arXiv:2302.00286v2 [cs.SD] 2 Feb 2023

JOINTIST：通过联合训练同步提升多乐器转录与音乐源分离

Kin Wai Cheuk[1*] **Keunwoo Choi**[2] **Qiuqiang Kong**[3] **Bochen Li**[3] **Minz Won**[3] **Ju-Chiang Wang**[3] **Yun-Ning Hung**[3] **Dorien Herremans**[1]

[1]新加坡科技设计大学，新加坡[2]Gaudio Lab，韩国[3]字节跳动，美国

摘要

本文介绍了 Jointist，这是一种乐器感知型多乐器框架，能够对音频片段中的多种乐器进行转录、识别和分离。Jointist 由一个乐器识别模块构成，该模块为另外两个模块提供条件信息：转录模块输出特定乐器钢琴卷帘，声源分离模块则利用乐器信息和转录结果。对转录模块和声源分离模块进行联合训练，可提升这两项任务的性能。乐器模块是可选的，也可由人类用户直接控制。因此，Jointist 是一种灵活的用户可控框架。

我们提出的这一具有挑战性的问题定义，使模型在现实世界中极具实用价值，因为现代流行音乐通常由多种乐器构成。然而，这一创新也要求我们以新的视角来评估此类模型。在实验中，我们从多个方面评估了所提出的模型，为多乐器转录提供了新的评估视角。我们的主观听音研究表明，Jointist 在流行音乐上的表现达到了最先进水平，优于 MT3 等现有多乐器转录模型。我们在若干下游任务上开展了实验，发现使用 Jointist 获得的转录结果后，所提出的方法使转录性能提升了 1 个百分点以上（ppt.），音源分离的信源失真比（SDR）提高了 5，强拍检测提升了 1.8 个百分点，和弦识别提升了 1.4 个百分点，调性估计提升了 1.4 个百分点。
演示请访问 https://jointist.github.io/Demo。

1 引言

转录，或自动音乐转录（AMT），是一项音乐分析任务，旨在使用乐谱或 MIDI（乐器数字接口）文件等符号记谱来表示音频录音（Benetos等人，2013；2018；Piszczalski与Galler，1977）。自动音乐转录在音乐信息检索系统中可以发挥重要作用，因为符号信息——例如音符的音高、时值和力度——决定了我们音乐感知的很大一部分。成功的自动音乐转录应以具有音乐意义的符号格式提供去噪后的音乐版本，从而降低许多音乐信息检索任务的难度，例如旋律提取（Ozcan等人，2005）、和弦识别（Wu与Li，2018）、节拍跟踪（Vogl等人，2017）、作曲家分类（Kong等人，2020a）、（Kim等人，2020）以及情感分类（Chou等人，2021）。此外，高质量的自动音乐转录系统还可用于构建大规模数据集，正如Kong等人（2020b）所做的那样。反过来，这可以加速基于神经网络的音乐信息检索系统的发展，因为这类系统通常使用其他情况下难以获得的音频对齐符号数据进行训练（Brunner等人，2018；Wu等人，2020a；Hawthorne 等人，2018）。目前，唯一可用的流行音乐数据集是Manilow

*在字节跳动实习期间完成

等人（2019）构建的Slakh2100 数据集。缺少面向流行音乐的大规模音频对齐符号数据集，阻碍了其他使用符号化音乐表示进行训练的音乐信息检索系统的发展。

在自动音乐转录系统早期的研究中，问题通常被狭义地定义为转录单一目标乐器，通常是钢琴（Klapuri & Eronen, 1998）或鼓（Paulus & Klapuri, 2003），且输入音频仅包含该乐器。这一强假设在当时不可避免，但其局限性显而易见：模型无法用于现代流行音乐，而流行音乐占人们收听的音乐的大多数。换言之，要应对自动音乐转录的实际应用场景，就必须开发多乐器转录系统。近期的例子包括 Omnizart（Wu等人，2021年；2020b）和 MT3（Gardner 等，2021），我们将在第 2 节中讨论。

然而，多乐器转录的发展才刚刚起步。这类系统的开发和评估仍面临若干挑战。尤其是，多乐器音频录音中的乐器数量并不固定。一首流行歌曲中的乐器数量可能从几种到十几种不等。因此，让模型为每首乐曲转录预先设定的固定数量的乐器，会限制其能力。相比之下，能够适应目标乐器数量变化的模型会更加稳健、实用。这表明，在开发和评估过程中，我们可能需要考虑乐器识别以及乐器特定行为。

受上述近期趋势和现存问题的启发，我们提出了 Jointist——一个包含乐器识别、声源分离以及转录的框架。我们采用联合训练方案，以最大限度地提升转录和声源分离模块的性能。实验结果证明了将转录模型作为音乐信息检索系统预处理模块的实用性。该结果进一步支持这样一种观点：转录结果是一种符号表示，有别于典型的 i）基于音频的表示（频谱图）或 ii）高层特征（Choi 等，2017；Castellon 等，2021）。

本文的组织结构如下。我们首先在第 2 节简要概述自动音乐转录的背景。随后在第 3 节介绍我们的框架 Jointist。第 4 节介绍实验细节，第 5 节讨论实验结果。我们还在第 6 节探讨 Jointist 生成的钢琴卷帘在其他音乐信息检索任务中的潜在应用。最后，在第 7 节总结全文。

2 背景

尽管钢琴音乐的自动音乐转录（AMT）模型已经发展得较为成熟，并且能够达到较高的准确率（Benetos等人，2018年；Sigtia等人，2015年；Kim和Bello，2019年；Kelz等人，2019年；Hawthorne 等（2017年；2018年）；Kong 等人，2021年），但多乐器自动音乐转录（MIAMT）领域的研究相对较少。MusicNet（Thickstun等人，2016年；2017年）和Re-conVAT（Cheuk等人，2021a）是能够转录钢琴以外乐器的MIAMT系统，但它们的输出是包含单个通道中所有乐器音符的扁平化钢琴卷帘。换言之，它们不具备乐器感知能力。Omnizart（Wu 等，2019；2021年）具备乐器感知能力，但正如第5.3节所讨论的那样，随着乐器数量增加，其扩展性较差。MT3（Gardner 等（2021））目前是最先进的多乐器自动音乐转录模型。它将自动音乐转录建模为序列预测任务，其中序列由表示音符的令牌组成。通过采用名为T5（Raffel等人，2020年）的自然语言处理（NLP）模型结构，MT3表明，基于Transformer的序列到序列架构能够通过从多个乐器的多个数据集进行学习，实现成功的转录。

尽管已有研究尝试联合训练语音分离与识别模型（Shi 等人，2022年），但在音乐信息检索领域，将转录模型与音源分离模型联合训练仍十分有限。例如，Jansson 等人（2019年）利用基频估计来引导歌声分离。不过，他们仅在单声部演唱音轨上展示了该方法。本文中的 Jointist 框架将这一思路扩展到了多声部音乐。Manilow 等人（2020年）将联合转录与音源分离训练扩展到了多声部音乐，但其模型最多只能处理五种声源（钢琴、吉他、贝斯、鼓和弦乐），无法涵盖现实世界流行音乐的多样性。我们提出的方法在39种乐器上进行训练和评估，足以应对现实世界的流行音乐。Hung 等人（2021年）和 Chen 等人（2021a）也对少数几种乐器采用了联合转录与音源分离训练

。不过，他们只在训练期间将转录作为辅助任务，而在推理阶段不执行转录。另一方面，Tanaka 等（2020年）的模型只能进行转录。该模型采用频谱图与音高图联合聚类方法，以提高多乐器转录准确率。Lin 等（2021）提出了一种零样本转录与分离模型，但该模型仅在13种古典乐器上进行训练和评估。我们提出的 Jointist 框架解决了上述大多数问题。与现有模型不同，Jointist 具备乐器感知能力，只转录并分离输入混合音频中出现的乐器。

3 JOINTIST

我们设计了所提出的框架 Jointist，通过联合训练音乐转录和音源分离模块，提升这两项任务的性能。通过加入乐器识别模块，我们的框架具备乐器感知能力，并且兼容多达 39 种不同乐器，如附录中的表 11 所示。

如图 1 所示，Jointist 由三个模块组成：乐器识别模块 f_{IR}（图 2）、转录模块 f_{T} （图 3）以及音乐源分离模块 f_{MSS}（图 4）。 f_{IR} 和 f_{T} 使用相同的梅尔频谱图 X_{mel} 作为输入，而 f_{MSS} 使用短时傅里叶变换频谱图 X_{STFT} 作为输入。

Jointist 是一个灵活的框架，可以端到端训练，也可以单独训练。为便于评估，我们对 f_{IR} 进行单独训练。请注意，即使没有 f_{IR}，我们的框架仍然可以运行。由于 f_{T} 和 f_{MSS} 都接受独热向量 I^i_{cond} 作为乐器条件，用户可以轻松绕过f_{IR}，手动指定要转录的目标乐器，从而使 Jointist 成为一个可由人控制的框架。在第 5.2–5.3 节中，我们展示了 f_{T} 与 f_{MSS} 的联合训练能够使两个模块相互受益。在下文的讨论中，我们将带有 f_{IR} 模块的 Jointist 称为**端到端 Jointist**；不带 f_{IR} 模块的称为**可控 Jointist**。

3.1 乐器识别模块

附录中的图2展示了乐器识别模块$f_{\text{IR}}(X_{\text{mel}})$的模型架构。该架构的设计受 Won 等（2021a）提出的音乐标签 Transformer 启发。变换器网络（Vaswani 等人，2017年）已被证明适用于广泛的音乐信息检索任务（Chen 等人，2021b；Lu 等人，2021年；Huang 等人，2018年；Park 等人，2019年；Won 等人，2019年；Guo 等人，2022年；Won 等（2021a））。因此，我们将乐器识别模型设计为由卷积神经网络（CNN）前端和变换器后端构成。为防止过拟合，我们在每个卷积块后应用 Dropout，丢弃率为 0。2（Srivastava 等人，2014年）。我们开展了表1所示的研究，以确定最优的 Transformer 层数；最终确定为 4。乐器识别损失定义为 $L_{\text{IR}} = \text{BCE}(Y_{\text{cond}}, \hat{Y}_{\text{cond}})$，其中 BCE 表示二元交叉熵， $\hat{Y}_{\text{cond}}$ 是预测乐器的向量， Y_{cond} 是真实标签。随后，预测的 $\hat{Y}_{\text{cond}}$ 将被转换为 $\hat{I}^i_{\text{cond}}$，供 f_{T} 和 f_{MSS}后续使用。

3.2 转录模块

图 3 展示了转录模块的模型架构， $f_{\text{T}}(X_{\text{mel}}, \hat{I}^i_{\text{cond}})$。该模块改编自 Hawthorne 等人（2017）提出的 Onsets and Frames 模型。按照 Meseguer-Brocal 和 Peeters（2019）为音源分离提出的方法，该模块使用 FiLM（Perez 等人，2018）以独热乐器向量 $\hat{I}^i_{\text{cond}}$ 作为条件，从而能够控制要转录的目标乐器。为确保模型稳定性，采用教师强制训练，即使用真实值 I^i_{cond} 作为条件。推理时，我们基于 Kong 等人（2021）提出的思路，结合起始音卷帘 Y_{onset} 和帧卷帘 Y_{frame}，生成最终的转录结果 Y_{roll}。在此过程中，使用 $\hat{Y}_{\text{onset}}$ 过滤掉有噪声的 $\hat{Y}_{\text{frame}}$，从而得到更干净的 $\hat{Y}_{\text{roll}}$。请注意，$\hat{I}^i_{\text{cond}}$ 既可以由 f_{IR} 模块提供，也可以由人类用户提供。转录损失定义为 $L_{\text{T}} = \sum_{\text{j}} L_j$，其中 $L_i = \text{BCE}(\hat{Y}_j, Y_j)$ 是二元交叉熵， $j \in \{\text{onset}, \text{frame}\}$。



图1：所提出的 Jointist 框架。该框架可转录/分离多达39种不同的乐器，具体定义见附录中的表 11。 B：批次大小， L：音频长度， C：乐器类别， T：时间步数， K：预测乐器数量。虚线表示重复执行 K 次的迭代操作。建议以彩色查看。

3.3 声源分离模块

图 4 展示了声源分离模块的模型架构， $f_{\text{MSS}}(X_{\text{STFT}}, \hat{I}^i_{\text{cond}}, \hat{Y}^i_{\text{frame}})$，其设计受到 Jansson 等人（2017）以及 Meseguer-Brocal & Peeters（2019）的启发。除了 $\hat{I}^i_{\text{cond}}$（由 f_{IR} 生成或由人类用户提供）之外，我们的 f_{MSS} 还使用音乐的 $\hat{Y}^i_{\text{frame}}$。声源分离损失 $L_{\text{MSS}} = \text{L2}(\hat{Y}^i_S, Y^i_S)$ 定义为预测的声源波形 $\hat{Y}^i_S$ 与真实值声源波形 Y^i_S之间的 L2 损失。将转录输出 $\hat{Y}^{i\text{帧}} = f_{\text{T}}(X_{\text{mel}}, \hat{I}^i_{\text{cond}})$ 与 X_{STFT}结合时，我们探索了两种不同模式：求和 $g(\hat{Y}^{i\text{帧}}) + X_{\text{STFT}}$和拼接 $g(\hat{Y}^{i\text{帧}}) \oplus X_{\text{STFT}}$。其中， g 是一个线性层，可将 88 个 MIDI 音高 $\hat{Y}^{i\text{帧}}$映射到与 X_{STFT}相同的维度。关于这两种不同模式的消融研究见附录中的图9。

4 实验

4.1 数据集

Jointist 使用 Slakh2100 数据集（Manilow 等人，2019年）进行训练。该数据集由 Lakh 数据集（Raffel，2016年）的一部分合成而来：使用高质量的采样合成器渲染 MIDI 文件，采样率为 44.1 kHz。训练集、验证集和测试集分别包含1500、225和375首乐曲。Slakh2100 数据集的总时长为145小时。该数据集中每首乐曲的音轨数为4至48条，中位数为9条，因此适用于多乐器自动音乐转录。该数据集使用了167种不同的插件来渲染音频录音。

每个插件还分配有一个 MIDI 编号，因此我们可以利用此信息，将 167 种不同的插件映射到附录表 11 中定义的 39 种不同乐器。我们根据乐器类型映射 MIDI 编号。例如，尽管 MIDI 编号 0-3 表示不同类型的钢琴，我们仍将它们映射到钢琴。最后，原始 MIDI 编号仅涵盖 0-127 个通道。我们增加一个额外的通道（128）来表示鼓，以便模型也能转录这种乐器。

我们还进行了主观评估，以衡量转录输出的感知质量，因为已有研究表明，较高的客观指标不一定意味着更好的感知质量（Simonetta 等人，2022年；Luo 等人，2020年）。考虑到编曲和作曲复杂度的多样性，我们从 4 个公开数据集 Isophonics（Mauch 等人，2009年）、RWC-POP（Goto 等人，2002年）、JayChou29（Deng & Kwok，2017年）和 USPOP2002（Berenzweig 等人，2004年）中选取了 20 首完整的流行歌曲。20 首歌曲及参与者的元数据列于附录 C 的表8和图5中。

4.2 训练

训练 Jointist 框架的各个模块时，首先将音频录音重采样至 16 千赫兹，这一采样率足以捕捉钢琴最高音高（4,186 赫兹）的基频以及部分泛音 C_8（Hawthorne 等人，2017年；Kong 等人，2021年）。按照输入特征的一些惯例（Hawthorne 等人，2017年；Kong 等人，2021年；Cheuk 等人，2021a），

对于乐器识别和转录，我们使用对数梅尔频谱图，窗口大小为 2,048 个采样点，跳跃长度为 160 个采样点，并采用 229 个梅尔滤波器组。对于音源分离，我们使用短时傅里叶变换，窗口大小为 1,024，跳跃长度为 160。此配置生成的频谱图每秒包含 100 帧。受内存限制，我们随机抽取一段 10 秒的混合音频来训练模型。我们使用学习率为 0.001 的 Adam优化器（Kingma与Ba，2014年）训练 f_{IR}和 f_{T}。对于 f_{MSS}，经过初步实验后，选择了学习率 1×10^{-4}。三个模块均在两块 32GB Tesla V100 图形处理器上进行训练，每块的批次大小为六。所有实验及音频处理均使用 PyTorch 和 Torchaudio（Paszke等人，2019年；Yang等人，2022年）完成。

The 总体训练目标 L 是三个模块损失的总和，即 $L = L_{\text{IR}}+L_{\text{T}}+L_{\text{MSS}}$.

4.3 评估配置

4.3.1 客观评估

各项指标的详细说明见附录 B 节。图 8 中使用 Sigmoid输出 Y_{cond} 计算 $\hat{}f_{\text{IR}}$ 的平均精度均值。在实际应用中，我们需要对 Y_{cond} 应用一个阈值，以获得可供 f_{T} 和 f_{MSS} 用作条件的独热向量。因此，我们还在表1中报告了阈值为 0.5 时的 F1 分数。我们选择该阈值是为了简化实验，尽管可以针对每种乐器调整阈值，以进一步优化指标（Won等人，2021b）。为了了解模型在不同阈值下的性能，我们还在表1中报告了平均精度均值（mAP）。

对于 f_{T}，我们提出了一种新的按乐器划分的指标（图9和图10），以更好地衡量模型在多乐器转录任务中的性能。现有文献大多采用整体指标或按乐曲评估（Hawthorne 等人，2017年；Gardner 等，2021；Kong 等人，2021；Cheuk 等，2021a）。虽然这能大致反映转录效果，但无法说明模型在哪些乐器上表现特别好或特别差。由于逐帧指标不能反映感知转录准确度（Cheuk 等，2021b），我们仅在表2中报告逐音符（N.）和带偏移量的逐音符（N&O）指标。我们研究了端到端 Jointist 和人工可控 Jointist 的性能。

对于 f_{MSS}，我们还报告了按乐器划分的指标，以便更好地了解模型在各乐器上的性能（图13和表4）。

4.3.2 主观评估

如引言中所述，成功的自动音乐转录应能提供清晰且符合音乐逻辑的音乐符号表示。这意味着，在转录准确的情况下，由转录所得 MIDI 文件合成的音频应与原始录音相似。因此，我们请参与者使用数字音频工作站（DAW），最好是在 Mac OS 上使用 GarageBand，聆听由默认软件乐器合成的音频，并查看乐曲中每种乐器对应的 MIDI 音符显示。

我们重点比较 Jointist 与 MT3 在主观评估中的表现。为了生成 20 首选定歌曲的测试 MIDI 文件，我们使用端到端 Jointist，并按照 (Gardner 等，2021)[1] 复现了 MT3 模型。我们提出了四个评估方面（详细说明见附录 B.4）：**乐器完整性**、**乐器音符连续性**、**整体音符准确度**和**整体聆听体验**。参与者在 DAW 中导入一首乐曲的原始音频，并搭配两份匿名 MIDI 转录之一（由 Jointist 或 MT3 生成），以确保信号同步，从而能够来回对照音频与 MIDI。随后，参与者按照附录 B.5 中的说明，根据上述方面对两份转录按 1 到 5 分进行评分。我们要求参与者多关注重要乐器，例如鼓、贝斯、吉他和钢琴，因为这些乐器在塑造歌曲的流派、节奏、和声和旋律感知方面发挥着更重要的作用。

[1] https://github.com/magenta/mt3/blob/main/mt3/colab/music transcription with transformers.ipynb

	mAP		F1	
#层数（#参数）	宏平均	加权平均	宏平均	加权平均
1（78.0M）	71.8	91.6	61.5	85.4
2（78.8M）	72.1	92.1	65.0	86.2
3（79.6M）	73.7	92.2	63.7	86.9
4（80.3M）	**77.4**	**92.6**	**70.3**	**87.6**

表1：不同 Transformer 层数下的乐器识别准确率。

5 结果

5.1 乐器识别

表1展示了使用不同数量的 Transformer编码器层时，乐器识别模块 f_{IR} 的平均精度均值和F1分数。随着层数增加，平均精度均值和 F1分数均有所提升。使用四层 Transformer编码器层时，平均精度均值和 F1分数达到最高。由于 Slakh2100 数据集中的乐器类别不平衡，对于训练样本不足的乐器类别（如单簧管、小提琴或竖琴），我们的 f_{IR} 表现相对较差，如附录 E 中的图 8 所示。因此，加权F1/平均精度均值高于宏平均F1/平均精度均值。尽管如此，我们认为，我们的 f_{IR} 已足以为鼓、贝斯和钢琴等常见乐器生成可靠的乐器条件。

请注意，人类用户可以直接向 Jointist 提供独热乐器条件，从而轻松绕过此模块。即使人类用户不想接管 f_{IR}，Jointist 仍可通过激活全部 39 种乐器条件来运行。因此， f_{IR} 是该框架中的可选模块，因此我们仅对其进行最低限度的评估。

5.2 转录

表2展示了不同模型的逐音符（N.）以及带偏移量的逐音符（N&O）转录F1分数。除了仅训练 f_{T}（T）之外，我们还探索了联合训练f_{T} 和 f_{MSS} （TS）的可能性，并研究其对转录准确率的影响。

评估转录模块 f_{T}时，可能会出现 f_{IR} 预测了混音中不存在的乐器的情况，从而导致该乐器的F1分数未定义。鉴于人类用户可以绕过 f_{IR}，我们假设真实乐器标签 I^i_{cond} 就是人类用户想要转录的目标乐器。

将 f_{T} 独立训练 1,000 个训练轮次后，我们可以达到 23.2 的逐乐器F1分数。我们原以为，联合训练 f_{T} 和 f_{MSS} 会带来更好的表现，因为这些模块能够相互促进。令人惊讶的是，从头开始联合训练 f_{T} 和 f_{MSS}，得到的逐乐器F1分数反而更低，仅为 15.8。只有先使用预训练的 f_{T} 训练 500 个训练轮次，再继续联合训练 f_{T} 和 f_{MSS}，才能获得更高的 F1 分数 24.7。我们推测这一现象有两个原因。首先，我们认为这是因为联合训练早期，由 f_{T} 生成的噪声输出 $\hat{} Y^i_{\text{frame}}$ 会使 f_{MSS} 混乱。 f_{MSS} 的信源失真比（SDR）也呈现出相同的规律，相关讨论见第5.3节。其次，多任务训练的表现往往不如针对单一任务训练模型，这在一定程度上是由于难以平衡多个目标。我们的模型可能就是这种情况的一个例子。

比较条件策略时，求和模式与拼接模式之间的差异非常细微。就分段 F1 分数而言，求和模式仅比拼接模式高 0.1；而在扁平 F1 和乐器级 F1 方面，拼接模式则比求和模式高 0.001。我们认为，模型已经学会利用钢琴卷帘来增强梅尔频谱图。因此，将线性投影后的钢琴卷帘与梅尔频谱图相加，便足以实现这一目标，所以使用拼接模式时并未观察到明显提升。

与现有方法相比，就乐器级 F1而言，我们的模型（24.8）大幅优于 Omnizart（Wu 等，2019）（1.9）。附录 E 中图9-12的乐器级转录分析表明，Jointist 能更好地处理严重偏斜的数据分布

	扁平 F1		片段 F1		实例 F1	
模型	N.	N&O	N	N&O	N	N&O
Wu 等（2019）	26.6	13.4	11.5	6.30	4.30	1.90
Gardner 等（2021）	76.0	57.0	N.A.	N.A.	N.A.	N.A.
T	57.9	25.1	59.7	27.2	48.7	23.2
iT	58.0	25.4	N.A	N.A	N.A	N.A
pTS(s)	58.4	26.2	61.2	28.5	50.6	24.7
ipTS(s)	58.4	**26.5**	N.A	N.A	N.A	N.A
TS(s)	47.9	18.6	45.7	18.4	34.9	15.8
pTS(c)	**58.4**	26.3	**61.3**	**28.6**	**50.8**	**24.8**
TS(c)	47.7	18.6	46.0	18.0	35.5	16.2

表2：现有最先进模型与我们提出的 Jointist 的转录准确率。“T”和“S”分别表示经过训练的 f_{T} 和 f_{MSS} 模块。前缀“p-”表示转录模块经过预训练，“i-”表示使用 f_{IR} 来获取乐器条件；无前缀的模型表示由人类用户完全控制，其目标乐器与真实乐器完全一致。(s)和(c)分别表示将钢琴卷帘求和或拼接后输入频谱图。

	乐器完整性	音符连续性	总体音符准确率	总体乐器表现
MT3	2.66 ± 1.04	2.64 ± 1.04	2.76 ± 1.05	2.60 ± 1.06
Jointist	$\mathbf{2.87} \pm 0.96$	$\mathbf{2.92} \pm 1.03$	$\mathbf{2.93} \pm 1.02$	$\mathbf{2.91} \pm 1.03$
p-值	0.004	0.0001	0.02	2.7e-05

表3：基于20名参与者的平均主观评分（1至5分），每位参与者均对20首歌曲进行评分。底行呈现了MT3与Jointist之间比较的T检验 p值。

（图6）。Omnizart 仅在鼓和贝斯等标签充足的乐器上表现良好，而随着乐器标签可用性的降低，Jointist 仍能保持有竞争力的转录准确率。这一结果证明，采用控制机制的模型（Jointist）比输出乐器数量固定的模型（Omnizart）具有更好的可扩展性。

尽管我们提出的框架（58.4）在逐音符 F1 分数上不及 MT3 模型（76.0），但在主观评估的所有方面，Jointist 均优于 MT3（见表3）。我们注意到，MT3 虽然能够捕捉古典音乐中清晰明确的音符，但在音色复杂的流行音乐中表现不佳。有时，MT3 中的语言模型会压过声学模型，生成一连串录音中并不存在的意外转录音符（示例请见我们的演示页面[1]）。MT3 在延续乐器音符方面也不够稳定：录音中的一串音符在转录结果中可能会在不相关的乐器之间来回跳转，导致主观评估中“音符连续性”的感知评分明显偏低。Jointist 在这些方面比 MT3 更为稳健。此外，Jointist 比 MT3 更灵活，人类用户可以选择自己感兴趣的目标乐器。例如，如果只指定“钢琴”这一条件，Jointist 就只会转录钢琴；而 MT3 则会转录音频片段中找到的所有乐器。

为了测试所提出框架的鲁棒性，我们比较了f_{IR} 生成的 I^i_{cond} （表2中以“i-”为前缀）与人工指定条件下（假设目标乐器是用户想要转录的乐器）的扁平转录结果。两种方法得到的扁平F1分数相近，这表明 f_{IR} 足以挑选出转录所需的所有乐器。由于假阳性钢琴卷帘和假阴性钢琴卷帘的 F1 分数未定义，我们无法报告f_{IR} 生成的 I^i_{cond}的逐乐曲和逐乐器 F1 分数。Jointist 生成的端到端转录样本可在线获取[2]。

[2]https://jointist.github.io/Demo/

模型	乐器	乐曲	声源
S	1.52	3.24	3.03
TS	1.86	3.55	3.32
pTS	**2.01**	**3.72**	**3.50**
TS 上限	4.06	4.96	4.81

表4：不同模型的信源失真比（SDR）。T 表示转录器，S 表示分离器。“p-”前缀表示预训练模型。最后一行展示了使用真实钢琴卷帘进行音源分离时的上限。

5.3 音源分离

表 4 显示，当 f_{MSS} (TS) 与 f_{T}联合训练时，其表现优于独立训练的f_{MSS} (S)。与第 5.2 节中的讨论类似，从头开始训练 TS 并不能取得最佳结果，因为训练初期含噪转录输出会干扰 f_{MSS} 模块。可以看出，使用预训练的 f_{T}时， f_{MSS} 能够取得更高的信源失真比（pTS）。我们在第 5.2 节和第 5.3 节中的实验结果均表明，联合训练 f_{T} 和 f_{MSS}有助于两个模块摆脱局部极小值，因此，与独立训练相比，能够取得更好的性能。完整的按乐器划分的信源失真比分析见附录 E 的图 13，Jointist 生成的分离音频样本可在线获取[1]。

频谱图 X_{STFT} 和后验图 $\hat{Y}^i_{\text{frame}}$ 可以通过“sum”（s）或“cat”（c）进行组合。不过，前者与后者在乐器/乐曲/声源信源失真比方面仅有细微差异（2.01/3.72/3.50 dB 与 1.92/3.75/3.52 dB）。除了 $\hat{Y}^i_{\text{frame}}$，二值钢琴卷帘 $\hat{Y}^{i\text{卷帘}}$也可以作为我们 f_{MSS}的条件，但这并未带来更好的结果（附录 F 中的表 9）。我们还探究了使用真实值 Y^i_{T} 时音源分离性能的上限（表 4 最后一行）。当 f_{MSS} 能够获得准确的转录结果时，其信源失真比会大幅提升。这表明，自动音乐转录是一项重要的音乐信息检索任务，可能使音乐源分离等其他下游任务受益。在下一节中，我们将探讨 Jointist 生成的转录输出的应用。

6 JOINTIST 的应用

6.1 弱拍、和弦与调性估计

直观来看，符号信息有助于拍点与强拍追踪、和弦估计和调性估计。已有研究指出，音符的时间信息与节拍密切相关（Matthew E. P. Davies，2021年）。另一方面，强拍对应小节边界，且通常伴随着和声变化（Durand et al., 2016）。钢琴卷帘中包含的音高信息，能够明确提供音乐调性与和弦的信息（Pauws, 2004; Humphrey & Bello, 2015）。因此，由于强拍与和声变化相关，这些信息也为强拍建模提供了有力线索。基于这些发现，我们尝试应用所提出的混合表示，以提升这些任务的性能。实验设置和所用数据集的详细说明见附录G。
表5展示了各项任务的评估结果。我们报告和弦任务的逐帧大调/小调评分、强拍跟踪的F1分数，以及歌曲级准确率。我们发现，采用混合表示的模型在全部三项任务中都能稳定优于仅使用频谱图的模型。这得益于钢琴卷帘的优势：它能向具备频率感知能力（即沿频率轴不具备平移不变性）的 SpecTNT 提供明确的节奏和和声信息。

6.2 音乐分类

我们进行了实验，以检验 Jointist 生成的钢琴卷帘是否也有助于音乐分类。
MagnaTagATune 数据集（Law等人，2009年）是自动音乐标签研究中广泛使用的基准。我们按照 Won等人（2020b）的先前工作，使用了标签频次排名前 50 的 $\approx$21k 条音轨。

评估数据	输入	弱拍 (F1分数)	和弦 (大调/小调)	调性 (Acc)
Isophonics	仅音频	72.8	79.8	72.8
	混合	**74.6**	**81.2**	**75.2**
RWC-POP	仅音频	62.0	76.6	-
	混合	**66.3**	**78.5**	-

表 5：在两个数据集上各项任务中使用和不使用钢琴卷帘表示的对比。

输入	模型	ROC-AUC	PR-AUC
音频 Only	Pons & Serra (2019)	91.06	44.93
	Lee 等人（2017）	90.58	44.22
	Won 等人（2020b）	91.29	46.14
	Won 等人（2020a）	91.27	46.11
仅钢琴卷帘	Transformer	89.38	40.63
混合	Transformer	90.90	44.00

表 6：MagnaTagATune 数据集上的音乐标签结果。仅使用音频的基线系统包括 MusiCNN（Pons & Serra (2019)）、样本级 CNN（Lee 等人（2017））、短片段 ResNet（Won 等人（2020b））和谐波 CNN（Won 等人（2020a））。

由于使用符号数据进行音乐分类这一领域鲜有探索，我们设计了一种基于音乐标签 Transformer（Won等人，2021a）的新架构。我们的钢琴卷帘输入尺寸为 (B, 39, 2913, 88)，其中 B 为批次大小，39 为乐器数量，2913 为时间步数量，88 为 MIDI 音符区间数量。CNN 前端包含 3 个带有残差连接的卷积块，后端 Transformer 与音乐标签 Transformer（Won等人，2021a）完全相同。我们报告受试者工作特征曲线下面积（ROC-AUC）和精确率-召回率曲线下面积（PR-AUC）作为评估指标。

如表 6 所示，同时使用音频和钢琴卷帘特征的混合模型表现不及纯音频模型。因此，我们进一步测试了仅将钢琴卷帘作为输入是否足以进行音乐分类。令人惊讶的是，仅使用钢琴卷帘也取得了有竞争力的结果。这为将音乐分类应用于纯符号音乐数据集（或使用 Jointist 等框架将音频数据集转换为符号数据）开辟了新的可能性。将符号特征作为额外模态引入分类器模型，也让我们能够发现新的方面。实验过程中，我们观察到以下现象：

• 对于仅使用钢琴卷帘的方法，当乐器标签属于 Jointist 明确训练过的 39 种乐器之一（例如大提琴、小提琴、西塔琴），或流派与 Slakh2100 数据集高度相关（例如摇滚、电子舞曲）时，其模型性能与仅使用音频的方法相当。

• 当标签与这 39 种乐器无关（例如女声、男声；相比之下，吉他、钢琴与其相关），或标签涉及无法通过钢琴卷帘捕捉的声学特征（例如安静）时，仅使用钢琴卷帘的模型性能会下降。

Whi尽管这些结论很有趣，但这项工作仍属探索性质，深入探索留待 ft as f未来工作。

7 结论

本文介绍了 Jointist，这是一个用于乐器识别、转录和音源分离的灵活框架。在该框架中，人类用户可以轻松指定自己感兴趣的目标乐器，使 Jointist 仅转录和分离这些目标乐器。Jointist 采用联合训练，使转录和音源分离的性能相互受益。在大量实验中，我们证明，Jointist 的表现优于现有的多乐器自动音乐转录模型。在听音研究中，我们确认，Jointist 生成的转录结果与 MT3 等最先进模型（Hawthorne 等人，2021年）不相上下。随后，我们探索了该框架的不同实际应用场景，并表明，使用所提出框架生成的转录结果有助于提升模型在弱拍跟踪、和弦识别和调性估计等任务中的性能。

未来，Jointist 还可以进一步改进，例如像 Hawthorne 等人（2021年）那样，用 Transformer 替换卷积循环神经网络。我们的实验表明，将符号表示用于标注等任务具有很大潜力。因此，我们希望看到更多尝试使用符号表示来补充甚至取代音频表示，从而推动构建更完善的多任务音乐分析系统。

参考文献

Emmanouil Benetos、Simon Dixon、Dimitrios Giannoulis、HolgerKirchhoff 和 AnssiKlapuri。自动音乐转录：挑战与未来方向。智能信息系统期刊，41(3)：407–434，2013年。

EmmanouilBenetos、SimonDixon、Zhiyao Duan 和 Sebastian Ewert。自动音乐转录：概述。*IEEE* 信号处理杂志，36(1)：20–30，2018年。

Adam Berenzweig、Beth Logan、Daniel PW Ellis 和 Brian Whitman。声学与主观音乐相似度度量的大规模评估。计算机音乐期刊，第 63–76 页，2004年。

S. B¨ock 和 M. EP Davies。解构、分析、重构：如何改进速度、节拍和强拍估计。载于第 *21* 届国际音乐信息检索学会会议，*ISMIR 2020*，2020年。

Gino Brunner、Andres Konrad、Yuyi Wang 和 Roger Wattenhofer。Midi-vae：利用音乐的力度变化和乐器配置建模，并应用于风格迁移。*arXiv* 预印本 *arXiv:1809.07600*，2018年。

John Ashley Burgoyne、Jonathan Wild 和 Ichiro Fujinaga。用于音频和弦识别和音乐分析的专家级真实值集。在*ISMIR*，第 11 卷，第 633–638 页，2011年。

Rodrigo Castellon、Chris Donahue 和 Percy Liang。编码音频语言建模学会了对音乐信息检索有用的表征。*arXiv*预印本 *arXiv:2107.05677*，2021年。

Ke Chen、Xingjian Du、Bilei Zhu、Zejun Ma、Taylor Berg-kirkpatrick 和 Shlomo Dubnov。通过从弱标注数据进行基于查询的学习，实现零样本音频源分离。*arXiv* 预印本 *arXiv:2112.07891*，2021a。

Ting Chen、Saurabh Saxena、Lala Li、David J Fleet 和 Geoffrey Hinton。Pix2seq：一种用于目标检测的语言建模框架。*arXiv* 预印本 *arXiv:2109.10852*，2021b。

Tian Cheng、Simon Dixon 和 Matthias Mauch。一种用于自动音乐转录的确定性退火 EM 算法。国际音乐信息检索学会会议，第 475–480 页，2013年。

Kin Wai Cheuk、Dorien Herremans 和 Li Su。Reconvat：一种面向低资源真实世界数据的半监督自动音乐转录框架。载于第 *29* 届 *ACM* 国际多媒体会议论文集，第 3918–3926 页，2021a。

Kin Wai Cheuk、Yin-Jyun Luo、Emmanouil Benetos 和 Dorien Herremans。通过加性注意力重新审视起始与帧模型。载于 国际神经网络联合会议论文集，即将发表。IEEE，2021b。数字对象标识符：10.1109/SPW.2018.00014。

Keunwoo Choi、Gy¨orgy Fazekas、Mark Sandler 和 Kyunghyun Cho。用于音乐分类和回归任务的迁移学习。载于第*18*届国际音乐信息检索学会会议（*ISMIR 2017*），第141–149页。国际音乐信息检索学会，2017年。

Yi-Hui Chou、I Chen、Chin-Jui Chang、Joann Ching、Yi-Hsuan Yang 等。MidiBERT-Piano：用于符号音乐理解的大规模预训练。*arXiv* 预印本 *arXiv:2107.05223*，2021年。

Junyoung Chung、Caglar Gulcehre、KyungHyun Cho 和 Yoshua Bengio。门控循环神经网络的实证评估序列建模中的表现。*arXiv* 预印本 *arXiv:1412.3555*，2014年。

Jun-qi Deng 和 Yu-Kwong Kwok。采用均等机会训练方案的大词汇量自动和弦估计。载于*Proc. ISMIR*，第 531–536 页，2017年。

Simon Durand、Juan P Bello、Bertrand David 和 Ga¨el Richard。用于强拍跟踪的特征自适应卷积神经网络。载于*2016 IEEE*国际声学、语音与信号处理会议 *(ICASSP)*，第 296–300 页。IEEE，2016年。

Josh Gardner、Ian Simon、Ethan Manilow、Curtis Hawthorne 和 Jesse Engel。MT3：多任务多轨音乐转录。*arXiv* 预印本 *arXiv:2111.03017*，2021年。

Masataka Goto、Hiroki Hashiguchi、Takuichi Nishimura 和 Ryuichi Oka。RWC音乐数据库：流行音乐、古典音乐和爵士音乐数据库。在*ISMIR* 会议论文集，第 2 卷，第 287–288 页，2002 年。

Fabien Gouyon。一种节奏描述的计算方法——用于计算节奏周期函数的音频特征及其在速度诱导和音乐内容处理中的应用。庞培法布拉大学，2006年。

Rui Guo、Ivor Simpson、Chris Kiefer、Thor Magnusson 和 Dorien Herremans。Musiac：一种可扩展的生成式框架，用于具有多层级控制的音乐补全应用。在国际音乐、声音、艺术与设计计算智能会议（*EvoStar* 的组成部分），第 341–356 页。Springer，2022年。

Stephen W Hainsworth 和 Malcolm D Macleod。将粒子滤波应用于音乐速度轨迹跟踪。*EURASIP*信号处理进展期刊，2004(15):1–11，2004年。

Curtis Hawthorne、Erich Elsen、Jialin Song、Adam Roberts、Ian Simon、Colin Raffel、Jesse Engel、Sageev Oore 和 Douglas Eck。Onsets and Frames：双目标钢琴转录。*arXiv*预印本 *arXiv:1710.11153*，2017年。

Curtis Hawthorne、Andriy Stasyuk、Adam Roberts、Ian Simon、Cheng-Zhi Anna Huang、Sander Dieleman、Erich Elsen、Jesse Engel 和 Douglas Eck。借助 MAESTRO数据集实现因子化钢琴音乐建模与生成。*arXiv*预印本 *arXiv:1810.12247*，2018年。

Curtis Hawthorne、Ian Simon、Rigel Swavely、Ethan Manilow 和 Jesse Engel。使用 Transformer 进行序列到序列钢琴转录。*arXiv*预印本 *arXiv:2107.09142*，2021年。

Andre Holzapfel、Matthew EP Davies、Jos´e R Zapata、Jo˜ao Lobato Oliveira 和 Fabien Gouyon。用于节拍跟踪评估的选择性采样。*IEEE* 音频、语音与语言处理汇刊，20(9)：2539–2548，2012年。

Cheng-Zhi Anna Huang、Ashish Vaswani、Jakob Uszkoreit、Ian Simon、Curtis Hawthorne、Noam Shazeer、Andrew M Dai、Matthew D Hoffman、Monica Dinculescu 和 Douglas Eck。音乐变换器：生成具有长期结构的音乐。载于国际学习表征会议，2018年。

Eric J Humphrey 和 Juan Pablo Bello。关于自动和弦估计的四个及时见解。载于*ISMIR* 论文集，第10卷，第673–679页，2015年。

Yun-Ning Hung、Gordon Wichern 和 Jonathan Le Roux。转录即所需：学习以评分为监督分离音乐混合信号。载于*ICASSP 2021-2021 IEEE*国际声学、语音与信号处理会议（*ICASSP*），第46–50页。IEEE，2021年。

Yun-Ning Hung、Ju-Chiang Wang、Xuchen Song、Wei-Tsung Lu 和 Minz Won。使用时频变换器对节拍和强拍进行建模。发表于*ICASSP 2022-2022 IEEE*国际声学、语音与信号处理会议 *(ICASSP)*，第 401–405 页。IEEE，2022年。

Sergey Ioffe 和 Christian Szegedy。批归一化：通过减少内部协变量偏移加速深度网络训练。发表于国际机器学习会议，第 448–456 页。PMLR，2015年。

Andreas Jansson、Eric Humphrey、Nicola Montecchio、Rachel Bittner、Aparna Kumar 和 Tillman Weyde。使用深度U-Net卷积网络进行歌声分离。第*18*届国际音乐信息检索学会会议，*ISMIR 2017*，2017年。

Andreas Jansson、Rachel M. Bittner、Sebastian Ewert 和 Tillman Weyde。采用深度U-Net架构进行联合歌声分离与基频估计。载于*2019*年第*27*届欧洲信号处理会议（*EUSIPCO*），第1–5页，2019年。数字对象标识符：10.23919/EUSIPCO.2019.8902550。

Rainer Kelz、Sebastian B¨ock 和 Gerhard Widmer。深度复调钢琴音符转录。载于*ICASSP 2019—2019*年 *IEEE* 国际声学、语音与信号处理会议（*ICASSP*），第246–250页。IEEE，2019年。

Jong Wook Kim 和 Juan Pablo Bello。用于改进起音与帧音乐转录的对抗学习。国际音乐信息检索学会会议，第 670–677 页，2019年。

Sunghyeon Kim、Hyeyoon Lee、Sunjong Park、Jinho Lee 和 Keunwoo Choi。使用符号表示进行深度作曲家分类。*ISMIR* 最新成果／演示环节，2020年。

Diederik P Kingma 和 Jimmy Ba。Adam：一种随机优化方法。国际学习表征会议（*ICLR*）第 *3* 届会议论文集，2014年。

Anssi Klapuri 和 Antti Eronen。音乐自动转录。载于斯德哥尔摩音乐声学会议论文集，第 6–9 页。Citeseer，1998年。

Qiuqiang Kong、Keunwoo Choi 和 Yuxuan Wang。基于 MIDI 的大规模作曲家分类。*arXiv* 预印本 *arXiv:2010.14805*，2020a。

Qiuqiang Kong、Bochen Li、Jitong Chen 和 Yuxuan Wang。GiantMIDI-Piano：用于古典钢琴音乐的大型 MIDI 数据集。*arXiv* 预印本 *arXiv:2010.07061*，2020b。

Qiuqiang Kong、Bochen Li、Xuchen Song、Yuan Wan 和 Yuxuan Wang。通过回归起音与结束时间，实现带踏板的高分辨率钢琴转录。*IEEE/ACM* 音频、语音与语言处理汇刊，29：3707–3717，2021年。

Florian Krebs、Sebastian B¨ock 和 Gerhard Widmer。音乐音频中用于拍点与强拍追踪的节奏模式建模。载于*ISMIR*，第 227–232 页。Citeseer，2013 年。

Edith Law、Kris West、Michael I Mandel、Mert Bay 和 J Stephen Downie。利用游戏评估算法：以音乐标签为例。载于国际音乐信息检索学会会议（*ISMIR*），2009年。

Huu Le、Rasmus Kjær Høier、Che-Tsung Lin 和 Christopher Zach。AdaSTE：一种用于训练二值神经网络的自适应直通估计器。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第 460–469 页，2022年。

Jongpil Lee、Jiyoung Park、Keunhyoung LukeKim 和 Juhan Nam。使用原始波形进行音乐自动标注的样本级深度卷积神经网络。*arXiv* 预印本 *arXiv:1703.01789*，2017年。

Liwei Lin、Qiuqiang Kong、Junyan Jiang 和 Gus Xia。用于零样本音乐源分离、转录和合成的统一模型。载于国际音乐信息检索学会（*ISMIR*），2021年。

Wei-Tsung Lu、Ju-Chiang Wang、Minz Won、Keunwoo Choi 和 Xuchen Song。SpecTNT：用于音乐音频的时频 Transformer。*arXiv* 预印本 *arXiv:2110.09127*，2021年。

Yin-Jyun Luo、Yuen-Jen Lin 和 Li Su。迈向富有表现力的人声演唱校正：关于人声旋律提取评估指标的感知有效性。*arXiv* 预印本 *arXiv:2010.12196*，2020年。

Ethan Manilow、Gordon Wichern、Prem Seetharaman 和 Jonathan Le Roux。为音乐源分离砍出一块 Slakh：用于研究训练数据质量和数量影响的数据集。发表于*IEEE* 音频与声学信号处理应用研讨会论文集（*WASPAA*）。IEEE，2019年。

Ethan Manilow、Prem Seetharaman 和 Bryan Pardo。多种复调乐器与打击乐器混合音频的同步分离与转录。载于*ICASSP 2020-2020 IEEE*国际声学、语音与信号处理会议*(ICASSP)*，第 771–775 页。IEEE，2020年。

Ugo Marchand 和 Geoffroy Peeters。摇摆比估计。载于数字音频效果 *2015 (Dafx15)*，2015年。Magdalena Fuentes、Matthew E. P. Davies、Sebastian B ock。速度、节拍与弱拍估计。https://tempobeatdownbeat.github.io/tutorial/intro.html，2021年。URL https:// tempobeatdownbeat.github.io/tutorial/intro.html。M. Mauch、C. Cannam、M. Davies、S. Dixon、C. Harte、S. Kolozali、D. Tidhar 和 M. Sandler。OMRAS2 元数据项目 2009年。载于*ISMIR* 最新成果与演示，2009年。Gabriel Meseguer-Brocal 和 Geoffroy Peeters。条件化 U-Net：在 U-Net 中引入一种用于多源分离的控制机制。*arXiv* 预印本 *arXiv:1907.01277*，2019年。Giyasettin Ozcan、Cihan Isikhan 和 Adil Alpkocak。MIDI 音乐文件中的旋律提取。载于第七届 *IEEE* 多媒体国际研讨会 *(ISM'05)*，第 8 页起。IEEE，2005年。Jonggwon Park、Kyoyun Choi、Sungwook Jeon、Dokyun Kim 和 Jonghun Park。用于音乐和弦识别的双向变换器。*arXiv* 预印本 *arXiv:1907.02698*，2019年。Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga 等。PyTorch：一种命令式风格的高性能深度学习库。神经信息处理系统进展，第 32 卷，2019年。Jouni Paulus 和 Anssi Klapuri。打击乐音频信号转录中的基于模型的事件标注。载于国际数字音频效果会议论文集 *(DAFX)*，第 73–77 页。Citeseer，2003年。Steffen Pauws。从音频中提取音乐调性。载于第五届国际音乐信息检索学会会议，2004年。F. Pedregosa、G. Varoquaux、A. Gramfort、V. Michel、B. Thirion、O. Grisel、M. Blondel、P. Prettenhofer、R. Weiss、V. Dubourg、J. Vanderplas、A. Passos、D. Cournapeau、M. Brucher、M. Perrot 和 E. Duchesnay。Scikit-learn：Python 机器学习。机器学习研究期刊，12：2825–2830，2011年。Ethan Perez、Florian Strub、Harm De Vries、Vincent Dumoulin 和 Aaron Courville。FiLM：通过通用条件层进行视觉推理。载于*AAAI* 人工智能大会论文集，第 32 卷，2018年。Martin Piszczalski 和 Bernard A Galler。自动音乐转录。计算机音乐期刊，第 24–31 页，1977年。Jordi Pons 和 Xavier Serra。MusiCNN：用于音乐音频标注的预训练卷积神经网络。*arXiv* 预印本 *arXiv:1909.06654*，2019年。Colin Raffel。用于序列比较的基于学习的方法及其在音频到 *MIDI* 对齐与匹配中的应用。哥伦比亚大学，2016年。Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J Liu。探索统一文本到文本变换器迁移学习的极限。机器学习研究期刊，21：1–67，2020年。Jing Shi、Xuankai Chang、Shinji Watanabe 和 Bo Xu。从头开始训练：语音分离与识别的单阶段联合训练。计算机语音与语言，76：101387，2022年。ISSN 0885-2308。Siddharth Sigtia、Emmanouil Benetos 和 Simon Dixon。用于复调钢琴音乐转录的端到端神经网络。*IEEE/ACM* 音频、语音与语言处理汇刊，24：927–939，2015年。Federico Simonetta、Federico Avanzini 和 Stavros Ntalampiras。用于评估自动音乐转录再合成效果的感知度量。多媒体工具与应用，第 1–21 页，2022年。

Nitish Srivastava、Geoffrey Hinton、Alex Krizhevsky、Ilya Sutskever 和 Ruslan Salakhutdinov。Dropout：一种防止神经网络过拟合的简单方法。机器学习研究期刊，15(1):1929–1958，2014年。

Keitaro Tanaka、Takayuki Nakatsuka、Ryo Nishikimi、Kazuyoshi Yoshii 和 Shigeo Morishima。基于频谱图和音高图的深度球面聚类的多乐器音乐转录。发表于国际音乐信息检索学会会议（*ISMIR*），蒙特利尔，加拿大，2020年。

John Thickstun、Za¨ıd Harchaoui 和 Sham M. Kakade。从头开始学习音乐特征。发表于*ICLR*，第 abs/1611.09827 卷，2016年。

John Thickstun、Za¨ıd Harchaoui、Dean P. Foster 和 Sham M. Kakade。监督式音乐转录中的不变性与数据增强。*2018*年 *IEEE*国际声学、语音与信号处理会议（*ICASSP*），第 2241–2245 页，2017年。

Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。《注意力就是你所需的一切》。神经信息处理系统进展，30，2017年。

Emmanuel Vincent、R´emi Gribonval 和 C´edric F´evotte。盲音频源分离的性能测量。*IEEE*音频、语音与语言处理汇刊，14(4)：1462–1469，2006年。

Richard Vogl、Matthias Dorfer、Gerhard Widmer 和 Peter Knees。利用卷积循环神经网络联合建模节拍与鼓声进行鼓声转录。发表于第*18*届国际音乐信息检索学会会议，*ISMIR 2017*，第150–157页，2017年。

Ju-Chiang Wang、Yun-Ning Hung 和 Jordan BL Smith。捕捉副歌、主歌、前奏或其他部分：运用结构功能分析歌曲。发表于*ICASSP*论文集，第416–420页，2022年。

Minz Won、Sanghyuk Chun 和 Xavier Serra。利用自注意力实现可解释的音乐标签标注。*arXiv* 预印本 *arXiv:1906.04972*，2019年。

Minz Won、Sanghyuk Chun、Oriol Nieto 和 Xavier Serrc。用于音频表征学习的数据驱动谐波滤波器。发表于*ICASSP 2020-2020 IEEE*国际声学、语音与信号处理会议 *(ICASSP)*，第 536–540 页。IEEE，2020a。

Minz Won、Andres Ferraro、Dmitry Bogdanov 和 Xavier Serra。基于卷积神经网络的自动音乐标签标注模型评估。*arXiv* 预印本 *arXiv:2006.00751*，2020b。

Minz Won、Keunwoo Choi 和 Xavier Serra。半监督音乐标签 Transformer。发表于国际音乐信息检索学会会议 *(ISMIR)*，2021a。

Minz Won、Janne Spijkervet 和 Keunwoo Choi。《音乐分类：超越监督学习，迈向现实世界应用》。https://music-classification.github.io/tutorial，2021b。数字对象标识符：10.5281/zenodo.5703779。网址：https://music-classification.github.io/tutorial。

Xianchao Wu、Chengyuan Wang 和 Qinying Lei。基于 Transformer-XL、使用多个时值音符序列的音乐生成。《*arXiv*预印本 *arXiv:2007.07244*》，2020a。

Yiming Wu 和 Wei Li。采用 MIDI训练的深度特征和 BLSTM-CRF序列解码模型的自动音频和弦识别。《*IEEE/ACM* 音频、语音与语言处理汇刊》，27(2)：355–366，2018年。

Yu-Te Wu、Berlin Chen 和 Li Su。采用语义分割的复调音乐转录。载于《*ICASSP 2019 - 2019 IEEE*国际声学、语音与信号处理会议 *(ICASSP)*》，第166–170页，2019年。数字对象标识符：10.1109/ICASSP.2019.8682605。

Yu-Te Wu、Berlin Chen 和 Li Su。基于自注意力的实例分割多乐器自动音乐转录。*IEEE/ACM* 音频、语音与语言处理汇刊，28:2796–2809，2020b。

Yu-Te Wu、Yin-Jyun Luo、Tsung-Ping Chen、I Wei、Jui-Yang Hsu、Yi-Chin Chuang、Li Su 等。Omnizart：用于自动音乐转录的通用工具箱。*arXiv*预印本 *arXiv:2106.00497*，2021年。

Yao-Yuan Yang、Moto Hira、Zhaoheng Ni、Artyom Astafurov、Caroline Chen、Christian Puhrsch、David Pollack、Dmitriy Genzel、Donny Greenberg、Edward Z Yang 等。Torchaudio：音频与语音处理的构建模块。载于*ICASSP 2022-2022 IEEE*国际声学、语音与信号处理会议 *(ICASSP)*，第 6982–6986 页。IEEE，2022年。

Penghang Yin、Jiancheng Lyu、Shuai Zhang、Stanley J. Osher、Yingyong Qi 和 Jack Xin。《理解训练激活量化神经网络中的直通估计器》。发表于国际学习表征会议，2019年。网址：https://openreview.net/forum? id=Skh4jRcKQ。

A 模型架构

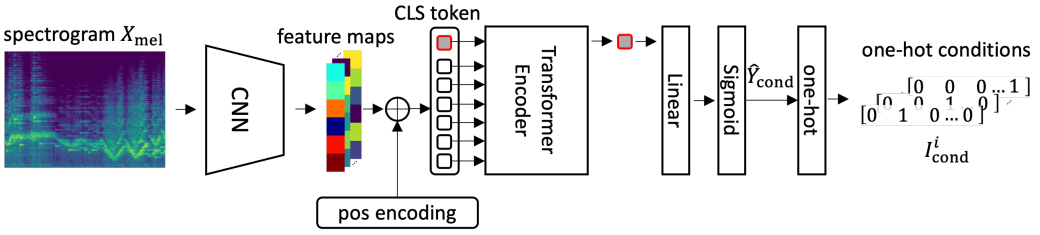

图 2： f_{IR} 模块的模型架构，由 CNN 骨干网络和 Transformer 编码器组成。Transformer 编码器中的分类（CLS）标记用于提取频谱图中出现的乐器。CNN 前端包含六个卷积块，输出通道数为 $[64, 128, 256, 512, 1024, 2048]$。每个卷积块包含两个卷积层，卷积核大小为 (3,3)，步幅为 (1,1)，填充为 (1,1)，后接平均池化 (2,2)。使用 Sigmoid 激活将输出转换为概率。最后，对 Sigmoid 输出应用阈值（$p > 0.5$），得到作为乐器条件的独热向量。

请注意，由于乐器条件具有独热性质，人类用户可以轻松接管f_{IR}，并指定感兴趣的目标乐器。否则，可以使用全部 39 种乐器条件进行完整转录，或者利用 f_{IR} 自动确定音频中出现的乐器。这使 Jointist 成为一个灵活的框架。

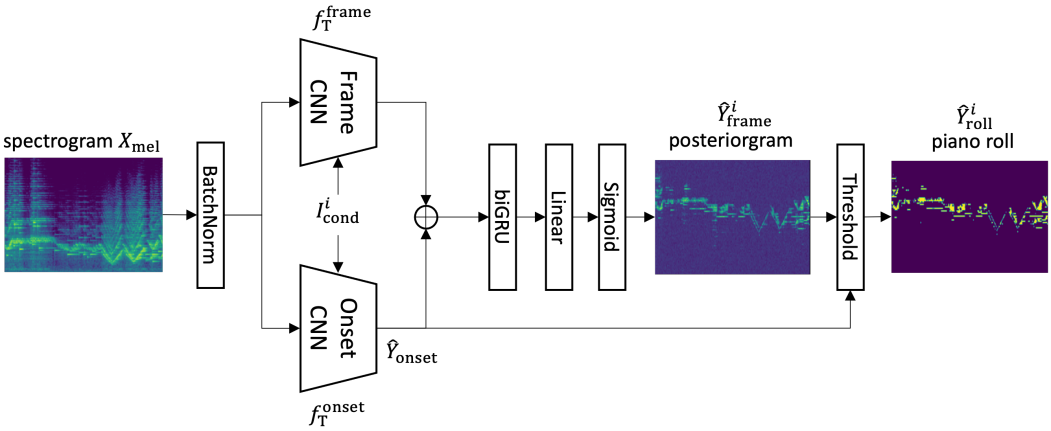

图 3： f_{T} 模块的模型架构。它由一个批归一化层（Ioffe与Szegedy，2015年）、一个帧模型 $f_{\text{T}}^{\text{frame}}$ 和一个起始模型 $f_{\text{T}}^{\text{onset}}$ 组成。 $f_{\text{T}}^{\text{frame}}$和 $f_{\text{T}}^{\text{onset}}$均采用相同的架构，由四个条件卷积块组成，这些卷积块通过 FiLM 层接收 $\hat{I}^i_{\text{cond}}$，后接一个带 ReLU 激活的线性层、一个隐藏维度为 256 的双向GRU（biGRU）层（Chung等人，2014年），以及一个带 Sigmoid 激活的线性层，该层将隐藏维度映射为 88 个音符。每个条件卷积块包含两个带批归一化和 ReLU 激活的二维卷积层，后接一个尺寸为 (2,2) 的平均池化层。

来自$f_{\text{T}}^{\text{frame}}$和$f_{\text{T}}^{\text{onset}}$的输出拼接在一起，得到一个维度为 172 的张量。然后将该张量输入另一个隐藏维度为 256 的双向 GRU 层，随后通过一个带 sigmoid 激活函数的线性层，将隐藏特征映射回 88 维，作为后验图 $\hat{Y}^i$帧。

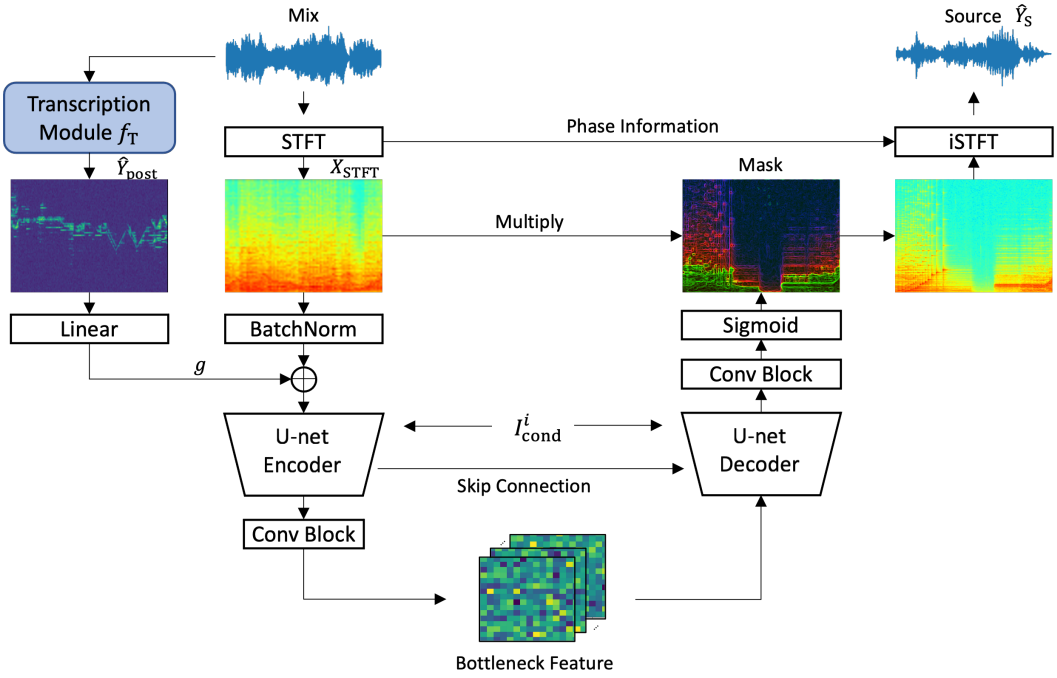

图 4： f_{MSS} 模块的模型架构以 U-net 为主要组件，输入为X_{STFT} 。U-net 中加入了 FiLM，使输出声源 $\hat{Y}^i_S$ 可由 $\hat{I}^i_{cond}$控制。使用线性层将 $\hat{Y}^i_{frame} = f_T(X_{mel}, \hat{I}^i_{cond})$映射为张量 g，其形状与X_{STFT}相同。然后，U-net编码器将经批归一化的 X_{STFT} 和 g 合并后的特征作为输入。表 9 研究了不同的合并方法，例如求和和拼接，以及频谱图块。在频谱图块方法中，我们将后验图特征 g 直接相加到

U-net编码器由六个卷积块组成，其输出通道数为 [32, 64, 128, 256, 384,384]。每个卷积块包含两个无偏置卷积层，卷积核大小为 (3,3)，步幅大小为 (1,1)，填充为 (1,1)，并采用平均池化尺寸为 (2,2) 的平均池化，随后连接一个批归一化层。卷积层的输出与 FiLM 条件相结合，方式同 Meseguer-Brocal & Peeters（2019年）中的方法。编码器之后还有一个额外的卷积块，其输入和输出通道数均为 384，卷积核大小为 (2,2)，填充为 (1,1)。

U-net解码器采用与 U-net编码器类似的架构来逆向执行该过程。它由六个转置卷积块组成，其输出通道数为 [384, 384, 256, 128, 64, 32]。每个卷积块包含一个无偏置转置卷积层，卷积核大小为 (3,3)，步幅大小为 (1,1)，填充为 (0,0)，随后连接一个批归一化层。同样，解码器之后还有一个卷积块，其输入和输出通道数均为 32，卷积核大小为 (1,1)，填充为 (0,0)。

U-netDecoder 的输出是一个取值介于 [0,1]之间的频谱图掩码。将原始幅度短时傅里叶变换（混合信号）与该掩码相乘，即可得到分离后的幅度短时傅里叶变换（声源）。为了将其逆变换回波形，我们使用逆短时傅里叶变换，该过程还需要相位信息。由于分离后的幅度短时傅里叶变换不包含相位，我们使用混合信号短时傅里叶变换中的原始相位信息。

我们还针对使用连续值后验图进行钢琴卷帘实验 $Y^i_{post} \in \{0,1\}$ 与表 9 中的图进行钢琴卷帘实验 $\in \{0,1\}$ 相比，我们发现使用后验图通常优于使用钢琴卷帘。

B 评估指标

B.1 平均精度均值

我们首先使用 scikit-learn 软件包（Pedregosa 等，2011）计算每个音频样本 s 的乐器级平均精度（AP_s^i），其定义如下：

$$\text{AP}_s^i = \sum_k (R_k^i - R_{k-1}^i) P_k^i, \tag{1}$$

其中， P_k^i 和 R_k^i 分别是表 11 中定义的乐器 i 在第 k个阈值下的精确率和召回率。

然后，可以使用以下公式计算乐器 i 在大小为 N 的测试集上的平均精度均值（mAP^i）：

$$\text{mAP}^i = \frac{1}{N} \sum_{s=1}^{N} \text{AP}_s^i \tag{2}$$

该指标衡量不同阈值下的乐器识别性能，因此不受阈值选择错误的影响。由于不同乐器标签的分布存在偏斜（如图6所示），我们同时报告**宏平均**和**加权平均**平均精度均值。**宏平均精度均值**的计算方式是对不同乐器的平均精度均值进行不加权平均：

$$\text{macro mAP} = \frac{\sum_{i=1}^{I} \text{mAP}^i}{I}, \tag{3}$$

其中， I 是乐器总数，根据表 11，该数值为39。宏平均精度均值容易受到只有少量标签的乐器影响，例如巴松管、大提琴和竖笛，因此当数据分布严重偏斜时，它并非可靠的指标。但它能让我们了解模型在所有乐器类别上的表现。宏平均评分较低意味着模型在某些特定乐器类别上的性能较差。

为考虑数据分布的不均衡，我们还报告**加权平均精度均值**，其计算方式如下

$$\text{weighted mAP} = \frac{\sum_{i=1}^{I} (\text{mAP}^i \cdot w(i))}{I}, \tag{4}$$

其中， $w(i)$ 是第 i 个乐器的权重，与其在测试集中的标签数量成正比。与宏平均指标相比，这一指标更可靠，因为它对标签较多的类别赋予更高权重，对标签较少的类别赋予较低权重。换言之，在标签较少的乐器类别中出现一次预测错误，不会严重影响加权平均精度均值。

结合图 8 报告这两个指标后，我们可以看出，模型性能与标签可用性高度相关，如图6所示。

B.2 音符级转录 F1分数

我们使用 mir eval 中标准的转录精确率、召回率、F1 重叠函数，计算每个音频样本 s中每种乐器 i 的音符级 F1 分数（记作 F_s^i）。我们采用标准的音符级 F1 设置：只有当预测音高与真实标签一致，且预测起始时间与真实起始时间相差不超过 50 毫秒时，才将其视为正确转录。

我们的**逐乐曲逐音符 F1分数**评分计算如下

$$\text{F1}_{\text{piece}} = \frac{1}{S} \sum_s \frac{1}{I_s} \sum_i F_s^i, \tag{5}$$

其中， I_s 是一个变量，表示音频样本 s中真实乐器的数量，而 S 是一个常数，表示测试集中音频样本的总数。虽然

这一指标反映了模型的整体性能，但无法说明模型特别擅长或不擅长处理哪种乐器。

To m衡量不同乐器上的性能，我们提出了**按乐器逐音符**
核心F1分数，其定义如下

$$\mathrm{F1}_{\text{instrument}} = \frac{1}{I}\sum_i \frac{1}{S_i}\sum_s F_s^i, \tag{6}$$

其中， S_i 是一个变量，表示包含乐器 i的音频样本数量；$I = 39$ 是一个常数，表示我们的模型支持的乐器数量，如表 11 所示。请注意，尽管（Cheng 等，2013）提到了按乐器进行评估，但他们的每个音频样本始终包含 10 种乐器，因此其评估方法比我们的方法简单得多。我们的样本中，每个样本包含的乐器数量不尽相同。因此，对 Slakh2100 进行按乐器评估，并不像（Cheng 等，2013）那样直接。

B.3 信源失真比（信源失真比）

我们采用了（Vincent 等，2006）中定义的标准信源失真比，

$$\mathrm{SDR} = 10\log_{10}\left(\frac{\|s_{\text{target}}\|^2}{\|s_{\text{target}} - \hat{s}\|^2}\right), \tag{7}$$

其中，s_{target} 是真实波形，$\hat{s}$ 是预测波形。由于 Slakh2100 中每个音频样本 s 的乐器数量 I_s 各不相同，因此我们需要定义新的指标来处理这种情况。为便于下文讨论，我们将样本s中乐器 i 的 SDR_s^i 定义为 SDR。

按声源计算的 SDR 定义为数据集中所有声源 N_{all} 的平均 SDR，

$$\mathrm{SDR}_{\text{source}} = \frac{1}{N_{\text{all}}}\sum_s\sum_i \mathrm{SDR}_s^i. \tag{8}$$

这是最直观、但信息量最少的指标。
为了了解模型在每个音频样本（乐曲）上的表现，我们将**逐曲 SDR** 定义如下

$$\mathrm{SDR}_{\text{piece}} = \frac{1}{N_S}\sum_s \frac{1}{N_s^I}\sum_i \mathrm{SDR}_s^i, \tag{9}$$

其中， N_s^I是音频样本（乐曲） s 中的乐器数量， N_S 是测试集中的音频样本（乐曲）总数。该指标使我们能够了解模型在每首乐曲上的性能。

同样，为了了解模型在每种乐器上的性能，我们使用**逐乐器 SDR** 。其定义如下：

$$\mathrm{SDR}_{\text{instrument}} = \frac{1}{I}\sum_i \frac{1}{N_S^i}\sum_s \mathrm{SDR}_s^i, \tag{10}$$

其中， N_S^i 表示包含乐器 i 和 $I = 39$ 的乐曲数量，如公式 6 所示。

B.4 主观指标

• **乐器完整性**：对重要乐器的识别有多准确？如果漏掉了某种重要乐器，或错误地将某种乐器列为主导乐器，则其乐器完整性被视为较差。

• **乐器音符连续性**：每种重要乐器的音符连续性如何？例如，如果系统将一串钢琴音符在不同片段中分别归为吉他、管风琴和钢琴，则钢琴的音符连续性被视为较差。

• **整体音符准确度**：整体聆听重要乐器时，其音符有多准确？是否遗漏了重要音符或插入了错误音符？

• **整体聆听体验**：转录的整体质量如何？

B.5 转录主观评估的评分标准

表7列出了第4.3.2节提出的四个方面的评分标准。

评分	**乐器完整性标准**
5	完美。
4	有些不重要的乐器被遗漏或误识别，但不会影响整体聆听体验。
3	有些重要的乐器被遗漏或误识别，并对整体聆听体验造成了一定影响。
2	更多重要的乐器被遗漏或误识别，并明显降评估整体聆听体验。
1	识别出的乐器完全错误。
评分	**乐器音符连续性标准**
5	完美。
4	一些不重要乐器的音符连续性较差，但不会影响整体聆听体验。
3	一些重要乐器的音符不连续，对整体聆听体验有一定影响。
2	更多重要乐器的音符不连续，明显降低了整体聆听体验。
1	音符被随机分配给乐器，无法归纳出任何规律。
评分	**整体音符准确度的评判标准**
5	完美。
4	一些不重要的乐器的音符准确度较低，但不会影响整体聆听体验。
3	一些重要乐器的音符不准确，并对整体产生中等程度的影响聆听体验。
2	更重要的乐器所演奏的音符不准确，显然会降低整体聆听体验。
1	音符完全杂乱无章，无法从这些音符中辨认出歌曲。
评分	**整体聆听体验的评判标准**
5	优秀
4	Good
3	Fair
2	Poor
1	糟糕

表7：主观转录质量的评分标准。

C 主观评估元数据

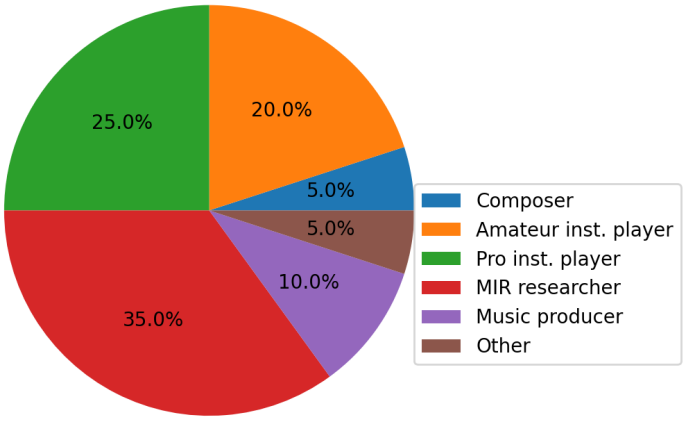

图5：主观评价参与者的背景音乐。

艺术家	歌曲标题（或 ID）
	Isophonics
披头士	《Let It Be》
披头士	《Lucy In The Sky With Diamonds》
披头士	《Yellow Submarine》
迈克尔·杰克逊	《Black or White》
迈克尔·杰克逊	我想要你回来
迈克尔·杰克逊	《墙外》
皇后乐队	《波希米亚狂想曲》
皇后乐队	我想要挣脱束缚
	RWC-POP
RWC	RM-P004
RWC	RM-P033
RWC	RM-P047
RWC	RM-P083
RWC	RM-P096
	JayChou29
周杰伦	超人不会飞
周杰伦	给我一首歌的时间
周杰伦	菊花台
	USPOP2002
PapaRoach	最后的手段
Radiohead	业力警察
瑞奇·马丁	疯狂人生
辣妹组合	成为一体

表8：用于主观评估的20首完整音轨的元数据。Isophon- ics、JayChou29 和 USPOP2002 的音频可能可在 https://github.com/krist311/ chords-recognition 获取。

D 数据分布

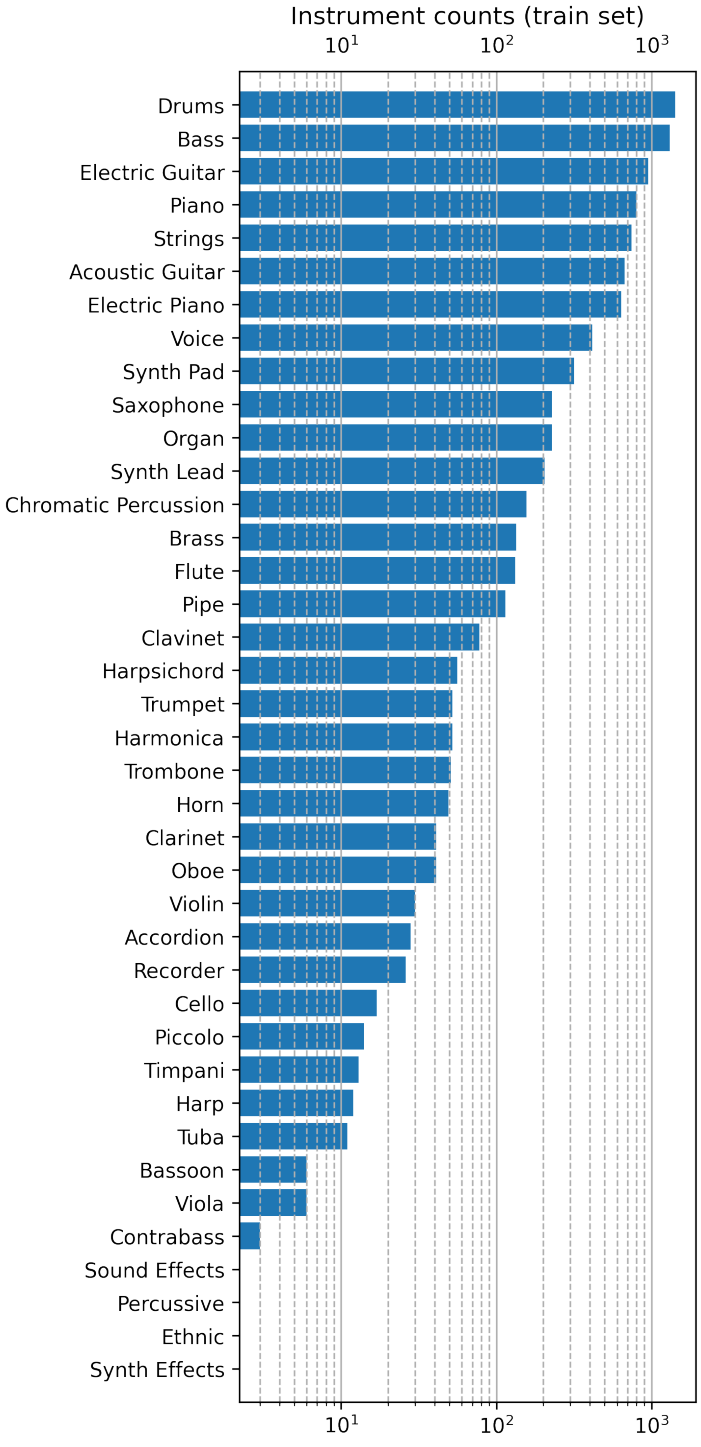

图6：Slakh2100 训练集的乐器标签数量。标签数量采用对数刻度。

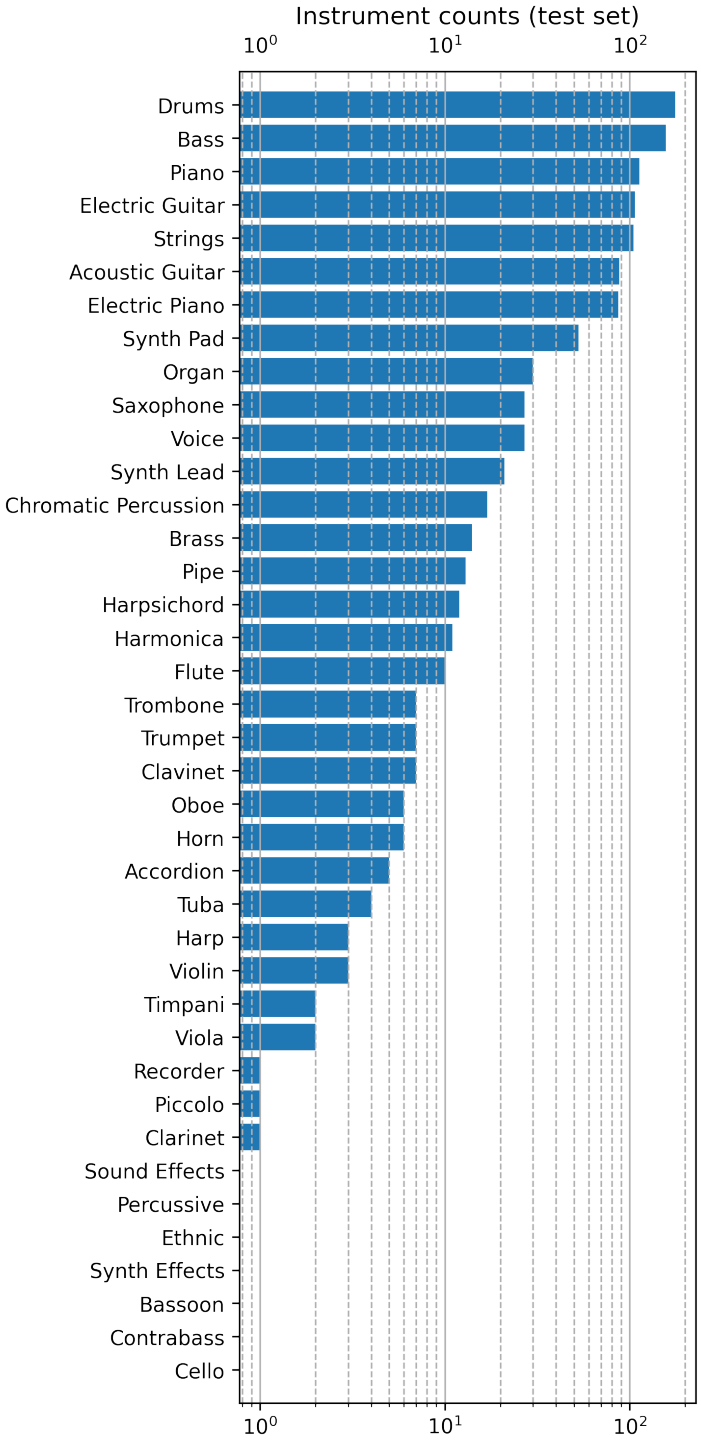

图 7：Slakh2100 测试集的乐器标签数量。标签数量采用对数刻度 .

E 逐乐器性能

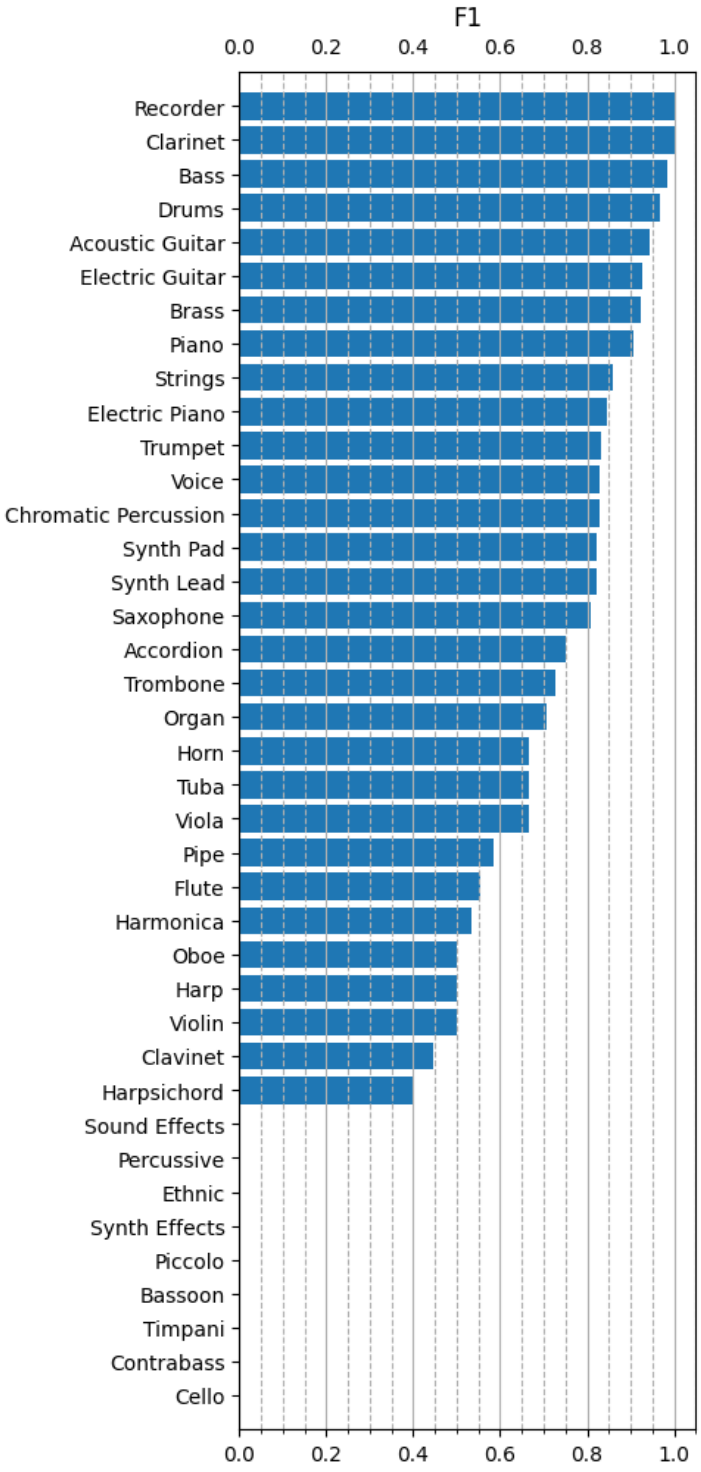

图 8：Slakh2100 测试集中 f_{IR} 模块的逐乐器F1分数。

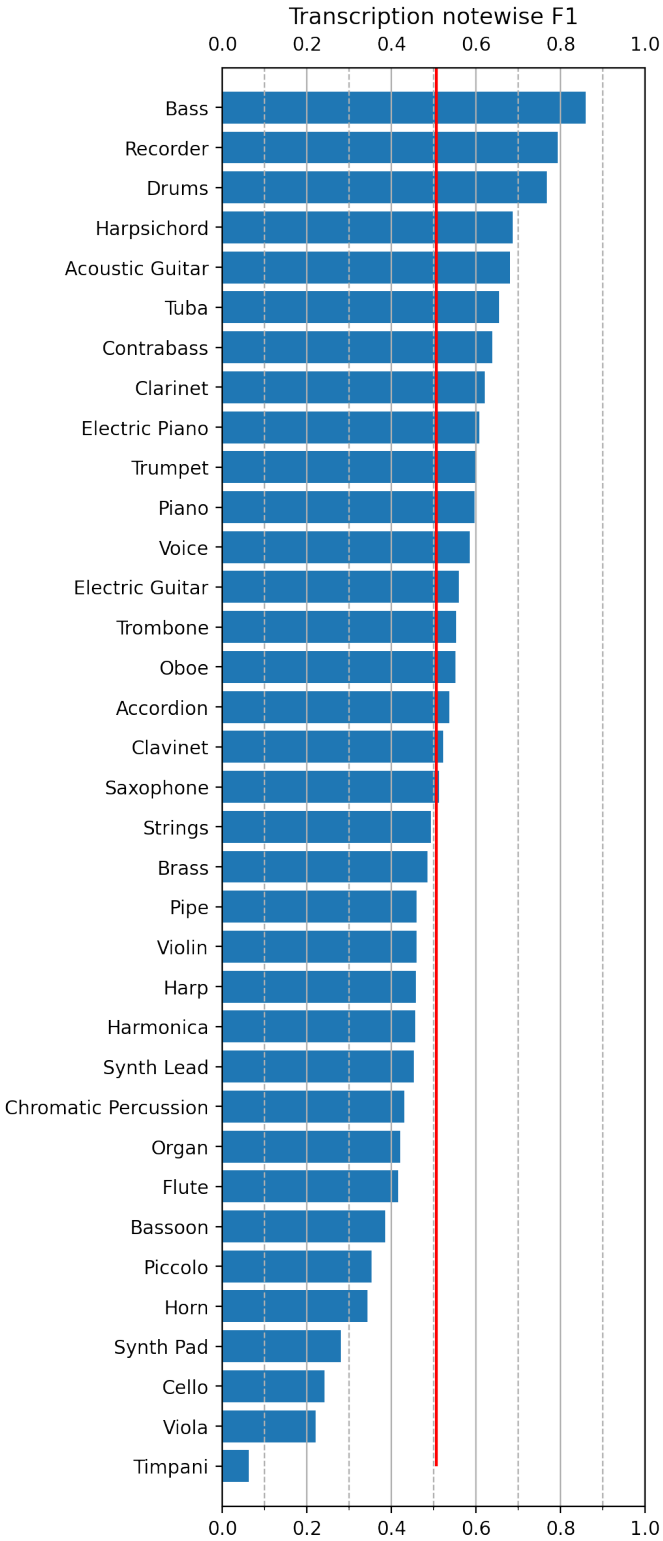

图9：Slakh2100 测试集中 f_{T} 模块各乐器的音符F1分数。红线表示平均 F1 分数。

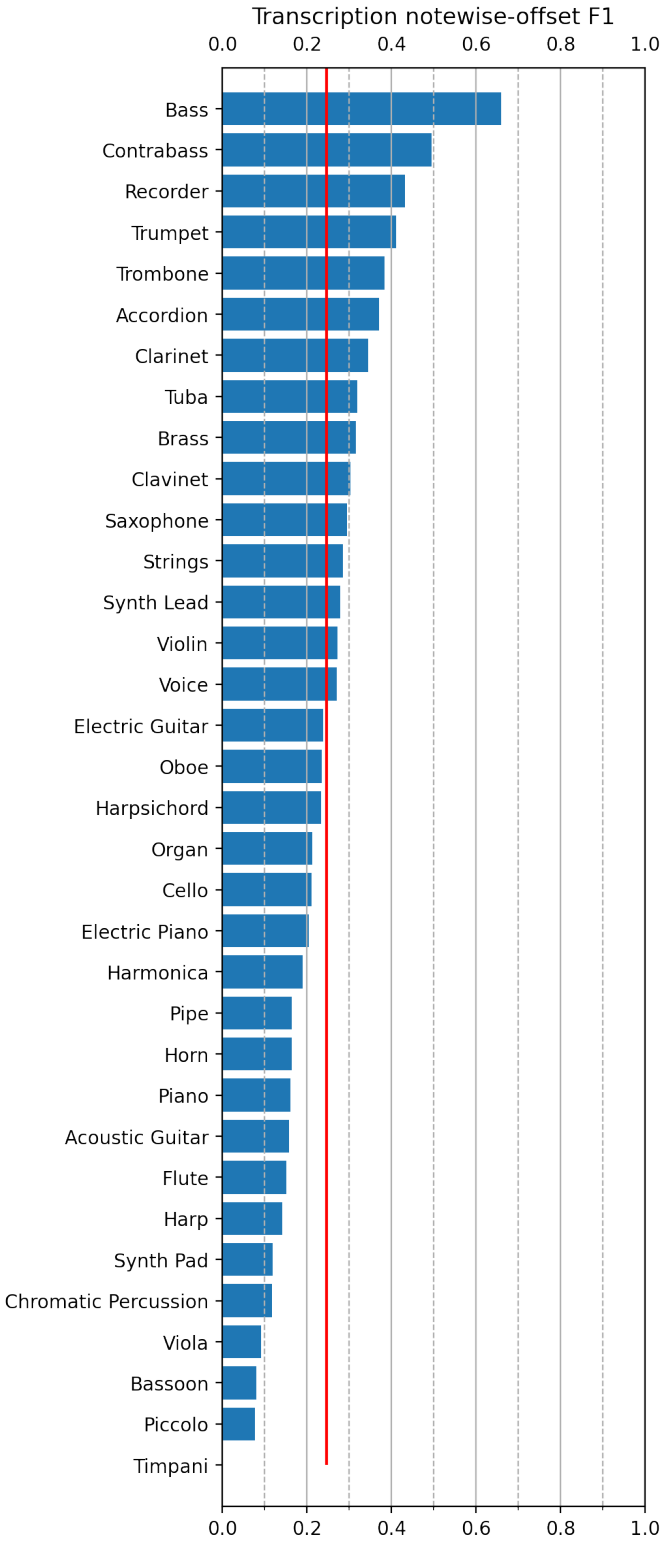

图10：Slakh2100 测试集中 f_{T} 模块按乐器划分的带偏移音符F1分数。红线表示平均 F1 分数。

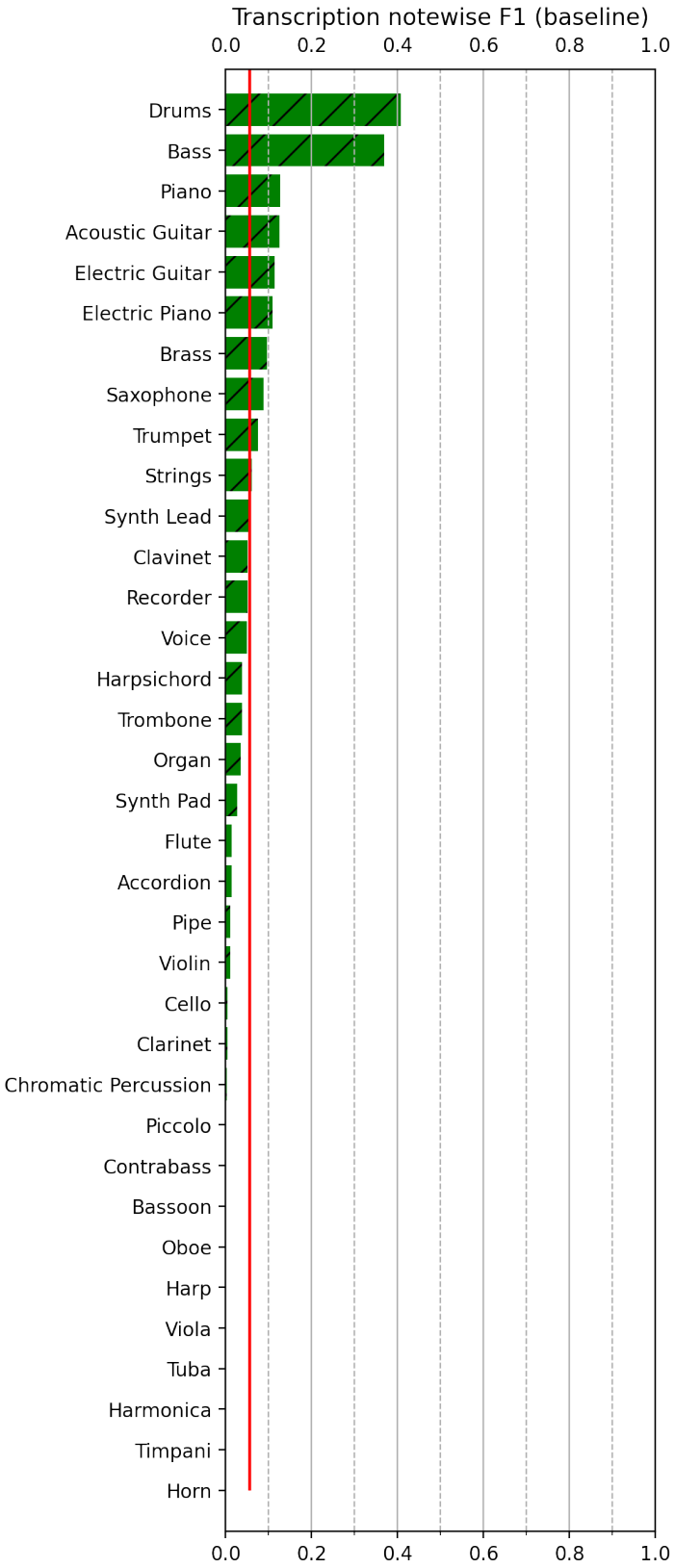

图 11：基线模型 Wu 等人（2021年）在 Slakh2100 测试集上各乐器的音符 F1 分数（有偏移量和无偏移量）。红线表示平均 F1 分数。

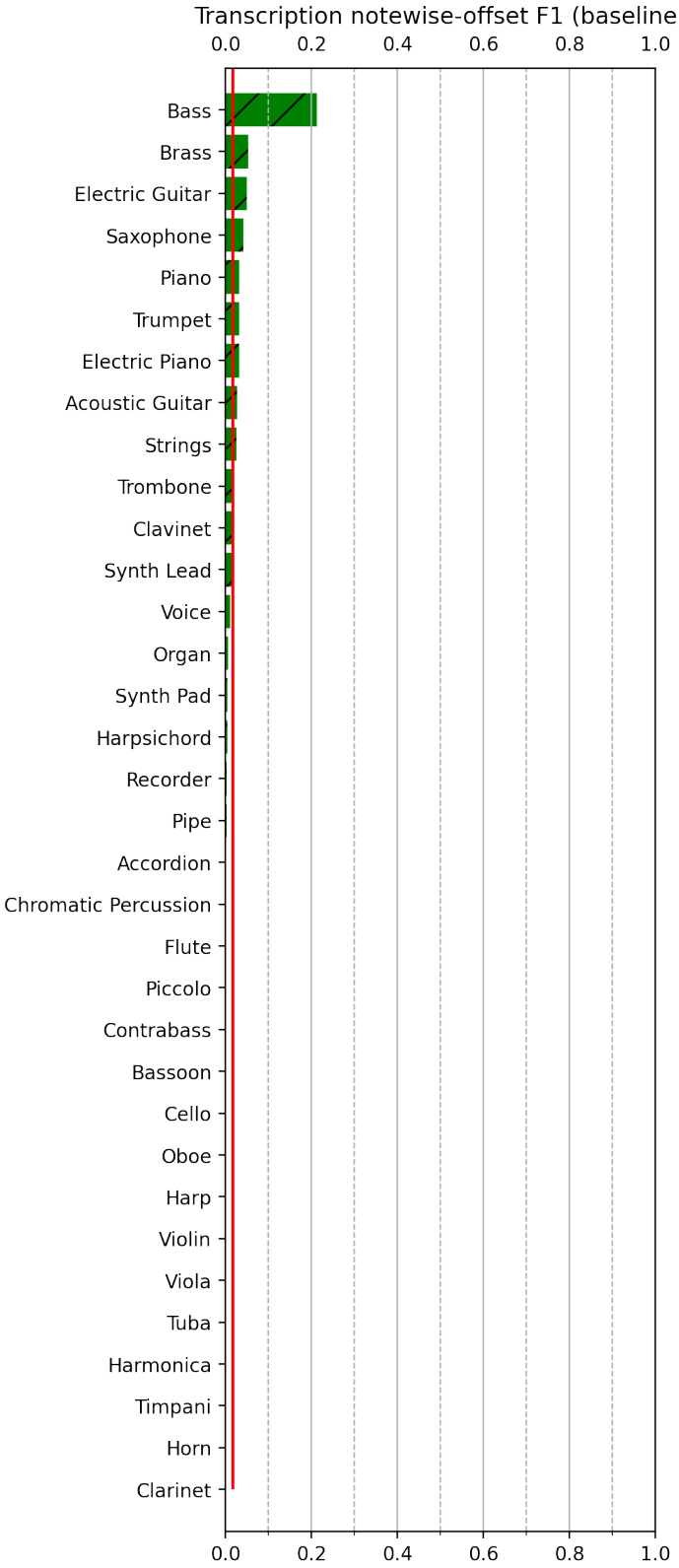

图 12：基线模型 Wu 等人（2021年）在 Slakh2100 测试集上的各乐器音符 F1分数（计入和未计入偏移量）。红线表示平均 F1 分数。

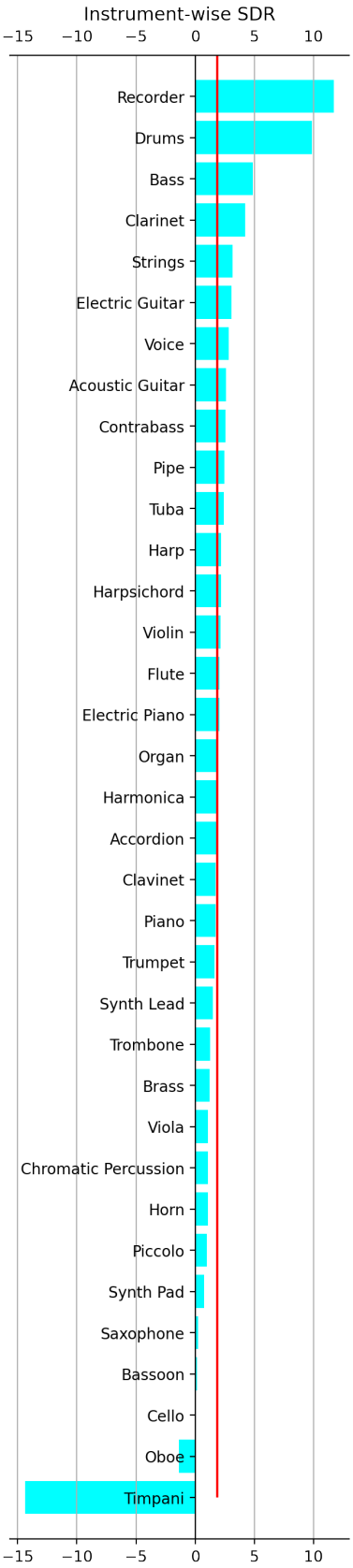

图 13：Slakh2100 测试集中 f_{T} 模块按乐器划分的源失真比（SDR）F1 分数。红线表示平均 F1 分数。29

F 消融研究

		全长度信源失真比			10 秒信源失真比		
特征合并	模型	inst	乐曲	声源	inst	乐曲	声源
sum	TS（后验）	1.86	3.55	3.32	3.47	5.60	4.54
	pTS（后验）	**2.01**	3.72	3.50	**3.69**	5.86	4.81
	TS+直通估计（二元）	1.30	3.30	3.06	2.40	5.13	4.16
	pTS+直通估计（二值）	1.67	3.30	3.06	3.29	5.36	4.41
拼接	TS（后验）	1.80	3.53	3.31	3.21	5.50	4.51
	pTS（后验）	1.92	**3.75**	**3.52**	3.50	**6.03**	**4.90**
	TS+直通估计（二值）	1.89	3.60	3.37	3.63	5.70	4.66
	pTS+直通估计（二值）	1.72	3.47	3.22	3.18	5.78	4.61
规格补丁	TS（后验）	1.79	3.46	3.24	3.35	5.30	4.43
	pTS（后验）	1.91	3.61	3.40	3.58	5.52	4.65
	TS+直通估计（二值）	1.74	3.46	3.24	3.50	5.28	4.42
	pTS+直通估计（二值）	1.71	3.47	3.23	3.28	5.40	4.53

表 9：当 f_{MSS} 将 f_{T} （T）与 f_{MSS} （S）联合训练时所达到的信源失真比（SDR）的消融研究。与表2类似，前缀“p-”表示转录模块经过预训练。在本研究中，我们还假设真实乐器就是人类用户希望转录的目标乐器，与表2中的设定相同。

我们研究了频谱图 X_{STFT} 与转录特征 g 之间的三种合并模式（图 4）。在“总和”模式下，批归一化层之后的 X_{STFT} 直接与 g相加。生成的合并张量与 X_{STFT} 和 g的维度相同。在“拼接”模式下，X_{STFT} 与 g 拼接在一起，形成一个频率维度为 X_{STFT} 和 g两倍的张量。“拼接”模式通常优于“总和”模式，但计算复杂度更高。我们还探索了“规格补丁”模式：在批归一化之前，将 g 直接加到 X_{STFT} 上。在这种合并模式下，模型预测掩码会应用于修改后的频谱图 $X_{\text{STFT}} + g$。我们想知道 g 能否用于增强与目标乐器 i相对应的 X_{STFT} 中的谱特征。结果表明，这种合并模式在三种模式中表现最差。由于“总和”和“拼接”模式之间的信源失真比差异并不显著，我们选择采用“总和”模式。

表2显示，使用真实钢琴卷帘时，由于其本质上是二值的，音源分离性能有了显著提升。为了弄清性能提升究竟是源于钢琴卷帘的二值特性，还是源于完全准确的转录音符，我们还尝试了两种转录输出形式：后验图 $Y^{i}_{\text{帧}} \in [0,1]$和钢琴卷帘 $Y^{i}_{\text{卷帘}} \in \{0,1\}$。将 $Y^{i}_{\text{帧}}$ 转换为 $Y^{i}_{\text{卷帘}}$时，需要进行阈值化，这会破坏其中的梯度信息。为了保留梯度信息，我们采用直通估计（STE）Yin 等人（2019年）；Le 等人（2022年）作为阈值化方法。结果表明，使用二值钢琴卷帘 Y^{i}_{roll} 并不比使用后验图 Y^{i}_{frame}有优势。

G 弱拍、和弦与调性估计的实验设置 S

在音频处理中，我们使用 Won 等人（2020a）提供的六通道谐波频谱图（128个频率箱）。我们将 Jointist 生成的钢琴卷帘简化为两个通道（乐器索引0–37作为通道0，索引38作为通道1）。随后，我们使用一维卷积层将钢琴卷帘投影到与频谱图相同的频率维度。由此得到的混合表示是频谱图与钢琴卷帘的拼接，即8通道表示。

我们选择了最先进的 Transformer架构 SpecTNT（Lu 等人（2021）），用于对音频录音中的时间音乐事件进行建模（Hung 等人（2022）；Wang 等人（2022））。时间戳和标签标注被转换为学习目标的时间激活曲线，供

与 Lu 等人（2021）和 Hung 等人（2022）的做法相同，使用 SpecTNT。下方表 10 总结了我们为这三项任务配置的 SpecTNT。

我们遵循 Hung 等人（2022）的方法，联合训练节拍与强拍跟踪任务，但重点关注强拍评估，因为它比节拍跟踪更具挑战性（B¨ock & Davies, 2020; Durand 等人, 2016）。和弦估计任务包含 24 种大调和小调三和弦类别，调性估计任务包含 24 种大调和小调调性类别；两个任务还都设有一个“无”类别。调性与和弦任务分别进行训练和评估。

对于强拍跟踪，我们使用 7 个数据集：Ballroom (Krebs 等人, 2013)、Hainsworth (Hainsworth & Macleod, 2004)、SMC (Gouyon, 2006)、Simac (Holzapfel 等人, 2012)、GTZAN (Marchand & Peeters, 2015)、Isophonics (Mauch 等人, 2009) 和 RWC-POP (Goto 等人, 2002)。我们使用 Isophonics 和 RWC-POP 进行评估，其余 6 个数据集用于训练。对于和弦估计，我们使用 Billboard (Burgoyne 等人, 2011)[3], Isophonics 和 RWC-POP。我们使用 Isophonics 和 RWC-POP 进行评估，其余 2 个数据集用于训练。对于调性估计，我们使用 Isophonics 进行评估，使用 Billboard 进行训练。

Task	输入长度	(k, d)	(h_k, h_d)
弱拍	6 秒	(128, 96)	(4, 8)
和弦	12 秒	(64, 256)	(8, 8)
Key	36 秒	(128, 32)	(8, 4)

表1：我们在各项任务中使用的 SpecTNT 参数，其中 k 和 d 表示频谱和时间特征维度；h_k 和 h_d 分别表示频谱和时间 Transformer 编码器的注意力头数量。

[3]由于缺少 Billboard 的音频文件，我们从互联网重新下载了缺失的文件。

MIDI 编号	MIDI 乐器	MIDI 音色	我们的映射	我们的编号
0-3	钢琴	大钢琴/明亮钢琴/酒吧钢琴	钢琴	0
4-5	钢琴	电钢琴 1-2	电钢琴	1
6	钢琴	大键琴	大键琴	2
7	钢琴	电翼琴	电翼琴	3
8-15	合唱打击乐	钢片琴、钟琴、音乐盒, 颤音琴、马林巴琴、木琴, 管钟 ，扬琴	合唱打击乐	4
16-20	管风琴	拉杆 ，打击乐 ， 摇滚、教堂, 簧风琴	管风琴	5
21	管风琴	手风琴	手风琴	6
22	管风琴	口琴	口琴	7
23	管风琴	探戈手风琴	手风琴	6
24-25	吉他	原声吉他（尼龙弦、钢弦）	原声吉他	8
26-31	吉他	电吉他（爵士、清音、闷音、 过载、失真、泛音）	电吉他	9
32-39	Bass	原声/电贝斯/击弦贝斯/合成贝斯	Bass	10
40	弦乐	小提琴	小提琴	11
41	弦乐	中提琴	中提琴	12
42	弦乐	大提琴	大提琴	13
43	弦乐	低音提琴	低音提琴	14
44	弦乐	弦乐震音	弦乐	15
45	弦乐	拨奏弦乐	弦乐	15
46	弦乐	管弦乐竖琴	Harp	16
47	弦乐	定音鼓	定音鼓	17
48-51	合奏	原声/合成弦乐合奏 1-2	弦乐	15
52-54	合奏	啊声/哦声/合成音色人声	人声	18
55	合奏	管弦乐重击音	弦乐	15
56	铜管乐器	小号	小号	19
57	铜管乐器	长号	长号	20
58	铜管乐器	Tuba	Tuba	21
59	铜管乐器	弱音小号	小号	19
60	铜管乐器	圆号	Horn	22
61-63	铜管乐器	原声/合成铜管乐器	铜管乐器	23
64-67	Reed	高音、中音、次中音、上低音萨克斯	萨克斯管	24
68	Reed	Oboe	Oboe	25
69	Reed	英国管	Horn	22
70	Reed	巴松管	巴松管	26
71	Reed	单簧管	单簧管	27
72	Pipe	短笛	短笛	28
73	Pipe	长笛	长笛	29
74	Pipe	竖笛	竖笛	30
75-79	Pipe	排箫 ，吹奏瓶 ，尺八 ， 哨笛 ，陶笛	Pipe	31
80-87	合成器主音	主音1-8	合成器主音	32
88-95	合成器铺底	铺底音色1-8	合成器铺底	33
96-103	合成器效果音	效果音1-8	合成器效果音	34
104-111	民族乐器	西塔琴、班卓琴、三味线、筝, 卡林巴琴、风笛、民间小提琴、唢呐	民族乐器	35
112-119	打击乐	叮当铃、阿哥哥铃、钢鼓、木鱼, 太鼓 ，旋律通鼓 ， 合成鼓	打击乐	36
120-127	音效	吉他品格噪声 ，呼吸噪声 ， 海浪声、鸟鸣声、电话铃声, 直升机、掌声、枪声	音效	37
128	鼓	鼓	鼓	38

表 11：我们实验中使用的乐器映射。我们的映射不如 MIDI 程序号细致，但比 MIDI乐器代码更精细，因此得到 39 种不同乐器。