MuseAgent-1：乐谱与演奏音频的交互式基于 grounding 的多模态理解

Qihao Zhao[1*]、Yun qi Cao[2*], Yangyu Huang[3],Hui Yi Leong[4],Fan Zhang[2],Kim-Hui Yap[1],Wei Hu [2†]

[1]南洋理工大学[2]北京化工大学[3]微软[4]
芝加哥大学

arXiv:2601.11968v1 [cs.MM] 17 Jan 2026

摘要

尽管多模态大语言模型（MLLMs）近期取得了进展，但它们理解音乐并与之交互的能力仍然有限。音乐理解需要基于 grounding 对符号乐谱和富有表现力的演奏音频进行推理，而通用型多模态大语言模型往往因感知 grounding 不足而难以胜任。我们推出了 MuseAgent，这是一种以音乐为中心的多模态智能体，通过由乐谱图像和演奏音频得出的结构化符号表示来增强语言模型。通过集成光学音乐识别和自动音乐转录模块，MuseAgent 能够针对细粒度音乐内容进行多步推理与交互。为系统评估音乐理解能力，我们进一步提出了 MuseBench，这是一个涵盖文本、图像和音频模态下的音乐理论推理、乐谱解读及演奏层面分析的基准测试。实验表明，现有多模态大语言模型在这些任务上的表现不佳，而 MuseAgent 则取得了显著提升，凸显了结构化多模态 grounding 对交互式音乐理解的重要性。

1 引言

> “缪斯女神啊，高贵的天才啊，请助我一臂之力！”
> 但丁，《神曲 · 地狱篇》，第二歌，第 7 行

音乐是一个兼具结构性与表现力的领域，因此成为检验人工智能的理想试验场。它涵盖记谱乐谱等符号表征，以及演奏等表现性声音实现，而二者分别需要不同的感知与推理能力。音乐既是一套形式系统，也是一种情感媒介，这使人工智能模型必须以高度精确性和对语境的细致把握，跨模态进行推理（Essid 和 Richard，2012年；Corrêard 等人，2021）。这些要求暴露了当前多模态系统的根本局限，尤其是缺乏对符号表示和时间表示进行结构化锚定的大语言模型。

在西方曲目中，钢琴音乐占据着独特的核心地位，这得益于数百年来形成的标准化记谱法和丰富的演奏档案。其宽广的音域、密集的复调以及双手协调，使钢琴音乐成为多模态推理系统极具挑战性的压力测试（Hawthorne 等，2019a）。理解钢琴音乐需要联合建模复杂的符号结构——如音高、节奏和力度——以及时值、奏法和弹性速度等富有表现力的演奏属性。这种整合对于转录、伴奏、数字归档和音乐教育等交互式应用至关重要（Benetos 等，2019）。

尽管各项独立任务已取得显著进展，现有方法仍不足以实现全面且交互式的音乐理解。先前研究推进了光学音乐识别（OMR）乐谱识别、音频自动音乐转录（AMT）等单模态问题。MAESTRO 等大规模数据集（Hawthorne 等，2019a）以及 MusicLM 等生成模型（Agostinelli 等，2023）表明，对符号和声学音乐表征进行建模是可行的，但仍无法支持细粒度的跨模态推理。近期的多模态大语言模型（MLLM），包括 GPT-4o（OpenAI，2023）和 Gemini（GeminiTeam，2023），虽展现出通用跨模态能力的潜力，但在需要精确符号解析或长篇演奏分析的音乐理解任务上表现不佳。这种不足主要源于缺乏领域特定的感知 grounding，以及无法在高时间分辨率下对齐符号模态与声学模态。

近年来，若干系统尝试弥合

*贡献相同。
†电子邮件：huwei@mail.buct.edu.cn

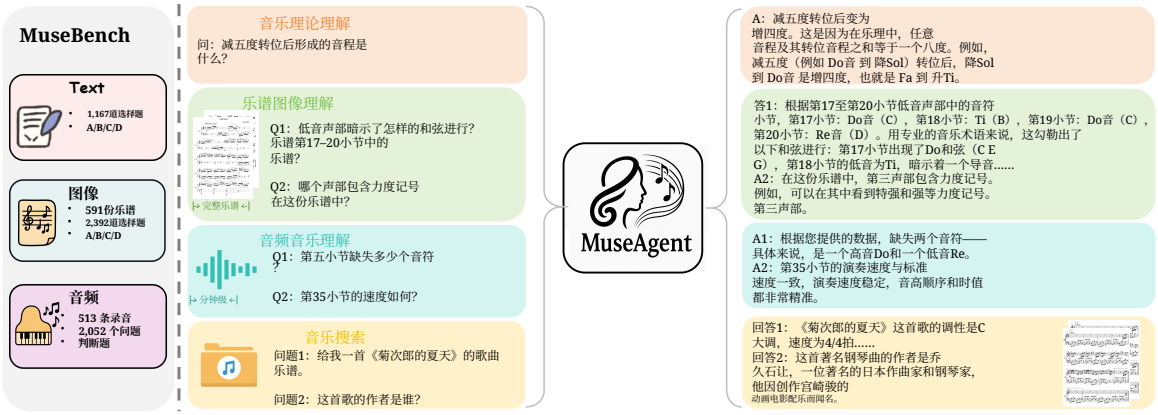


图 1：**MuseBench** 和**MuseAgent**概览。MuseBench包含涵盖文本、图像和音频模态的多模态音乐理解任务，涉及乐理、乐谱分析和演奏诠释。MuseAgent通过乐谱符号识别、音频对齐和检索模块整合这些模态，使大语言模型能够回答复杂的音乐问题。

这一差距，方法是为大语言模型接入外部工具。AudioGPT（Huang 等人，2024年）和 MusicAgent（Yu 等人，2023）是早期探索具备音乐感知能力的智能体的代表，它们将语言模型与领域专用组件相结合。然而，这些系统往往难以处理复杂乐谱记谱、长时值的演奏录音以及结构化交互，也缺乏针对音乐理解的系统化评估方案。与此同时，检索增强生成（RAG）（Lewis 等人，2020年；Guu 等人，2020年；Borgeaud 等人，2022）的进展表明，将语言模型建立在结构化外部表征之上，能够显著提升领域特定推理能力并减少幻觉。现有的多模态 RAG 框架（Shuster 等人，2022；Luo 等人，2023；Liu 等人，2023b）则主要关注文本—视觉—语音场景，很少涉及音乐领域，而在该领域中，符号乐谱必须与富有表现力的音频紧密对齐。

为应对这些挑战，我们提出了 MuseAgent，这是一种专为音乐理解与交互设计的多模态检索增强智能体。MuseAgent 将大型语言模型与专用感知前端相结合：(i) 逐小节光学音乐识别模块，可将乐谱图像转换为 ABC记谱法等符号表征（Yuan 等人，2024年a）；(ii) 基于自动音乐转录的演奏分析模块，可将音频录音与 MusicXML乐谱对齐，并以结构化 JSON格式提取表现特征；(iii) 检索模块，支持显式和隐式访问符号与音频库。这些组件将语言模型 grounding 在结构化多模态表征中，而记忆库则支持围绕音乐内容进行长上下文、多轮推理。

评估此类音乐理解智能体是一项不小的挑战。现有数据集和基准测试主要聚焦于转录或生成，无法评估智能体跨符号乐谱与演奏音频进行交互式推理的能力。为实现系统化评估，我们推出了 MuseBench，这是首个专为评估多模态音乐理解智能体而设计的基准测试。MuseBench以钢琴曲目为核心，包含乐谱–音频对齐、演奏错误检测和表现偏差分析等任务，并将 MAESTRO 等资源重新用于高层次推理任务，以评估智能体式音乐理解。

我们在 MuseBench 上进行的实验表明，通用多模态大语言模型在细粒度符号推理和演奏层面音乐推理方面的能力有限。相比之下，MuseAgent 在具有挑战性的图像和音频任务上取得了显著提升，验证了结构化感知落地和基于智能体的多模态推理对于交互式音乐理解的有效性。

2 相关工作

音乐理解系统与智能体。 近年来，ReAct、Auto-GPT 和 Gorilla 等通用智能体取得了进展，展示了工具增强推理在复杂任务中的有效性。在音乐领域，MuseNet（Payne，2019）和 Mu-

sicLM（Agostinelli 等人，2023）主要专注于符号音乐或音频音乐生成，而非音乐理解或交互式推理。近期面向音乐的多模态语言模型采用了领域特定的训练策略。例如，MusiLingo（Deng 等人，2024年）通过指令微调音频—语言建模，面向音乐描述生成和问答；MuMu-LLaMA（Liu 等人，2024年）在单一多模态框架中统一音乐音频、图像和语言；NotaGPT（Tang等人，2025年）则通过视觉—语言模型专注于音乐记谱理解。SymphonyNet（Liu 等人，2022）进一步展示了管弦乐规模的符号音乐生成。

尽管这些领域专用模型凸显了多模态对齐和专门训练的价值，但它们通常采用单体式架构，且不显式支持结构化交互、工具调用或多步推理。与此同时，MusicAgent（Yu 等人，2023）和 Audio-GPT（Huang 等人，2024年）等基于智能体的系统开始将大型语言模型与领域专用工具相结合，以处理音乐输入。尽管前景可期，这些系统并非为细粒度音乐理解而设计，而此类理解需要对符号乐谱和富有表现力的演奏音频进行联合推理；它们也不以复杂的钢琴曲目为目标。相比之下，我们提出的*MuseAgent* 通过引入一种以音乐为中心的智能体架构，推进了这一研究方向。该架构整合了结构化感知落地、基于检索的推理，以及跨符号和声学模态的智能体编排。

音乐理解基准。开发音乐理解智能体的一大关键挑战在于评估。现有音乐数据集，如 MAESTRO（Hawthorne 等，2019b）和 URMP（Li 等，2018年），提供了对齐的乐谱—音频配对数据，广泛用于转录、对齐和生成任务。然而，这些资源缺乏面向任务的评估协议，无法评估高层次推理或交互式理解。近期的多模态问答数据集，包括 MUSIC-AVQA（Li 等，2022）和 MuChoMusic（Weck 等，2024年），推动音乐基准迈向问答场景，但并未强调对符号乐谱的细致解读或细腻的演奏层面推理。相比之下，我们提出的*MuseBench* 旨在专门用于评估音乐理解智能体，考察其在一套统一的以钢琴为中心的任务中，联合基于文本、乐谱图像和演奏音频进行推理的能力。

通用多模态语言模型。GPT-4o（OpenAI，2023）、Gemini（Anil 等人，2023）、Qwen（Team，2024年）和 LLaVA（Liu 等人，2023a）等通用多模态语言模型，在文本—视觉基准测试中表现出色。然而，近期研究（Weck 等人，2024年）表明，这些模型在音乐理解任务上表现不佳，尤其是在面对结构化音乐记谱或富有表现力的演奏音频时。由于缺乏显式的符号落地，这类模型往往依赖语言先验，或产生关于音乐内容的幻觉。这些局限凸显了对具备音乐感知能力的智能体的需求；此类智能体整合领域专用感知与结构化推理，我们的 MuseAgent 正是一个例子。

我们工作的独特性。总而言之，先前的音乐理解研究主要聚焦于孤立模态、单体领域模型或面向生成的目标，而现有基准测试缺乏评估音乐领域交互式基于智能体的推理能力的手段。我们的工作独树一帜，将以音乐为中心的多模态智能体与面向智能体的基准测试相结合。MuseAgent 能够对符号乐谱和富有表现力的演奏音频进行结构化推理，而 MuseBench 则提供了一个统一的评估框架，专门用于评估此类能力。二者共同为研究细粒度、交互式多模态音乐理解奠定了全新的基础。

3 MuseAgent

我们提出了**MuseAgent**，这是一种面向结构化、交互式音乐理解的多模态检索增强智能体。与固定的感知—推理流水线不同，MuseAgent采用智能体编排循环，由大语言模型根据用户意图动态协调领域专用感知模块、检索操作和符号推理。该设计使MuseAgent能够围绕符号乐谱和富有表现力的演奏音频开展多步推理，从而应对面向智能体的音乐理解任务所带来的挑战，例如MuseBench中定义的任务。

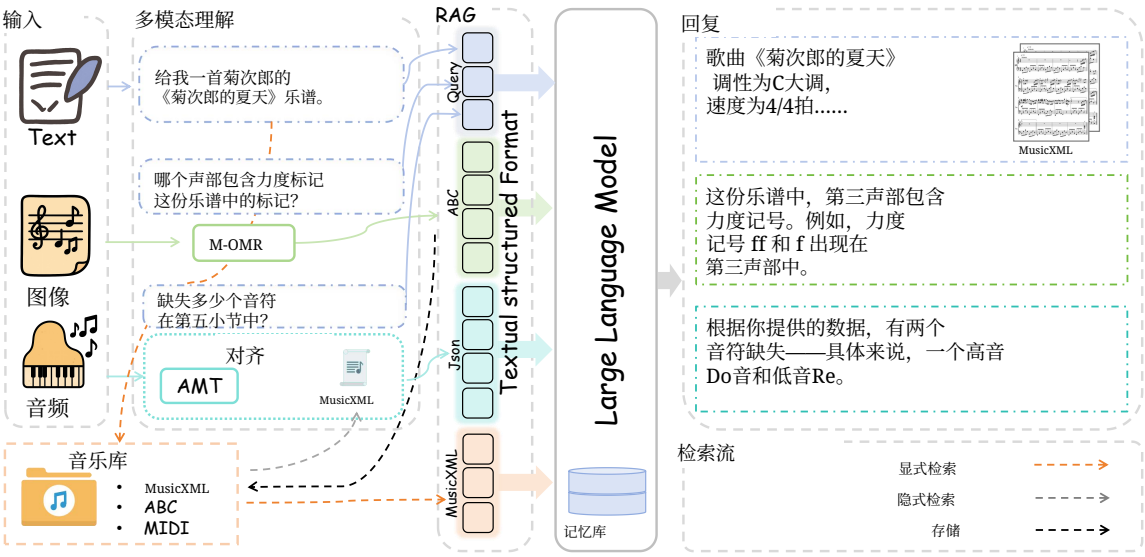


图2：MuseAgent框架将M-OMR、自动音乐转录和音乐检索（显式/隐式）整合到一个基于大语言模型（LLM）的统一系统中。每个感知模块都会将原始多模态输入转换为结构化符号表示（例如 ABC、MusicXML、JSON），并将其纳入检索增强生成（RAG）流水线。大语言模型充当智能体控制器，根据用户意图动态协调各模块的使用；记忆库则支持多轮对话，并检索先前的输出。

如图 2所示，MuseAgent整合了三个核心组件：(i) 感知定位模块，将原始多模态输入转换为结构化符号表示；(ii) 检索增强生成（RAG）机制，以外部音乐知识为推理提供依据；以及 (iii) 记忆库，支持长上下文、多轮交互。

3.1 智能体工作流

MuseAgent的核心是一个基于大语言模型的控制器，它负责解读用户查询，并动态确定要调用哪些感知或检索模块。例如，涉及和声结构或记谱法的查询会触发逐小节光学音乐识别模块，而有关时值、速度稳定性或演奏准确性的问题则会调用基于自动音乐转录的对齐。检索既可由用户请求显式触发，也可由智能体在需要补充符号上下文时隐式启动。这种智能体工作流支持跨模态的灵活、意图驱动推理，而非依赖静态、预定义的处理流水线。

3.2 逐小节光学音乐识别

多模态大语言模型（MLLMs）在音乐理解方面面临的一项关键挑战，是高分辨率乐谱图像与音乐分析所需的符号推理之间存在模态差距。与自然图像不同，乐谱结构密集且具有特定领域属性，其中编码了音高、节奏和力度等层级元素，而通用视觉语言模型难以直接解读这些内容。

为应对结构化乐谱识别的挑战，我们提出了一种基于“分而合”策略的逐小节光学音乐识别（M-OMR）模块，用作感知落地模块。首先，利用谱表线和小节线等视觉布局线索，将输入的乐谱图像划分为一个个独立的小节。每个小节都被视为一个独立的视觉单元，并通过局部识别编码为符号表示。随后，将这些小节级输出组合起来，重建完整的音乐作品。最终结果以 ABC记谱法表示；这种记谱法是一种紧凑且结构化的符号格式，非常适合用于下游语言模型处理。

不同于以往的光学音乐识别方法 NotaGPT（Tang等人，2025年），该方法在音符级别切分乐谱图像，并依赖冻结的视觉和文本编码器；我们采用逐小节分割策略。两者的差异如图3所示。我们将每个小节视为一个语义单元，从而保留音乐结构，并减少过度细粒度切分带来的噪声。我们训练了一个基于 ResNet 的（He等人，2016）视觉编码器，

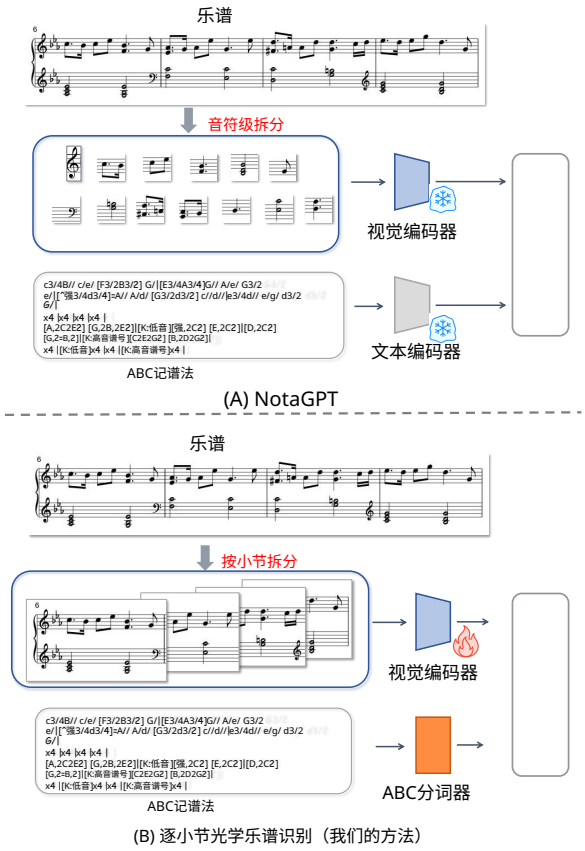


图3：(A)*NotaGPT* 与 (B) 我们提出的 逐小节光学乐谱识别 方法的比较。火焰符号表示可训练模块，雪花符号表示冻结组件。

并与长短期记忆网络（Yu 等人，2019）联合训练，以捕捉乐谱序列内部的依赖关系。此外，我们专门设计了一个适用于 ABC 记谱法表示的 ABC分词器。该分词器涵盖数百种音乐专属构造（例如调性、拍号、和弦）的 ABC 记谱法变体。与通用文本编码器相比，它生成的词元序列更紧凑、结构意义更明确。更多详情见附录D。

3.3 自动音乐转录与对齐

为了理解富有表现力的演奏音频，MuseAgent 将自动音乐转录（AMT）模块和音频到乐谱对齐组件作为音频感知基础模块。AMT 模块通过常数Q变换提取时频特征（Schörkhuber and Kla-puri，2010），并应用神经转录模

els.

型，将原始音频转录为符号表示（例如 MusicXML）。随后，使用分层隐马尔可夫模型（H-HMM）（Nakamuraet al.，2015）将生成的符号序列与参考乐谱进行时间对齐。该模型能够稳健应对表现性时值变化、装饰音，以及重复或跳过等结构偏差。对齐过程会生成 JSON 格式的结构化输出，记录起音时值、音符对应关系和表现参数。

随后，这些对齐结果会与用户提示以及从音乐库中隐式检索到的 MusicXML 文件融合，作为检索增强生成（RAG）模块的输入。RAG 组件将这些多模态元素整合为内容更丰富的提示，使语言模型能够基于符号和听觉演奏数据进行推理。实现细节，包括模型架构和训练配置，见附录 E。

3.4 音乐检索模块

MuseAgent 支持从大型符号音乐库中进行显式和隐式检索，音乐库包含 ABC、MusicXML 和 MIDI 等格式。

对于**显式检索**，用户可以直接使用自然语言查询（例如，“给我一首《菊次郎的夏天》的歌曲”）来获取匹配的乐谱。对于**隐式检索**，系统会根据音频和乐谱上下文执行内部搜索，选择相关的符号文件（例如与音频配对的 MusicXML），并将其整合到 RAG 流程中。

与传统信息检索方法不同，这里的检索嵌入在智能体循环中：大语言模型可以直接回应用户查询，也可以隐式调用检索 API，为其推理提供依据。这种设计实现了面向多模态音乐的智能体检索增强生成。

3.5 乐理理解与对话上下文

除了上述能力外，MuseAgent 还利用大语言模型中蕴含的内在音乐知识来回答乐理问题（例如：“将减五度转位后会得到什么音程？”或“C 大调音阶中以 E 音为起始音的是哪种调式？”）。这项能力的效果可能因所使用的大语言模型而异。有关不同大语言模型乐理理解能力的评估，请参阅第5.1.2节。

MuseAgent 还支持多轮对话的记忆功能。为此，我们维护了一个轻量级的**记忆库**，用于存储中间模块输出、检索到的文件以及先前的模型响应。记忆库不仅支持多轮推理，还能检索先前的结构化输出。

4 MuseBench

为系统性评估音乐理解智能体，我们推出了**MuseBench**，这是一个旨在评估音乐领域结构化推理与交互式推理能力的多模态基准测试。MuseBench 面向融合感知、检索和基于语言的推理能力的智能体，并评估其对**文本**、**乐谱图像**和**演奏音频**进行联合推理的能力。MuseBench 不以孤立的识别或生成技能为衡量目标，而是专注于高层次的音乐理解任务；这些任务需要符号落地、跨模态对齐和多步推理——这些能力是 MuseAgent 等智能体框架的核心。

4.1 数据来源

为支持面向智能体的评估，MuseBench 由高质量的符号、视觉和声学音乐数据构建而成。该基准测试涵盖多种风格、时期和难度等级，包括巴洛克时期、古典时期、浪漫主义时期及当代钢琴曲目。数据集概览见图1，更多来源详情请参阅附录B.1。

4.2 数据集构建

4.2.1 预处理

我们最初从多个开放存储库收集了 ~3,000 份候选乐谱（参见附录B.1）。我们根据完整性、可读性和分辨率质量对乐谱进行筛选。经过归一化处理（调整分辨率、去除背景噪声和五线谱线校正）后，保留了 ~600 张乐谱图像。在音频方面，我们收集了 513 段高质量演奏录音。我们将每个音频文件统一为相同的采样率，并进行后期处理（降噪、归一化），以确保音质清晰。为避免侵犯版权，我们仅纳入依据公共许可协议发布的录音，或由音乐家明确提供并附有书面同意的录音。

4.2.2 标注

每张乐谱图像都与一个 ABC 格式的符号文件配对，其中包含标题、作曲家、调性、拍号、音符时值和节奏等元数据。专业音乐家还对技术难度和演奏相关要素进行了标注。每首曲目都与专业钢琴演奏录音进行对齐，并转换为带有小节级乐谱—音频对齐信息的 MusicXML，从而形成用于后续任务构建的标准化元数据池。所有标注均由受过培训的音乐家完成，他们至少接受过五年正规音乐教育。为确保可靠性，多位标注人员对标签进行了交叉验证，Cohen's κ 达到 0.87。

4.2.3 定义

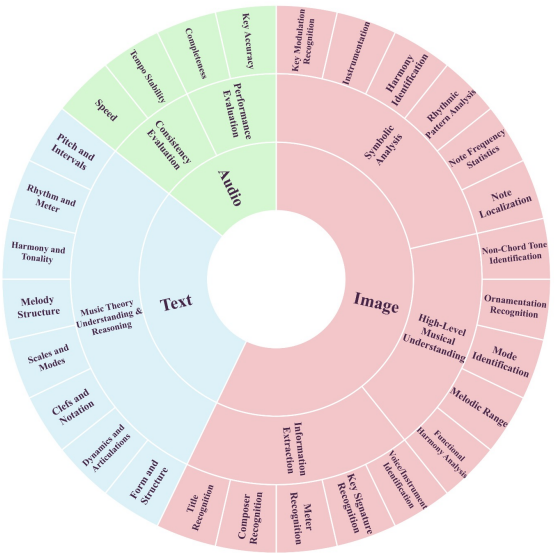


图 4：MuseBench 中问题的分布。该基准测试包含三个模态下的 28 种任务类型。任务分布相对均匀，以确保评估均衡。

为了评估多模态大语言模型在音乐理解和推理方面的能力，我们与专业音乐家合作设计了**MuseBench**这一基准测试。它围绕三个核心维度构建：（1）**音乐理论理解**，（2）**乐谱理解**，以及（3）**演奏音频分析**。任务定义和评估标准均经专家共识确立，以确保相关性和严谨性。

附录中详细介绍了各子任务及其设计依据C。如图 4 所示，我们共定义了 **28 项具体任务**，涵盖三种模态和六个子维度，确保问答对和任务格式的分布均衡。我们的

设计借鉴了 MMMU（Yue 等人，2024年）和 OmniBench （ Li 等人，2024 年）等评测框架，同时明确考虑任务难度和模态多样性，以支持稳健的基准测试。

5 实验

5.1 MuseBench 上的比较

5.1.1 基线

我们在 MuseBench 上将 MuseAgent 与 17 个具有代表性的多模态语言模型进行比较，以考察基于智能体的 grounding 对音乐理解的有效性。所选基线涵盖多种模型范式，包括：**(i) 通用语言模型**，如 GPT-4.1、GLM-4 和 Phi-4；**(ii)全模态模型**，包括 GPT-4o、Gemini 2.5-Pro 和 Qwen2.5-Omni；**(iii) 视觉—语言模型**，如 LLaVA、VisualGLM 和 Qwen2.5-VL（ 7B/32B/72B）；**(iv) 具备音频处理能力的模型**，包括 Qwen2-audio 和 MuMu-LLaMA；以及 **(v)音乐专用模型**，如用于音乐记谱理解的 NotaGPT。

尽管这些模型在模态覆盖范围和领域专精方面各不相同，但它们主要被设计为单体模型，缺乏明确的智能体编排、结构化感知落地或多步检索推理。这组多样化的基线使我们能够系统地评估，相较于非智能体及 grounding 能力较弱的多模态系统，MuseAgent 的智能体设计在细粒度音乐理解任务上的优势。

表 1：MuseBench 性能比较。MuseAgent 显著提升了图像和音频理解能力，优于全模态和音乐专用基线模型。

模态	模型	准确率（%）
Text	GPT-4.1（OpenAI，2025年）	**86.7**
	GPT-4o（OpenAI，2023）	85.5
	Gemini2.5-Pro（Comanici 等，2025年）	83.9
	GPT-4.1-mini（OpenAI，2025年）	80.5
	GPT-4.1-nano（OpenAI，2025年）	73.3
	GLM-4-PLUS（GLM 等人，2024年）	78.2
	GLM-4-FlashX（GLM 等人，2024年）	60.3
	Qwen2.5-72B（Team，2024年）	81.4
	Qwen2.5-32B（Team，2024年）	79.8
	Qwen2.5-7B（Team，2024年）	71.3
	Qwen2.5-Omni-7B（Xu 等，2025年）	63.3
	Phi-4-14B（Abdin 等，2024年）	67.2
	随机	25.0
音频	MuseAgent（使用 GPT-4.1）	**79.1**
	MuseAgent（使用 GPT-4o-mini）	78.9
	MuseAgent（使用 GLM-4-FlashX）	77.2
	MuseAgent（使用 GPT-4.1-Nano）	63.9
	GPT-4o (OpenAI，2024年)	55.9
	Gemini2.5-Pro (Comanici 等，2025年)	53.1
	MuMu-LLaMA(Liu 等人，2024年)	51.7
	Qwen2-audio（Team，2024年）	51.4
	Qwen2.5-Omni-7B(Xu 等，2025年)	50.6
	随机	50.0
图像	MuseAgent（使用 GPT-4.1）	**74.1**
	MuseAgent（使用 GLM-4-FlashX）	72.7
	NotaGPT-7B (Tang等人, 2025年)	68.1
	GPT-4.1 (OpenAI，2025年)	66.1
	GPT-4o（OpenAI，2024年）	64.2
	GPT-4.1-mini（OpenAI，2025年）	54.8
	Gemini2.5-Pro (Comanici 等，2025年)	62.1
	Qwen2.5-VL-72B (Team, 2024年)	58.9
	Qwen2.5-VL-32B (Team，2024年)	55.7
	Qwen2.5-Omni-7B(Xu 等，2025年)	44.6
	LLaVA-v1.5-13B (Liu 等人，2023a)	38.9
	GLM-4V-9B（GLM 等人，2024年）	37.1
	随机	25.0

5.1.2 结果分析

文本模态。 在纯文本任务中，通用语言模型表现出色。GPT-4.1 的准确率最高（86.7%），紧随其后的是 GPT-4o（85.5%）和 Gemini 2.5-Pro（83.9%）。模型规模扩大带来的提升较为有限，Qwen2.5-72B（81.4%）和 Qwen2.5-32B（79.8%）的结果便体现了这一点，这表明音乐理论问题在很大程度上契合大型语言模型的原生能力。

音频模态。 相比之下，通用全模态模型难以进行演奏级别的音频推理。GPT-4o（55.9%）和 Gemini 2.5-Pro（53.1%）的表现仅略高于随机水平，而 Qwen2.5-Omni（50.6%）和专门处理音频的 Qwen2-Audio（51.4%）也表现出类似的局限性。这些结果表明，仅凭原始音频输入不足以实现细粒度音乐理解。通过明确结合自动音乐转录和乐谱–音频对齐，配备 GPT-4.1 的 MuseAgent 达到了 79.1% 的准确率，表明结构化感知落地对于演奏表现的推理至关重要。

图像模态。 视觉—语言模型在理解符号音乐记谱方面存在显著困难。LLaVA（38.9%）和 GLM-4V（37.1%）等模型表现不佳，反映出密集且具有领域特性的乐谱表示所带来的挑战。GPT-4o（64.2%）和 Gemini 2.5-Pro（62.1%）等更大型的全模态模型表现有所提升，但准确率仍然有限，GPT-4.1 的准确率为 66.1%。结合所提出的逐小节光学乐谱识别（M-OMR）模块后，MuseAgent 的准确率达到 74.1%，超过了通用模型以及 NotaGPT（68.1%）等音乐专用基线。

图5：图像到ABC转换比较。评估结果基于NotaGPT中引入的标准化基准测试（Tang等人，2025年）。

模型	Levenshtein距离
VisualGLM-6B	643.72
DeepSeek-VL-7B-Chat	308.27
LLaVA-v1.5-13B	147.47
LLaVA-v1.6-Vicuna-13B	918.94
Qwen-VL	439.82
NotaGPT-7B	59.47
Gemini-pro-vision	354.30
GPT-4V	655.45
M-OMR（我们的方法）	**18.39**

图6：开源模型与基于API的模型对比。

模型	LSA	ROUGE-1	ROUGE-L	METEOR	Avg
InternVL-Chat-v1.5	14.96	19.71	13.32	19.68	16.92
VisualGLM-6B	10.36	21.61	13.21	18.19	15.84
DeepSeek-VL-7B-base	9.92	16.43	11.60	13.81	12.94
InstructBLIP-Vicuna-7B	8.28	22.23	14.93	16.74	15.55
InstructBLIP-Vicuna-13B	8.37	20.29	14.18	14.17	14.25
Qwen-VL	9.58	15.21	10.37	12.56	11.93
Qwen-VL-Chat	9.66	16.80	11.37	14.42	13.06
NotaGPT-7B	12.46	22.63	15.53	18.34	17.24
Gemini-pro-vision	15.88	22.21	15.09	**20.31**	18.37
GPT-4V	14.03	18.49	11.36	19.94	15.96
GPT-4o	**15.92**	18.27	11.35	20.26	16.45
MuseAgent（搭配 M-OMR）	15.75	**24.92**	**15.76**	20.17	**19.15**

5.2 M-OMR 与视觉语言模型的演奏评估

为单独评估 MuseAgent 中结构化视觉感知的贡献，我们进一步将所提出的 M-OMR 模块与最先进的视觉语言模型（VLMs）在乐谱理解任务上的表现进行比较。遵循 NotaGPT（Tang等人，2025年）的评测方案，我们考察两种设置：（i）**封闭集**将乐谱转换为 ABC记谱法，并使用莱文斯坦距离进行评估；（ii）**开放集**视觉音乐分析，使用包括 LSA、ROUGE 和 METEOR 在内的语义指标进行评估。这种受控比较使我们能够评估，逐小节符号层面的落地是否比整体式视觉—语言建模更有利于结构化音乐理解。

5.2.1 封闭集图像到 ABC 记谱法

我们评估了八种具有代表性的多模态大语言模型，包括基于 API 的模型（例如 GPT-4V、Gemini Pro）和开源模型（例如 LLaVA、VisualGLM、Qwen-VL-32B、NotaGPT-7B）。如表 5 所示，M-OMR 的 Levenshtein距离最低（18.39），远胜于 NotaGPT-7B（59.47）和 LLaVA-13B（147.47）等所有基线。这些结果表明，M-OMR 在符号记谱转换方面具有更高的结构准确性，凸显了其在封闭集任务中的稳健性。

5.2.2 开放集乐谱理解

对于开放集任务，我们比较各模型在语义相似度和内容相关性方面的表现（表6）。结合 M-OMR 的 MuseAgent 平均得分最高（19.15），优于 GPT-4o（16.45）和 Gemini Pro（18.37）等强大的 API 基线，也优于开源视觉—语言模型。各项指标上的提升都很稳定：更高的 ROUGE-1 和 METEOR 反映出更全面的内容覆盖和更流畅的表达，而 LSA 的提升则凸显了 M-OMR 捕捉细腻音乐语义的能力。综合来看，这些结果证明，M-OMR 是 MuseAgent 中用于乐谱解读的稳健可靠方案。

6 结论

我们推出了 **MuseBench**，一个全面的多模态音乐理解基准测试；以及 **MuseAgent**，一个集成了符号乐谱解析和演奏音频转录的模块化智能体。MuseBench 涵盖乐理、乐谱和演奏三个维度的 28 项任务，为评估多模态大语言模型的推理能力提供了严格的测试平台。实验表明，尽管通用大语言模型在基于文本的任务上表现出色，但在精细的乐谱和音频理解方面仍有困难。通过整合 M-OMR 和自动音乐转录等模态专属模块，MuseAgent 在图像和音频模态上均取得了显著提升，证明了具备领域知识的感知前端不可或缺。这些发现凸显了通用模型单纯扩大规模的局限，并证实了模块化集成在复杂音乐推理中的有效性。我们希望这项工作能为未来的人工智能辅助音乐分析、音乐创作和音乐教育研究奠定基础，并推动多模态基准测试从文本、视觉和语音扩展到丰富的音乐领域。

局限性

我们的基准测试和模型目前聚焦于钢琴相关任务，这是经过审慎考量的设计选择，旨在利用大规模数据的可获得性以及钢琴记谱法的标准化特性。虽然这使得评估能够在受控条件下进行，但也限制了对其他乐器和记谱系统的直接覆盖。

尽管如此，只要具备适当的特定任务数据，MuseBench 和 MuseAgent 都经过设计，可扩展至其他领域。

参考文献

Marah Abdin、Jyoti Aneja、Harkirat Behl、Sébastien Bubeck、Ronen Eldan、Suriya Gunasekar、Michael Harrison、Russell J. Hewett、Mojan Javaheripi、Piero Kauffmann、James R. Lee、Yin Tat Lee、Yuanzhi Li、Weishung Liu、Caio C. T. Mendes、Anh Nguyen、Eric Price、Gustavo de Rosa、Olli Saarikivi及其他8位作者。2024年。Phi-4 技术报告。预印本，arXiv:2412.08905。

Andrea Agostinelli、Timo I. Denk、Zalán Borsos、Jesse Engel、Mauro Verzetti、Antoine Caillon、Qingqing Huang、Aren Jansen、Adam Roberts、Marco Tagliasacchi、Matt Sharifi、Neil Zeghidour及Chris- tian Frank。2023。MusicLM：从文本生成音乐。*arXiv* 预印本 *arXiv:2301.11325*。

Rohan Anil、Sebastian Borgeaud、Jean-BaptisteAlayrac 等 1 位作者。2023年。Gemini：功能强大的多模态模型系列。*arXiv* 预印本 *arXiv:2312.11805*。

Emmanouil Benetos、Simon Dixon、Dimitrios Giannoulis、Hanno Kirchhoff 和 Anssi Klapuri。2019年。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。

Sebastian Borgeaud、Arthur Mensch、Jordan Hoffmann、Trevor Cai、Eliza Rutherford、Katie Millican、George van den Driessche、Jean-Baptiste Lespiau、Bogdan Damoc、Aidan Clark 等 1 位作者。2022年。通过从数万亿个词元中检索来改进语言模型。*arXiv* 预印本 *arXiv:2112.04426*。

Gheorghe Comanici、Eric Bieber、Mike Schaekermann、Ice Pasupat、Noveen Sachdeva、Inderjit Dhillon、Marcel Blistein、Ori Ram、Dan Zhang、Evan Rosen 等 1 位作者。2025年。Gemini 2.5：凭借先进推理、多模态、长上下文和新一代智能体能力，突破前沿。*arXiv* 预印本 *arXiv:2507.06261*。

Florian Corrêard、Yuki Wang、Sebastian Ewert、Manuel Moussallam 和 Mark Sandler。2021年。用于音乐音频表征学习的跨模态变换器。第*22*届 *ISMIR* 论文集。

Zihao Deng、Yinghao Ma、Yudong Liu 等1人。2024年。MusiLingo：利用预训练语言模型连接音乐与文本，实现音乐描述生成和查询响应。载于*2024*年北美计算语言学协会年会论文集（*NAACL*）。

Slim Essid 和 Gaël Richard。2012年。《音乐信息检索：从符号到语义音频分析》。*IEEE*信号处理杂志，29(1)：118–129。

Gemini Team。2023。Gemini：一系列能力出众的多模态模型。*arXiv* 预印本 *arXiv:2312.11805*。Team GLM、Aohan Zeng、Bin Xu、Bowen Wang、Chen- hui Zhang、Da Yin、Diego Rojas、Guanyu Feng、Han- lin Zhao、Hanyu Lai、Hao Yu、Hongning Wang、Jiadai Sun、Jiajie Zhang、Jiale Cheng、Jiayi Gui、Jie Tang、Jing Zhang、Juanzi Li，以及另外37位作者。2024年。Chatglm：从 glm-130b 到 glm-4 全工具系列大语言模型。预印本，arXiv:2406.12793。Kelvin Guu、Kenton Lee、Zora Tung、Panupong Pasu- pat 和 Ming-Wei Chang。2020年。Realm：检索增强型语言模型预训练。载于国际机器学习会议。

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

Curtis Hawthorne、Andriy Stasyuk、Adam Roberts、Ian Simon、Cheng-Zhi Anna Huang、Sander Dieleman、Erich Elsen、Jesse Engel 和 Douglas Eck。2019a。借助 MAESTRO 数据集实现因子化钢琴音乐建模与生成。载于国际学习表征会议（*ICLR*）。

Curtis Hawthorne、Andriy Stasyuk、Adam Roberts、Ian Simon、Cheng-Zhi Anna Huang、Sander Dieleman、Erich Elsen、Jesse Engel 和 Douglas Eck。2019b。使用 MAESTRO数据集实现因子化钢琴音乐建模与生成。载于国际学习表征会议 *(ICLR)*。

Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。2016。用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议论文集，第770–778 页。

Rongjie Huang、Mingze Li、Dongchao Yang、Jia-tong Shi、Xuankai Chang、Zhenhui Ye、Yuning Wu、Zhiqing Hong、Jiawei Huang、Jinglin Liu、Yi Ren、Zhou Zhao 和 Shinji Watanabe。2024年。AudioGPT：理解并生成语音、音乐、声音和说话头像。载于第 *38* 届 *AAAI* 人工智能会议论文集 *(AAAI)*。即将发表，arXiv:2304.12995。

IMSLP / 彼得鲁奇音乐图书馆。2025年。国际乐谱图书馆项目（imslp）。访问日期：2025- 05-09。

Patrick Lewis、Ethan Perez、Aleksandra Piktus、Fabio Petroni、Vladimir Karpukhin、Naman Goyal、Sid- dharth Kulkarni、Alessandro Miaschi、Veselin Stoy- anov 和 Douwe Kiela。2020年。面向知识密集型自然语言处理任务的检索增强生成。发表于神经信息处理系统进展。

Bochen Li、Xinzhao Liu、KarthikDinesh 等。2018年。 创建一个多轨古典音乐

演奏数据集，用于多模态音乐分析：挑战、见解与应用。 *IEEE*多媒体汇刊，20(9):2382–2395。Guangyao Li、Yake Wei、Yapeng Tian 等。2022年。 在动态视听场景中学习回答问题。载于 *IEEE/CVF*计算机视觉与模式识别会议论文集（*CVPR*），第5672–5682页。

Yizhi Li、Ge Zhang、Yinghao Ma、Ruibin Yuan、Kang Zhu、HangyuGuo、Yiming Liang、JiahengLiu、Zekun Wang、Jian Yang 等人。2024年。OmniBench：迈向通用全语言模型的未来。*arXiv* 预印本 *arXiv:2409.15272*。

Haotian Liu、Chunyuan Li 和 Qiang Liu。2023a。LLaVA：大语言与视觉助手。*arXiv* 预印本 *arXiv:2304.08485*。

Jiafeng Liu、Yuanliang Dong、Zehua Cheng 等人。2022年。基于置换不变语言模型的交响乐生成。载于国际音乐信息检索学会会议论文集（*ISMIR*）。

Shansong Liu、Zhenhua Ma、YixiaoZheng 等人。2024年。MuMu-LLaMA：通过大语言模型实现多模态音乐理解与生成。*arXiv* 预印本 *arXiv:2412.06660*。

Zekai Liu、Yiming Yu、Yizeng Li、Chen Sun、Hao Ma 和 Tie-Yan Yu。2023b。视觉RAG：面向视觉语言模型的检索增强生成。*arXiv* 预印本 *arXiv:2305.17590*。

Zujie Luo、Shuo Yang、Jingang Yuan、Yankai Lin 和 Maosong Sun。2023。面向知识密集型任务的多模态检索增强生成。载于计算语言学协会 *EMNLP 2023* 论文集。

MuseScore团队。2025年。MuseScore官方网站。访问日期：2025年5月9日。

Tomohiko Nakamura、Eita Nakamura 和 Shigeki Sagayama。2015 年。包含错误以及任意重复和跳过的音乐演奏实时音频到乐谱对齐。*IEEE/ACM* 音频、语音与语言处理汇刊，24(2)：329–339。OpenAI。2023 年。GPT-4 技术报告。*arXiv* 预印本 *arXiv:2303.08774*。OpenAI。2024年。GPT-4o mini：推进高性价比智能。OpenAI。2025年。在 API 中推出 GPT-4.1。Christine Payne。2019 年。MuseNet。OpenAI博客。Christian Schörkhuber 和 Anssi Klapuri。2010 年。用于音乐处理的常数Q变换工具箱。载于第七届声音与音乐计算会议，西班牙巴塞罗那，第 3–64 页。SMC。

Kurt Shuster、Samuel Humeau、Stephen Roller、Emily Dinan 和 Jason Weston。2022 年。寻求知识的语言模型：面向多模态问答的模块化搜索与生成。*arXiv* 预印本 *arXiv:2203.05482*。Sima Siami-Namini、Neda Tavakoli 和 Akbar Siami Namin。2019 年。长短期记忆网络和双向长短期记忆网络在时间序列预测中的表现。载于*2019 IEEE* 国际大数据会议（*Big Data*），第 3285–3292 页。IEEE。

Mingni Tang、JiajiaLi、Lu Yang、ZhiqiangZhang、Jing- hao Tian、Zuchao Li、Lefei Zhang 和 Ping Wang。2025年。Nota：面向视觉大语言模型的多模态音乐记谱理解。预印本，arXiv:2502.14893。

Qwen团队。2024年。Qwen2.5：一系列基础模型。访问日期：2025年5月15日。

Mutopia计划。2025年。Mutopia计划：免费乐谱。访问日期：2025年5月9日。

Rejin Varghese 和 M. Sambath。2024年。YOLOv8：一种性能和鲁棒性均得到提升的新型目标检测算法。载于 *2024*年数据工程与智能计算系统进展国际会议（*ADICS*）。IEEE。

Chris Walshaw。2021年。abc音乐标准 2.1（2011年12月）。

Benno Weck、Ilaria Manco、Emmanouil Benetos 和 Elio Quinton。2024年。MuChoMusic：评估多模态音频语言模型的音乐理解能力。*arXiv* 预印本 *arXiv:2408.01337*。

Jin Xu、Zhifang Guo、Jinzheng He、Hangrui Hu、Ting He、Shuai Bai、Keqin Chen、Jialin Wang、Yang Fan、Kai Dang及其他1位作者。2025年。Qwen2.5-omni技术报告。*arXiv* 预印本 *arXiv:2503.20215*。

Dingyao Yu、Kaitao Song、Peiling Lu、Tianyu He、Xu Tan、Wei Ye、Shikun Zhang 和 Jiang Bian。2023。MusicAgent：利用大语言模型进行音乐理解与生成的人工智能智能体。 *arXiv* 预印本 *arXiv:2310.11954*。Yong Yu、Xiaosheng Si、Changhua Hu 和 Jianxun Zhang。2019。循环神经网络综述：长短期记忆单元与网络架构。*Neural compu-tation*，31(7):1235–1270。Ruibin Yuan、Hanfeng Lin、Yi Wang、Zeyue Tian、Shangda Wu、Tianhao Shen、Ge Zhang、Yuhang Wu、Cong Liu、Ziya Zhou、Liumeng Xue、Ziyang Ma、Qin Liu、Tianyu Zheng、Yizhi Li、Yinghao Ma、Yim- ing Liang、Xiaowei Chi、Ruibo Liu 及另外 13 位作者。2024a。ChatMusician：以内在方式理解和生成音乐。载于 计算语言学协会论文集：*ACL 2024*，第 6252–6271 页，泰国曼谷。计算语言学协会。

Ruibin Yuan、Hanfeng Lin、Yi Wang、Zeyue Tian、Shangda Wu、Tianhao Shen、Ge Zhang、Yuhang Wu、Cong Liu、Ziya Zhou 及另外 1 位作者。2024b。Chatmusician：以内在方式理解和生成音乐，并使用大语言模型。*arXiv* 预印本 *arXiv:2402.16153*。

Xiang Yue、Yuansheng Ni、Kai Zhang、Tianyu Zheng、Ruoqi Liu、Ge Zhang、Samuel Stevens、Dongfu Jiang、Weiming Ren、Yuxuan Sun 等另外 1 位作者。2024年。Mmmu：面向专家级通用人工智能的海量多学科多模态理解与推理基准测试。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 9556–9567 页。

Li Yujian 和 Liu Bo。2007。一种归一化莱文斯坦距离指标。*IEEE*模式分析与机器智能汇刊，29(6):1091–1095。

Zhiyuan Zhao、Hengrui Kang、Bin Wang 和 Conghui He。2024年。Doclayout-yolo：通过多样化合成数据和由全局到局部的自适应感知，增强文档版面分析。*arXiv*预印本 *arXiv:2410.12628*。

A 附录

B 数据集详情

B.1 数据来源与筛选标准

为确保多样性与法律合规性，我们选取的乐谱满足以下条件之一：(i) 属于公有领域作品（作曲家于1954年之前去世）；或 (ii) 明确以知识共享等开放许可发布。未包含过去75年内的任何受版权保护的材料。所有录音均来自公有领域数据集，或由签署知情同意协议的专业音乐家提供。

该数据集包括：

- **乐谱图像：**
  - **MuseScore**（MuseScore团队，2025年）：~300由社区创建的乐谱（以知识共享许可重新发布），包括现代音乐和流行音乐。– **IMSLP**（IMSLP / 彼得鲁奇音乐图书馆，2025年）：~200 涵盖1600年至1920年的高分辨率古典乐谱，均保证属于公有领域。– **Mutopia项目与古腾堡计划**（Mutopia计划，2025年）：~100 采用公共许可协议的乐谱，涵盖巴赫、莫扎特、贝多芬和肖邦的经典作品。

- **文本描述：** 每份乐谱的元数据包括标题、作曲家、调性、拍号和节奏结构，这些信息会从符号文件中自动提取，并由专业标注人员核验。

- **音频文件：** 513 段钢琴演奏录音，涵盖古典时期、现代音乐和流行音乐类型。所有录音均为公有领域作品（来自较早的档案来源），或由演奏者直接提供，并依据知识共享许可协议授权。

B.2 基准测试许可与使用

MuseBench 的所有组成部分均在严格遵守法律和伦理规范的前提下发布：

- **乐谱** 仅取自公有领域库（IMSLP、Mutopia、古腾堡计划），或取自 MuseScore 上依据知识共享许可协议授权作品的贡献者。仅收录于 1954 年之前逝世的作曲家的作品。

- **音频录音** 来自公有领域档案，或由专业音乐家在知情同意并遵循知识共享许可协议的情况下直接提供。不收录过去 75 年内受版权保护的录音。

- **标注** （ABC、MusicXML、任务提示）由受过培训的音乐家准备。标注人员均提供了知情同意，标注者间一致性达到 $\kappa = 0.87$。

许可协议：MuseBench 根据**知识共享署名-非商业性使用 4.0 国际许可协议**发布，仅供非商业研究和教育用途。单独乐谱或录音的再分发或再利用必须遵守其原始来源的许可协议。论文被接收后，我们将公开发布所有数据、标注和评估脚本。

C 详细基准测试构建

C.1 详细任务定义

音乐理论理解。此维度侧重于对符号化和概念性音乐知识的文本理解，包含两个子任务：

- **音乐理论识别：** 评估对基本乐理概念的理解，包括调号、拍号、音符时值和节奏结构。

- **音乐理论推理：** 涉及需要对音乐符号描述进行深入推理的问题，例如确定和声进行或识别音乐形式。

乐谱理解。 这一维度评估模型根据乐谱图像理解记谱音乐的能力，包括：

- **信息提取：** 从视觉输入中转录谱号、调号和速度标记等基本音乐元数据。
- **符号分析：** 理解音符符号、它们在空间和节奏上的关系，以及基于谱表的结构元素。
- **高层次诠释：** 分析音乐语境中的表现性或风格线索，例如奏法、乐句处理和功能角色。

演奏音频分析。 这一维度评估模型分析真实演奏录音中表现性和结构性特征的能力，包括：

- **演奏评估：** 评判音乐演奏的准确性和完整性，包括节奏精确度、力度变化以及奏法清晰度。
- **一致性评估：**分析演奏中的速度稳定性、音高一致性，以及表现性过渡的流畅度。

C.2 生成

基于标注元数据，我们构建了多模态问答对。首先结合基于规则的模板和 GPT-4o（OpenAI，2024年）提示生成候选问题，以提高语言表达的多样性和自然度。真实答案通过检索、计算和统计，从符号元数据中确定性地得出，从而确保结果可复现。随后，专业音乐家审阅并完善了所有问答对，以保证其正确性、清晰度和难度均衡。

最终生成的数据集涵盖文本、图像和音频模态，提供全面的多模态评估。不同于 MusicTheoryBench（Yuan 等人，2024b）等以往数据集，后者仅专注于基于文本的符号音乐理论问题，且仅包含几百个示例，MuseBench 引入了需要对乐谱、音频和元数据进行联合推理的多模态问答任务。该过程产生了一个大规模基准测试，包括 591 张乐谱图像、513 份音频录音和 2,052 个专家验证的问答对。

C.3 数据集合规性与许可

为确保符合道德与法律要求，MuseBench 的所有组成部分均来自公有领域的资源库（如 IMSLP、Mutopia、古腾堡计划）或采用知识共享许可的平台（如 MuseScore），从而确保完全符合法律规定。演奏录音要么属于公有领域，要么是在明确同意下依据知识共享许可提供的。关于数据来源、筛选标准和许可条款的更多详情，请参见附录 B.1。

通过整合文本、图像和音频模态中的任务，**MuseBench**数据集为多模态大语言模型提供了一个全面的评估平台，涵盖音乐理解的方方面面——从乐理理解到对真实演奏特征的评估。

D M-OMR 的实现细节

M-OMR 模块采用“拆分—合并”策略，作为专门用于乐谱的视觉编码器。

拆分。 首先，通过结合五线谱线检测和小节线定位，将输入的乐谱图像分割为单个小节。随后，将每个分割后的小节视为独立的视觉单元进行局部识别。具体而言，采用基于 YOLOv8 的检测器（Varghese 和 Sambath，2024年）来识别并定位每个小节，

处理。 使用 ResNet-50 主干网络（He 等人，2016）将每个小节图像编码为高维嵌入，以捕捉音乐符号的细粒度视觉特征，包括谱号、五线谱、小节线、调号和拍号。随后，使用基于 LSTM 的解码器（ Yu 等人，2019 ）将这些嵌入依次解码为 ABC 格式符号序列，该解码器经过音符级转录训练。

合并。将小节级符号序列汇总，以重构完整的乐曲。在此步骤中，

表 2：MuseBench 中定义的任务。每项任务的问题格式均从格式池中随机选取。问题类型“选择题”和“判断题”分别表示多项选择题和判断正误。

模态	能力维度	子任务	示例问题	Type
Text	音乐理论理解与推理	音高与音程	增六度音程包含多少个半音?	MCQ
		节奏与拍号	哪种拍号表示复四拍子?	MCQ
		和声与调性	哪个枢纽和弦可以实现从C大调到G大调的平滑转调?	MCQ
		旋律结构	在乐段结构中，用以描述解决第一乐句的第二乐句的是什么?	MCQ
		音阶与调式	从D音开始的多利亚调式源自哪个大调音阶?	MCQ
		谱号与记谱法	在中音谱号中，第四间上的音级是什么?	MCQ
		力度与奏法	演奏者应如何演奏标有渐强记号的断奏音符?	MCQ
		曲式与结构	哪种结构模式最能描述奏鸣回旋曲式?	MCQ
图像	信息提取	标题识别	这首作品的标题是什么?	MCQ
		作曲家识别	标题为《古典拉格》的作品，其作曲家是谁?	MCQ
		拍号识别	标题为 The Waltz on my bum 的作品，其拍号是什么?	MCQ
		调号识别	乐谱开头部分的调号是什么?	MCQ
		声部/乐器识别	哪些声部被分配到低音谱号?	MCQ
	符号分析	音符定位	声部 V:1 首次演奏包含 E	MCQ
		音符频率统计	自然音高高于中央C? 在声部V:3中，哪一个音级作为发声音符出现的频率最高	MCQ
		节奏型分析	（不包括休止符和装饰音）? 旋律线（V:1）的前四个小节中，主要使用哪种节奏型	MCQ
		和声辨认	主要用于? 第18小节中，女高音（V:1）和低音（V:3）构成了什么和弦?	MCQ
		配器	这份乐谱中哪些五线谱使用低音谱号?	MCQ
	高层次音乐理解	功能和声分析	在降 E 大调中，升高的 A 音（â）的和声功能是什么?	MCQ
		调式识别	考虑调号和临时记号，暗示的是哪种调式?	MCQ
		旋律音域	声部 V:1 的旋律音域是什么?	MCQ
		装饰音识别	哪种装饰音的运用最为一致?	MCQ
		非和弦音识别	声部1的第2小节中，哪个非和弦音起经过音的作用?	MCQ
		调性转调识别	从降 E 大调转调发生在哪个小节?	MCQ
音频	演奏评估	调性准确度	第11小节证实调性完全准确。	T/F
		完整性	第8小节没有缺失音符。	T/F
	一致性评估	速度稳定性	第11小节的速度符合参考稳定性。	T/F
		速度	第8小节的速度明显慢于要求。	T/F

预处理过程中提取的拍号会与每个小节对齐，以确保一致的节奏语境。最终输出是一份格式规范的ABC记谱表示，同时保留时间结构和记谱正确性。

近期研究已证明，基于 YOLO 的模型在结构化文档分析任务中（Zhao 等人，2024年）。

随后，将每个分割后的小节图像输入 ResNet-50（He 等人，2016）编码器，以获得潜在视觉嵌入$\mathbf{x}_t$。解码器采用单向长短期记忆网络（LSTM）实现，并以自回归方式逐个生成相应的 ABC 序列标记。

$$\mathbf{h}_t = \text{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t; \theta), \tag{1}$$

解码所有小节后，通过以下方式重建符号输出：

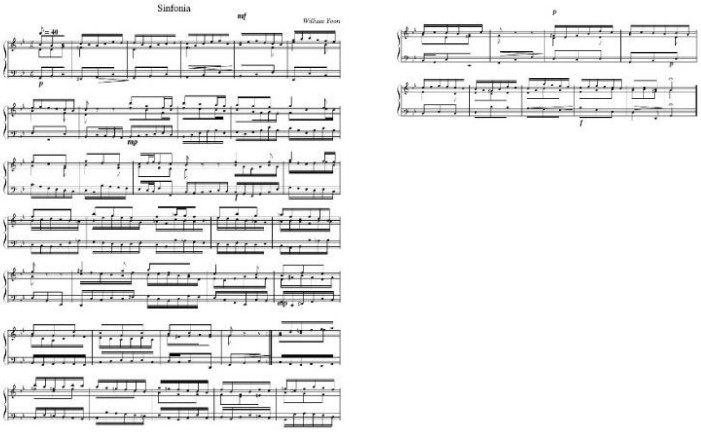
$$\text{ABC}_{\text{full}} = \text{ConcatMeasures}\left\{(\text{ABC}_i, \text{TimeSig}_i)\right\}_{i=1}^{n}, \tag{2}$$

其中，TimeSig_i 是预先检测到的小节 i的拍号，n 是小节总数。

数据集。为构建稳健的光学音乐识别（OMR）模块，我们整理了一个来自 MuseScore 平台的大型数据集，其中包含超过 80,000 份乐谱。我们首先将每份乐谱从 MusicXML 格式转换为 ABC 记谱法（Walshaw，2021），随后将其渲染为 SVG 图像。为进一步提高数据多样性和模型稳健性，我们通过随机打乱并替换 ABC 小节进行结构化数据增强，从而构建出包含 230 万个 ABC 样本的合成语料库。生成图像后，我们采用基于 YOLO 的（Varghese 和 Sambath，2024年）分割方法，自动检测并提取 SVG 中的单个小节，最终获得超过**1000 万**组成对的图像与小节数据。

训练配置。训练进行了 100 个周期，批量大小为 12，学习率为 1e-4。我们的模型在留出的验证集上达到了近乎完美的准确率（约 98 %），这证明了训练流程的规模和有效性。

ABC记谱法的可视化示例见图7。从图中可以看出，配备 M-OMR 模块的 MuseAgent 能够准确地将整份乐谱转录为 ABC记谱法。相比之下，其他大语言模型难以提取完整且精确的 ABC记谱表示，往往会遗漏结构或符号细节。



X:1T:交响曲C:William YoonL:1/4Q:1/8=40M:3/8I:换行 $K:Bb%%staffscale 0.65%%score { (1 2) | (3 4) } V:1 高音谱号V:2 高音谱号V:3 低音V:4 低音
V:1
B/A/B/ | c/4B/4A/4c/4e/4d/4 | d/4B/4G/4B/4c/4d/4 |!<(! c/4A/4c/4e/4f/4c/4!>(! | !mf! d/4e/4f/4d/4c/4B/4 | g/4a/4b/4g/4f/4e/4 | d/4e/4d/4c/4B/4A/4 |$ B/!>(! z/ z/!>)! b/f/d/4f/4 | b/a/g/4f/4 | g/f/e/ | f d/ | e/c/A/ | f/d/B/ |$ c/4d/8e/8d/c/ | B/ z/ z/ e e/ | g f/4e/4 | f/d/B/ | b/g/e/ | f/_a/4g/4f/ |$ e/b/g/ | _a/4f/4c/4f/4d/4B/4 | e/ z/ z/ | ^f f/ | g/d/B/ | c B/ | A/d/c/ |$ B/G/d/ | c/G/c/ | B/c/A/ |!>(! G/ z/ z/!>)! :| !mp! c/4d/4=e/4f/4g/4a/4 | b/4g/4=e/4g/4b/4a/4 | g/4f/4=e/4d/4c/4B/4 |$ 强/4=e/4d/4c/4B/4A/4 | d/4B/4F/4B/4d/4c/4 |=e/4^c/4e/4g/4f/4d/4 | B/4c/4B/4A/4G/4c/4 |!>(! F/ z/ z/!>)! | !p! ^F/4D/4F/4A/4d/4c/4 | B/4c/4B/4A/4B/4G/4 | c/4d/4e/4d/4c/4B/4 |$ A/4G/4^F/4A/4d/4c/4 | !<(! B/4G/4g/4a/4e/4d/4!<)! | c/4d/4e/4d/4c/4B/4 |!>(! B/c/A/!>)! !tp! !延长记号!G3/2 |]
V:2
D/D/G/ | ^F3/2 | D3/2 | F3/2 | B F/ | e B/ | A F/ |$ D/ x d/B/F/4B/4 | d/c/B/4A/4 | B/G/4F/4G/4E/4 | F/B/c/4B/4 | A F/ | B F/ |$ G/F/E/ | D/ x B/4G/4E/4G/4B/4G/4 | e/4d/4c/4d/4e/4B/4 | B/4c/4d/4c/4B/4 | A/4 | B/4e/4g/4e/4B/4G/4 | _A/4F/4c/4B/4A/4B/4 |$ e/4d/4c/4B/4 | A/4G/4 | _A/4B/4A/4F/4B/4A/4 | G/ x | c/4d/4c/4B/4A/4d/4 | B/4c/4B/4A/4G/4D/4 | E/4C/4E/4G/4F/4E/4 | D/4^F/4A/4F/4D/4F/4 |$ G/4D/4B,/4D/4G/4F/4 | E/4D/4C/4D/4E/4C/4 | G/4^强/4G/4A/4^强/ | G/ x :| A/c/强/ | =e/c/e/ | g/=e/c/ |$ A/强/强/ | 强/D/=E/ | A/G/强/ | G/D/=E/ | 强/ x | C D/ | G D/ | E C/ |$ D/C/D/ | D/B/G/ | E/G/A/ | G/A/^强/ | G3/2 ||
V:3
!p! G,,/B,,/D,/ | ^F,,/A,,/D,/ | G,,/B,,/D,/ | A,,/C,/F,/ | B,,/D,/F,/ | E,,/E,/C,/ | F,/F,,/E,/ |$ B,,/D,/4C,/4B,,/4A,,/4 !mp! B,,/4C,/4D/4E,/4F,/4D,/4 | B,,/4D,/4F,/4A,/4B,/4F,/4 | G,/4E,/4G,/4B,/4E/4B,/4 | D/4B,/4F,/4B,/4D/4B,/4 | C/4A,/4F,/4E,/4D,/4C,/4 | D,/4B,,/4B,/4A,/4G,/4F,/4 |$ E,/4D,/8C,/8F,/F,,/ | B,,/4A,,/4B,,/4C,/4D/4C,/4 强! B,,/4G,,/4E,,/4G,,/4B,,/4E,/4 | G,/4F,/4E,/4D,/4E,/4C,/4 | D,/4C,/4B,,/4D,/4F,/4A,/4 | G,/4F,/4E,/4D,/4E,/4F,/4 | D,/4B,,/4B,/4_A,/4G,/4F,/4 |$ E,/4D,/4F,/4E,/4D,/4E,/4 | C,/4 A,,/4B,,/4B,/4A,/4B,/4 | E,/8,,/E,/ | D,,/4D,/4C,/4E,/4D,/4^强/4 | G,/4D,/4B,,/4D,/4G,/4F,/4 | E,/4C,/4G,,/4E,,/4C,/4B,,/4 | A,,/4D,,/4^F,,/4A,,/4D,/4C,/4 |$ B,,/4G,,/4A,,/4B,,/4A,,/4G,,/4 | C,/4D,/4E,/4D,/4E,/4C,/4 | D,/4C,/4D,,/ | G,,/B,,/D,/ :| C,/4A,,/4G,,/4A,,/4F,,/4A,,/4 | C,/4F,/4-E,/4D/4C,/4B,,/4 | C,/4B,/4A,/4G,/4F,/4-E,/4 |$ F,/4C,/4A,,/4C,/4F,/4C,/4 | D,/4B,,/4D,/4F,/4-E,/4C,/4 | A,,/4G,/4A,/4G,/4F,/4D/4 | B,,/4G,/4C,/C,/ | F,/C,/F,/ | D,/C,/D,/ | G,,/B,,/D,/ | C,/E,/C,/ |$ D,/D/C/ | B,/A,/G,/ | 强! C/G,/E,/ | D, D,/ | 延长记号!G,,3/2 ||
V:4
xx/|xx/|xx/|xx/|xx/|xx/|xx/|x2x|xx/|xx/|xx/|xx/|xx/|x2x|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/:|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/|xx/

E 自动音乐转录与音频到乐谱对齐的实现细节

本附录详细介绍 MuseAgent 中使用的自动音乐转录（AMT）和音频到乐谱对齐模块的实现。虽然我们采用了受先前研究启发的方法（Hawthorne 等，2017年；Nakamura 等，2015），但我们报告了所有相关的架构和配置细节，以便于复现和下游集成。

E.1 自动音乐转录（AMT）

输入表示。我们使用常数Q变换（CQT）从原始音频中提取时频特征。采用 CNN–BiLSTM（Siami-Namini 等，2019）架构，同时预测起始和逐帧音符激活，遵循 Onsets-and-Frames（Hawthorne 等，2017年）的结构。

形式化地，给定输入特征 $\mathbf{X}_t$，起始点概率的预测如下：

$$\mathbf{O}_t = \sigma(\text{BiLSTM}(\text{CNN}(\mathbf{X}_t))), \tag{3}$$

逐帧激活的计算如下：

$$\mathbf{F}_t = \sigma(\text{BiLSTM}([\text{CNN}(\mathbf{X}_t), \mathbf{O}_t])). \tag{4}$$

参数列于表3中。

表3：自动音乐转录的常数Q变换配置

参数	数值
采样率	16 kHz
跳跃长度	512 个采样点
频率箱	88（涵盖 A0–C8）
每八度音程箱数	12
窗函数	Hann
归一化	对数幅度

网络架构。自动音乐转录模型通过以下方式处理常数Q变换输入：

- **CNN前端：** 3个卷积层（卷积核大小：3×3，步幅：1，填充：1），每层后接ReLU和批归一化。
- **双向LSTM层：** 一个双向长短期记忆网络，每个方向有128个隐藏单元。
- **起始头：** 带有 Sigmoid 激活函数的全连接层，用于预测逐帧音符起始。
- **帧头：** 与此类似的层，以起始特征为条件，用于预测逐帧音符激活。

训练细节。

- **损失：** 分别对起始和帧预测独立应用二元交叉熵损失。
- **优化器：** Adam，学习率为 1×10^{-4}。
- **训练轮数：** 在 MAESTRO-V3 上训练 50 轮（Hawthorne 等，2019a）。
- **批量大小：** 8。

后处理。二元预测的阈值设为 0.5。使用以下启发式方法，将起始激活值和帧激活值合并为 MIDI 音符事件：- 若起始激活值超过阈值，则判定音符起始。- 若连续帧中预测均为激活状态，则延长音符时长。

最终输出以 MIDI 或 MusicXML 格式导出。需要进行符号对齐时，我们还会将其转换为 ABC 记谱法。

E.2 音频到乐谱对齐

模型概述。我们采用（Nakamura 等，2015）提出的双层隐马尔可夫模型方法，可稳健地对齐符号乐谱与自动音乐转录生成的音频事件。对齐过程通过最大化给定观测声学特征时乐谱位置的后验概率来实现 p_t，其中观测到的声学特征为x_t：

$$p(p_t \mid x_t) = \frac{p(x_t \mid p_t) \cdot p(p_t)}{p(x_t)}. \tag{5}$$

结构。

- **顶层隐马尔可夫模型：** 对乐谱位置之间的转移进行建模（例如，小节或音符组之间的转移）。
- **底层隐马尔可夫模型：** 捕捉音符内部的细粒度时间动态（起始、延续、静音）。

观测模型。给定乐谱位置 p_t，观测到声学特征 x_t 的似然 $p(x_t \mid p_t)$由高斯混合模型（GMM）建模：

- 组件数量：8
- 协方差：对角

• 输入：经主成分分析降维的常数Q变换（维度 = 30）

•训练：基于对齐的乐谱—音频对进行期望最大化

转移模型。 我们定义一个转移矩阵A，支持以下操作：

- **自循环：** 保持在当前音符位置。
- **向前转移：** 按顺序正常推进。
- **向后跳转：** 重复乐段或进行纠正。
- **向前跳跃：** 跳过乐段。

这些转移以概率形式编码：

$$A_{ij} = p(p_t = j \mid p_{t-1} = i), \quad \text{with non-zero mass for } |i-j| \le 1.$$

推断。 我们使用维特比解码来计算概率最高的对齐路径：

$$p^*_{1:T} = \arg\max_{p_{1:T}} \prod_{t=1}^{T} p(x_t \mid p_t) \cdot A_{p_{t-1},p_t}$$

该算法能够有效处理表现性节奏、遗漏、重复和错误音符，因此在真实演奏场景中进行对齐时具有很强的鲁棒性。

E.3 自动音乐转录与对齐算法的评估结果

本节介绍自动音乐转录（AMT）和对齐算法的评估结果，这些结果以 JSON 格式记录。结果详细评估了多项转录指标，例如总体准确率、音符匹配、速度、稳定性和速度同步。

随机样本中测量 37 的评估结果总结如下表：

指标	数值
总体评估（整体评价）	0.9252619743347168
音符评估（音符评价）	1.0
速度评估（评估速度）	1.0
稳定性评估（稳定性评价）	0.7282252907752991
速度同步（节奏同步评价）	1.0
额外音符数	0
匹配音符数	2
缺失音符数	0

表4：随机样本测量 37 的评估结果

此外，下图 8 直观展示了不同转录指标下的演奏比较。

F 评估详情

F.1 基准测试中的评估详情

我们展示了用于三种模态的不同提示：文本、图像和音频。下表总结了每种模态对应的具体提示。

F.2 对比实验中使用的评估指标

在本附录中，我们介绍了 M-OMR 所采用的评估指标，并将其与其他将图像转换为 ABC 记谱文本的模型进行比较，使用Levenshtein距离进行评估。此外，我们还使用语义相似度和词语匹配指标分析音乐内容。

Levenshtein距离。 Levenshtein距离（Yujian和Bo，2007）用作将图像转换为ABC记谱文本的评估指标。它指将模型响应转换为正确答案序列所需的最少单字符操作次数。

设 D 为一个大小为 $(|R|+1) \times (|A|+1)$ 的矩阵，其中 $|R|$ 和 $|A|$ 分别表示响应序列和答案序列的长度。 $D[i][j]$表示 R 的前 i 个字符与A的前 j 个字符之间的最小编辑距离。

D 的后续数值通过以下递推关系计算：

$$D[i][j] = \min \begin{cases} D[i-1][j] + 1 & \text{(delete)} \\ D[i][j-1] + 1 & \text{(insert)} \\ D[i-1][j-1] + \text{cost} & \text{(substitute)} \end{cases}$$

其中，当 $R[i-1] = A[j-1]$时，代价为 0；否则为 1。

语义相似度与词语匹配指标。我们的实验还采用两类指标：语义相似度和词语匹配，用于分析音乐内容。

对于语义相似度，我们采用潜在语义分析（LSA），通过计算向量之间的余弦相似度来衡量文本的语义相似度。余弦相似度定义如下：

对于词语匹配，我们采用以下指标：

- **ROUGE-1**：计算生成文本与参考文本之间匹配的单元语法数量。

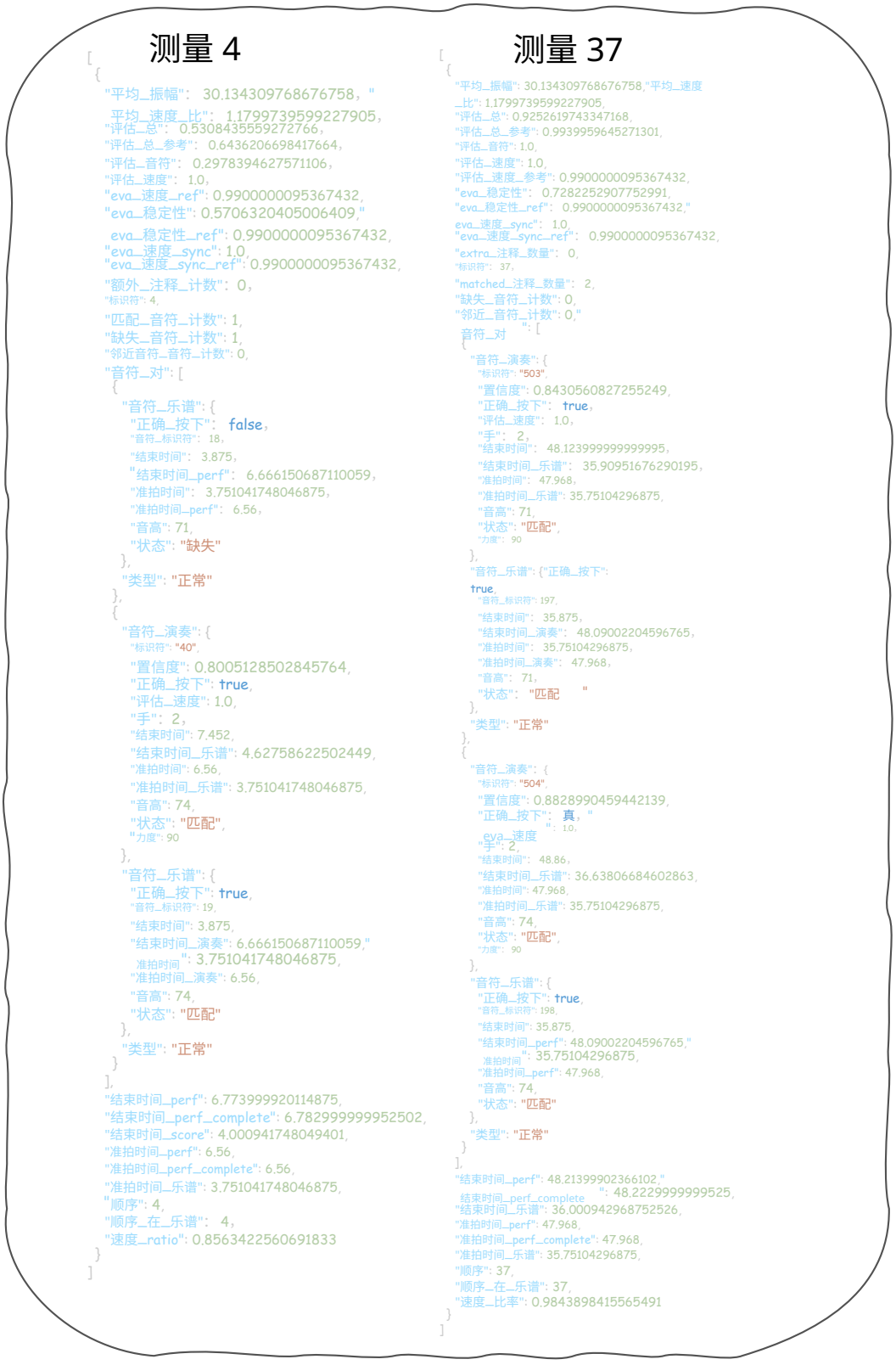


图 8：不同转录指标下的演奏性能比较结果

• **ROUGE-L**：衡量生成文本与参考文本之间的最长公共子序列（LCS）匹配程度。

• **METEOR**：计算同义词匹配，并结合一元语法匹配、最长公共子序列和同义词匹配。