arXiv:2602.03023v1 [cs.SD] 3 Feb 2026

R借助音乐元数据大语言模型重新思考音乐描述生成

Irmak Bukey ♯♭* *Zhepei Wang* ♭ *Chris Donahue* ♯ *Nicholas J. Bryan*♭

♯ 卡内基梅隆大学 ♭ Adobe Research

摘要

音乐描述生成，即生成音乐的自然语言描述这一任务，对音乐理解和可控音乐生成都很有用。然而，训练字幕生成模型通常需要高质量的音乐描述数据，而这类数据相对于元数据（例如流派、情绪等）较为稀缺。因此，人们通常使用大型语言模型（大语言模型）根据元数据合成字幕，从而为字幕生成模型生成训练数据，不过这一过程会施加固定的风格化，并将事实信息与自然语言风格纠缠在一起。作为一种更直接的方法，我们提出基于元数据的描述生成。我们训练一个元数据预测模型，从音频推断详细的音乐元数据，然后在推理时通过预训练大语言模型将其转换为富有表现力的描述。与在由元数据派生的大语言模型生成的描述上训练的强端到端基线相比，我们的方法：（1）相比端到端字幕生成模型，能以更少的训练时间取得相当的性能；（2）提供在训练后轻松改变风格化的灵活性，使输出字幕能够针对特定的风格与质量要求进行定制；（3）可以同时以音频和部分元数据作为提示，从而实现强大的元数据填补或补全——这是整理音乐数据时的常见任务。

索引词— 音乐理解、元数据、多模态大语言模型、音乐描述生成、音乐标签标注

1. 引言

大型语言模型（大语言模型）已被证明是面向音频与音乐的音乐理解、推理、描述生成和生成任务的强大基础[1, 2, 3, 4, 5, 6, 7, 8]。尤其是多模态大语言模型（MLLMs），通过利用配对的音频与文本描述对强大的大语言模型进行微调，在音乐相关任务上取得了显著进展 [5, 6, 8, 9, 10, 11]。然而，由于用文本描述音乐较为复杂、许可费用高昂，以及/或网页抓取存在法律和/或伦理问题，此类方法在获取配对训练数据方面仍然受限。因此，准确、信息丰富且多样的标注或“字幕”至关重要。

以往研究探索了多种策略，用于合成或增强音乐描述数据集，以供训练使用。一种方法是使用现有的音乐描述数据集，通常会合并多个不同来源的数据集 [12, 13]。然而，这些单独的数据集通常规模较小，因此限制了它们在训练稳健模型方面的作用。另一种方法是利用大语言模型生成合成描述。有些方法会扩充现有字幕或基于关键词的描述 [2, 4, 11, 14, 15]，另一些方法则提示大语言模型转换现有音乐元数据，和/或通过算法从音乐信息检索算法中提取元数据（例如每分钟节拍数）来添加所谓的伪标签，从而创建内容丰富的句子级描述或指令微调数据集 [5, 7, 16]。这有助于生成内容更丰富的真实标注描述，但仍不尽如人意，因为对音乐的人类描述和/或伪标签描述通常不够准确，也不够全面。此外，许多下游任务并不需要字幕，只需要丰富的元数据，因此音乐描述生成模型经过训练后，在语言风格、细节程度和/或事实内容方面可能缺乏灵活性和可控性。

在这项工作中，我们提出了音乐元数据 大语言模型——一种音乐描述生成的替代方法，如图 1（右下）所示，它能够轻松适应不同的自然语言风格。具体而言，我们在元数据预测任务上对预训练大语言模型进行指令微调，使其能够直接从音频中推断词汇开放集的结构化音乐元数据。随后，我们通过文本提示指令，使用同一个预训练大语言模型将预测出的元数据转换为字幕。此外，我们训练的元数据大语言模型可接收零个或多个输入元数据字段，以及音频输入，因此能够利用该模型为现有音乐数据库填补缺失的元数据字段。在评估中，我们将所提方法与一个强大的端到端基线进行比较；该基线使用由元数据生成的大语言模型生成的描述进行训练。我们还测试了事后风格化的影响，并展示了除音频输入外加入部分元数据的效果。总的来说，我们发现，所提出的方法具有竞争力，训练速度更快，在描述风格和质量方面更加灵活，并且能够利用部分元数据改善元数据预测；而采用固定描述训练的端到端字幕生成模型则不具备这些优势。我们的贡献总结如下:

1. 我们提出了一种用于音乐的音频到元数据再到描述的方法，与音频到描述的方法相比，该方法性能相当且训练速度更快。2. 我们证明，事后将元数据转换为描述能够提升整体描述生成性能，并带来更高的灵活性。3. 我们表明，我们的音频到元数据模型能够有效利用部分元数据进行元数据填补，从而提升预测性能。

2. 相关工作

早期的音乐描述生成工作包括 MusiCNN+LSTM [17]，该方法将描述表述为音乐标签的串联，缺乏自然语言的表现力；以及 MusCaps [18]。较新的工作包括 Qwen2-Audio [2], SALMONN [3], Audio Flamingo 3 [4], LLark [5], Lp-MusicCaps [7], 和 ViML [19]，它们通过利用大语言模型，将现有元数据、人工标注描述和/或内容分析算法生成的伪标签改写为指令遵循格式，从而构建用于音乐理解的训练数据。尽管这种方法能够通过自然语言实现可扩展的监督，并推动通用音乐理解和推理任务的发展，但生成的描述通常缺少配器、调性和速度等音乐细节，而这些细节对于具有细粒度语义控制的下游应用十分重要。基于检索的

*在 Adobe Research 实习期间完成的工作。

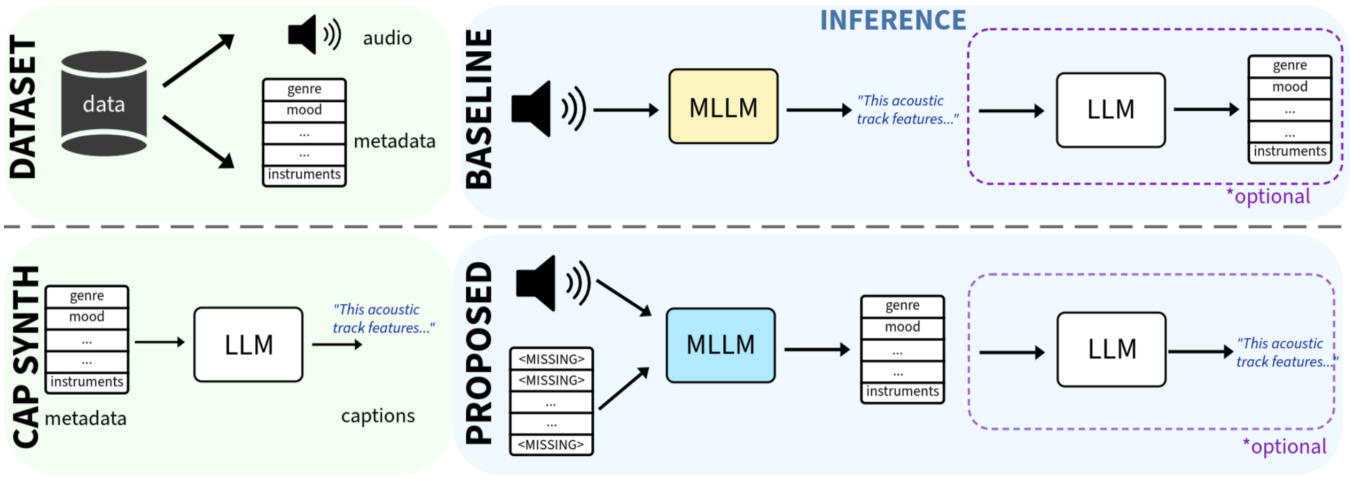


图 1。（左上）数据集格式。我们的训练数据集由配对的音乐和元数据标注组成。（左下）元数据到字幕合成。我们使用基于文本的大语言模型，将音乐元数据转换为自然语言描述。（右上）典型音乐描述生成方法的推理流程：训练字幕多模态大语言模型来生成字幕。还可以选择使用基于文本的大语言模型，从生成的字幕中提取元数据。（右下）所提出的元数据多模态大语言模型的推理流程：预测全部或缺失的元数据字段。还可以选择使用基于文本的大语言模型，将预测的元数据转换为字幕。

工作（如 TTMR++ [20] ）同样使用元数据和大语言模型来丰富训练描述，但并未训练模型生成详细描述。

与此同时，在图像字幕生成领域，研究人员已探索将结构化元数据表示作为图像描述生成流程的中间环节。例如，图像到场景图再到字幕的流程 [21, 22] 能够生成更丰富的字幕，进而提升图像生成质量 [23]。随着语言模型不断改进，第二阶段的负担有所减轻，这使得这种方法日益具有现实意义 [24]。然而，据我们所知，音乐描述生成领域尚未探索类似方法。

3. 方法

我们分两个阶段生成字幕。首先，根据音频和可选的部分输入元数据预测结构化元数据。其次，使用基于大语言模型的转换阶段将预测的元数据转换为音乐字幕，并进行风格化。

3.1. 元数据预测模型

我们使用音频—元数据对对现代的仅解码器预训练纯文本大语言模型进行指令微调，使其能够根据音频执行元数据预测。由于我们从预训练的纯文本大语言模型开始，因此采用两阶段微调流程，使其适应音频输入和结构化输出生成。

在第一阶段，我们赋予纯文本大语言模型多模态音频理解能力。为此，我们首先使用量化音频编码器 [25] 将音频输入编码为离散词元序列，然后将音频词元映射为大语言模型中的保留文本词元。随后，我们让预训练大语言模型同时适应自监督音频与语言续接任务，从而得到一个预训练的音频文本多模态大语言模型。

在第二阶段，我们针对元数据预测任务对音频多模态大语言模型进行指令微调。我们的训练样本由 JSON 格式的元数据与音频对组成，如图 1（左上）所示。该目标在我们的指令中有详细表述提示，是以结构化格式预测元数据字段，同时保持所提供的字段不变。我们的设置使模型能够执行元数据填补，从而有效地学习从音频以及（可选的）部分上下文信息中推断出一整套音乐属性。在推理时，我们向模型提供音频和一个空的元数据字典作为输入，促使其以结构化方式生成全套元数据字段。

3.2. 元数据到字幕转换

给定预测元数据后，我们使用原始的纯文本大语言模型将其转换为字幕。我们通过精心设计的指令引导模型生成富有表现力且连贯的字幕，使其反映预测的元数据字段，同时避免凭空捏造预测元数据输出中不存在的信息。此外，我们还可以在事后轻松调整元数据到字幕提示，通过加入上下文学习示例 [26, 27]，针对特定的问题领域或目标风格化进行定制。这与现有的在训练之前根据元数据合成字幕的方法形成对比。在下文的实验中，我们将探究事后字幕转换的不同变体，以评估其对字幕质量和可控性的影响。

3.3. 元数据填补

除了预测音乐元数据和音乐字幕这一标准用例外，我们还探讨音乐元数据填补任务——根据音频和不完整的元数据子集预测完整元数据。元数据填补是音乐数据整理中的常见实际任务，对于重新训练新的字幕生成模型和/或元数据模型都很有用。为支持填补，我们在训练数据中人为屏蔽可用的元数据字段，并提示模型根据音频令牌和被屏蔽的元数据生成完整元数据。我们指出，将这项任务纳入核心方法十分自然；但对字幕生成模型而言，实现起来则更加困难，因为很难即时修改预生成字幕以省略特定的元数据字段。

4. 实验

4.1. 实验设置

我们将元数据描述生成方法与基线字幕生成模型进行比较，后者通过音频-描述配对数据进行端到端训练。所有模型均针对 1）元数据预测和 2）字幕生成进行评估。

数据集。 我们使用一个内部授权的器乐数据集训练模型。该数据集包含约 2.5 万小时的音乐及其对应的元数据标注，字段包括流派、情绪、关键词、速度、调性、能量和乐器。请注意，元数据通常并不完整，因为 23% 的音乐条目缺少一个或多个字段。训练音频样本是从这些曲目中随机选取的 10 秒音频片段。对于元数据预测评估，我们使用该内部数据集中的一个留出子集，其中包含 5,000 首曲目。对于描述生成评估，我们使用来自非人声子集的音乐-字幕配对数据：MusicCaps 数据集中的 2,185 首曲目 [12, 28], ，以及 Song Describer 数据集中的 446 首曲目 [13]。

基线方法。 我们训练了两个端到端音乐描述生成模型，作为比较基线。我们使用与我们的方法相同的元数据数据集，提示 Gemma3-12B-it 分别按照 MusicCaps 和 SongDescriber 字幕的风格，通过提炼元数据来生成描述。由此得到两组（音频，合成描述）配对数据，每组都对应一种特定的描述生成风格。随后，我们使用这些配对数据对适配音频的模型进行指令微调，得到两个基线描述生成器：一个用于生成 MusicCaps 风格的描述，另一个用于生成 Song Describer 风格的描述。

训练配置。 对于所有模型，我们使用指令微调 Gemma3-1B-it [29], 这一仅解码器文本大语言模型，并采用上述方法赋予其多模态音频能力。对于元数据到描述和描述到元数据的转换，我们使用原始 Gemma3-1B-it 模型生成提示，不进行任何额外的微调。所提出的元数据模型和基线描述生成器均在 4 块 NVIDIA A100 GPU 上进行训练，并采用提前停止；元数据模型的检查点位于第 161,000 次迭代，两个描述生成器的检查点则均位于第 347,600 次迭代。对于音频自编码器，我们采用基于 DAC [30], 的架构，但使用时间下采样更激进 [31]的编码器，该编码器包含 32 个通道，帧率为 21 Hz，并进一步使用码本大小为 1024 [25]的码本进行量化。

4.2. 评估任务

我们评估音频到元数据模型和音频到描述基线模型在元数据和描述生成任务上的表现，并使用大语言模型使它们适应各自未接受过训练的任务。

元数据评估。 对于所提出的方法，我们提示模型生成完整的结构化元数据，并评估其对流派、情绪、乐器和关键词等语义字段的预测结果。对于基线输出，我们通过提示大语言模型回答特定字段问题（例如：“根据字幕推断，这首曲目的流派是什么？”），从生成的字幕中提取元数据。这样，我们便可以为基线字幕生成模型的输出构建结构化元数据，并为元数据评估做好准备。评估时，我们使用固定模板将每个预测字段和真实标注字段转换为句子（例如，乐器预测 /电吉他、鼓、贝斯/ 会被转换为句子 “这首曲目使用了电吉他、鼓和贝斯等乐器。”）。我们计算句子-BERT（SBERT）嵌入 [32]在预测结果与真实标注之间的相似度，以处理长度不同的输入、同义词和开放式词汇。

表 1。使用 SBERT 相似度评估的元数据预测以及 MusicCaps 风格（MC）和 Song Describer 风格（SD）字幕生成模型的性能。数值越高越好。

模型	流派	Mood	乐器	关键词	Avg
MC 字幕生成器	0.556	0.673	**0.677**	0.614	0.630
SD 字幕生成器	**0.562**	0.687	0.676	**0.618**	**0.636**
元数据（我们的方法）	0.548	**0.711**	0.675	0.566	0.625

表 2。在风格匹配和跨风格／跨数据集的描述生成设置下，使用 SBERT 相似度评估描述，评估数据集为 Mu- sicCaps (MC) 和 Song Describer (SD)。数值越高越好。

风格	模型	MC	SD	Avg
匹配	图注生成器	**0.478**	0.468	**0.407**
	元数据（我们的方法）	0.443	0.454	0.392
跨域	图注生成器	0.441	**0.469**	0.405
	元数据（我们的方法）	0.439	0.462	0.395

描述评估。 对于描述评估，我们分别针对我们的生成流程和基线方法，计算预测字幕与参考字幕之间的 SBERT 相似度。为更全面地分析风格的影响，我们探索了两种互补的跨评估设置：(1) 针对所提出的元数据到描述生成流程，采用跨风格提示，即我们的模型为 MusicCaps 音频生成元数据，但使用 Song Describer 风格的提示将元数据转换为描述（反之亦然）；(2) 针对基线字幕生成模型，采用跨数据集评估，即在 MusicCaps 风格字幕上训练的字幕生成模型在 Song Describer 数据集上进行评估（反之亦然）。

4.3. 风格与提示变体

我们方法的一项优势在于，无需重新训练即可灵活适应新的描述风格。为探索这一点，我们尝试了其他元数据到描述提示，并评估其对字幕质量的影响。作为对比，我们还将事后风格化提示应用于基线描述生成器的输出，以评估直接修改描述能否实现类似的风格控制。具体而言，我们重点考察上下文学习的不同变体，或在大语言模型提示中提供丰富示例。我们尝试使用固定示例集、随机抽样示例集，以及将元数据标签作为上下文学习示例集的一部分。

在这些实验中，我们使用风格相似度指标来衡量不同风格提示对描述生成性能的影响。我们结合了三种互补指标：BM 25 [33], a，一种基于词袋检索的度量，用于捕捉词汇重叠和内容相关性；长度相似度，根据描述长度差异通过平滑指数函数计算；以及词性（POS）直方图相似度，用于比较预测字幕与参考字幕的句法结构。我们还报告所有语义和风格指标的平均值。

最后，我们设置实验，以评估模型在推理时被提示部分元数据时的元数据预测性能，模拟部分元数据信息可用、并可用于填补缺失字段的场景。

表3。不同风格提示对描述生成性能的影响。“图注生成器”结果表示匹配风格条件下两个基线描述生成器的表现。对于所提出的方法，我们在元数据到描述阶段尝试了不同的风格提示。“固定1-shot”使用固定的上下文学习示例（参考风格描述），而非随机抽取的示例。“元数据1-shot”则在示例中包含元数据标签。对于基线字幕生成模型，我们使用等效的风格提示对预测描述进行事后编辑。数值越高越好。

方法	SBERT-Sim			BM25			长度			POS			Avg
	MC	SD	Avg	MC	SD	Avg	MC	SD	Avg	MC	SD	Avg	
图注生成器	0.478	0.468	0.473	0.177	0.104	0.141	0.293	0.122	0.208	0.805	**0.724**	**0.765**	0.396
+更短的提示词	0.477	0.466	0.472	0.174	0.101	0.138	0.293	0.138	0.216	0.803	0.722	0.763	0.397
+固定1-shot	0.476	0.467	0.472	0.174	0.104	0.139	0.296	0.138	0.217	0.804	**0.724**	0.764	0.398
元数据	0.443	0.454	0.449	0.201	**0.111**	0.156	0.205	0.165	0.185	0.768	0.702	0.735	0.381
+更短的提示词	0.450	0.463	0.457	0.175	0.089	0.132	0.252	0.234	0.243	0.774	0.707	0.741	0.393
+固定1-shot	0.475	**0.475**	0.475	0.190	0.059	0.125	**0.319**	0.412	0.366	0.787	0.694	0.741	0.426
+元数据1-shot	**0.495**	0.471	**0.483**	**0.317**	0.045	**0.181**	0.284	**0.454**	**0.369**	**0.810**	0.656	0.733	**0.442**

表4。部分元数据补全的 SBERT 得分。% 表示其他字段可用的比例（预测字段始终被遮蔽）。我们的方法-0% 等同于表1的第一行。数值越高越好。

模型	%	流派	Mood	乐器	关键词
Gemma3-1b	50%	0.504	0.666	0.657	0.543
Ours	0%	0.548	0.711	0.675	0.566
	25%	0.638	0.743	0.754	0.618
	50%	0.679	0.765	0.78	0.645
	75%	0.715	0.789	0.807	0.671
	100%	**0.731**	**0.798**	**0.817**	**0.686**

5. 结果与分析

首先，我们通过实证表明，所提出的音乐元数据大语言模型在元数据预测和字幕生成两项任务上均取得了与基线音频到描述的方法相当的性能，且训练成本显著更低。表1报告了使用SBERT相似度指标对流派、情绪、乐器和关键词等字段的元数据预测结果。我们的方法在平均意义上取得了与基线描述生成器相当的性能，情绪预测更优，而关键词预测较差。我们在表2的描述生成评估中观察到类似的性能对比。我们将基线描述生成器具有竞争力的表现部分归因于：它们在由我们流程所使用的相同元数据生成的字幕上训练，从而使两种对比方法的训练样本高度一致。此外，一个关键因素是基础大语言模型的归纳偏置——它在与基线描述生成器更为相近的自然语言数据上预训练，而非我们的模型所训练的更为受限且结构化的JSON数据。这种归纳偏置很可能使基线描述生成器受益。尽管如此，我们的模型在元数据与字幕生成上均取得了相近性能，而训练所用GPU小时数仅为基线方法的46.3%。

从表 2 的“Cross”行中，我们还观察到，使用 MusicCaps 风格的参考字幕来描述 Song Describer 音频，并不会导致性能下降，而反向设置则会。这可能是因为 MusicCaps 字幕往往比通常更短、描述更不具体的 Song Describer 字幕更加详细，因此在不同数据集上仍能保持不错的性能。

接下来, 我们展示了所提出的音频到元数据的流水线 ine 为事后风格化提供了灵活性，并进一步提升了描述生成性能。在表 3 中，我们展示了提示工程对所提出方法的影响。随着我们不断优化元数据到描述的转换提示，我们观察到输出质量和风格持续改善，描述生成性能提升了 20% 以上，且无需重新训练。相比之下，将类似的提示用于对基线图注生成器的输出进行事后编辑，并未带来相当的提升，变化微乎其微，这凸显了我们方法的优势。

最后，我们展示了模型的另一项关键优势：利用部分元数据提升元数据预测性能，并实现元数据填补的能力。在表 4 中，我们通过逐步增加推理时可用的部分元数据量，评估模型的元数据预测性能。作为基准对比，我们还纳入了一个大语言模型基线（Gemma3-1b-it），并评估其在提示中提供 50% 可用字段时的元数据预测性能。结果表明，使用部分元数据提示我们的模型，能显著提升预测性能，平均提升 21%，单个字段最高提升 33%。对于使用元数据和/或字幕来控制音乐生成模型等下游任务，能够填补高质量元数据以补全数据库条目是一项显著优势。端到端字幕生成模型无法实现类似的部分提示和定向预测。

6. 结论

本文提出了一种灵活有效的音乐描述生成方法：训练音频到元数据预测模型，并在推理时利用预训练的大语言模型，将丰富的结构化元数据预测结果转换为字幕。事后将元数据转换为字幕的能力，提升了控制描述风格的灵活性，同时也改善了整体描述生成性能。此外，在输入部分元数据的提示下，我们的模型能够提升元数据预测性能，从而有效地进行元数据填补。这是端到端描述流程不具备的能力，却对音乐数据整理极具价值。我们的结果证明了这一元数据驱动流程的有效性和灵活性，同时也凸显了其相较于传统描述方法的若干关键优势。

对于未来的工作，我们认为有许多令人期待的方向。我们很期待探索使用动态变化的结构化元数据方案进行训练，以及采用长度可控的字段，以便在推理时从单标签字段适应到多标签字段。

7. 参考文献

[1] Yuan Gong, Hongyin Luo, Alexander H. Liu, Leonid Karlin- sky, and James Glass, 《聆听、思考与理解》，发表于*ICLR*，2024年。[2] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin 等, 《Qwen2-Audio 技术报告》，*arXiv:2407.10759*，2024年。[3] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma 和 Chao Zhang, 《SALMONN：迈向大型语言模型的通用听觉能力》，发表于*ICLR*，2024年。[4] Arushi Goel, Sreyan Ghosh, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Huck Yang 等, 《Audio Flamingo 3：借助完全开放的大型音频语言模型推进音频智能》，*arXiv:2507.08128*，2025年。[5] Josh Gardner, Simon Durand, Daniel Stoller 和 Rachel Bit- tner, 《LLark：一种用于音乐的多模态指令遵循语言模型》，发表于*ICML*，2024年。[6] Mengjie Zhao, Zhi Zhong, Zhuoyuan Mao, Shiqi Yang, Wei-Hsiang Liao, Shusuke Takahashi 等, 《OpenMU：您的音乐理解瑞士军刀》，*arXiv:2410.15573*，2024年。[7] SeungHeon Doh, Keunwoo Choi, Jongpil Lee 和 Juhan Nam, 《LP-MusicCaps：基于大语言模型的伪音乐描述》，发表于国际音乐信息检索学会会议（*ISMIR*），2023年。[8] Ruibin Yuan, Hanfeng Lin, Yi Wang, Zeyue Tian, Shangda Wu, Tianhao Shen, Ge Zhang 等, 《Chatmusician：借助大语言模型从本质上理解和生成音乐》，发表于计算语言学协会：*ACL 2024*，2024年。[9] Shansong Liu, Atin Sakkeer Hussain, Chenshuo Sun 和 Ying Shan, 《Music Understanding LLaMA：通过问答和描述生成推进文本到音乐的生成》，*arXiv:2308.11276*，2023年。[10] Zihao Deng, Yinghao Ma, Yudong Liu, Rongchen Guo, Ge Zhang, Wenhu Chen 等, 《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》，发表于计算语言学协会北美分会（*NAACL*），2024年。[11] Shansong Liu, Atin Sakkeer Hussain, Chenshuo Sun 和 Ying Shan, 《M^2 UGen：借助大型语言模型的强大能力实现多模态音乐理解与生成》，*arXiv:2311.11255*，2023年。[12] Andrea Agostinelli, Timo I Denk, Zal´an Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang 等, 《MusicLM：从文本生成音乐》，*arXiv:2301.11325*，2023年。[13] Hector Martinez, Hendrik Koops 和 Anja Volk, 《歌曲描述数据集：用于自动音乐描述生成的数据集》，发表于国际音乐信息检索学会会议（*ISMIR*），2022年。[14] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick 和 Shlomo Dubnov, 《通过特征融合和关键词到描述增强进行大规模对比式语言-音频预训练》，发表于*ICASSP*，2023年。[15] Ilaria Manco, Justin Salamon 和 Oriol Nieto, 《增强、删除与交换：提升大语言模型生成描述的多样性，以实现高效的音乐-文本表征学习》，*arXiv:2409.11498*，2024年。[16] Abhinaba Roy、Renhang Liu、Tongyu Lu 和 Dorien Herremans, 《Jamendomaxcaps：一个包含推算元数据的大规模音乐描述数据集》，*arXiv:2502.07461*，2025年。[17] Tian Cai、Michael I Mandel 和 Di He, 《将音乐自动标注作为描述生成》，载于音乐与音频自然语言处理研讨会（*NLP4MusA*），2020年。[18] Ilaria Manco、Emmanouil Benetos、Elio Quinton 和 Gy¨orgy Fazekas, 《Muscaps：为音乐音频生成字幕》，载于国际神经网络联合会议（*IJCNN*）。IEEE，2021年。[19] Daniel McKee、Justin Salamon、Josef Sivic 和 Bryan Russell, 《通过提示类比为视频提供语言引导的音乐推荐》，载于*IEEE*计算机视觉与模式识别会议（*CVPR*），2023年。[20] SeungHeon Doh、Minhee Lee、Dasaem Jeong 和 Juhan Nam, 《利用微调的大语言模型和元数据丰富音乐描述，以实现文本到音乐检索》，载于*ICASSP*，2024年。[21]Xu Yang、Kaihua Tang、Hanwang Zhang 和 Jianfei Cai, 《为图像字幕生成自动编码场景图》，*arXiv:1812.02378*，2018年。[22] Xu Yang、Hanwang Zhang 和 Jianfei Cai, 《为图像字幕生成自动编码并蒸馏场景图》，*IEEE*模式分析与机器智能汇刊，2022年。[23] James Betker、Gabriel Goh、Li Jing、Tim Brooks、Jianfeng Wang、Linjie Li、Long Ouyang、Juntang Zhuang、Joyce Lee、Yufei Guo 等, 《通过改进字幕提升图像生成效果》，2023年。[24] Xu Yang、Jiawei Peng、Zihua Wang、Haiyang Xu、Qinghao Ye、Chenliang Li、Songfang Huang、Fei Huang、Zhangzikang Li 和 Yu Zhang, 《将视觉场景图转换为图像字幕》，载于计算语言学协会，2023年。[25] Jonah Casebeer、Ge Zhu、Zhepei Wang 和 Nicholas J. Bryan, 《一种生成优先的神经音频自编码器》，载于*ICASSP*，2026年。[26] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan 等, 《语言模型是少样本学习者》，载于*NeurIPS*，2020年。[27] Qingxiu Dong、Lei Li、Damai Dai、Ce Zheng、Jingyuan Ma、Rui Li、Heming Xia、Jingjing Xu、Zhiyong Wu、Baobao Chang 等, 《上下文学习综述》，*arXiv:230100234*，2022年。。[28] Zach Evans、Julian D Parker、CJ Carr、Zack Zukowski、Josiah Taylor 和 Jordi Pons, 《Stable Audio Open》，载于*ICASSP*，2025年。[29] G Team, 《Gemma 3：一种高效的多模态开放模型》，*arXiv:2503.19786*，2025年。[30] Rithesh Kumar、Prem Seetharaman、Alejandro Luebs、Ishaan Kumar 和 Kundan Kumar, 《利用改进的RVQGAN实现高保真音频压缩》，*NeurIPS*，2023年。[31] Dimitrios Bralios、Jonah Casebeer 和 Paris Smaragdis, 《Re-bottleneck：神经音频自编码器的潜在表示重构》，载于*MLSP*，2025年。[32] Nils Reimers 和 Iryna Gurevych, 《Sentence-BERT：使用孪生BERT网络生成句子嵌入》，载于自然语言处理实证方法会议（*EMNLP*），2019年。[33] Stephen E. Robertson、Steve Walker、Susan Jones、Micheline Hancock-Beaulieu 和 Mike Gatford, 《TREC-3中的Okapi》，载于文本检索会议（*TREC-3*），1995年。