arXiv:2601.09385v1 [cs.SD] 14 Jan 2026

SLAM-LLM：面向语音、语言、音频与音乐处理的模块化开源多模态大语言模型框架及最佳实践

Ziyang Ma、Guanrou Yang、Wenxi Chen、Zhifu Gao、Yexing Du、Xiquan Li、Zhisheng Zheng、HainaZhu、Jianheng Zhuo、Zheshu Song、Ruiyang Xu、Tiranrui Wang、Yifan Yang、Yanqiao Zhu、Zhikang Niu、Liumeng Xue、Yinghao Ma、Ruibin Yuan、Shiliang Zhang、Kai Yu、Eng Siong Chng、Xie Chen

摘要—**近年来，LLaVA 等开源多模态大语言模型（MLLM）框架迅速发展，为人工智能开发者和研究人员提供了便捷的起点。然而，大多数 MLLM 框架都将视觉作为主要输入模态，对语音、音频和音乐模态的深入支持十分有限。这种情况阻碍了音频语言模型的发展，也迫使研究人员投入大量精力进行代码编写和超参数调优。我们推出了 SLAM-LLM，这是一个旨在训练定制化多模态大语言模型的开源深度学习框架，专注于语音、语言、音频与音乐处理。SLAM-LLM 提供了不同编码器、投影器、大语言模型和参数高效微调插件的模块化配置。SLAM-LLM 还针对主流任务提供了详细的训练与推理方案，以及基于大语言模型的高性能检查点，包括自动语音识别（ASR）、自动音频描述（AAC）和音乐描述（MC）。其中一些方案已经达到或接近最先进的性能，部分相关技术也已被学术论文接收。我们希望 SLAM-LLM 能够加速研究人员的迭代、开发、数据工程和模型训练。我们致力于通过这一开源框架持续推动基于音频的多模态大语言模型发展，并呼吁社区为基于大语言模型的语音、音频与音乐处理贡献力量。**

索引词—**多模态大语言模型、大语言模型、框架、工具包、语音处理、语言处理、音频处理、音乐处理**

I. 引言

大语言模型（LLM）的快速发展推动了多模态学习系统的开发

Ziyang Ma、Guanrou Yang、Wenxi Chen、Xiquan Li、Haina Zhu、Jianheng Zhuo、Zheshu Song、Ruiyang Xu、Yifan Yang、Yanqiao Zhu、Zhikang Niu、Kai Yu 和 Xie Chen 任职于中国上海交通大学计算机学院、教育部人工智能重点实验室及 X-LANCE实验室。Zhifu Gao 和 Shiliang Zhang 任职于中国杭州阿里巴巴集团通义实验室。Yexing Du 任职于中国广东鹏城实验室。ZhishengZheng 任职于美国德克萨斯州的德克萨斯大学奥斯汀分校。Tianrui Wang 任职于中国天津大学。Liumeng Xue 和 Ruibin Yuan 任职于中国香港科技大学。Yinghao Ma 任职于英国伦敦玛丽女王大学。Ziyang Ma 和 Eng Siong Chng 任职于新加坡南洋理工大学。Xie Chen 还任职于中国上海创智学院。

本研究获得国家自然科学基金委员会（编号：*U23B2018*）、上海市科技重大专项（项目编号：*2021SHZDZX0102*）以及长三角科技创新共同体联合研究项目（*2024CSJGG01100*）的资助。

通讯作者：教授 . *Xie Chen* . *e*-邮箱：*chenxie95@sjtu .edu.cn*

开源地址：*https://github.com/X-LANCE/SLAM-LLM*

这些框架整合了文本、视觉和音频等多种形式的输入。尽管 LLaVA [1]和 OpenFlamingo [2] 等近期的开源框架在视觉语言建模方面展现出卓越的能力，但它们仍然主要以视觉为中心。这些框架对非视觉模态的支持有限，尤其是在语音、音频和音乐处理方面。缺乏原生支持给从事听觉任务研究的研究人员带来了巨大挑战，往往迫使他们通过繁琐的适配改造基于视觉的系统，导致效率低下，开发流程也支离破碎。

为解决这些局限性，我们推出了 **SLAM-LLM：一个面向多模态大语言模型（MLLM）的模块化开源框架，重点关注语音、音频和音乐**。SLAM-LLM 旨在降低构建、训练和部署处理音频相关模态的基于大语言模型的系统的门槛。其核心架构采用简洁的编码器–投影器–大语言模型模块化设计，可通过 YAML 配置文件灵活定制模型组件。该框架支持广泛的预训练编码器（例如 Whisper [3],HuBERT [4], BEATs [5], MERT [6]）、投影模块（多层感知机层、卷积神经网络层、Q-Former [7]）以及大语言模型骨干网络（例如 LLaMA [8], [9], Vicuna [10], 通义千问 [11]）。框架还集成了参数高效微调（ PEFT ）策略，例如低秩适配（LoRA）和前缀微调，因此适用于低资源和领域特定适配。SLAM-LLM 的设计还解决了多模态建模中的常见工程挑战，例如处理变长输入和高效微调，同时支持快速原型开发和可复现性。

除了架构上的灵活性，SLAM-LLM 还为多项关键任务提供了全面的训练方案和预训练检查点，包括：自动语音识别（ ASR）、语音到文本翻译（S2TT）、语音情感描述（SEC）、自动音频描述（AAC）和音乐描述（MC）。针对这些任务开展的大量实验揭示了一系列重要见解。

简而言之，SLAM-LLM 做出了若干重要贡献：

1）用于开发音频相关模态基于大语言模型的模型的统一、灵活且可扩展的框架；2）广泛支持多种编码器、投影器和大语言模型的模块化组件；

3）涵盖语音、音频和音乐任务的精选训练与推理配置方案库；4）在各项基准测试中取得最先进的结果，尤其是在语音识别和音频描述生成方面；5）开源实现，鼓励社区协作并推动快速创新。

SLAM-LLM 弥合了当前多模态大语言模型生态系统中的一大空白，并推动通用音频语言模型（音频语言模型）的未来发展。通过开源该框架及配套配置方案，SLAM-LLM 邀请更广泛的社区参与这一重要但发展尚不充分的多模态人工智能领域，共同推动创新、加速进步。

II. 多模态大语言模型框架

大语言模型的发展催生了一个迅速扩展的开源工具包生态系统，旨在支持大语言模型在各种模态下的训练、微调和部署。尤其是，近期多模态学习热潮推动了多个框架的出现，使大语言模型的能力超越了纯文本理解。

在这些工具包中，已有若干通用大语言模型基础设施框架被提出，以促进模型开发与定制。值得注意的是，LLaMA-配置方案 [12],是 Meta 发布的 LLaMA 模型系列官方实现与用法集合，为基于文本的模型和视觉语言模型提供端到端支持。LLaMA-Adapter [13] 通过引入视觉语言对齐机制，提出了一种参数高效微调策略，可将冻结的 LLaMA 模型适配为支持指令遵循，并在扩展版本中支持视觉模态。LLaMA2-Accessory [14] 进一步拓展了这一系列工作，提供了全面的工具箱，支持单模态和多模态设置下的预训练、微调与推理，并原生支持集成 CLIP [15] 和 DINOv2 [16]等视觉编码器。与此同时，LLaMA-Factory [17] 为 100 多种基础模型提供统一且可扩展的训练接口，其中包括通过特定任务适配器和微调模板支持图像、音频和视频任务的多模态分支。此外，LMFlow [18]和 LitGPT 提供高效的微调与推理解决方案，重点关注内存优化、可扩展性以及对多模态输入的支持。

与这些专注于基础设施的工具包不同，另一类研究致力于开发专为视觉语言模型（VLMs）设计的端到端多模态大语言模型。MiniGPT-4 [19] 展示了，通过轻量级投影层将预训练视觉编码器与冻结的语言模型对齐，便可实现强大的视觉语言对话能力。LLaVA [1] 采用两阶段训练策略，将视觉特征对齐与指令微调相结合，从而打造出功能强大的开源视觉对话模型。TinyLLaVA [20] 在此基础上着重提升模型效率，以大幅减少参数量为代价，仍实现了具有竞争力的多模态性能。与此同时，OpenFlamingo [2] 以开放方式重新实现了 DeepMind 的 Flamingo [21]架构，支持小样本学习，适用于由交错图文对组成的序列，并提供多种规模的预训练模型。

然而，现有的大多数工具包主要面向文本或视觉任务，对语音、音频和音乐等非视觉模态的支持有限，甚至完全不支持。因此，语音识别、音频描述生成或音乐分析等听觉领域的研究人员，往往不得不改造那些架构和流水线与视觉或文本紧密耦合的工具，这需要投入大量适配工作。这种缺乏原生支持的情况阻碍了高效音频语言模型的构建。ESPnet[1] 等工具包专注于端到端语音系统，Fairseq[2] 专注于语音序列建模，但它们并未采用以大语言模型为中心的设计，而这正是 SLAM-LLM 所带来的范式转变。该框架通过模块化设计，将语音、音频和音乐编码器与大语言模型无缝集成，从而弥补了这一不足。在 SLAM-LLM 中，所有听觉任务都统一为自回归生成过程，使研究人员能够借助大语言模型强大的文本能力，同时以模块化方式融入经过验证的语音处理最佳实践，以提升生成性能。其灵活的架构支持轻松定制编码器、投影器和大语言模型，并兼容参数高效微调 [22],FSDP [23], 和 DeepSpeed [24]。凭借全面的训练和推理方案，SLAM-LLM 加速了研究，并通过开源发布推动社区协作开发。

III. SLAM-LLM 的设计

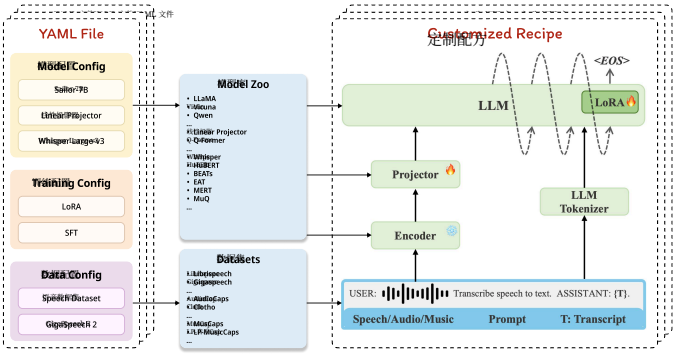


图 1：SLAM-LLM 框架中的简要工作流程。通过在 YAML 文件中指定模型、训练和数据配置，可以高效满足定制需求。图中示例展示了一个基于大语言模型的东南亚自动语音识别模型，该模型使用东南亚语言大语言模型 Sailor-7B 和相应的语音数据集 GigaSpeech 2。

A. SLAM-LLM 概述

SLAM-LLM 允许用户通过 YAML 文件配置定制化的多模态大语言模型，从而大幅减轻模型构建的负担。用户可在 YA ML

[1]https://github.com/espnet/espnet[2] https://github.com/facebookresearch/fairseq

文件中指定所需的模型组件、训练策略和数据格式，使 SLAM-LLM 能够根据具体的定制需求调整训练和推理。

图1展示了 SLAM-LLM 如何使用基于大语言模型的低资源语言自动语音识别模型运行。具体而言，在训练期间，YAML 文件将 Whisper Large-v3 [3]指定为语音编码器，将线性层指定为投影器，并将东南亚语言模型 Sailor-7B [25] 指定为大语言模型，同时使用 GigaSpeech 2 [26] 数据集进行 LoRA 微调。此设置有效地配置了一个基于大语言模型的东南亚自动语音识别模型。对于训练完成的模型，只需保存经过调优的参数（例如投影器和 LoRA），并将其视为生成资产。在推理期间，可以从模型库中获取其他组件，并根据配置将其与生成资产组装在一起，从而大幅提升开发过程中的灵活性。

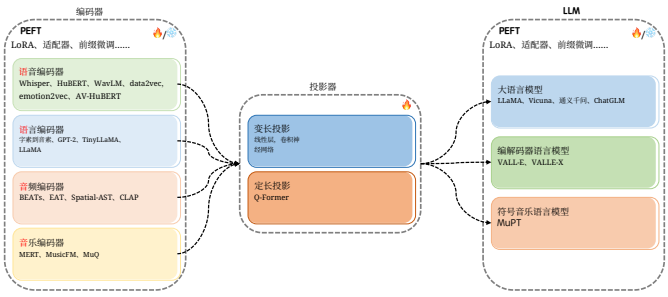


图2：SLAM-LLM 将编码器-投影器-大语言模型这三个部分的组件模块化，并根据训练和推理时的配置进行组装。

B. 模块化设计

SLAM-LLM 的一项显著特征是其模块化设计，因此它属于一个框架，而非单纯的模型。如图2所示，SLAM-LLM 将多模态大语言模型拆分为三个彼此解耦的组件：编码器、投影器和大语言模型。每个组件是否需要调优，以及如何调优，均可根据训练配置单独指定。

更具体地说，编码器充当信号感知器，支持多种用于语音、语言、音频和音乐的编码器。投影器连接编码器与大语言模型。SLAM-LLM 支持可变长度投影（例如线性投影器），也支持固定长度投影（例如 Q-Former）。大语言模型利用其丰富的知识和生成能力来产出高质量结果。在 SLAM-LLM 中，支持多种语言模型，包括文本语言模型、编解码器语言模型和符号音乐语言模型，以满足语音、音频和音乐领域的定制化需求。

基于 SLAM-LLM 的模块化设计，我们提供了多种用于实现不同任务的配置方案和检查点。在以下实验部分，我们详细展示了大多数经典任务的基础结果，尽可能优先采用主流模型组件，并开展了大规模实验。通过比较和分析各种实验结果，我们得出了若干经验性见解。这些见解不仅为优化现有系统提供了指导，也为开发新组件奠定了基础。此外，这些结果还可作为可靠的基线参考文本，供未来研究和应用参考。

C. 任务支持

虽然 SLAM-LLM 主要旨在促进基于大语言模型的理解任务（即将音频输入映射为自然语言文本），但它已扩展支持语音生成和多轮对话，以进一步提升其作为通用框架的实用性。对于生成任务，该框架支持 VALL-E X [27]，等基于编解码器的文本转语音（TTS）模型，这得益于其对编解码器语言模型的原生支持。此外，SLAM-LLM 集成了 SLAM-Omni [28]，这是一个端到端的多轮语音对话系统，可处理连续的语音表征，并生成离散语音词元作为输出。这种设计展现了该框架支持复杂、以交互为导向的应用的灵活性，拓展了自动音频描述生成或语音识别等静态任务的范畴。

IV. 基于大语言模型的语音处理

A. 自动语音识别（*ASR*）

a) 模型设置：我们研究了两类语音编码器：在大规模语音-文本配对数据上训练的监督模型，以及在无标注语音上训练的自监督模型。对于监督式编码器，我们使用 Whisper [3]系列（*tiny* 至*large-v2*），仅保留编码器作为特征提取器。对于自监督编码器，我们评估了不同规模的 HuBERT [4]和 WavLM [29]，包括预训练版本和微调版本。基础模型在 960 小时的 LibriSpeech [30]，数据上训练，而较大的模型则使用 LibriLight [31]（HuBERT 使用 6 万小时数据；WavLM 使用 9.4 万小时的 +VoxPopuli [32] + GigaSpeech [33] 数据）。HuBERT 还包含 X-Large 变体，属于目前公开可用的最大规模模型之一。对于大语言模型，我们同时考虑了经过监督微调和基于人类反馈的强化学习（RLHF）的预训练模型和聊天模型。预训练模型包括*TinyLLaMA*（1.1B）[34] 和*LLaMA-2*（7B）[9]；聊天模型包括*TinyLLaMA-Chat*（1.1B）、*LLaMA-2-Chat*（7B）和*Vicuna*（7B）[10]。所有编码器的输出频率均为 50 Hz，在输入大语言模型之前会降采样至 10 Hz。投影器的隐藏层大小为 2048；编码器输出维度和大语言模型输入维度因模型而异。训练方案和模型检查点可在此处获取[3]。为满足真实世界部署的实际需求，我们报告了使用 7B 规模大语言模型骨干网络执行一项代表性任务所需的计算资源：尽管不同实验的训练配置可能略有差异，以确保取得最佳性能，但其基础资源需求基本相同。使用 4 × NVIDIA A100（80GB）图形处理器，仅更新投影器并冻结大语言模型和编码器，在 1,000 小时语音数据上训练一个基于大语言模型的模型 1 个轮次，约需 4 小时。

b) 数据集：我们使用LibriSpeech [30] 基准，其中包含960小时的训练数据，未采用数据增强或拼接。验证和测试在10小时的dev-other、test-clean和test-other子集上进行。

[3]https://github.com/X-LANCE/SLAM-LLM/tree/main/examples/asr_librispeech

表 I：用于执行基于大语言模型的自动语音识别的不同有监督语音编码器与大语言模型组合。我们展示了不同规模的大语言模型上使用不同尺寸的 Whisper 模型所得的结果。

语音编码器	预训练模型				聊天模型					
	TinyLLaMA-1.1B		LLaMA-2-7B		TinyLLaMA-Chat-1.1B		LLaMA-2-Chat-7B		Vicuna-v1.5-7B	
	干净	其他	干净	其他	干净	其他	干净	其他	干净	其他
Whisper-tiny	12.72	21.64	16.16	25.17	9.55	21.01	8.97	18.77	7.07	16.01
Whisper-base	7.35	15.89	17.46	21.84	7.03	15.92	6.37	12.98	5.07	13.07
Whisper-small	6.61	11.81	6.41	10.88	5.94	11.5	4.51	8.94	4.19	9.50
Whisper-medium	4.65	8.95	3.35	6.10	5.01	8.67	2.71	6.37	2.72	6.79
Whisper-large	4.39	8.22	3.01	7.15	4.33	8.62	2.72	6.79	2.58	6.47

c) 不同有监督语音编码器上的实验：表 I 展示了不同有监督语音编码器与大语言模型组合下的自动语音识别结果，并突出了几个关键发现。首先，随着Whisper编码器规模增大，自动语音识别性能有所提升。例如，在使用TinyLLaMA-1.1B时，test-clean的词错误率从12.72（Whisper-tiny）降至4.39（Whisper-large），表明更大的编码器能够更好地捕捉语音特征。然而，随着规模增大，收益会逐渐减少：从Whisper-medium（4.65）到Whisper-large（4.39）的词错误率提升幅度很小。其次，在所有编码器规模下，聊天模型始终优于预训练大语言模型。例如，在使用Whisper-large时，LLaMA-2-7B在test-clean上的词错误率为3.01，而LLaMA-2-Chat-7B将其提升至2.72。这表明，监督微调增强了模型将语音嵌入视为一种“语言”的能力，并使其在微调过程中受益于类翻译学习。最后，在固定使用Whisper-large的情况下，Vicuna- 7B取得了最佳结果，在test- clean上的词错误率达到2.58，优于LLaMA-2-7B（3.01）和LLaMA-2- Chat-7B（2.72）。这表明，在用户共享的对话数据上进行微调的Vicuna具有更强的泛化能力。

d) 不同自监督语音编码器的实验：表 II 使用 Vicuna-7B-v1.5 作为固定的大语言模型，对用于基于大语言模型的自动语音识别的自监督语音编码器进行了分析。随着编码器规模增大，词错误率总体有所改善，这与监督模型中观察到的趋势一致。在基础规模（95M 个参数、768 的隐藏层大小）下，HuBERT 和 WavLM 的表现与 Whisper-small 不相上下，自监督学习并未展现出明显优势。然而，在 300M 个参数规模下，WavLM Large 的表现超过了表 I 中所有监督式编码器，包括 Whisper-medium（300M）和 Whisper-large（600M）；相比之下，HuBERT 从 Base 扩展到 Large 所带来的提升较为有限。在 LibriSpeech 上微调 HuBERT Large 后，性能显著提升，在 test-clean 上的词错误率达到 2.10%，在 test-other 上达到 4.26%，优于 WavLM Large。进一步扩展至 HuBERT X-Large（1B 个参数，经过微调）后，取得了最佳结果：在 test-clean 上的词错误率为 1.84%，在 test-other 上为 3.39%，相较于 Whisper-large 分别降低了 28.7% 和 47.4%。这些发现凸显了扩展自监督学习编码器规模并对其进行微调，对于基于大语言模型的自动语音识别具有显著益处。

e) 与非大语言模型自动语音识别模型的比较：表 III 比较了我们的基于大语言模型的自动语音识别模型（SLAM-LLM）与最先进的基于神经网络的系统，包括在 LibriSpeech-960 上训练的专用模型、在大规模无标注语音上预训练的自监督模型，以及在多语言标注数据集上训练的通用模型。对于 LibriSpeech-960 专用模型，我们将其与 ContextNet [35]、Conformer [36]、Branchformer [37]（ESP-net）和 Zipformer [38]（K2）进行比较。这些模型采用了大量

表 II：探索采用不同自监督学习（SSL）语音编码器的基于大语言模型的自动语音识别性能。LS-960 指 LibriSpeech 960 小时数据集。FT 指微调。

语音编码器	编码器参数量	隐藏 Size	投影器参数量	词错误率(%) ↓ test-clean	test-other
HuBERT Base	94.70M	768	16.26M	4.43	10.72
WavLM 基础版	94.38M	768	16.26M	4.14	9.66
HuBERT Large	316.61M	1024	18.88M	4.53	8.74
+ LS-960 微调	316.61M	1024	18.88M	2.10	4.26
WavLM Large	315.45M	1024	18.88M	2.13	4.73
+ LS-960FT 315.45M		1024	18.88M	1.96	4.18
HuBERT X-Large 964.32M		1280	21.50M	2.41	4.49
+ LS-960FT 964.32M		1280	21.50M	1.84	3.39

系统级技术——SpecAugment、速度扰动、EMA平均——并且通常会融合在 LibriSpeech 文本上训练的领域内语言模型。尽管没有采用这些工程优化，我们的基于大语言模型的模型仍优于这些专用系统中的最佳者。然而，使用 Fairseq 实现的 HuBERT-large/x-large 和使用 UniSpeech 实现的 WavLM-large 等自监督学习模型，在更大的无标注数据集上预训练并搭配领域内语言模型后，在含噪 test-other 子集上的表现优于我们的模型。这些提升很大程度上归功于语言模型集成，这表明针对特定域对大语言模型进行微调也可能带来类似的改进，但代价是泛化能力下降。与通用模型相比，我们的系统优于 Whisper-large-v2、更强的 Whisper-large-v3，以及顶尖的学术基线 OWSM-v3.1。这些结果凸显了我们基于大语言模型的方法的有效性：无需繁重的工程优化或大规模预训练，即可取得具有竞争力乃至更优的自动语音识别性能。

f) 少数民族语言数据集上的实验：除英语和中文等高资源语言外，低资源语音识别仍是一项重大挑战。本实验针对泰语（th）、越南语（vi）和印度尼西亚语（id）等东南亚少数民族语言，使用 GigaSpeech 2 [26]进行训练和评估。为模拟真实世界的低资源条件，每种语言的训练数据均限制为 200 小时。我们的模型使用 Whisper large-v3 编码器 [3],ReLU激活的多层感知机层作为投影器，并采用 Sailor-7B [25]作为大语言模型。为提升性能，我们对编码器进行部分微调：冻结前 30 层，仅更新最后两层。对于大语言模型，我们在每个自注意力层的查询和数值投影上应用 LoRA 适配器。表 IV 汇报了低资源自动语音识别任务的结果。对于泰语和越南语，基于大语言模型的模型（Whisper 编码器 +Sailor-7B）优于 Whisper 和 Whisper-finetune。在印度尼西亚语上，尽管其表现不及 Whisper-finetune，但仍然

表 III：与 LibriSpeech 上以往基于神经网络的模型相比，专用模型仅使用 960 小时的 LibriSpeech 数据集进行训练，通常采用基于 LibriSpeech 文本和转录文本构建的域内语言模型。基于自监督学习的模型是在大规模未标注数据上预训练、再通过监督进行微调的自监督模型。通用模型指在大量带标签的语音-文本对上训练的通用系统。为确保结果可复现，文中提供了代码仓库链接。“我们的方法”表示表 II 中我们最佳系统的结果。

模型	实现	语音数据（小时） 无标签/有标签	词错误率（%）↓ test-clean	test-other
专用模型				
ContextNet-large [35]	-	-/960	2.1	4.6
+领域内语言模型			1.9	4.1
Conformer-large [36]	-	-/960	2.1	4.3
+领域内语言模型			1.9	3.9
Branchformer-large [37]	ESPnet	-/960	2.4	5.5
+领域内语言模型			2.1	4.5
Zipformer-large [38]	K2	-/960	2.0	4.4
+领域内语言模型			1.9	3.9
基于自监督学习的模型				
wav2vec 2.0-large [39]	Fairseq	60k/960	2.2	4.5
+领域内语言模型			1.8	3.3
HuBERT-large [4]	Fairseq	60k/960	2.2	4.5
+领域内语言模型			1.9	3.3
HuBERT-x-large [4]	Fairseq	60k/960	2.0	3.7
+领域内语言模型			1.8	2.9
WavLM-large [29]	UniSpeech	94k/960	2.7	5.0
+领域内语言模型			1.8	3.2
通用模型				
Whisper-large-v2 [3]	OpenAI/Whisper	-/680k	2.7	5.2
Whisper-large-v3 [3]	OpenAI/Whisper	-/1000k	2.0	3.9
OWSM-v3.1 [40]	ESPnet	-/180k	2.4	5.0
基于大语言模型的模型				
Ours	SLAM-LLM	-/960	1.8	3.4

优于基础版 Whisper 模型。这些结果表明，在资源受限的环境下，引入大语言模型能够有效利用丰富的文本知识来提升自动语音识别性能。针对东南亚语言进行微调的 Sailor-7B 进一步提升了性能。与基于 Vicuna 的模型相比，基于 Sailor 的系统在三种语言上均取得了更好的结果，凸显了在低资源场景中选择特定语言大语言模型的价值。

表 IV：泰语（th）、越南语（vi）和印度尼西亚语（id）等低资源语言中，Whisper 与基于大语言模型的自动语音识别在词错误率/汉字错误率方面的比较。“+微调”指使用与基于大语言模型的自动语音识别模型相同数量的低资源语言训练数据进行微调的模型。

模型	语言（词错误率%）↓ th	vi	id
Whisper 编码器 + Whisper 解码器	20.44	17.94	20.03
+ 微调	17.08	16.15	**17.13**
Whisper 编码器 + Vicuna-7B	17.25	17.95	19.83
Whisper 编码器 + Sailor-7B	**16.13**	**15.10**	19.48

g) 代码切换数据集上的实验：我们还在 ASRU 2019 中英代码切换挑战赛数据集上进行了实验 [41], 。该数据集包含 200 小时的代码切换语音和 500 小时的纯普通话语音。本研究仅使用其中 200 小时的代码切换子集进行训练，并使用 20 小时的测试集进行评估。我们的系统采用 Whisper Large-v3 作为语音编码器，Qwen2-7B 作为大语言模型解码器，并使用轻量级线性投影器对齐语音和文本模态。为提升性能，我们在大语言模型每个自注意力层的查询矩阵和值矩阵上应用 LoRA 适配器。表 V 显示，在代码切换测试集的所有指标上，基于大语言模型的模型均优于 Whisper Large-v3，MER 相对降低了 20.2%，凸显了基于大语言模型的架构的有效性。

表 V：Whisper Large-v3 与基于大语言模型的自动语音识别在中英代码切换自动语音识别任务中的比较。MER 表示中文字符和英语单词的混合错误率。

模型	中文汉字错误率	英文词错误率	MER
Whisper large-v3	7.85	35.18	10.01
Ours	**5.97**	**26.84**	**7.99**

B. 上下文自动语音识别（*CASR*）

1）任务设置：我们研究两类上下文自动语音识别任务：**视觉上下文语音识别**和**上下文偏置语音识别**，如图 3 所示。在视觉上下文自动语音识别中，利用从演示文稿幻灯片中提取的文本关键词来提高会议内容的转写准确率。每个语音片段都与预处理后的光学字符识别结果和特定幻灯片关键词配对。在上下文偏置自动语音识别中，会提供一份预先定义的列表，其中包含数百至数千个偏置词条，例如命名实体、联系人姓名或歌曲名称，以帮助识别罕见或领域专用词汇。*2*）视觉上下文语音识别：

a）数据集：我们使用 SlideSpeech [42] 数据集进行训练和评估。这个大规模视听语料库由 YouTube 会议视频构建而成，并提供与同步幻灯片对齐的高质量转写语音。它包含 720p 视频、16kHz 音频、预处理后的光学字符识别结果，以及按片段提取的关键词，可支持多模态自动语音识别任务。该数据集提供两个训练集：大型训练集（L95），共 473 小时；较小的子集（S95），共 161 小时。开发集和测试集分别包含 5.07 小时和 8.75 小时。语音与幻灯片之间的对齐使该数据集非常适合文本增强多模态自动语音识别，尤其适用于纠正常被传统系统误识别的领域专用或专有术语。

b) 模型设置：我们采用官方的 WavLM Large 模型作为语音编码器、Vicuna-7B 作为大语言模型解码器，并使用轻量级线性投影器将语音特征映射到大语言模型输入空间。投影器首先通过一维卷积将 50Hz 特征下采样至 10Hz，然后经过两个隐藏维度为 2048 的线性层。训练方案和模型检查点可在此处找到[4]。

c) 实验：表 VI 报告了在 SlideSpeech 数据集上的性能，采用词错误率（WER）、偏置词错误率（B-WER）、无偏置词错误率（U-WER）和关键词召回率进行评估。我们在 L95/S95 上训练的基线模型，词错误率分别为 9.4%/11.7%，相较于 SlideSpeech

[4]https://github.com/X-LANCE/SLAM-LLM/tree/main/examples/mala 自动语音识别 SlideSpeech

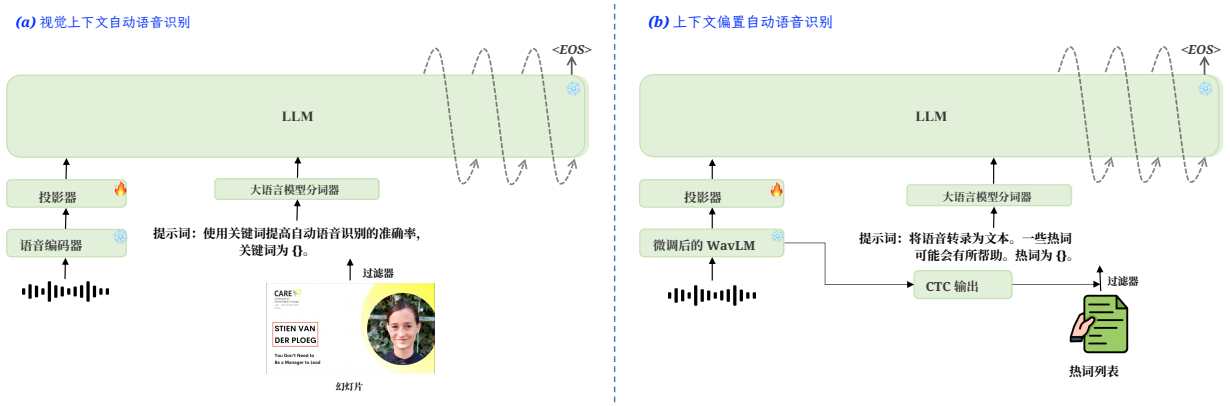


图 3：(a) SLAM-LLM 框架中基于大语言模型的视觉上下文自动语音识别。(b) SLAM-LLM 框架中基于大语言模型的上下文偏置自动语音识别。

表 VI：在 S95/L95 数据集上训练，并在开发集/测试集上评估有无上下文关键词时的自动语音识别结果，与先前基于神经网络的模型进行比较。

训练	模型	上下文相关关键词	Dev				Test			
			WER	B-WER	U-WER	召回率 ↑	WER	B-WER	U-WER	召回率 ↑
S95 (161h)	SlidesSpeech [42]	✗	21.05	31.27	20.29	68.76	21.22	26.60	20.83	73.51
	CPP [42]	✓	20.80	28.61	20.22	71.48	20.95	24.05	20.73	76.10
	LCB-net [43]	✓	18.80	27.90	18.11	72.09	19.21	23.70	18.89	76.48
	Ours	✗	11.57	16.23	**11.28**	83.83	11.80	13.52	11.71	86.71
	Ours	✓	**11.14**	**8.92**	11.36	**91.44**	**11.26**	**7.67**	**11.52**	**92.50**
L95 （473小时）	SlidesSpeech [42]	✗	13.09	16.13	12.87	83.90	12.89	12.70	12.90	87.43
	CPP [42]	✓	12.64	12.39	12.66	87.64	12.38	9.32	12.60	90.86
	LCB-net [43]	✓	12.21	12.12	12.21	87.98	12.02	9.03	12.24	91.12
	Ours	✗	9.38	11.98	9.19	88.08	9.34	9.52	9.33	90.64
	+ LoRA	✗	8.82	9.62	8.77	90.38	8.61	7.34	**8.72**	92.84
	Ours	✓	8.91	6.07	9.13	94.02	9.14	5.47	9.42	94.87
	+ LoRA	✓	**8.30**	**5.22**	**8.53**	**94.87**	**8.46**	**4.89**	8.73	**95.31**

上下文语音识别基线，分别相对降低了 27.9%/44.7%。加入关键词提示后，词错误率进一步降至 9.0%/11.2%，提升了 3.6%/4.1%。值得注意的是，L95 上的 B-WER 从 10.8% 降至 5.8%，S95 上则从 14.9% 降至 8.3%；与此同时，召回率分别从 89.4% 提升至 94.4%，以及从 85.3% 提升至 92.0%。U-WER 保持稳定，表明关键词提示得到了有效利用，且未损害通用转录效果。使用 LoRA 适配器对大语言模型进行微调后，结果进一步提升。在 L95 上，词错误率从 9.4% 降至 8.7%；使用关键词提示时则降至 8.4%。与 CPP [42],LCB-Net [43]，以及 SlideSpeech 基线等先前的上下文语音识别系统相比，我们基于大语言模型的方法取得了显著更低的词错误率，凸显了其利用上下文实现准确转录的卓越能力。

3）上下文偏置语音识别：

a) 数据集：我们使用 LibriSpeech 语料库进行训练和评估，遵循以往的工作 [44], [45]。基于大语言模型的自动语音识别模型使用完整的 960 小时训练集进行训练，并采用官方的 WavLM Large 作为预训练编码器。在标准的 dev-clean/dev-other 和 test-clean/test-other 集上进行评估。对于上下文语音识别，我们采用 [44]，中的人工偏置列表，其中出现频率最高的 5,000 个训练词语被标记为常见词，其余则标记为罕见词。每个测试时偏置列表都包含参考文本中的罕见词，以及从包含 209.2K 个词的罕见词词汇表中抽取的干扰词。通过改变干扰词的数量，构建大小为 $N = 100$、500、1000、2000 的列表。

b) 模型设置：我们使用 CTC 损失，在 960 小时的 LibriSpeech 训练集上对官方 WavLM Large 模型（315.5M 个参数）进行全量微调。微调后的模型用作语音编码器。其余架构与视觉上下文语音识别设置相同，采用 Vicuna-7B（6.7B）作为大语言模型解码器，并使用轻量级线性投影器（15.7M）。训练方案和模型检查点可在此处找到[5]。

c) 不同偏置列表的实验：表 VII 展示了我们的 CTC 辅助的基于大语言模型的上下文自动语音识别（CASR）模型在 LibriSpeech 上的表现，评估指标为词错误率、B-WER 和 U-WER。对于非上下文语音识别，使用预训练的 WavLM Large 编码器，在 test-clean/test-other 上的词错误率分别为 2.13%/4.73%。微调后性能略有提升，词错误率降至 2.11%/4.20%，作为我们的基线。在上下文语音识别中，使用空字符串作为提示词时，test-clean 上的词错误率/B-WER 为 1.96%/9.33%，test-other 上为 4.18%/20.02%，表明即使没有明确提供热词，模型仍具有较强的鲁棒性。提供真实热词后，词错误率/B-WER 进一步降至 1.13%/2.78% 和 2.68%/6.00%，代表性能上限。在实际场景中，偏置列表混合相关术语与干扰词时，最佳表现

[5]https://github.com/X-LANCE/SLAM-LLM/tree/main/examples/ 上下文语音识别

出现在使用 100 词列表时，词错误率/B-WER 降至 1.27%/3.67 % 和 2.72%/8.02%，相较于基线，词错误率/B-WER 分别相对降低 39.81%/63.37% 和 35.24%/61.37%，而 U-WER 保持稳定。随着列表规模增大，性能略有下降，但即使使用 2,000 词列表，模型在 test-clean 上仍达到 1.38%/4.41%，在 test-other 上达到 3.20%/10.02%，保持了显著的性能提升。这些结果表明，模型具备强大的能力，能够有效过滤并利用偏置信息。

表 VII：基于大语言模型的上下文语音识别在 LibriSpeech 测试集上的性能。“无偏置”表示添加空字符串，“偏置列表”指纳入从完整偏置列表中过滤出的热词，“真实标注热词”表示在推理时纳入准确的真实热词。

WavLM 编码器	提示词	test-clean			test-other		
		词错误率	B-WER	U-WER	词错误率	B-WER	U-WER
预训练	✗	2.13	10.15	1.20	4.73	22.43	2.84
CTC 微调	✗	2.11	10.02	1.20	4.20	20.76	2.43
CTC 微调	无偏置	1.96	9.33	1.11	4.18	20.02	2.49
	100 偏置词	1.27	3.67	1.00	2.72	8.02	2.16
	500 偏置词	1.33	3.92	1.03	3.04	9.04	2.40
	1000 偏置词	1.33	4.16	1.00	2.99	9.33	2.31
	2000 偏置词	1.38	4.41	1.03	3.20	10.02	2.47
	真实标注热词	**1.13**	**2.78**	**0.94**	**2.68**	**6.00**	**2.32**

表 VIII：采用人工偏置列表的基于神经网络的上下文自动语音识别模型性能对比，这些模型由 [44] 提出，使用 Librispeec 测试集，偏置列表大小 N 设为 1000。

模型	test-clean			test-other		
	词错误率	B-WER	U-WER	词错误率	B-WER	U-WER
CPPNet [45]	3.81	11.40	2.90	8.75	25.30	6.90
深度偏置+BPB [46]	3.47	7.70	3.00	7.34	15.80	6.40
$\text{TCPGen}_{\text{+GNN enc.}}$ [47]	3.10	6.70	-	7.90	17.80	-
GA-CTC [48]	2.40	6.30	2.00	6.20	15.20	5.20
$\text{TCPGen}_{\text{+p+phn-awareQ}}$ [49]	2.20	4.60	-	6.00	12.30	-
DB-NNLM [44]	2.14	6.70	1.60	6.35	17.20	5.10
我们的方法 [50]	**1.33**	**4.16**	**1.00**	**2.99**	**9.33**	**2.31**

d) 与非基于大语言模型的上下文自动语音识别模型的比较：表 VIII 展示了使用 [44] 中提出的人工偏置列表，在 Librispeech 测试集上与各种基于神经网络的上下文自动语音识别模型进行的性能比较，其中偏置列表大小设为 1000。我们基于大语言模型的方法显著优于传统的基于神经网络的方法。

C. 视觉语音识别（*VSR*）

视觉语音识别（VSR）旨在通过分析说话者面部的视觉线索来转录语音，如图所示

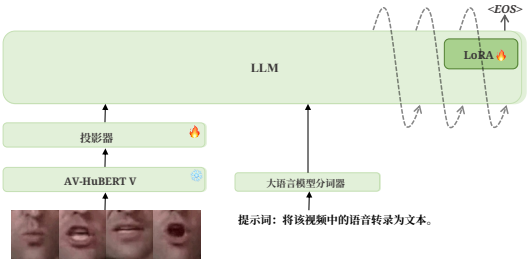


图 4：SLAM-LLM 框架中基于大语言模型的视觉语音识别（VSR）模型架构。

4. 我们在 SLAM-LLM 框架中探索基于大语言模型的视觉语音识别。

a) 数据集：我们在大规模音视频语音识别基准数据集 LRS 3 [51]，上开展实验。该数据集包含超过 400 小时的 TED 和 TEDx 视频，并配有对齐字幕和词边界。数据集分为三个子集：119k 条语音（407 小时）用于预训练，32k 条语音（30 小时）用于训练验证，1,452 条语音（1 小时）用于测试。由于头部姿态、光照、说话风格和说话人的差异很大，LRS3 带来了重大挑战。

表 IX：基于 SLAM-LLM 框架的视觉语音识别模型在 LRS3 数据集上的性能。

模型	编码器	解码器	准则	可训练参数	WER
AV-HuBERT [52]	$\text{AV-HuBERT}_{\text{V}}$	Transformer	CTC+注意力	325	28.6
Ours		$\text{Vicuna}_{\text{v1.5-7B}}$	仅解码器	49	28.3

b) 模型设置：我们使用官方的 AV-HuBERT Large 模型（477.3M），该模型在 LRS3 和 VoxCeleb2 [53]（1,759 小时未标注数据）上进行预训练，并在 433 小时带标注的 LRS3 数据上针对视觉语音识别任务进行微调。Vicuna-7B 作为大语言模型解码器，轻量级线性投影器作为适配器。为提升性能，我们在大语言模型每个自注意力模块中的键、查询、值和输出投影层应用 LoRA 适配器，秩设为 32、alpha 设为 32、dropout 设为 0.05。这额外引入了 33.6M 个可训练参数。训练方案和模型检查点可在此处找到[6]。

c) 实验：表 IX 展示了我们在 LRS3 数据集上训练和评估的基于大语言模型的视觉语音识别模型的性能。在使用相同数量的带标注和未标注训练数据时，我们的 SLAM-VSR 模型取得了更优的词错误率 28.3；相比之下，针对视觉语音识别任务微调的 AV-HuBERT 基线模型表现较差，而我们的模型仅有 4,900 万个可训练参数，充分体现了基于大语言模型架构的高效性。

D. 语音到文本翻译（*S2TT*）

a) 模型设置：SLAM-LLM 中基于大语言模型的语音到文本翻译（S2TT）模型采用冻结的 Whisper 编码器和 Qwen2-7B 大语言模型，唯一的可训练组件是 Q-Former 投影。编码器提取语音特征，并通过 Q-Former 对其进行压缩，使其与大语言模型输入空间对齐。随后，大语言模型根据拼接后的音频和文本嵌入生成文本输出。训练方案和模型检查点可在此处找到[7]。

b) 数据集：我们使用 CoVoST-2 [54] 进行训练，并使用 MuST-C [55] 进行零样本评估。

[6]https://github.com/X-LANCE/SLAM-LLM/tree/main/examples/vsr LRS3[7] https://github.com/X-LANCE/SLAM-LLM/tree/main/examples/st covost2

表 X：该提示设计旨在对三个任务进行指令微调：自动语音识别、多模态机器翻译和语音识别翻译。我们设计了简洁而有效的提示来区分这些任务。

Task	指令	预测
ASR	<|英语|>	明天会下雨吗?
多模态机器翻译	明天会下雨吗? <|de|>	明天会下雨吗? <|de|>明天会下雨吗?
SRT	<|de|>	明天会下雨吗? <|de|>明天会下雨吗?

表 XI：CoVoST-2 和 MuST-C 数据集上的语音翻译 BLEU 分数。我们在德语（De）、日语（Ja）和中文（Zh）上进行了实验。我们用下划线标出此前最先进水平的基线，并用 **粗体** 标出超越最先进水平的表现。

En→X	CoVoST-2				MuST-C(zero-shot)			
	De	Ja	中文	平均值	德语	Ja	中文	平均值
级联式语音翻译方法								
Whisper+NLLB$_{3.3b}$ [56]	-	19.0	32.0	-	-	-	13.5	-
Whisper+通义千问$_{7B\text{-}chat}$ [57]	25.3	22.7	43.6	30.5	22.8	-	-	-
端到端方法								
SALMONN [58]	18.6	22.7	33.1	24.8	-	-	-	-
BLSP-KD [57]	24.4	21.3	41.3	29.0	20.7	-	-	-
SeamlessM4T-V2 [59]	37.0	23.5	34.6	31.7	-	-	18.1	-
Qwen2-Audio [60]	29.9	28.6	45.2	34.5	22.4	7.8	18.3	16.1
Ours-7B	**28.7**	**30.8**	**47.7**	**35.7**	**18.3**	**11.3**	**21.2**	**16.9**

c) 实验：我们采用多模态思维链（CoT）方法，将语音翻译（ST）任务分解为两个依次进行的阶段：先进行自动语音识别（ASR），再进行多模态机器翻译（MMT），统称为语音识别与翻译（SRT），如表 X 所示。这种方案支持端到端生成转录文本和译文。我们在 CoVoST-2 上采用课程学习进行三阶段监督微调，并在 MuST-C 上进行零样本评估。如表 XI 所示，我们的模型在 CoVoST-2 上达到最先进水平（SOTA），并在 MuST-C 英译中零样本任务中优于现有方法。

E. 语音情感描述（*SEC*）

a) 任务设置：语音情感描述（SEC）旨在用自然语言描述语音情感，为传统的语音情感识别（SER）提供一种更细腻的替代方案。传统方法通常依赖固定的情感类别，难以捕捉人类情感的复杂性。SECap [61] 首次采用编码器–投影器–大语言模型框架实现 SEC，生成丰富的情感描述。SEC 已支持 PerceptiveAgent [62], 等应用，该应用能够提升对话中的情感意识；也支持 AVI-Talking [63], ，后者能够增强说话人脸动画的表现力。除此之外，SEC 在许多尚未探索的应用领域也大有潜力。

b) 数据集：我们使用约 40 小时的内部情感语音数据训练模型。数据集中的每个音频片段都标注了 1 至 3 条与情绪相关的描述，用自然语言描述其中表达的情感内容。这些描述经过精心整理，旨在捕捉超越传统类别标签的多样情感细微差别，使模型能够学习细粒度的情感表征。

c) 模型设置：SECap [61]采用 HuBERT [4] 作为音频编码器、中文 LLaMA-2 解码器以及 Q-Former Bridge-Net 来提取与情感相关的声学特征，取得了当前最先进的 SEC 性能。然而，要实现其所报告的性能提升，该方法依赖复杂的训练策略——语音–转录互信息学习（STMIL）和语音–描述对比学习（SCCL）。为增强内在的情感感知能力，同时简化训练流程，我们将音频编码器替换为 emotion2vec [64], ，这是一种专为情感表征设计的自监督模型。我们采用 Vicuna-7B 作为大语言模型，并保留 Q-Former 作为投影模块。训练方案和模型检查点可在此处找到[8]。

d) 实验：遵循 SECap 中的评估协议，我们在由 [61], 发布的 EMOSpeech 测试集中的 600 个句子上测试模型，并基于 MACBERT [65] 报告句子相似度（SIM）作为主要指标。表 XII 对比了 SLAM-LLM 中基于大语言模型的声音事件分类模型与 SECap 系列模型，以及 SECap 论文中报告的 HTSAT–BART 基线。在不使用任何辅助目标的情况下，我们的方法取得了 71.10% 的 SIM，优于原始 SECap，并缩小了与工程设计最复杂的 SECap 变体之间的差距。相较于 HTSAT–BART 基线，提升超过 7 个百分点，凸显了将自监督学习（SSL）emotion2vec 编码器与强大的大语言模型解码器相结合的有效性。

表 XII：使用 SLAM-LLM 框架进行语音情感描述的性能。不同模块的具体信息见表中。除我们的方法外，其他结果均来自 SECap 论文。

模型	编码器	解码器	SIM(%)
基线 [61]	HTSAT	BART	63.95
SECap [61]	HuBERT	Chinese-LLaMA2-7B	67.29
+STMIL			68.75
+SCCL			71.95
Ours	emotion2vec	Vicuna-v1.5-7B	71.10

F. 主要结论

从上述实验中可以得出几项关键结论，为未来基于大语言模型的语音任务研究与开发提供指导：

- **更大的模型能够提升性能**。显然，更大的语音编码器和更大的大语言模型都能持续带来更好的自动语音识别性能。
- **聊天大语言模型优于预训练大语言模型**。聊天模型（例如 Vicuna-7B）在自动语音识别任务中始终优于预训练大语言模型，尤其是在搭配更大的语音编码器时。
- **规模足够大时，自监督编码器表现更优**。自监督编码器优于监督式编码器

[8] https://github.com/X-LANCE/SLAM-LLM/tree/main/examples/sec 声音事件分类 emotioncaps

当达到足够大的规模时。微调这些自监督编码器能显著提升性能；尽管我们的实验有限，但我们认为，使用领域内文本数据微调大语言模型也能提升性能。

- **领域专门化大语言模型是得力的助手**。例如，使用基于低资源语言训练的大语言模型，可以提升基于大语言模型的低资源语言自动语音识别的效果。

- **Whisper模型在截断方面存在局限**。Whisper模型会将语音输入填充至30秒，在截断时性能会下降。将语句填充至30秒也会显著增加大语言模型后训练阶段的计算需求。我们建议使用具有可变长度嵌入的自监督模型作为编码器，以缓解这一问题。

有关子任务的更多详细信息和具体技术内容，请参阅 SLAM-LLM 系列的研究 [50], [66], [67] 。

V. 基于大语言模型的音频与音乐处理

A. 自动音频描述（AAC）

自动音频描述（AAC）旨在根据输入音频生成细粒度的自然语言描述，是音频处理中的一项关键任务。如图 5 所示，我们使用 SLAM-LLM 框架，重点关注两种主流的自动音频描述范式：（1）**原始设置**，也称为全监督音频描述生成，即使用配对音频文本数据训练模型。（2）**零样本设置**，即仅使用文本数据训练模型，并期望模型在推理时以零样本方式为音频片段生成描述。

a) 数据集：我们在音频描述生成实验中使用了四个主要的音频文本数据集：Clotho [68], AudioCaps [69],WavCaps [70], 和 MACS [71]。对于 Clotho，我们使用了 2.1 版，其中包含时长为 15 至 30 秒的音频片段。该数据集包括 3,839 个训练样本、1,045 个验证样本和 1,045 个评估样本，每个音频片段配有五条描述。AudioCaps 包含超过 50,000 段源自 AudioSet [72]、时长为 10 秒的音频片段。该数据集分为训练集（49,274 段片段，每段配有一条描述）、验证集（494 段片段，每段配有五条描述）和测试集（957 段片段，每段配有五条描述）。WavCaps 包含从多个来源收集的 403,050 段音频片段，包括 AudioSet、BBC Sound Effects、FreeSound 和 SoundBible。MACS 包含 3,930 个时长为 10 秒的音频文件，每个文件对应 2 至 5 条描述，主要录制于三种声学环境中（机场、公共广场和公园）。在实验中，我们使用了 Clotho、AudioCaps 和 MACS 的训练集，以及完整的 WavCaps 数据集进行预训练。此外，我们还使用释义方法扩充了 Clotho训练集。该方法借鉴了机器翻译中的回译 [73] 技术，以便在预训练期间扩充数据集。

b) 评估指标： 为了评估生成的音频描述的质量，我们使用了多种标准音频描述生成指标：METEOR [74], CIDEr [75], SPICE [76], SPIDEr [77], SPIDEr- FL [78] 和 FENSE [78]。METEOR 考虑单元准确率、召回率、同义词和词干提取。CIDEr 使用 TF-IDF 衡量 n 元语法共识。SPICE 对比生成描述与参考文本描述的语义图。SPIDEr 通过线性组合 CIDEr 和 SPICE，实现平衡评估。SPIDEr- FL 进一步纳入了 FENSE 的流畅度检测；FENSE 使用 Sentence-BERT 衡量语义相似度，并结合流畅度错误检测。

c) 模型设置：在原始设置中，我们使用冻结的 EAT 模型 [79] 作为音频编码器，提取细粒度音频表示，随后对其进行下采样，并通过线性投影器与大语言模型嵌入对齐。具体而言，投影器使用两层线性层将 50Hz 特征下采样至 10Hz，中间隐藏层维度为 2048。在解码时，大语言模型 Vicuna [10], 基于这些拼接后的表示生成描述，并通过 LoRA 进行高效微调。在推理过程中，系统使用不同的束宽度进行束搜索，生成多个候选描述，并采用 CLAP-Refine [80]策略选出与音频最匹配的描述作为最终输出。训练方案和模型检查点可在此处找到[9]。

对于零样本音频描述，我们使用冻结的 CLAP[81] 模型作为音频与文本编码器。训练期间，首先由 CLAP 模型的文本分支对原始描述进行编码，然后通过两层线性投影器与大语言模型对齐。经过 LoRA [82]方法高效微调的大语言模型，会学习根据映射后的 CLAP 潜在表示以及从数据存储库检索到的编码提示，重构真实描述。推理期间，CLAP 模型的文本分支被音频分支取代，系统以零样本方式描述音频片段。采用基于投影的解码 [83] 和检索增强生成（RAG），以缩小模态差距并提升描述质量。音频嵌入被投影到文本嵌入空间，同时从数据存储库检索相似描述作为提示，引导大语言模型。训练方案和模型检查点可在此处找到 [10]。

d) 基础音频描述实验：对于原始音频描述生成系统，我们先使用预训练数据集对模型进行训练，随后分别在 AudioCaps 和 Clotho 数据集上进行微调。此外，我们还在这两个数据集上开展了全面的消融研究，以评估各个组件的影响。

表 XIII 比较了我们的模型与此前最先进的音频描述生成系统的性能。我们纳入了三个采用非基于大语言模型解码的模型：EnCLAP [84],WavCaps [70], 和 Wu 等人 [85]；以及两个采用基于大语言模型解码的模型：Tang 等人 [86] 和 LOAE [87]。总体而言，基于大语言模型的模型往往能够生成质量更高的描述，这可能得益于其更强的推理能力。与此前的模型相比，我们提出的系统在 Clotho 和 AudioCaps 数据集上均展现出持续的改进。在 Clotho 上，它在所有音频描述生成指标上都取得了最高分，其中 FENSE（54.0%）的提升尤为显著，超过了 LOAE 和

[9] https://github.com/X-LANCE/SLAM-LLM/tree/main/examples/slam 自动音频描述 [10] https://github.com/X-LANCE/SLAM-LLM/tree/main/examples/drcap 零样本自动音频描述

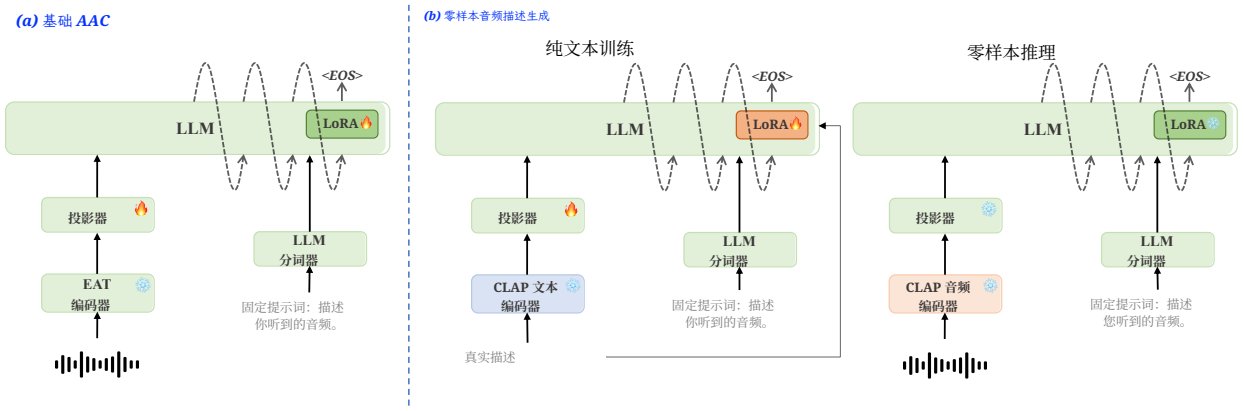


图 5：（a）SLAM-LLM 框架中基于大语言模型的原始自动音频描述生成；（b）SLAM-LLM 框架中基于大语言模型的零样本自动音频描述生成

表 XIII：Vanilla （完全监督）音频描述模型在 Clotho 和 AudioCaps 评估集上的性能比较。所使用的预训练数据集包括 AudioCaps（AC）、Clotho（CL）、WavCaps（WC）、MACS（MA）、LibriSpeech（LS） [30],和 GigaSpeech（GS） [33]。报告的指标包括 METEOR（MT）、CIDEr（CD）、SPICE（SC）、SPIDEr（SD）、SPIDEr-FL（SF）和 FENSE（FS）。CL_C 和 CL_P 分别表示采用 ChatGPT 混合方法和释义改写方法增强的 Clotho 训练集。所有指标均以数值越高表示性能越好。

模型	预训练数据	Clotho 评估 (%)						AudioCaps 评估 (%)					
		MT	CD	SC	SD	SF	FS	MT	CD	SC	SD	SF	FS
EnCLAP-large [84]	AC+CL	18.6	46.4	13.3	29.9	28.9*	50.7*	25.5	80.3	18.8	49.5	49.9*	65.5*
WavCaps† [70]	AC+CL+WC	18.5	48.8	13.3	31.0	29.6*	50.1*	25.0	78.7	18.2	48.5	48.3*	64.2*
Wu 等人 [85]	AC+CL_C	19.3	50.6	14.6	32.6	32.6	53.6	-	-	-	-	-	-
Tang 等人 [86]	AC+CL+WC+LS+GS	-	-	-	31.8	-	-	-	-	-	50.6	-	-
LOAE [87]	AC+CL+WC	19.7	51.4	14.7	33.1	32.8	53.2	26.7	81.6	19.3	50.5	50.4	66.2
Ours	AC+CL $_P$+WC+MA	**19.7**	**51.5**	**14.8**	**33.2**	**33.0**	**54.0**	**26.8**	**84.1**	**19.4**	**51.8**	**51.5**	**66.8**

*对于开源模型，我们使用评估划分评估了原始论文中未报告的指标。

†选取 WavCaps 在两个数据集上的最佳性能进行比较（模型架构可能有所不同）。

以及 Wu 等人的模型。在 AudioCaps 上，改进更加显著：我们的模型获得了 84.1% 的 CIDEr 分数，显著超过 LOAE（81.6%）；同时取得了 66.8% 的 FENSE 分数，优于所有其他模型。

表 XIV 展示了在 AudioCaps 和 Clotho 上开展的全面消融研究结果。该研究探讨了不同音频编码器、音频编码器的不同版本（预训练版本与在 AudioSet 上微调的版本）、是否进行预训练、参数高效微调（例如 LoRA）的使用情况，以及不同的解码策略（束搜索与 CLAP-Refine）所产生的影响。结果表明，经过微调的音频编码器，尤其是在分类任务中表现强劲的编码器，能够显著提升自动音频描述质量。对基于大语言模型的 AAC 模型进行预训练可以进一步提升性能，而 LoRA 则能够提高训练效率。此外，CLAP-Refine 解码方法通过增强与输入音频之间的对齐，持续改善描述质量，从而获得更高的 CIDEr 和 SPIDEr 分数。

e) 零样本音频描述生成实验：为评估零样本音频描述生成的性能，我们在域内和跨域场景下进行了实验。对于域内场景，系统按照标准的训练集、验证集和测试集划分，在同一数据集上进行训练和评估。对于跨域场景，系统在源数据集的训练集上进行训练，并在评估集上进行评估

表 XIV：AudioCaps 和 Clotho 上的消融研究。*PT*表示预训练模型，*FT* 表示微调版本。BS 表示束搜索解码，CR 表示 CLAP-Refine。

数据集	音频预训练参数高效微调			解码	指标（%）
	编码器	模型	(LoRA)		机器翻译 字幕描述 语义相似度 语义差异
AudioCaps	$BEATs_{PT}$	✗	✗	BS	23.2 69.6 17.2 43.4
	$BEATs_{FT}$	✗	✗	BS	23.9 69.8 17.3 43.5
	EAT_{PT}	✗	✗	BS	23.5 67.0 17.3 42.2
	EAT_{FT}	✗	✗	BS	25.0 75.3 18.5 46.9
	EAT_{FT}	✗	✓	BS	26.0 80.0 18.3 49.2
	EAT_{FT}	✓	✓	BS	26.1 82.7 **19.5**51.1
	EAT_{FT}	✗	✓	CR	26.6 81.4 19.4 50.4
	EAT_{FT}	✓	✓	CR	**26.8 84.1** 19.4 **51.8**
Clotho	EAT_{FT}	✗	✗	BS	17.1 34.1 11.3 22.7
	EAT_{FT}	✗	✓	BS	17.0 38.9 12.0 25.4
	EAT_{FT}	✓	✓	BS	19.2 49.8 14.4 32.1
	EAT_{FT}	✗	✓	BS	19.4 51.0 14.6 32.8
	EAT_{FT}	✓	✓	CR	**19.7 51.5 14.8 33.2**

目标数据集的。具体而言，对于 Clotho 评估，源数据集为 AudioCaps；对于 AudioCaps 评估，源数据集为 Clotho。

表 XV 展示了我们的零样本音频描述生成模型与此前最先进方法的性能比较。ZerAuCap[88] 使用 CLAP 引导大语言模型生成描述，

表 XV：零样本音频描述系统的性能比较。指标说明：MT：METEOR，CD：CIDEr，SP：SPICE，SD：SPIDEr，FS：FENSE。所有指标的数值越高，性能越好。

方法	场景	Clotho 评估（%） MT	CD	SP	SD	FS	AudioCaps 评估 （%） MT	CD	SP	SD	FS
ZerAuCap [88]	In	9.4	14.0	5.3	9.7	-	12.3	28.1	8.6	18.3	-
WSAC [89]	In	17.4	37.1	12.3	24.7	-	24.1	63.3	17.3	40.3	-
Zhang 等人 [90]	域	17.5	41.1	12.2	26.7	48.8	22.0	64.4	15.6	40.0	-
Ours		**18.2**	**43.8**	**13.3**	**28.5**	**53.0**	**25.3**	**70.5**	**18.0**	**44.2**	**66.2**
WSAC [89]	跨域	12.0	20.6	8.2	14.4	-	17.3	25.6	12.0	18.8	-
Zhang 等人 [90]	域	13.2	24.8	9.3	17.1	-	18.2	33.7	12.4	23.0	52.1
Ours		**15.0**	**33.3**	**10.4**	**21.8**	**52.2**	**22.9**	**44.3**	**17.0**	**30.6**	**62.6**

表 XVI：针对我们的零样本音频描述生成模型 DRCap，在 AudioCaps（域内）和 AudioCaps ⇒ Clotho（跨域）上的消融研究。PD 表示投影解码。

主要组件	AudioCaps（%） 机器翻译	字幕描述	SP	语义差异	特征选择	AudioCaps⇒Clotho（%） 机器翻译	字幕描述	SP	语义差异	特征选择
Ours	**25.5**	**70.5**	18.0	**44.2**	**66.2**	**15.0**	**33.3**	**10.4**	**21.8**	**52.2**
- 不使用检索增强生成	25.0	69.2	**18.4**	43.7	65.5	14.2	30.1	10.2	20.1	51.3
- 不使用 LoRA	23.7	64.7	16.4	40.5	64.1	14.1	29.8	9.8	19.8	51.1
- 不使用投影解码	19.9	31.2	13.4	22.3	55.4	13.2	22.8	8.6	15.7	46.4

WSAC [89] 以 CLAP 嵌入为条件，采用前缀语言建模范式训练文本解码器，而 Zhang 等人 [90] 则设计了软提示和硬提示，以弥合 CLAP 的音频嵌入与文本嵌入之间的模态差距。如表 XV 所示，在域内场景中，我们的模型大幅超越了所有具有竞争力的零样本音频描述系统，且表现可与其他全监督方法媲美。在跨域场景中，该模型在所有指标上均取得最先进的结果，凸显了其稳健的域迁移能力。此外，我们发现，在这两种场景中，我们的模型在 FENSE [78] 得分方面均优于其他方法。我们推测，这一优势源于模型能够利用 CLAP 语义丰富的联合多模态空间，从而生成更精细的描述。

表 XVI 展示了对零样本音频描述生成模型的不同核心组件进行的全面消融研究，涵盖域内和跨域场景。研究探讨了检索增强生成（RAG）的贡献、LoRA 适配器的使用，以及投影解码（PD）的效果。结果表明，基于投影的解码有助于缩小模态差距，LoRA 可提升训练效率和质量，而检索增强生成也能提升模型性能，尤其是在跨域场景中。

B. 音乐描述（*MC*）

a) 数据集：我们使用 LP-MusicCaps-MC 数据集[91]进行训练，并在 LP-MusicCaps-MC 的标准测试集上进行测试。所有训练所用的音乐音频均被切分为 10 秒的片段。由于难以获取 LP-MusicCaps-MSD 数据集的原始音频，我们只能使用 LP-MusicCaps-MC 数据集训练模型，而该数据集是 LP-MusicCaps 数据集的一个小型子集。不过，我们在实验中证明，即使模型仅使用 LP-MusicCaps-MC 进行训练，也能取得与现有的基于 LP-MusicCaps-MSD 预训练的模型相当的结果 [91], [92] 。

b) 模型设置： 我们使用预训练模型作为音乐编码器，并使用 Vicuna-7b-v1.5 [10]作为大语言模型（LLM），训练期间二者均保持冻结。我们使用线性投影器，该投影器是唯一可训练的组件。我们考虑两种不同类型的音乐编码器。第一种是**帧级编码器**，它从音乐音频中提取具有时间维度的特征，通常通过自监督学习（SSL）进行训练，例如 MERT [6], MusicFM [93], 和 MuQ [94]。第二种是**序列级编码器**，它直接从音乐中提取特征，得到固定维度的特征（例如 512 或 1024 维），通常通过对比学习进行训练，例如 Laion-CLAP [81], Microsoft-CLAP [95],和 MuQ-MuLan [94]。对于帧级编码器，投影器会将帧级特征下采样至 0.5 Hz，作为输入送入大语言模型；例如，10 秒的音乐对应 5 个 token。对于序列级编码器，我们不进行下采样，而是在线性层投影后直接输入大语言模型，这意味着序列级编码后的特征只有 1 个 token。训练方案和模型检查点可在此处找到[11]。

c) 实验：在表 XVII 中，我们将不同音乐编码器在音乐描述任务上的效果与现有模型进行了比较。请注意，表中的 SLAM-LLM 模型均仅使用规模较小的 LP-MusicCaps-MC 数据进行训练。对于帧级编码器，MuQ 的表现优于 MusicFM 和 MERT，这与这些自监督学习模型的基本性能一致。对于序列级编码器，令人意外的是，尽管序列级特征只有 1 个 token，但其表现通常优于帧级特征。其中，MuQ-MuLan 在 BLEU、METEOR、ROUGE-S 和 BERT-S 四项指标上表现最佳。这意味着，即使只有一个嵌入，使用对比学习训练的模型也比使用自监督学习训练的编码器更适合音乐描述任务。我们还注意到，尽管 SLAM-LLM 仅使用了少量 LP-MusicCaps-MC 数据，但其结果与 MusCaps [96], LP-MusicCaps [91],和 MusiLingo[92],等使用额外预训练数据进行训练的模型相当。

C. 要点

根据上述实验，以下几点关键启示可为基于大语言模型的音频与音乐任务的未来研究与开发提供指导：

- **基于大语言模型的模型优于非大语言模型基线**。我们的探索进一步推动了这一趋势在自动音频描述任务中的发展，在 Clotho 和 AudioCaps 上均取得最先进结果。
- **微调编码器至关重要**。 像 EAT 这样的音频编码器，在分类任务上进行微调后，始终优于预训练或优化程度较低的同类编码器。

- **预训练和参数高效微调可提升质量与效率**。模型预训练结合 LoRA，能够以更低的训练成本生成更优质的描述。

[11]https://github.com/X-LANCE/SLAM-LLM/tree/main/examples/mc 音乐描述caps

表 XVII：音乐描述实验的结果，均在 LP-MusicCaps-MC测试集上评估。灰色高亮的行表示使用额外数据或私有数据进行预训练的模型。

模型		训练数据集	BLEU	METEOR	ROUGE-S	BERT-S
MusCaps		私有	10.2	17.0	22.2	83.5
LP-MusicCaps		MSD+MTT+音乐描述		22.4	21.5	87.8
MusiLingo		MSD+音乐描述		21.6	21.7	86.8
Ours	编码器Frame-wise					
	MERT	MC	22.2	11.6	16.8	86.3
	MusicFM	MC	25.1	18.6	19.2	86.2
	MuQ	MC	27.0	19.2	19.1	86.8
	编码器Sequence-wise					
	Laion-CLAP	MC	25.9	19.6	20.9	86.6
	Microsoft-CLAP 2023	MC	27.5	20.3	**21.1**	86.8
	MuQ-MuLan	MC	**27.9**	**20.5**	**21.1**	**86.9**

- **检索增强生成可提升模型性能**。通过获取外部知识来描述未见过的声景，检索增强生成能够提升模型的效果。

- **投影解码可以缩小模态差距**。 为了弥合 CLAP 中的模态差距，基于投影的解码效果显著；而在零样本设置中直接使用音频编码器，表现并不理想。

- **单个词元也足以胜任**。 在音乐描述任务中，信息相对稀疏，强编码器（MuQ）能够有效地将相关内容浓缩为单个词元，并将其传达给大语言模型解码器。

- **最佳投影器的选择因任务类型而异**。对于自动语音识别和视觉语音识别等需要严格单调对齐和时间序列保留的任务，线性投影器更为合适；而对于声音事件分类和自动音频描述等需要全局语义感知和跨模态上下文感知的任务，Q-Former桥接器则更有效。

SLAM-LLM系列各子任务的研究中提供了更详细且针对具体技术的信息 [80], [97]。

VI. 结论

SLAM-LLM为语音、语言、音频和音乐处理量身打造了模块化、灵活且开源的框架，推动了多模态大语言模型领域的重大进展。凭借编码器–投影器–大语言模型架构，SLAM-LLM能够针对自动语音识别、声音事件分类、自动音频描述及其他众多任务，高效地进行定制和部署。实验结果表明，SLAM-LLM中的配置方案不仅在多个基准测试中取得了具有竞争力或达到最先进水平的表现，也为基于大语言模型的音频处理社区提供了启发。通过降低入门门槛并促进社区协作，SLAM-LLM有望加速音频语言建模领域的研究与创新，并为多模态人工智能开辟新的可能性。

参考文献

[1] H. Liu、C. Li、Q. Wu 和 Y. J. Lee，“视觉指令微调”，载于 *NeurIPS* 论文集，2023年。

[2] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe 等人，“OpenFlamingo：用于训练大型自回归视觉语言模型的开源框架”，CoRR，第 abs/2308 01390 卷，2023年。.[3] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey 和 I. Sutskever，“通过大规模弱监督实现稳健的语音识别”，载于 ICML 论文集，2023年。[4] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov 和 A. Mohamed，“HuBERT：通过隐藏单元掩码预测进行自监督语音表征学习”，载于 IEEE/ACM 音频、语音与语言处理汇刊论文集，2021年。[5] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen 和 F. Wei，“BEATs：使用声学分词器进行音频预训练”，载于 ICML 论文集，2023年。[6] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Xiao 等人，“MERT：采用大规模自监督训练的声学音乐理解模型”，载于 ICLR 论文集。OpenReview.net，2024年。[7] J. Li, D. Li, S. Savarese 和 S. Hoi，“BLIP-2：利用冻结图像编码器和大语言模型引导语言—图像预训练”，载于 ICML 论文集，2023年。[8] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozi`ere, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave 和 G. Lample，“LLaMA：开放且高效的基础语言模型”，CoRR，第 abs/2302.13971 卷，2023年。[9] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov 等人，“LLaMA 2：开放式基础模型与微调聊天模型”，CoRR，第 abs/2307.09288 卷，2023年。[10] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu 等人，“Vicuna：一款以 90%* ChatGPT 质量惊艳 GPT-4 的开源聊天机器人”，https://vicuna.lmsys.org，2023年。[11] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang 等人，“通义千问技术报告”，arXiv 预印本 arXiv:2309.16609，2023年。[12] meta llama，“llama-recipes”，https://github.com/meta-llama/llama-recipes，2024年。[13] R. Zhang, J. Han, C. Liu, P. Gao, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li 和 Y. Qiao，“Llama-adapter：通过零初始化注意力高效微调语言模型”，arXiv 预印本 arXiv:2303.16199，2023年。[14] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue 等人，“LLaMA-Adapter v2：参数高效的视觉指令模型”，载于 ICLR 论文集，2024年。[15] A. Radford, J. W. Ki m, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark 等人，“通过自然语言监督学习可迁移视觉模型”，载于 IC ML 论文集。PmLR，2021年，第 8748–8763 页。[16] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khali- dov, P. Fernandez 等人，“DINOv2：无需监督学习稳健的视觉特征”，TMLR，2024年。[17] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, Z. Feng 和 Y. Ma，“LlamaFactory：统一高效地微调 100+ 语言模型”，载于 ACL 论文集，2024年。[18] S. Diao, R. Pan, H. Dong, K. Shum, J. Zhang, W. Xiong 和 T. Zhang，“LMFlow：用于微调和推理大型基础模型的可扩展工具包”，载于 NAACL 论文集，2024年，第 116–127 页。[19] D. Zhu, J. Chen, X. Shen, X. Li 和 M. Elhoseiny，“MiniGPT-4：利用先进的大语言模型增强视觉语言理解”，载于 ICLR 论文集，2024年。[20] B. Zhou, Y. Hu, X. Weng, J. Jia, J. Luo, X. Liu, J. Wu 和 L. Huang，“TinyLLaVA：小规模大型多模态模型框架”，CoRR，第 abs/2402.14289 卷，2024年。[21] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds 等人，“Flamingo：用于小样本学习的视觉语言模型”，载于 NeurIPS 论文集，2022年。[22] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul 和 B. Bossan，“Peft：先进的参数高效微调方法”，https: //github.com/huggingface/peft，2022年。[23] Y. Zhao, A. Gu, R. Varma, L. Luo, C.-C. Huang, M. Xu, L. Wright 等人，“PyTorch FSDP：全分片数据并行扩展实践”，Proc. VLDB Endow.，第 16 卷，第 12 期，第 3848–3860 页，2023年。[24] J. Rasley, S. Rajbhandari, O. Ruwase 和 Y. He，“Deepspeed：系统优化助力训练参数量超过 1000 亿的深度学习模型”，载于 SIGKDD 论文集，2020年，第 3505–3506 页。[25] L. Dou, Q. Liu, G. Zeng, J. Guo, J. Zhou, X. Mao, Z. Jin, W. Lu 和 M. Lin，“Sailor：面向东南亚的开放语言模型”，载于 EMNLP 论文集，2024年，第 424–435 页。[26] Y. Yang, Z. Song, J. Zhuo, M. Cui, J. Li, B. Yang, Y. Du 等人，“Gigaspeech 2：不断发展的大规模多领域自动语音识别语料库

面向低资源语言，采用自动爬取、转录和优化技术”，载于 ACL 论文集，维也纳，2025年。[27] Z. Zhang, L. Zhou, C. Wang, S. Chen, Y. Wu, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li 等人，“用自己的声音说外语：跨语言神经编解码器语言建模”，arXiv 预印本 arXiv:2303.03926，2023年。[28] W. Chen, Z. Ma, R. Yan, Y. Liang, X. Li, R. Xu, Z. Niu, Y. Zhu, Y. Yang, Z. Liu 等人，“SLAM-Omni：采用单阶段训练的音色可控语音交互系统”，载于计算语言学协会会议论文集：ACL 2025，2025年，第 2262–2282 页。[29] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao 等人，“WavLM：面向全栈语音处理的大规模自监督预训练”，载于 JSTSP 论文集，2022年。[30] V. Panayotov, G. Chen, D. Povey 和 S. Khudanpur，“Librispeech：基于公有领域有声读物的自动语音识别语料库”，载于 ICASSP 论文集，2015年。[31] J. Kahn, M. Rivi`ere, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazar´e, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen 等人，“Libri-light：有限监督或无监督自动语音识别基准”，载于 ICASSP 论文集，2020年。[32] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino 和 E. Dupoux，“Voxpopuli：用于表征学习、半监督学习和解释的大规模多语言语音语料库”，载于 ACL 论文集，2021年。[33] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang 等人，“Gigaspeech：不断发展的多领域自动语音识别语料库，包含 10,000 小时转录音频”，载于 Interspeech 论文集，2021年。[34] P. Zhang, G. Zeng, T. Wang 和 W. Lu，“TinyLLaMA：开源小型语言模型”，arXiv 预印本 arXiv:2401.02385，2024年。[35] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang 和 Y. Wu，“Contextnet：通过全局上下文改进用于自动语音识别的卷积神经网络”，载于 Interspeech 论文集，2020年，第 3610–3614 页。[36] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu 等人，“Conformer：用于语音识别的卷积增强型 Transformer”，载于 Interspeech 论文集，2020年。[37] Y. Peng, S. Dalmia, I. Lane 和 S. Watanabe，“Branchformer：并行多层感知机注意力架构，用于捕捉语音识别与理解中的局部和全局上下文”，载于 ICML 论文集，2022年。[38] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin 和 D. Povey，“Zipformer：更快、更出色的自动语音识别编码器”，载于 ICLR 论文集，2024年。[39] A. Baevski, Y. Zhou, A. Mohamed 和 M. Auli，“wav2vec 2.0：用于自监督语音表征学习的框架”，载于 NeurIPS 论文集，2020年。[40] Y. Peng, J. Tian, W. Chen, S. Arora, B. Yan, Y. Sudo, M. Shakeel, K. Choi, J. Shi, X. Chang, J. weon Jung 和 S. Watanabe，“OWSM v3.1：基于 e-branchformer 的更优、更快速开放式 Whisper 风格语音模型”，载于 Interspeech 论文集，2024年，第 352–356 页。[41] X. Shi, Q. Feng 和 L. Xie，“ASRU 2019 中英代码转换语音识别挑战赛：开放数据集、赛道、方法与结果”，arXiv 预印本 arXiv:2007.05916，2020年。[42] H. Wang, F. Yu, X. Shi 和 S. Zhang，“SlideSpeech：大型幻灯片增强型视听语料库”，载于 ICASPP 论文集，2024年。[43] F. Yu, H. Wang, X. Shi 和 S. Zhang，“LCB-NET：面向视听语音识别的长上下文偏置”，载于 ICASPP 论文集，2024年。[44] D. Le, M. Jain, G. Keren, S. Kim, Y. Shi, J. Mahadeokar, J. Chan 等人，“结合基于字典树的深度偏置与浅层融合的上下文相关流式端到端语音识别”，载于 Interspeech 论文集，2021年，第 1772–1776 页。[45] K. Huang, A. Zhang, Z. Yang, P. Guo, B. Mu, T. Xu 和 L. Xie，“利用上下文短语预测网络实现上下文相关端到端语音识别”，载于 Interspeech 论文集，2023年，第 4933–4937 页。[46] Y. Sudo, M. Shakeel, Y. Fukumoto, Y. Peng 和 S. Watanabe，“利用基于注意力的偏置短语增强束搜索实现上下文相关自动语音识别”，载于 ICASSP 论文集，2024年，第 10 896–10 900 页。[47] G. Sun, C. Zhang 和 P. Woodland，“结合图神经网络编码的树约束指针生成器，用于上下文语音识别”，载于 Interspeech 论文集，2022年，第 2043–2047 页。[48] J. Tang, K. Kim, S. Shon, F. Wu 和 P. Sridhar，“通过引导式注意力改进自动语音识别上下文偏置”，载于 ICASSP 论文集，2024年。[49] H. Futami, E. Tsunoo, Y. Kashiwagi, H. Ogawa, S. Arora 和 S. Watanabe，“ 面向基于前缀树的上下文自动语音识别的音素感知编码 ”，载于 ICASSP 论文集，2024 年。[50] G. Yang, Z. Ma, Z. Gao, S. Zhang 和 X. Chen，“基于 CTC 辅助大语言模型的上下文自动语音识别

[51] T. Afouras、J. S. Chung 和 A. Zisserman，《LRS3-TED：用于视觉语音识别的大规模数据集》，arXiv 预印本，2018年。[52] B. Shi、W.-N. Hsu、K. Lakhotia 和 A. Mohamed，《通过掩码多模态聚类预测学习音视频语音表征》，载于 ICLR 会议论文集，2022年。[53] J. S. Chung、A. Nagrani 和 A. Zisserman，《VoxCeleb2：深度说话人识别》，载于 Interspeech 会议论文集，2018年，第1086–1090页。[54] C. Wang、A. Wu、J. Gu 和 J. Pino，《CoVoST 2 与大规模多语言语音翻译》，载于 Interspeech 会议论文集，2021年，第2247–2251页。[55] M. A. Di Gangi、R. Cattoni、L. Bentivogli、M. Negri 和 M. Turchi，《MuST-C：多语言语音翻译语料库》，载于 NAACL 会议论文集，2019年，第2012–2017页。[56] M. R. Costa-juss`a、J. Cross、O. C.elebi、M. Elbayad、K. Heafield、K. Heffernan、E. Kalbassi、J. Lam、D. Licht、J. Maillard 等，《不让任何语言掉队：扩大以人为本的机器翻译规模》，arXiv 预印本 arXiv:2207.04672，2022年。[57] C. Wang、M. Liao、Z. Huang 和 J. Zhang，《BLSP-KD：通过知识蒸馏引导语言-语音预训练》，arXiv 预印本 arXiv:2405.19041，2024年。[58] C. Tang、W. Yu、G. Sun、X. Chen、T. Tan、W. Li、L. Lu、Z. Ma 和 C. Zhang，《SALMONN：迈向具备通用听觉能力的大语言模型》，载于 ICLR 会议论文集，2024年。[59] L. Barrault、Y.-A. Chung、M. C. Meglioli、D. Dale、N. Dong、M. Duppenthaler、P.-A. Duquenne、B. Ellis、H. Elsahar、J. Haaheim 等，《Seamless：多语言、富有表现力的流式语音翻译》，arXiv 预印本 arXiv:2312.05187，2023年。[60] Y. Chu、J. Xu、Q. Yang、H. Wei、X. Wei、Z. Guo、Y. Leng、Y. Lv、J. He、J. Lin 等，《Qwen2-Audio 技术报告》，arXiv 预印本 arXiv:2407.10759，2024年。[61] Y. Xu、H. Chen、J. Yu、Q. Huang、Z. Wu、S.-X. Zhang、G. Li、Y. Luo 和 R. Gu，《SECap：基于大语言模型的语音情感描述》，人工智能促进协会年会论文集，2024年。[62] H. Yan、Y. Zhu、K. Zheng、B. Liu、H. Cao、D. Jiang 和 L. Xu，《与类人智能体交谈：通过可感知的声学感知与反应实现共情对话》，载于 ACL 会议论文集，2024年，第15 009–15 022页。[63] Y. Sun、W. Chu、H. Zhou、K. Wang 和 H. Koike，《AVI-Talking：学习音视频指令以生成富有表现力的三维说话人脸》，IEEE Access，2024年。[64] Z. Ma、Z. Zheng、J. Ye、J. Li、Z. Gao、S. Zhang 和 X. Chen，《emotion2vec：语音情感表征的自监督预训练》，ACL 论文集，2024年。[65] Y. Cui、W. Che、T. Liu、B. Qin 和 Z. Yang，《采用全词掩码进行预训练的中文 BERT》，IEEE/ACM 音频、语音与语言处理汇刊论文集，2021年。[66] Z. Ma、G. Yang、Y. Yang、Z. Gao、J. Wang、Z. Du、F. Yu、Q. Chen、S. Zheng、S. Zhang 等，《一种极其简单且具备强大自动语音识别能力的大语言模型方法》，arXiv 预印本 arXiv:2402.08846，2024年。[67] G. Yang、Z. Ma、F. Yu、Z. Gao、S. Zhang 和 X. Chen，《MaLa-ASR：多媒体辅助的大语言模型自动语音识别》，Interspeech 论文集，2024年。[68] K. Drossos、S. Lipping 和 T. Virtanen，《Clotho：音频描述数据集》，载于 ICASSP 会议论文集。IEEE，2020年，第736–740页。[69] C. D. Kim、B. Kim、H. Lee 和 G. Kim，《AudioCaps：为真实场景中的音频生成描述》，载于 NAACL 会议论文集，2019年，第119–132页。[70] X. Mei、C. Meng、H. Liu、Q. Kong、T. Ko、C. Zhao、M. D. Plumbley、Y. Zou 和 W. Wang，《WavCaps：一个由 ChatGPT 辅助构建、用于音频-语言多模态研究的弱标注音频描述数据集》，arXiv 预印本 arXiv:2303.17395，2023年。[71] I. Mart´ín-Morat´o 和 A. Mesaros，《什么是真实标注？音频标记多标注者数据的可靠性》，载于欧洲信号处理会议论文集，2021年。[72] J. F. Gemmeke、D. P. Ellis、D. Freedman、A. Jansen、W. Lawrence、R. C. Moore、M. Plakal 和 M. Ritter，《AudioSet：音频事件本体与人工标注数据集》，载于 ICASSP 会议论文集。IEEE，2017年，第776–780页。[73] R. Sennrich、B. Haddow 和 A. Birch，《利用单语数据改进神经机器翻译模型》，载于 ACL 会议论文集，2016年，第86–96页。[74] S. Banerjee 和 A. Lavie，《METEOR：一种与人工判断相关性更高的机器翻译评估自动指标》，载于 ACL 会议论文集，2005年。[75] R. Vedantam、C. Lawrence Zitnick 和 D. Parikh，《CIDEr：基于共识的图像描述评估》，载于 IEEE/CVF 计算机视觉与模式识别会议论文集，2015年。[76] P. Anderson、B. Fernando、M. Johnson 等，《SPICE：语义命题图像描述评估》，载于欧洲计算机视觉会议论文集，2016年。[77] S. Liu、Z. Zhu、N. Ye、S. Guadarrama 和 K. Murphy，《通过对 SPIDEr 进行策略梯度优化改进图像描述生成》，载于 IEEE/CVF 国际计算机视觉会议论文集，2017年。

[78] Z. Zhou、Z. Zhang、X. Xu、Z. Xie、M. Wu 和 K. Q. Zhu，《能否使用图像描述指标评估音频描述？》，载于 ICASSP 会议论文集，2022年。[79] W. Chen、Y. Liang、Z. Ma、Z. Zheng 和 X. Chen，《EAT：采用高效音频 Transformer 的自监督预训练》，载于 IJCAI 会议论文集，2024年。[80] W. Chen、Z. Ma、X. Li、X. Xu、Y. Liang、Z. Zheng、K. Yu 和 X. Chen，《SLAM-AAC：通过释义增强和基于大语言模型的 CLAP-Refine 提升音频描述生成》，载于 ICASSP 会议论文集，2025年，第1–5页。[81] Y. Wu、K. Chen、T. Zhang、Y. Hui、T. Berg-Kirkpatrick 和 S. Dubnov，《通过特征融合和关键词到描述增强实现大规模对比式语言-音频预训练》，载于 ICASSP 会议论文集，2023年。[82] E. J. Hu、Y. Shen、P. Wallis、Z. Allen-Zhu、Y. Li、S. Wang、L. Wang 和 W. Chen，《LoRA：大语言模型的低秩适配》，载于 ICLR 会议论文集，2022年。[83] W. Li、L. Zhu、L. Wen 和 Y. Yang，《DeCap：通过纯文本训练解码 CLIP 潜变量，实现零样本描述生成》，载于 ICLR 会议论文集，2023年。[84] J. Kim、J. Jung、J. Lee 和 S. H. Woo，《EnCLAP：结合神经音频编解码器与音频-文本联合嵌入实现自动音频描述生成》，载于 ICASSP 会议论文集，2024年。[85] S.-L. Wu、X. Chang、G. Wichern、J.-w. Jung、F. Germain、J. Le Roux 和 S. Watanabe，《基于 BEATs、采用 INSTRUCTOR 嵌入监督和 ChatGPT 混合增强的音频描述模型》，技术报告，DCASE 挑战赛，2023年。[86] C. Tang、W. Yu、G. Sun、X. Chen、T. Tan、W. Li、L. Lu、Z. Ma 和 C. Zhang，《扩展大语言模型以用于语音与音频描述生成》，载于 ICASSP 会议论文集，2024年。[87] J. Liu、G. Li、J. Zhang、H. Dinkel、Y. Wang、Z. Yan、Y. Wang 和 B. Wang，《通过优化音频编码，利用大语言模型增强自动音频描述生成》，载于 Interspeech 会议论文集，2024年，第1135–1139页。[88] S. Deshmukh、R. Singh 和 B. Raj，《对比式音频-语言模型的领域自适应》，载于 Interspeech 会议论文集，2024年，第1680–1684页。[89] T. Kouzelis 和 V. Katsouros，《通过纯文本训练实现弱监督自动音频描述生成》，arXiv 预印本 arXiv:2309 12242，2023年。[90] Y. Zhang、X. Xu、R. Du、H. Liu、Y. Dong、Z.-H. Tan、W. Wang 和 Z. Ma，《使用软提示和硬提示实现零样本音频描述生成》，IEEE TASLP，第33卷，第2045–2058页，2025年。[91] S. Doh、K. Choi、J. Lee 和 J. Nam，《LP-MusicCaps：基于大语言模型的伪音乐描述生成》，arXiv 预印本 arXiv:2307.16372，2023年。[92] Z. Deng、Y. Ma、Y. Liu、R. Guo、G. Zhang、W. Chen、W. Huang 和 E. Benetos，《MusiLingo：通过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 NAACL 研究成果，2024年，第3643–3655页。[93] M. Won、Y.-N. Hung 和 D. Le，《音乐信息学基础模型》，载于 ICASSP 会议论文集。IEEE，2024年，第1226–1230页。[94] H. Zhu、Y. Zhou、H. Chen、J. Yu、Z. Ma、R. Gu、Y. Luo、W. Tan 和 X. Chen，《MuQ：采用梅尔残差向量量化进行自监督音乐表征学习》，CoRR，第abs/2501.01108卷，2025年。[95] B. Elizalde、S. Deshmukh、M. Al Ismail 和 H. Wang，《通过自然语言监督学习音频概念的 CLAP》，载于 ICASSP 会议论文集，2023年。[96] I. Manco、E. Benetos、E. Quinton 和 G. Fazekas，《MusCaps：为音乐音频生成描述》，载于 IJCNN 会议论文集。IEEE，2021年，第1–8页。[97] X. Li、W. Chen、Z. Ma、X. Xu、Y. Liang、Z. Zheng、Q. Kong 和 X. Chen，《DRCap：通过检索增强生成解码 CLAP 潜变量，实现零样本音频描述生成》，载于 ICASSP 会议论文集，2025年，第1–5页。