encoder+多尺度声学编码器”架构，实现音素对齐、音符转录、技术识别与风格标注的多任务集成。模型在COnPOff指标上达到71.0，RPA指标86.7，支持9类声乐技术识别，但参数量高达50.12M，推理速度慢，难以满足低资源设备部署需求。

本篇旨在尝试用简易框架完成STARS框架中的声乐技术标签分类任务，并非全自动ASA。

## 2.3 轻量化深度学习模型在其他音频分类中的应用

轻量化深度学习模型在音频分类中聚焦架构优化与特征筛选，通过深度可分离卷积、梅尔频谱裁剪等技术实现参数精简与维度降低。

其中CNN-RNN融合模型（含CRNN等衍生架构）在非声乐标注领域贡献显著，如Sebastian Murgul等人的CRNN模型：用4层卷积块提取吉他音频对数梅尔频谱特征，双向GRU捕捉扫弦与和弦的时序关联，和弦识别准确率达90%，远超传统CNN的73%-79%。在通用音频分类中，该架构也能精准区分乐器与环境音的频域及时序差异，小数据集下性能优于单一CNN或RNN。

轻量化与泛化平衡上，CNN-RNN通过架构精简与数据优化适配实际部署：Murgul的CRNN采用双分支共享卷积栈，参数量降至传统模型60%，训练仅需2小时，适配低资源设备；针对数据稀缺问题，结合合成与真实数据训练，使模型在麦克风噪声场景下和弦识别准确率仍达85%以上，较纯真实数据训练提升40%，为其他声学标注工程化落地提供可行路径。[12]

除此以外，LSTM-CNN融合架构有不错的兼容性。其也在语音情绪多标签的任务中取得了进展，例如电子科技大学采用与本篇类似的技术路径取得了不错的成绩。[4][17]

# 3 实验设计与模型架构

## 3.1 数据预处理

为提升模型的鲁棒性与轻量化程度，本文设计两套核心预处理策略，均经实验验证有效性。

### 3.1.1 噪声注入增强鲁棒性

采用频域偏移、幅度缩放、高斯噪声注入相结合的复合增强策略，模拟真实场景中音频信号的各类自然扰动与噪声干扰，仅对50%的训练样本随机应用增强，平衡原始数据与增强数据比例，避免过度扰动破坏核心特征： (a) 频域偏移：对梅尔频谱的频域轴进行±1范围内的随机偏移，偏移后的空白区域填充边缘均值，避免硬填充导致的特征突变，模拟频谱特征的自然偏移； (b) 幅度缩放：对频谱幅度进行0.9-1.1倍的随机缩放，模拟演唱力度的自然波动，保留音频核心能量特征；(c) 高斯噪声注入：叠加低强度高斯噪声（噪声标准差为原频谱标准差的5%），模拟环境背景噪声干扰，使模型学习区分有效音频信号与无用噪声，提升对复杂场景的适应能力。

实验表明，该复合增强策略可使模型对各类音频扰动的抗干扰能力显著提升，在含轻微噪声或未消除伴奏的测试场景中仍能完成分类任务，鲁棒性得到一定强化。

### 3.1.2 梅尔频谱裁剪优化模型尺寸

将原始梅尔频谱的频域范围裁剪为200Hz-5kHz关键频段（该频段包含声乐技术的核心特征，如真声的低频谐波、假声的高频成分），裁剪后梅尔倒谱频率特征维度从128维精简至74维。对于频域维度不足74的样本，通过频域后填充0值确保输入结构一致性。这一操作在不损失关键信息的前提下，显著降低特征冗余度，使模型参数量减少约42%，有效实现轻量化，同时提升模型训练与推理效率。

## 3.2 模型演进与架构设计

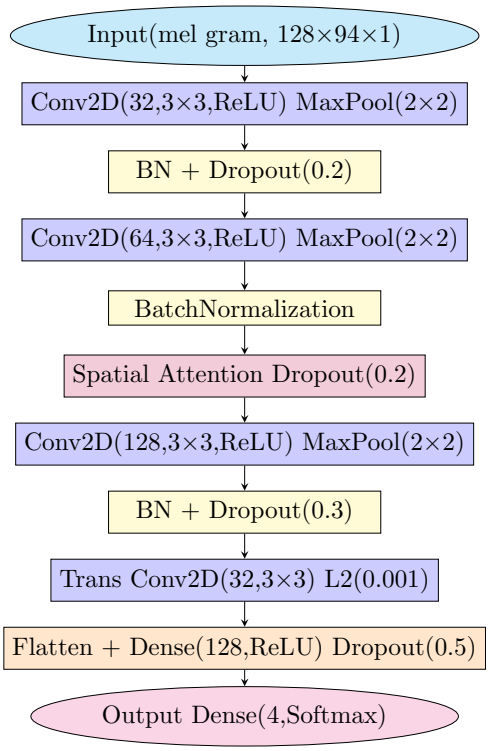
本文设计三级模型演进方案，从预实验到最终模型逐步优化，具体架构如下：

### 3.2.1 预实验模型：单一CNN架构

为检验仅使用CNN提取梅尔倒谱特征能否识别真声、混声、假声三类核心声乐发声模式，设计预实验并进行双重校验：(a) 核准梅尔频谱能否包含不同腔体共鸣位置发声技术的声学特征信息，测试该技术路径下CNN的分类准确率；(b) 通过反向传播输出训练后的CNN参数，观察CNN对不同腔体共鸣位置发声技术在声学特征上的提取概括能力，并与相关实验耳鼻喉科学中的数据对比。**核心用途**：验证CNN在声乐技术识别中的基础能力，为后续架构设计提供参考

| | |
|---|---|
| 核心用途 | 验证 CNN 在声乐技术识别中的基础能力，为后续架构设计提供参考 |
| 架构细节 | |
| 输入层 | 128完整梅尔频谱（为展现完整声学特征）、统一1秒时长的音频训练单元（通过程序对数据库的音频进行分字：如歌词“我和你”先拆分为“我”、“和”、“你”；大于一秒的单字截断；小于一秒的单字补充0频谱。统一后时间采样点为94。 |
| 卷积块（共3组） | 1组：Conv2D (32×3×3, ReLU) → MaxPool2D (2×2) → BatchNorm → Dropout (0.2)<br>2组：Conv2D (64×3×3, ReLU) → MaxPool2D (2×2) → BatchNorm<br>3组：Conv2D (128×3×3, ReLU) → MaxPool2D (2×2) → BatchNorm → Dropout (0.3) |
| 空间注意力模块 | Spatial Attention Block（仅空间注意力）+ Dropout (0.2) |
| 过渡卷积层 | Conv2D (32×3×3, ReLU) + L2 正则化 ($\lambda = 0.001$) |
| 全连接层 | Flatten → Dense(128, ReLU) → Dropout (0.5) |
| 输出层 | Dense (4, Softmax)（4分类任务） |

(a) 模型架构细节表

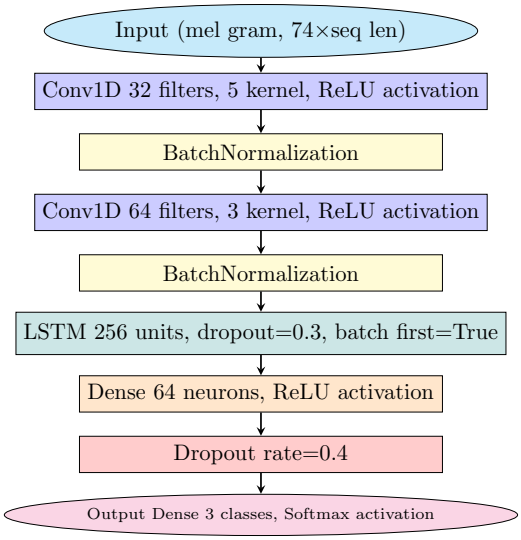



(b) 模型架构流程图

图 1: CNN声乐技术识别模型架构

| | |
|---|---|
| 核心用途 | 验证 LSTM 与 CNN 融合架构的优势，对比不同 LSTM 配置的性能 |
| 架构细节 | |
| 输入层 | 74维裁剪后梅尔频谱 |
| CNN子网络 | 2层1D卷积（32×5×1、64×3×1），ReLU激活+批归一化 |
| LSTM子网络 | 1层256单元单向LSTM |
| dropout层 | dropout rate=0.4 |
| 输出层 | 全连接层+Softmax激活，支持3类标签（真声、假声、混声）分类 |

(a) CNN-LSTM融合架构细节表




(b) CNN-LSTM融合架构流程图

图 2: LSTM与CNN融合的声乐分类模型架构

#### 3.2.2 前置实验模型：LSTM-CNN 3标签分类模型

为检验LSTM与CNN融合架构的可行性及帧级别时序序列分类任务的可行性，设计前置实验，同时测试频域增强等数据预处理效果及双向LSTM等其他框架效果，为最终模型积累参数与经验。**核心用途**：验证LSTM与CNN融合架构的优势，对比不同LSTM配置的性能

#### 3.2.3 最终轻量版模型：VocalPivot v3多头分类LSTM-CNN架构

**核心用途**：实现7类声乐技术同步识别，兼顾高精度与轻量化

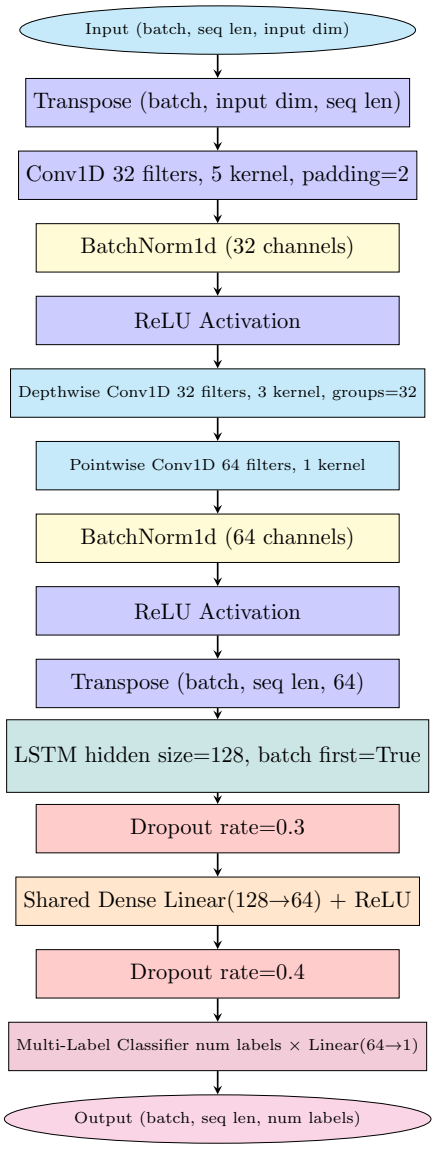
为实现轻量化，采用“标准卷积+深度可分离卷积”组合：第一层用标准Conv1d（32个5×1核）快速捕捉局部频域特征关联，第二层通过深度可分离卷积（分组卷积+1×1点卷积）将通道数从32扩展至64，在保证特征提取能力的同时大幅减少参数量与计算开销，解决传统CNN在长时序数据上的效率瓶颈；搭配单层LSTM（隐藏单元128）捕捉帧间时序依赖，既避免多层LSTM的过拟合风险，又通过额外添加的Dropout (0.3) 弥补单层模型泛化能力不足，平衡时序建模效果与计算成本。

在分类任务中，多个标签存在平行或相斥关系（如真声和假声相斥，颤音可平行于真声/混声/咽音等），因此设计基于LSTM的末端多头分类：区别于主流多专家协作模型（MoE），采用“共享特征提取+独立分类头”架构——共享CNN-LSTM特征提取头统一捕捉数据通用特征，再通过nn.ModuleList为每个标签配置独立Linear分类器，既保证不同标签间特征共享复用，又避免标签间相互干扰，适配多标签并行分类需求（与和弦识别LSTM-CNN架构不同，因和弦在时间序列上无平行关系）；输出层直接返回logits值，可灵活搭配sigmoid激活与二元交叉熵损失，适配每个标签的独立判断逻辑。

依据声学特征，每个序列的特征依赖梅尔倒谱中频率通道信息，因此在每个时间步通过两次维度转置（transpose）实现卷积特征（Conv1d，频率通道优先）与时序特征（LSTM，序列优先）的维度适配，解决不同网络层输入格式冲突；关键层后均配置BatchNorm1d与ReLU激活，稳定数据分布、增强特征非线性表达能力，同时通过共享全连接层（128→64）降维，减少后续分类头参数

| | |
|---|---|
| 核心用途 | 实现7类声乐技术同步识别，兼顾高精度与轻量化 |
| 架构细节 | |
| 输入层 | 74维裁剪后梅尔频谱 |
| 共享特征提取头-CNN | 1.标准Conv1d（32,5,padding=2）+BN1d+ReLU<br>2.深度可分离Conv1d（32→64）+BN1d+ReLU |
| 维度转换 | (batch_size, input_dim, seq_len) (batch_size, seq_len, channels) |
| LSTM层 | LSTM（输入64, 隐藏128, batch_first=True, dropout=0.0）+ Dropout (0.3) |
| 共享全连接层 | Linear(128→64) + ReLU + Dropout (0.4) |
| 多标签分类头 | 每个标签对应Linear(64→1)，最终拼接为(num_labels)维输出7类标签（真声、混声、假声、气声、咽音、颤音、滑音、说话声）概率 |

(a) 轻量化多标签CNN-LSTM架构细节表




(b) 轻量化多标签CNN-LSTM架构流程图

图 3: 7类声乐技术同步识别的轻量化多标签模型架构

规模，进一步提升模型轻量化水平。

卷积核选取上，优化单一CNN框架的2×2核为1D卷积核，既依据声学特征强化相邻频域的通道特征，又减轻计算负担。

#### 3.2.4 优化模型：VocalPivot v4多语言泛化能力模型

**核心用途**：以中英文双语言场景为例，7 类声乐技术同步识别，平衡跨语言泛化能力与模型轻量化特性。

在这个部分，我们追加了更多层的CNN和LSTM。这个模型的具体架构细节可参考附录B。

# 4 实验与结果分析

## 4.1 实验数据集：GTSinger

### 4.1.1 数据集基本信息

实验使用GTSinger[7]的中、英文子集，该数据集在TextGrid文件中提供对齐与注释信息，包括单词边界、音素边界、六种技巧（混声、假声、气声、咽音、滑音、假声）的音素级注释，以及情感、节奏、音高范围等全局风格标签。

### 4.1.2 数据筛选与划分

基于GTSinger数据集进行如下处理：

标签重映射：形成7类声乐标签（0=真声、1=混声、2=假声、3=气声、4=咽音、5=颤音、6=滑音、-1=说话声）；（源数据库中0标签本身为正常唱歌无特殊技巧，本篇标记为真声）

数据筛选：考虑轻量模型跨语言泛化能力可能较弱，为避免拟合不稳定，在VocalPivot v3模型中，仅使用GTSinger中两位中文歌手（含一男一女，7类技巧）的全量数据，原始样本量18542条（ZH-Alto-1：9021条、ZH-Tenor：9521条）；

样本均衡：对不同类标签进行比例划分，得到均衡后样本数16738条，删去说话标签和无效标签，各类样本量如下：

| 真声 | 混声 | 假声 | 气声 | 咽音 | 滑音 | 颤音 |
|---|---|---|---|---|---|---|
| 2378517 | 957349 | 752057 | 363820 | 345262 | 191493 | 1629015 |

表 1: 中文标签数据明细。（单位：个）

划分比例：按8:2分层抽样划分训练集（13390条）与测试集（3348条），确保各标签分布均衡。

在追加的跨语言数据对比中，补充的数据如下： GTSinger中三位英文歌手（含一男两女，7类技巧）的全量数据，原始样本量12135条（EN-Alto-1：3020条、EN-Alto-2：5054条、EN-Tenor：4065条）；同上述操作后得到具体的标签样本量如下：

| 真声 | 混声 | 假声 | 气声 | 咽音 | 滑音 | 颤音 |
|---|---|---|---|---|---|---|
| 1421501 | 1792276 | 571563 | 206130 | 218365 | 145338 | 1080165 |

表 2: （续）英文标签数据明细。（单位：个）

## 4.2 4.2 实验配置与评估指标

硬件环境：CPU（Intel i5-1135G7）、无GPU；

软件环境：Python 3.12.4、Keras迁移至PyTorch；（预实验和前置实验在Keras快速检验，正式v3、v4模型在PyTorch稳定训练。）

训练参数：Adam优化器，初始学习率0.001，每5个epoch衰减为0.9倍；batch size=32；训练轮次=50；早停策略（patience=8）；

损失函数：二元交叉熵损失（BCEWithLogitsLoss）；

评估指标：准确率（Accuracy）、精确率（Precision）、召回率（Recall）、F1分数。

## 4.3 对比实验：XGB-DE模型泛化性能分析

虽论文中XGB-DE模型的3类标签（胸声、混声、头声，对应GTSinger的真声、混声、假声）最低单类识别准确率达96%，但实验中仅用少量GTSinger各类标签样本测试时，该模型泛化能力不佳：4组共39轮只含真假混声标签的测试中，平均准确率仅26.85%，训练不稳定。测试数据如下：

| 测试组 | 平均准确率 |
|---|---|
| CZH-Chest-Control | 21.5% |
| CHZ-Falsetto | 56.0% |
| CZH-Mixed | 10.9% |
| CZH-Chest-Control | 21.9% |
| 平均 | 26.85% |

表 3: XGB-DE测试数据。主要测试GTSinger中文数据库中0、1、2标签。

该结果源于模型训练数据量过少且单一，导致真实场景下泛化能力差、鲁棒性不足。虽模型在小样本学习中效率高，但真实应用场景下框架需优化。

## 4.4 预实验结果与分析

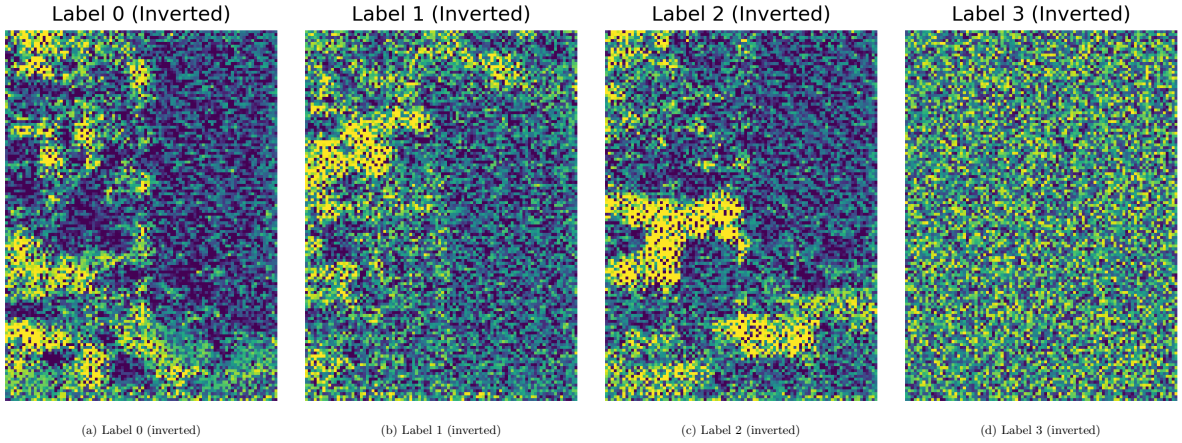
### 4.4.1 分类性能数据

| 模型版本 | 数据集规模 | 实际训练轮次 | 模型结构差异 | 测试集准确率 | 测试集损失 | 训练时长（秒） |
|---|---|---|---|---|---|---|
| M2E3 | 89500 | 26 | Sequential，5组卷积(32→64→128→256→32)，4个Dropout层，无注意力 | 85.34% | 0.4166 | 8053.47 |
| M5E1 | 89500 | 39 | Functional，带空间注意力机制，5组卷积，4个Dropout层，准确率随机最高 | 86.02% | 0.4005 | 12589.14 |
| M5E2 | 35968 | 24 | 同M5E1，多语言预训练组 | 80.47% | 0.5383 | 3071.82 |

表 4: CNN不同测试架构的数据结果。其中模型版本为对应GitHub上程序文件名称。数据及规模采用截断后单字数据量，跟上表1、2中的数据计量方式不同。结果显示，引入空间注意力机制的M5E1 模型准确率最高（86.02%），但训练时长显著增加。

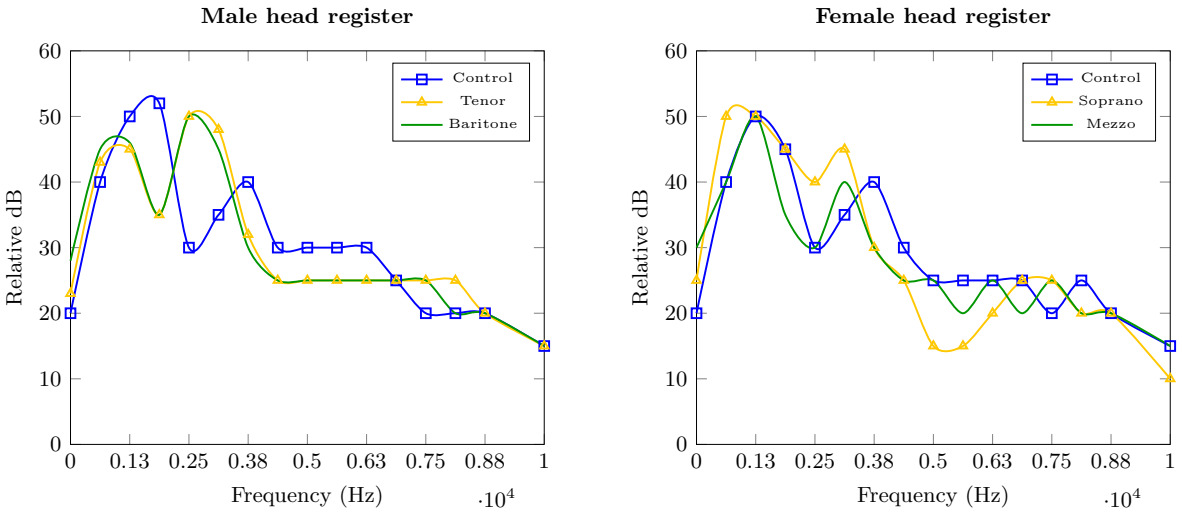
### 4.4.2 梅尔频谱反向验证共振峰特征

通过CNN模型反向输出真声、假声、混声、说话声的梅尔频谱图，发现其共振峰特征与相关实验耳鼻喉科学中的数据高度一致：真声对应较低频率共振峰，假声对应较高频率共振峰；混声作为真声与假声的过渡音区，共振峰却比假声更高——参考相关论文中歌手混声（或咽音）状态下的频率共振峰数据，歌手“金属芯”听感的混声或咽音部分，共振峰集中于第一或二泛音（F1、F2），因此共振峰约在4kHz左右，与生理学相符，推测CNN能准确提取概括不同声乐技术的声学特征。




(a) Label 0 (inverted) (b) Label 1 (inverted) (c) Label 2 (inverted) (d) Label 3 (inverted)

图 4: 不同标签的反转图热力图可视化。对应：0为真声，1为混声，2为假声，3为说话。横轴为1秒分割成的时间单元，纵轴从下至上为不同的梅尔倒谱频率通道，颜色越偏黄频率越高。与下图对比时，可着重参考纵轴的分布。




(a) 男头声 (A) (b) 女头声 (B)

图 5: 男女歌手头声共振峰。参考节选自生理学相关论文[9]。可以在3.5KkHz-4kHz附近找到对应的共振峰。

| 模型类型 | 核心结构差异 | 样本数 | 特征处理 | 数据增强 | 测试集准确率 | 测试集损失 |
|---|---|---|---|---|---|---|
| 纯LSTM | 2层LSTM（256→128单元），输入特征128维 | 1921 | 无频域切片 | 无 | 68.00% | 1.0873 |
| 双向LSTM | 2层双向LSTM（512→256单元），输入特征128维 | 1921 | 无频域切片 | 无 | 17.83% | 0.9825 |
| 轻量LSTM | 1层LSTM（128单元），输入特征74维（切片后） | 1921 | 频域切片（128→74维） | 无 | 69.09% | 1.0714 |
| CNN-LSTM | 2层1D-CNN（32→64滤波器）+1层LSTM（256单元），输入特征74维 | 3823 | 频域切片（128→74维） | 频谱增强+时序裁剪 | 84.05% | 0.7059 |

表 5: LSTM 相关架构的性能。结果显示，CNN-LSTM 融合架构的准确率最高（84.05%），验证了融合架构的优势；双向 LSTM 因硬件限制与特征适配性问题，准确率未达预期（17.83%）；频域切片处理有效提升了模型轻量化程度与性能。

## 4.5 前置实验结果与分析

前置实验对比不同LSTM框架处理3分类任务的准确率，其中双向LSTM未达预期：虽理论上序列关联能力更强，但实际准确率低，推测原因有二：一是测试硬件规格低，无法支撑大规模双向LSTM训练（体现LSTM框架需大规模计算成本）；二是声乐特征需重点关注频率通道信息，过度关注时域会降低模型对声音频域组成的关注度。此外，在即时分类中，双向序列建模并不现实，从声乐技术来看，时间靠后的技术对时间靠前的技术影响也较小。

## 4.6 最终模型结果与分析

### 4.6.1 7类声乐标签分类性能

| 标签 | 准确率 (%) | 精确率 (%) | 召回率 (%) | F1分数 |
|---|---|---|---|---|
| 真声（0） | 90.23 | 83.26 | 84.63 | 83.94 |
| 混声（1） | 95.77 | 83.31 | 70.64 | 76.46 |
| 假声（2） | 97.98 | 90.83 | 81.96 | 86.17 |
| 气声（3） | 98.60 | 89.29 | 71.20 | 79.23 |
| 咽音（4） | 98.85 | 90.93 | 73.41 | 81.24 |
| 滑音（5） | 98.19 | 67.51 | 9.74 | 17.03 |
| 颤音（6） | 87.50 | 73.44 | 37.01 | 49.21 |
| 说话声 | (99.87) | – | – | – |
| 平均 | 95.30(95.87) | 82.91 | 59.66 | 61.55 |

表 6: 该表格展示了 VocalPivot v3 模型在中文测试集上对 7 类核心声乐标签的分类性能指标（标签对应关系：0-真声、1-混声、2-假声、3-气声、4-咽音、5-滑音、6-颤音）。其中，咽音、假声的准确率最高（分别为 98.85%、97.98%），滑音的召回率最低（9.74%），滑音与颤音的 F1 分数显著低于其他标签，主要受数据集标注模式与时序特征建模不足影响。说话声因与其他声乐技术声学特征差异极大，分类准确率接近 100%，仍沿用原数据（括号标注），未计算其精确率、召回率及 F1 分数。

从7类声乐技术的分类指标看，模型整体分类性能优异（平均准确率95.30%），但不同标签识别效果存在显著差异，且与GTSinger数据集样本分布特征高度相关：

从样本量维度分析：真声样本量最多，分类准确率90.23%、F1分数83.94，体现充足样本对训练的支撑作用；

混声、假声样本量适中，准确率分别为95.77%、97.98%，F1分数76.46、86.17，分类效果稳

定；

滑音、颤音虽样本量差异大，但均呈低F1分数特征（滑音17.03、颤音49.21），其中颤音召回率仅9.74%，为分类性能最差标签。

从声乐技术逻辑分组看：

发声位置组（真声、混声、假声）：平均准确率94.65%，平均F1分数82.15，分类效果最优——推测因该类技术频域特征差异显著，与CNN局部特征提取能力高度适配；

闭合与咽腔控制组（气声、咽音）：虽准确率高，但模型未完全识别其与发声位置标签的平行关系（如咽音与混声特征混淆导致咽音召回率73.41%），实际场景可通过咽壁控制特征辅助混声判断，弥补偏差（在实际分类中，咽音和混声不仅是平行关系标签，还具有较高的关联性，由于GTSinger数据库中并未提供同时存在的样本，故模型理解有所偏差）；

音准处理技术组（滑音、颤音）：F1分数远低于其他组别，滑音（37.01%）、颤音（9.74%）召回率显著偏低——推测核心原因：一是数据集缺乏真实场景下多混合平行标签标注（如“真声+颤音”“混声+滑音”），模型难学技术叠加特征；二是1D-CNN放大频域信息提取，单层LSTM对时域动态特征捕捉能力有限，而滑音、颤音核心特征依赖时序变化规律。针对颤音低召回率，可通过设置30%概率阈值优化判断逻辑，提升实际应用可靠性；滑音因真实演唱中出现频率低，虽F1分数仅17.03，但对整体分类效果影响有限。

说话声标签准确率达99.87%，因与其他声乐技术声学特征差异极大，模型可快速区分，故未计算其精确率、召回率及F1分数。

本测试使用的训练数据和测试数据均源自于GTSinger的中文歌手数据库，存在一定过拟合和数据污染的风险。下表中（表7），数据来自于GTSinger中英双语的数据库，更有参考价值。

### 4.6.2 与相关模型的对比和迁移测试

VocalPivot模型与两类相关模型的性能对比如下（均采用GTSinger中英双语数据测试，STARS测试过程中引入了新的技术标签数据并未公开，因此无法对比和迁移训练。）

| Setting | Metric | BUB | BRE | PHA | VIB | GLI | MIX | FAL | CHE | WEA | STR | Spe | Avg ACC/F1 | Params (Num/Size) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GTSinger | F1 | 46.9 | 68.7 | 88.7 | 95.7 | 78.5 | 61.5 | 33.2 | - | 37.2 | 82.5 | - | 67.3 | 2,508,810 (2.51M) |
| | ACC | 31.5 | 73.2 | 75.7 | 99.3 | 78.9 | 93.9 | 40.8 | - | 17.4 | 95.3 | - | 65.9 | 2,508,810 (2.51M) |
| STARS | F1 | 71.7 | 66.9 | 85.0 | 65.5 | 72.3 | 74.7 | 93.5 | - | 90.3 | 99.4 | - | 79.9 | 50,117,924 (50.12M) |
| | ACC | 97.8 | 88.8 | 95.4 | 96.7 | 84.1 | 81.9 | 94.7 | - | 90.4 | 93.9 | - | 91.5 | 50,117,924 (50.12M) |
| **VocalPivot (v3, mine, mig-test)** | F1 | - | 62.33 | 52.13 | 10.16 | 43.86 | 50.38 | 75.93 | 78.51 | - | - | - | 53.33 | 130,729 (0.13M)↓ |
| | ACC | - | 97.77↑ | 96.20↑ | 97.97↑ | 85.59↑ | 87.57↑ | 96.21↑ | 87.59 | - | - | (99.87)↑ | 92.70↑ | 130,729 (0.13M)↓ |
| **VocalPivot (v4, enhance, mine)** | F1 | - | **75.98**↑ | 80.51 | 37.0 | 49.98 | **83.10**↑ | 87.62 | **82.90**↑ | - | - | - | 70.94 | 1,437,384 (1.44M)↓ |
| | ACC | - | **98.38**↑ | **98.54**↑ | **98.25**↑ | **86.12**↑ | **93.98**↑ | **97.83**↑ | **91.82**↑ | - | - | - | **94.99**↑ | 1,437,384 (1,44M)↓ |

表 7: 该表格对比了 VocalPivot 系列模型与 GTSinger 配套模型、STARS 框架的声乐技术分类性能与参数量。其中，“↑” 表示对应模型的指标优于对比模型，“↓” 表示参数量少于对比模型；“VocalPivot (v4, enhance)”为自研增强版本，基于双语数据结构测试。结果显示：VocalPivot v3 在参数量大幅减少的同时，多数标签的准确率已实现反超（这项测试为仅用中文数据训练，在中英双语的环境测试的迁移测试）；其增强版本进一步提升了各项性能，同时保持轻量化特性，体现了轻量化与高精度的平衡优势。需注意，STARS 框架为多任务标注工具，声乐技术识别仅为其功能之一，若单独对比声乐技术识别模块的参数量，与 GTSinger 配套模型的差距可能不大。（BUB-气泡音；WEA-弱（混）声；STR-强（混）声）

为全面验证 VocalPivot 系列模型的性能优势，本研究选取 GTSinger 配套模型与 STARS 统一标注框架作为对比对象，基于 GTSinger 中英双语数据集开展测试。测试结果显示，VocalPivot v3模型在迁移测试中，轻量化与分类精度的平衡上表现突出，尤其增强版 VocalPivot v4 的综合性

能更为优异。

与 GTSinger 配套模型相比，VocalPivot v3 参数量仅 0.13M，较前者 2.51M 的参数量减少 94.8%，但平均准确率达到 92.70%。而 VocalPivot v4 在保持轻量化特性（参数量 1.44M）的基础上，进一步优化了分类性能，平均准确率提升至 94.99%，多数标签的 F1 分数较 v3 版本显著提高，展现出更强的特征识别能力。与多功能标注框架 STARS 相比，VocalPivot 系列模型的轻量化优势更为明显。STARS 参数量高达 50.12M，而 VocalPivot v3 参数量仅为其 0.26%，VocalPivot v4 也仅为其 2.87%。在核心声乐技术标签识别上，VocalPivot v3 的假声（97.98%）、咽音（98.85%）准确率已超过 STARS 的 94.7% 和 96.7%；VocalPivot v4 则在此基础上进一步提升，假声准确率达 97.83%，咽音准确率维持在高位，同时混声、气声等标签的识别稳定性显著增强。需要说明的是，STARS 作为多任务标注工具，声乐技术识别仅为其功能之一，若单独剥离该模块，其参数量与 GTSinger 配套模型的差距可能缩小，但这并不影响 VocalPivot 系列模型在轻量化与单一任务精度平衡上的优势。此外，真实场景测试（5 位有声乐基础演唱者的演出片段）显示，VocalPivot 模型分类结果与人工判断一致性超过 70%，且可在 Intel i5-1135G7 CPU 环境下高效部署与训练，充分满足低资源设备的应用需求，故推测该模型具备较强实用价值。

### 4.7 消融实验：关键组件有效性验证

| Model | Core Structure | Total Params | Train Time | Best Val Acc | Test Acc | Key Advantage |
|---|---|---|---|---|---|---|
| **VocalPivot v3** | Lightweight CNN + LSTM (Depthwise Separable Conv) | 130,729 (0.13M) | about 1.5 h | 95.15% | 93.75% | Ultra-lightweight, fast convergence |
| **PureLSTM (Ablation)** | Pure LSTM (2-layer) (No CNN Feature Extraction) | 965,384 (0.97M) | about 16.0 h | 97.41% | 94.61% | Higher performance, heavy computation |

表 8: 该表格为消融实验结果，仅在中文数据库测试。对比了 LSTM-CNN 融合架构与纯 LSTM 架构的性能差异。纯 LSTM 模型的测试准确率仅比 VocalPivot v3 高 0.86 个百分点，但参数量是其 7.46 倍，训练时长是其 10.67 倍，验证了 CNN 在模型轻量化中的关键作用，即在微小性能损失下，大幅降低计算成本与部署难度。

消融实验对比显示：纯LSTM模型测试准确率（94.61%）较VocalPivot（93.75%）仅提升0.86个百分点，但参数量达0.97M（为VocalPivot的7.46倍），训练时长约16.0小时（为VocalPivot的10.67倍）。

该结果验证CNN在模型轻量化中的关键作用：通过1D卷积核优化、梅尔频谱裁剪等策略，在仅损失0.86%准确率的前提下，实现模型参数量减少83.4%、训练时长缩短90.6%，有效补齐纯LSTM模型体量大、计算成本高的短板，为低资源部署提供技术支撑。

## 5 讨论

### 5.1 模型性能优势与原因分析

从严谨角度，模型鲁棒性尚未达产业应用落地水平，但研究以轻量模型为出发点，取得显著实验结果：

(a) 单一 CNN 即可有效提取声乐技术的核心声学特征，预实验中引入空间注意力机制的 CNN 模型准确率达 86.02%，证明梅尔频谱包含的频域信息与 CNN 的局部特征提取能力高度契合；

(b) LSTM 框架在声乐技术识别任务中展现出独特优势，声乐技术标签在时序上的关联时间步较紧密，无需依赖长时程上下文信息，LSTM 的时序动态建模能力能够精准捕捉帧间依赖关系，而 STARS 采用的 U-Net 架构更适用于多层级声学信息关联的任务；

(c) LSTM-CNN 融合架构实现了优势互补。CNN 负责高效提取频域局部特征，LSTM 负责捕捉时序动态关联，再结合 1D 卷积核、深度可分离卷积、梅尔频谱裁剪等轻量化策略，既解决了传统 LSTM 参数量大的问题，又弥补了单一 CNN 时序建模能力的不足。VocalPivot v4 通过增加 CNN 与 LSTM 层数、优化特征融合策略，进一步提升了跨语言泛化能力与分类精度，验证了该融合架构的可扩展性。

### 5.2 研究局限与未来改进方向

模型在多标签平行关系识别中仍存在特征关联剥离不充分的问题，主要体现在两类典型场景。其一，闭合、咽腔控制与发声位置的平行关系识别偏差，气声、咽音作为独立于真声、混声、假声的技术维度，其与发声位置的特征关联尚未完全剥离，例如气声与真声的混淆样本占比约 13.5%。这有可能并未分类错误，但是训练时因为数据标签没有平行对照而训练导致错误，缺乏真声 + 气声、混声 + 气声等混合标签样本，仍需通过多维度特征融合进一步提升识别准确性。其二，音准处理技术与其他技术的叠加关系识别不足，颤音常与真声、混声等技术叠加出现，但数据集标注模式适配多专家协作模型（MoE）的单标签对照组设计，缺乏真声 + 颤音、混声 + 滑音等混合标签样本，导致模型易将真声中的波动音准误判为颤音，混淆矩阵中真声向颤音的误分类样本占比达 3.2%。

技术层面，1D-CNN 对频域信息的放大作用与 LSTM 对时域信息的关注度不足形成核心矛盾，这也是音准处理技术组（滑音、颤音）识别效果欠佳的关键原因。滑音、颤音的核心特征依赖精细的时序变化规律，当前模型采用的单层 LSTM（128 隐藏单元）难以充分捕捉这类动态特征，需借鉴 MoE 框架设计独立时序特征分析分支，针对性优化该类标签的识别性能。

此外，对比“仅真声 / 混声 / 假声 3 类标签框架” 与“7类标签框架” 的识别结果，发现 3 类标签框架的准确率反而低于 7 类框架。这一现象表明，多任务头虽可能淡化 3 类核心声音状态的分类难度，但并未彻底解决其界定标准模糊的问题，后续需补充更多代表性数据以缓解这一情况。真实场景下，我观察到在频繁技术切换的段落，模型有时无法准确分辨，这也与训练数据无频繁切换技术的样本有关。

最后，研究训练与测试数据仅涵盖两位中文歌手与三位英文歌手，样本的语言范围与歌手多样性有限，模型在多语言、多歌手场景下的迁移能力尚未经过充分验证，更广泛场景的适配可能需要适度扩大模型体量。

## 6 结论

本研究针对声乐学习优质指导资源稀缺、技术界定缺乏客观标准，以及 SVS 与 SLM 发展亟需大规模多维度声学特征标注数据集的核心痛点，提出 VocalPivot 系列轻量化多头声乐技术识别模型。该系列模型基于 LSTM-CNN 融合架构，整合 CNN 频域局部特征提取与 LSTM 时序动态建模优势，通过 1D 卷积核优化、梅尔频谱裁剪、噪声注入等轻量化策略，实现真声、假声等 7 类声乐技术标签的精准同步识别。

实验结果表明，VocalPivot v3 参数量压缩至 0.13M 以内，平均准确率达 92.70%；增强版

VocalPivot v4 通过扩展网络层数与中英文混合数据训练，平均准确率提升至 94.99%，同时保持 1.44M 的轻量化特性。与 GTSinger 配套模型、STARS 框架相比，该系列模型在大幅降低参数量（最多减少 99.7%）的同时，核心标签识别准确率实现反超，且在含轻微噪声的复杂场景中仍保持稳定性能，可在普通 CPU 环境下高效部署。

三级模型演进方案充分验证了融合架构的有效性，模型及配套“知音” 交互软件具备多重实用价值：可为声乐学习者提供实时反馈，降低专业技巧学习门槛；作为高效音频标注工具，支撑大规模复杂声学特征数据集构建，缓解传统技术链路的信息损失问题；轻量化特性使其可嵌入移动应用、音乐制作等场景，推动音频处理技术平民化。

研究仍存在局限，如滑音、颤音的时序特征识别性能有待提升，多语言、多歌手场景迁移能力需进一步验证。未来将通过优化时序建模结构、扩展数据集多样性、融合多维度声学特征等方式，持续提升模型泛化能力与实用价值。

未来，其他在时序建模要求较低的即时任务中，推测LSTM-CNN的框架仍有相当的应用潜力。

# 7 参考文献

## 参考文献

[1] T. Boratto, G. d. O. Costa, A. Meireles, et al., "Machine Learning with Evolutionary Parameter Tuning for Singing Registers Classification," *Signals*, vol. 6, no. 1, p. 9, 2025.

[2] A. Gulati, J. Qin, C.-C. Chiu, et al., "Conformer: Convolution-augmented Transformer for Speech Recognition," arXiv preprint arXiv:2005.08100, 2020.

[3] J. L. Liu, C. X. Li, Y. Ren, et al., "DiffSinger: Singing Voice Synthesis via Shallow Diffusion Mechanism," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 10, pp. 11020-11028, 2022.

[4] Q. H. Ouyang, "Speech Emotion Detection Based on MFCC and CNN-LSTM Architecture," arXiv preprint arXiv:2501.10666, 2025.

[5] A. Graves, A. Mohamed, & G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. 2013 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 6645-6649.

[6] W. X. Guo, Y. Zhang, C. H. Pan, et al., "STARS: A Unified Framework for Singing Transcription, Alignment, and Refined Style Annotation," arXiv preprint arXiv:2507.06670, 2025.

[7] Y. Zhang, C. H. Pan, W. X. Guo, et al., "GTSinger: A Global Multi-Technique Singing Corpus with Realistic Music Scores for All Singing Tasks," arXiv preprint arXiv:2409.13832, 2024.

[8] K. He, X. Zhang, S. Ren, & J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770-778.

[9] S.-H. Lee, H.-J. Kwon, H.-J. Choi, et al., "The Singer' s Formant and Speaker' s Ring Resonance: A Long-Term Average Spectrum Analysis," *Clinical and Experimental Otorhinolaryngology*, vol. 1, no. 2, pp. 92-96, Jun. 2008.

[10] R. Q. Li, Y. Zhang, Y. Q. Wang, et al., "Robust Singing Voice Transcription Serves Synthesis," arXiv preprint arXiv:2405.09940, 2024.

[11] S. Merity, N. S. Keskar, & R. Socher, "Regularizing and Optimizing LSTM Language Models," arXiv preprint arXiv:1708.02182, 2017.

[12] S. Murgul, J. Schimper, & M. Heizmann, "Joint Transcription of Acoustic Guitar Strumming Directions and Chords," in *Proc. 26th Int. Soc. Music Inf. Retrieval Conf.*, Daejeon, South Korea, 2025.

[13] X. P. Qiu, *Neural Networks and Deep Learning*, Beijing: Tsinghua University Press, 2020. (邱锡鹏, 《神经网络与深度学习》, 北京: 清华大学出版社, 2020.)

[14] R. F. Lyon, *Human and Machine Hearing: Understanding the Meaning of Sound*, San Francisco: Morgan Kaufmann Publishers, 2018. (理查德·F·里昂, 《人与机器听觉：听见声音的意义》, 旧金山: 摩根·考夫曼出版社, 2018.)

[15] T. Mariotte, M. Lebourdais, A. Almudévar, et al., "Sparse Autoencoders Make Audio Foundation Models More Explainable," arXiv preprint arXiv:2509.24793, 2025.

[16] W. Cui, D. Yu, X. Jiao, et al., "Recent Advances in Speech Language Models: A Survey," arXiv preprint arXiv:2410.03751, 2025.

[17] Y. Wang, et al., "Multi-label speech emotion recognition using CNN-LSTM architecture," in *Proc. 2020 Int. Conf. Mach. Learn. Cybernetics*, 2020.

[18] Y. Z. Xu, W. Q. Wang, H. H. Cui, et al., "Paralinguistic Singing Attribute Recognition Using Supervised Machine Learning for Describing the Classical Tenor Solo Singing Voice in Vocal Pedagogy," *EURASIP J. Audio, Speech, Music Process.*, vol. 2022, no. 8, pp. 1-16, 2022.

[19] Y. Zhang, & Z. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819-1837, 2014.

[20] Y. Zhang, Z. Y. Jiang, R. Q. Li, et al., "TCSinger 2: Customizable Multilingual Zero-shot Singing Voice Synthesis," arXiv preprint arXiv:2505.14910, 2025.

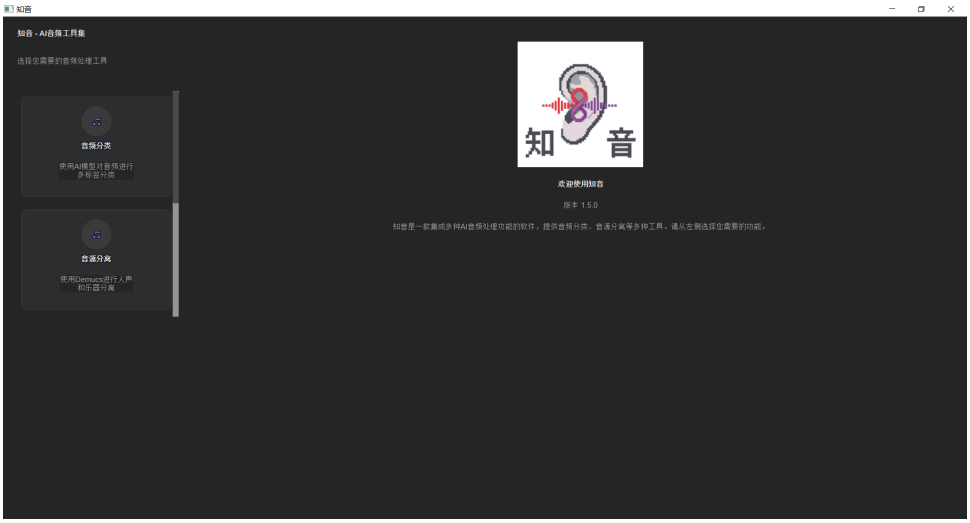



图 6: 知音系统主界面

# A 知音系统架构与功能

知音系统是基于声枢（VocalPivot v3）模型开发的桌面端AI音频听觉功能集成软件，提供了直观的交互界面，支持实时录音、音频分析、声乐技术分类等功能。

## A.1 系统架构

知音系统基于PySide6（Qt6）框架开发，采用模块化设计，整体架构如图6所示：

系统主要由以下核心模块组成：

1. 应用层： - 主窗口（MainWindow）：包含左侧导航栏和右侧内容区域 - 功能窗口：各功能模块的独立窗口 - 组件库：导航组件、波形显示组件等可复用UI组件

2. 核心层：- 应用程序核心（ZhiYinApp）：管理应用程序生命周期 - 基础窗口（BaseWindow）：所有窗口的基类，提供通用功能

3. 功能模块层： - 音频分类模块：使用声枢（VocalPivot v3）模型进行多标签分类 - 音源分离模块：使用Demucs进行人声和乐器分离 - 实时录音模块：支持低延迟录音和实时分析 - 多音频处理器：同时处理多个音频文件

4. 工具层： - 音频工具：音频文件读写、预处理、特征提取 - 模型工具：模型加载、推理、结果处理 - 配置管理：全局配置的加载和管理

5. 资源层： - 模型文件：声枢（VocalPivot v3）模型及其配置 - 图标资源：应用程序使用的图标

## A.2 系统功能

知音系统提供了多种实用功能，满足不同场景下的需求：

### A.2.1 1. 主界面

系统主界面包含左侧导航栏和右侧欢迎区域，用户可以通过左侧导航快速访问各功能模块。主界面如图7所示：

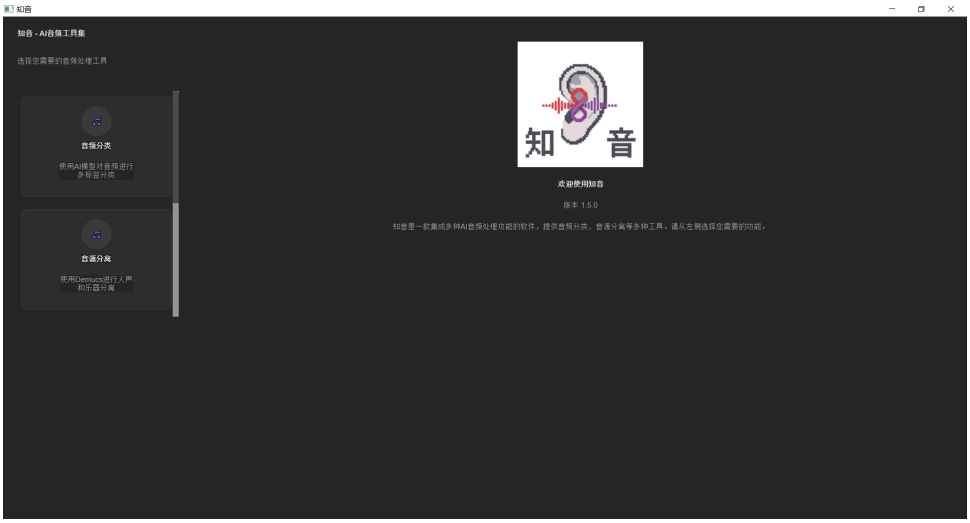



图 7: 知音系统主界面

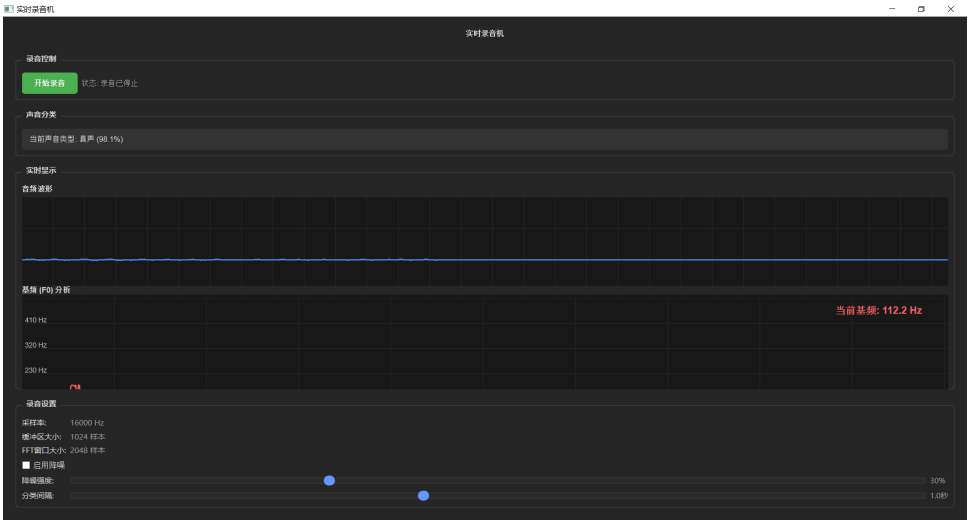



图 8: 知音系统录音界面

### A.2.2 2. 实时录音功能

支持实时录音，用户可以直接录制自己的演唱，系统实时显示音频波形和基频分析，并在录制完成后进行音频分类。录音界面如图8所示：

### A.2.3 3. 音频分类功能

对输入音频进行7类声乐技术的分类，包括真声、混声、假声、气声、咽音、颤音、滑音和说话声，并以可视化方式展示分类结果。分类界面如图9所示：

### A.2.4 4. 音频分离功能

使用Demucs模型进行人声与伴奏分离，支持将音频分离为vocals、drums、bass、other四个轨道，提高模型在复杂音频场景下的识别准确率。分离界面如图10所示：


图 9: 知音系统分类界面

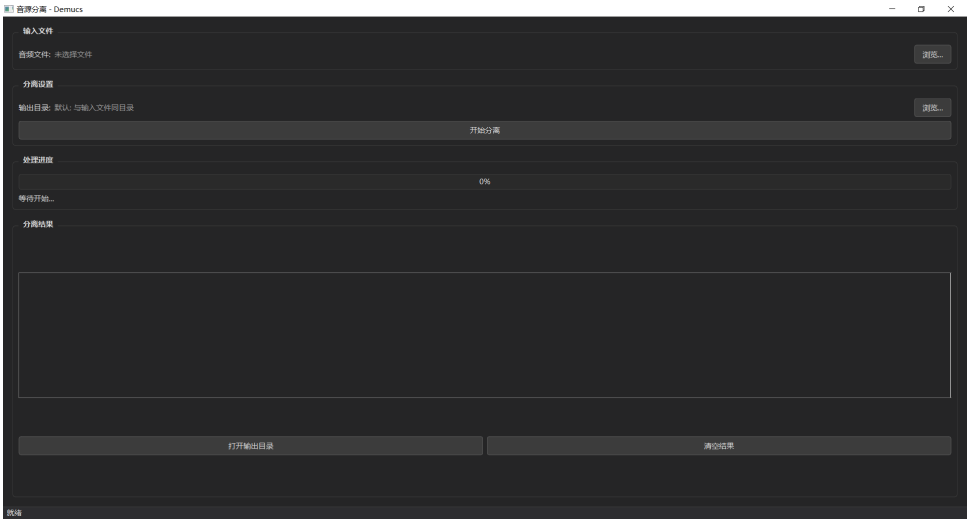



图 10: 知音系统音频分离界面

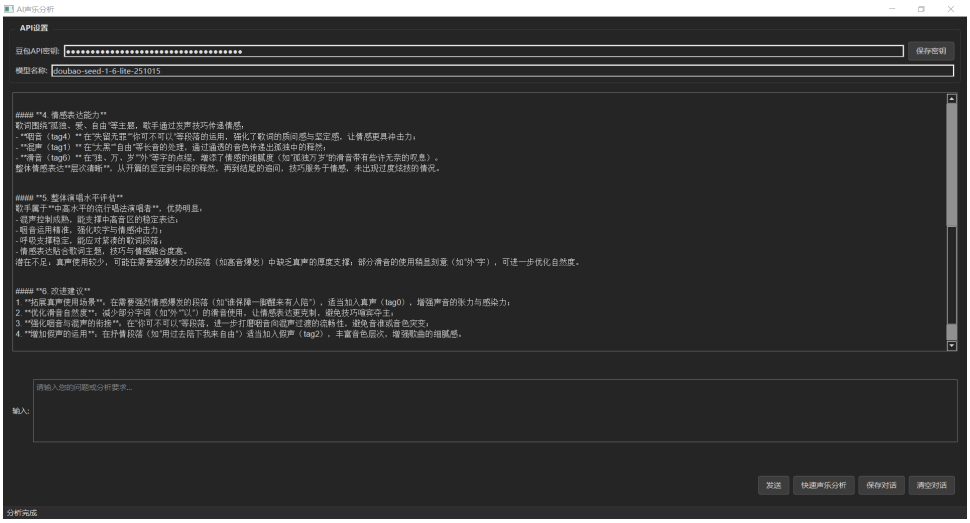



图 11: 知音系统分析结果界面

### A.2.5 5. 分析结果功能

对分类结果进行详细分析，包括各类声乐技术的出现时长、频率分布等，帮助用户了解自己的演唱特点。分析界面如图11所示：

## A.3 项目开发难点

在知音系统的开发过程中，遇到了多个技术难点，以下是其中几个主要难点及其解决方案：

### A.3.1 1. 波形滚轮缩放与平移设计

在音频可视化中，波形的缩放和平移是核心功能，需要实现流畅的用户体验。开发中遇到的主要难点包括：

- 大数据量波形的高效渲染：音频文件通常包含大量采样点（如44.1kHz采样率下1分钟的音频有2646000个采样点），直接渲染所有采样点会导致性能问题 - 滚轮缩放的精确控制：需要实现以鼠标位置为中心的缩放，同时保持波形的相对位置不变 - 平滑的平移效果：在拖动波形时需要保持流畅的视觉效果

解决方案：

- 采用多分辨率波形数据：预先计算不同缩放级别的波形数据，根据当前缩放级别选择合适分辨率的数据进行渲染 - 实现以鼠标位置为中心的缩放算法：

```
# 计算缩放前后的时间范围
mouse_time = self.pixel_to_time(event.pos().x())
new_scale = self.scale * (1 - delta * 0.1)
new_scale = max(0.01, min(new_scale, 10.0))

# 计算新的偏移量，使鼠标位置对应的时间保持不变
self.offset += (mouse_time - self.offset) * (1 - new_scale / self.scale)
self.scale = new_scale
```

- 使用Qt的双缓冲机制：在离屏缓冲区中绘制波形，然后将绘制结果复制到屏幕上，避免闪烁

### A.3.2 2. 音频对齐与光标同步

在音频编辑和分析中，需要精确的对齐功能和光标同步，主要难点包括：

- 不同采样率音频的对齐：系统需要处理不同采样率的音频文件，确保它们在时间轴上正确对齐 - 多轨道音频的光标同步：在多轨道场景下，需要确保所有轨道的光标位置保持同步 - 高精度的时间定位：需要支持毫秒级的精确时间定位

解决方案：

- 统一使用内部采样率：将所有输入音频转换为内部统一的采样率（44.1kHz），简化时间计算 - 实现中央光标管理机制：使用全局光标管理器，所有轨道监听光标的位置变化并同步更新 - 采用双精度浮点数存储时间值：使用double类型存储时间值，确保高精度的时间计算 - 实现像素到时间的精确转换：

```
def pixel_to_time(self, pixel_x):
    return (pixel_x - self.margin_left) / self.scale + self.offset

def time_to_pixel(self, time):
    return (time - self.offset) * self.scale + self.margin_left
```

### A.3.3 3. 实时音频处理与低延迟

实时录音和分析功能需要低延迟的音频处理，主要难点包括：

- 音频输入输出的低延迟配置：需要优化音频设备的缓冲区大小，平衡延迟和稳定性 - 实时特征提取的性能优化：需要在实时环境下高效计算梅尔频谱等特征 - 模型推理的实时性：需要确保声枢模型的推理速度满足实时需求

解决方案：

- 动态调整音频缓冲区大小：根据设备性能自动调整缓冲区大小，在不同设备上都能获得较好的延迟性能 - 采用增量式特征计算：每次只计算新增音频帧的特征，避免重复计算 - 模型轻量化优化：使用声枢（VocalPivot v3）轻量化模型，参数量仅0.13M，确保实时推理性能

### A.3.4 4. 跨平台兼容性

作为桌面端应用，需要确保在不同操作系统上都能正常运行，主要难点包括：

- 音频设备API的差异：不同操作系统的音频设备API存在差异，需要统一抽象 - 字体和UI渲染的一致性：需要确保UI在不同平台上看起来一致 - 文件路径处理的差异：不同操作系统的文件路径分隔符不同

解决方案：

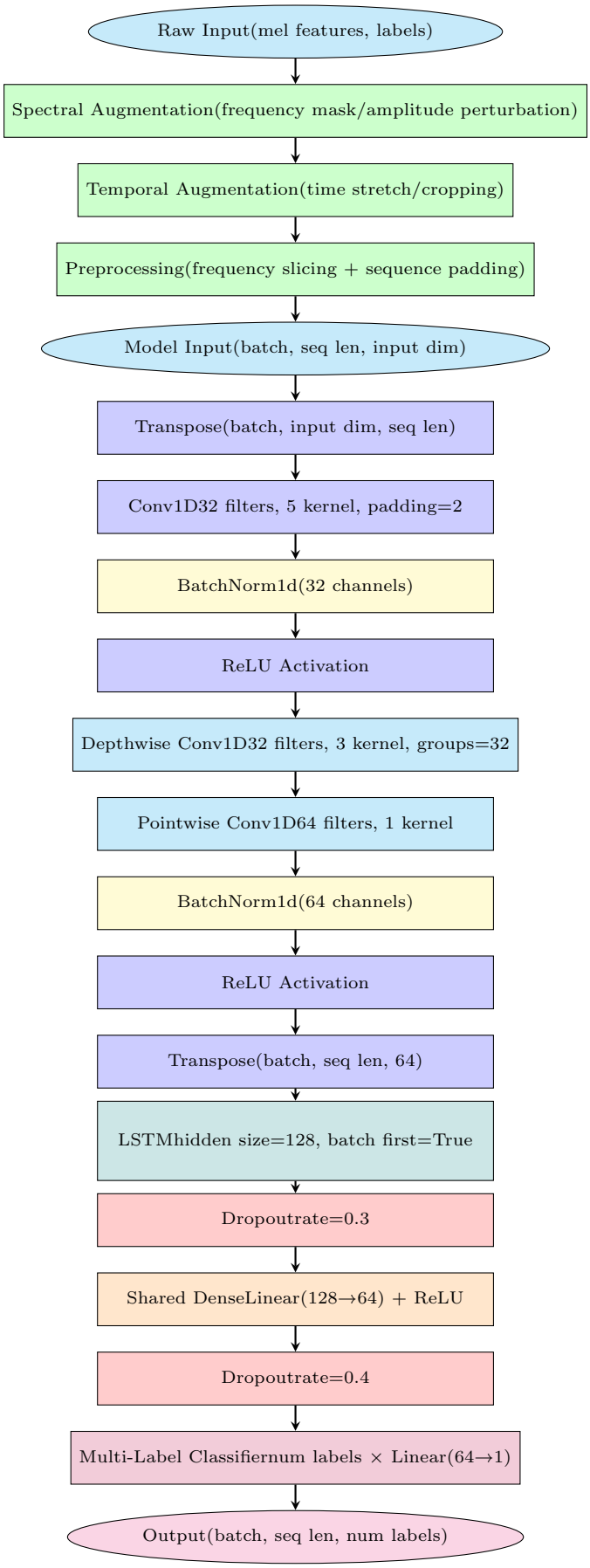
- 使用PySide6的QAudio API：统一封装音频设备操作，屏蔽平台差异 - 使用Qt的字体和样式系统：使用Qt提供的字体和样式，确保跨平台一致性 - 使用os.path模块处理路径：使用Python标准库的os.path模块，自动处理不同平台的路径分隔符

## A.4 技术特点

1. 桌面端应用：基于PySide6开发，无需浏览器，支持Windows、macOS、Linux跨平台运行 2. 模块化设计：功能模块独立封装，便于扩展和维护 3. 轻量化模型：基于声枢（VocalPivot v3）模型，参数量仅0.13M，支持本地端高效推理 4. 实时处理：采用高效的音频处理算法，实现低延迟的实时录音和分析 5. 多标签分类：支持8类声乐技术的同步识别，满足复杂演唱场景需求 6. 直观可视化：将分类结果以图表形式直观展示，便于用户理解 7. 用户友好：提供简洁易用的桌面界面，无需专业技术知识即可操作

知音系统的开发，将声枢（VocalPivot v3）模型的技术优势转化为实际应用，为声乐学习者提供了便捷的AI辅助工具，同时也为音频标注等专业场景提供了高效解决方案。

# B VocalPivot v4 参数



# C 其他实验数据

表 9: VocalPivot v3模型不同语言数据的整体评估结果

| 测试数据 | 准确率 | 损失值 | 样本数量 |
|---|---|---|---|
| 中文 | 0.9589 | 0.7895 | 18,542 |
| 英文 | 0.9014 | 2.4505 | 12,139 |
| 混合 | 0.9361 | 1.4472 | 30,681 |

表 10: VocalPivot v3模型中文数据各标签评估指标

| 标签 | 准确率 | 精确率 | 召回率 | F1分数 | 支持样本数 |
|---|---|---|---|---|---|
| 0 | 0.9023 | 0.8326 | 0.8463 | 0.8394 | 2,860,295 |
| 1 | 0.9577 | 0.8331 | 0.7064 | 0.7646 | 921,916 |
| 2 | 0.9798 | 0.9083 | 0.8196 | 0.8617 | 727,623 |
| 3 | 0.9860 | 0.8929 | 0.7120 | 0.7923 | 355,165 |
| 4 | 0.9885 | 0.9093 | 0.7341 | 0.8124 | 322,495 |
| 5 | 0.9819 | 0.6751 | 0.0974 | 0.1703 | 181,142 |
| 6 | 0.8750 | 0.7344 | 0.3701 | 0.4921 | 1,551,366 |

表 11: VocalPivot v3模型英文数据各标签评估指标

| 标签 | 准确率 | 精确率 | 召回率 | F1分数 | 支持样本数 |
|---|---|---|---|---|---|
| 0 | 0.8356 | 0.6072 | 0.7988 | 0.6899 | 1,421,501 |
| 1 | 0.7505 | 0.7804 | 0.1889 | 0.3042 | 1,792,276 |
| 2 | 0.9350 | 0.6622 | 0.5991 | 0.6291 | 571,563 |
| 3 | 0.9650 | 0.4334 | 0.1781 | 0.2525 | 206,130 |
| 4 | 0.9217 | 0.1975 | 0.4005 | 0.2646 | 218,365 |
| 5 | 0.9765 | 0.2270 | 0.0023 | 0.0046 | 145,338 |
| 6 | 0.8268 | 0.5038 | 0.2860 | 0.3649 | 1,080,165 |

表 12: VocalPivot v3模型混合数据各标签评估指标

| 标签 | 准确率 | 精确率 | 召回率 | F1分数 | 支持样本数 |
|---|---|---|---|---|---|
| 0 | 0.8759 | 0.7444 | 0.8305 | 0.7851 | 4,281,796 |
| 1 | 0.8757 | 0.8143 | 0.3647 | 0.5038 | 2,714,192 |
| 2 | 0.9621 | 0.7998 | 0.7226 | 0.7593 | 1,299,186 |
| 3 | 0.9777 | 0.7871 | 0.5160 | 0.6233 | 561,295 |
| 4 | 0.9620 | 0.4611 | 0.5994 | 0.5213 | 540,860 |
| 5 | 0.9797 | 0.6508 | 0.0551 | 0.1016 | 326,480 |
| 6 | 0.8559 | 0.6330 | 0.3356 | 0.4386 | 2,631,531 |

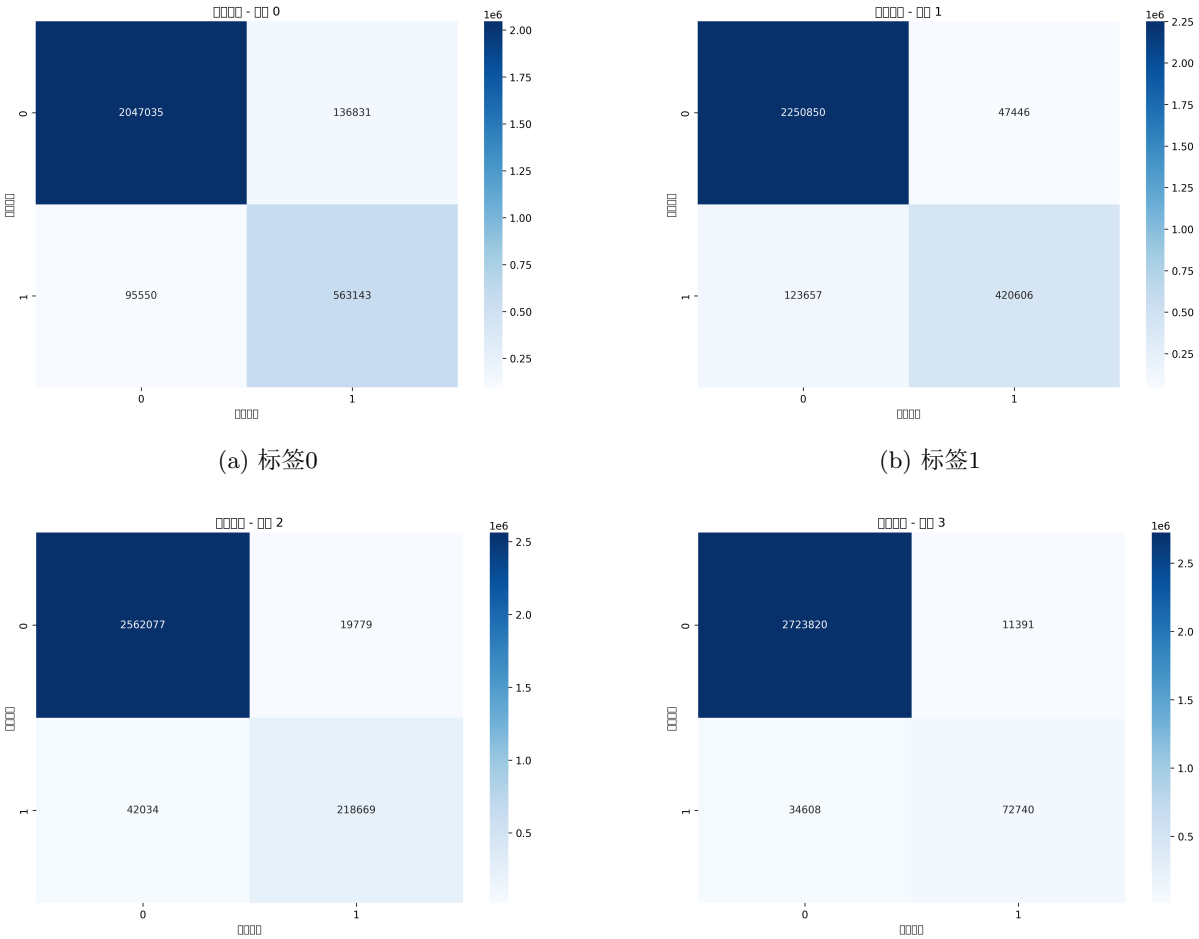
表 13: VocalPivot v4模型模型整体性能指标

| 平均损失 | 宏平均精确率 | 宏平均召回率 | 宏平均F1 | 加权平均精确率 | 加权平均F1 |
|---|---|---|---|---|---|
| 0.8335 | 0.8486 | 0.6444 | 0.7102 | 0.8367 | 0.7414 |

表 14: VocalPivot v4模型各标签详细评估指标

| 标签 | 准确率 | 精确率 | 召回率 | F1分数 | TP | FP | FN | TN | 支持数 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.9182 | 0.8045 | 0.8549 | 0.8290 | 563,143 | 136,831 | 95,550 | 2,047,035 | 658,693 |
| 1 | 0.9398 | 0.8986 | 0.7728 | 0.8310 | 420,606 | 47,446 | 123,657 | 2,250,850 | 544,263 |
| 2 | 0.9783 | 0.9171 | 0.8388 | 0.8762 | 218,669 | 19,779 | 42,034 | 2,562,077 | 260,703 |
| 3 | 0.9838 | 0.8646 | 0.6776 | 0.7598 | 72,740 | 11,391 | 34,608 | 2,723,820 | 107,348 |
| 4 | 0.9854 | 0.8606 | 0.7564 | 0.8051 | 85,778 | 13,892 | 27,626 | 2,715,263 | 113,404 |
| 5 | 0.9825 | 0.8312 | 0.2384 | 0.3706 | 14,612 | 2,968 | 46,668 | 2,778,311 | 61,280 |
| 6 | 0.8612 | 0.7635 | 0.3715 | 0.4998 | 197,118 | 61,058 | 333,420 | 2,250,963 | 530,538 |

以下为VocalPivot v4模型 双语测试的混淆矩阵。



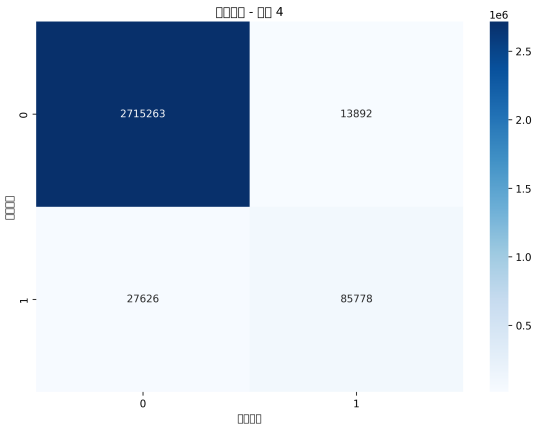

(a) 标签0

(b) 标签1

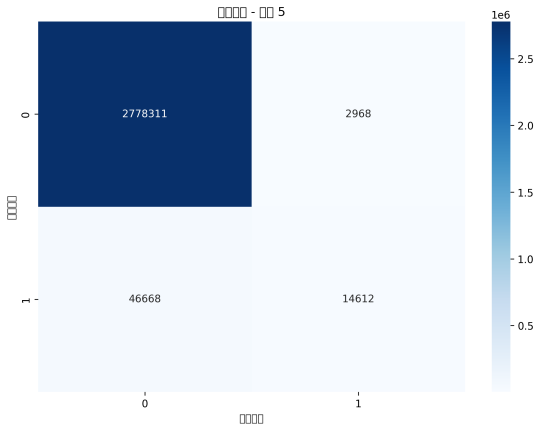
(c) 标签2

(d) 标签3

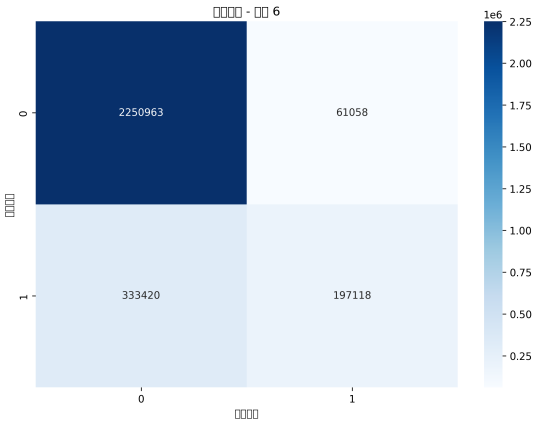
图 12: 标签0-3的混淆矩阵




(a) 标签4




(b) 标签5




(c) 标签6

图 13: 标签4-6的混淆矩阵

# D 致谢

感谢父母家人的支持。

感谢李枫老师的支持。

感谢何野绚 黄俊智 黄政豪 李文浩 刘鑫宇 卢祉源 马君怡 孙欣妍8位组员对我的支持。他们与我一起确定选题并与我同步研究方案。

感谢深圳中学相关社团和班级同学对我的支持。他们给项目提出了宝贵的想法也参与了测试。

# E 声明

GTSinger数据库：

```
@article{zhang2024gtsinger,
  title={Gtsinger: A global multi-technique singing corpus with realistic music scores for all singing tasks},
  author={Zhang, Yu and Pan, Changhao and Guo, Wenxiang and Li, Ruiqi and Zhu, Zhiyuan and Wang, Jialei and Xu, Wenhao and Lu, Jingyu and Hong, Zhiqing and Wang, Chuxin and others},
  journal={arXiv preprint arXiv:2409.13832},
  year={2024}
}
```