Day4

一般来说，训练数据集的构造和benchmark的设计与测试方面，相比于模型架构和训练范式来说，并不是主角。但是在多模态领域中，两篇文献中都着重强调了数据集的重要性。我想了想可能是和多模态属性，和模型的encode, align, interact and generate的复杂architecture有关。因此今天我将call back之前两篇综述中相关的总结，然后调研一下不同工作中是如何使用数据集、或者是构造数据集的，最后调研一些benchmark，着重看是选择了怎样的数据作为评测样本，以及使用了哪些指标来evaluate MLLM的。

MLLM综述

> A Survey on Multimodal Large Language Models

Pretrain

预训练本质是进行对齐模态。那么进行模态对齐训练的数据集一般分为两类：一类是粗粒度大规模数据，一类是

细粒度高质量数据。前者一般是公开网站爬取或者是开源的大数据集，但是有很多的噪声或者是要对MLLM的输入进行适配，因此需要进行很多预处理；而后者一般是通过一些template+LLM Captioner (如GPT-4V)构造出来的，对于对齐训练来说更有价值，且更适合用于grounding和缓解幻觉。

微调

在MLLM能够理解和对齐模态之后，需要能够让模型学会相应用户指令。于是一般这一块的指令微调训练数据为instruction、multimodal input 和 response 三元组。其中输入输出模态组合可以是自由的。一般改造认为有三条路线：

- Data Adaptation：改造已有的数据集，其中主要缺失的可能是template，那么可以使用instruction template pool，后者是人工写，或者是GPT生成，等等
- Self-Instruction：用 LLM/MLLM 生成更贴近真实对话的数据。LLaVA等工作将不同模态转化为元数据，然后使用GPT-4来生成多模态指令数据
- Data Mixture：MLLM需要能够保留原来的LLM的能力，因此一些工作会混合 language-only user-assistant conversation data 和 multimodal instruction data，主要是防止MLLM对原本LLM聊天对话能力的灾难性遗忘

对齐调优

> 使用RL等手段，引入人类偏好和幻觉纠正

这里的数据不再是普通问答，而是偏好数据：给定多模态 prompt 和两个回答，标注哪个更好。MLLM 场景中，这类数据尤其服务于减少幻觉、提高 helpfulness、faithfulness 和安全性。两篇综述提到 LLaVA-RLHF 使用人类偏好数据减少幻觉，RLHF-V 用 segment-level hallucination correction 构造细粒度偏好对。

数据集对 MLLM 训练的重要性和影响

- 如果模态之间发现对齐不是很好，那么很有可能是输入的模态训练对质量差
- 如果模型的模态细粒度能力有点低，则需要长且干净的数据更适合高分辨率训练，并有助于缓解幻觉；短且噪声大的数据适合低分辨率快速训练。有时高质量 caption 数据还可以支持解冻视觉编码器，从而得到更好的对齐。
- 如果发现MLLM不像一个聊天对象或者是助手，那么需要用指令微调
- 如果发现MLLM泛化能力不高，那么需要提升prompt diversity和task coverage，前者是对用户query更鲁棒，后者是让模态的理解和生成变的更鲁棒。比如视觉推理类数据往往比单纯 caption/QA 更能提升整体能力。
- 如果MLLM有幻觉和或者我们希望改善偏好对齐，那么需要使用偏好数据和纠正幻觉的pair data，来RL微调参数。
- 如果MLLM对于某一个模态的理解和生成有点弱，甚至可能有幻觉（推理无视高模态输入）（modality bias），有一种可能的原因是训练数据中该模态的占比很少，或者非常不平衡。

Omni综述

> From Specific-MLLMs to Omni-MLLMs: A Survey on MLLMs Aligned with Multi-modalities

数据集构造方法论

Omni不只要解决“图像/音频/视频 → 文本回答”，还要解决“多模态输入之间如何交互”以及“文本之外的模态如何生成”，也就是可以处理任意模态组合。

Alignment Data

X-Text paired data用于alignment pretraining，并且主要采用的是Text-Centric策略，不同模态对应到text里面。有一些模态很少有X-Text数据，那么常见策略是合成数据。有些工作还使用interleaved data，也就是包含了不止一种非语言模态的数据。

Instruction Data

Omni-MLLM 不仅使用 Specific-MLLM 的单模态指令数据，还要构造跨模态指令数据。三类方法:

- Template-based Construction。把已有 cross-modal downstream datasets 和预定义模板结合起来，构造跨模态 instruction。例如原始任务可能是 audio-visual QA、video-audio reasoning，现在通过模板改造成 instruction-following 格式。
- GPT Generation。先从标注数据集或预训练模型中提取元信息，例如 caption、object category、segmentation/meta information，再用强 LLM 生成 cross-modal instruction。这和 LLaVA 的自指令路线类似，但 Omni 更强调多模态之间的组合与交互。
- T2X Generation。把已有文本指令或图文指令，通过 TTS、DALL-E-3、MusicGen 等 Text-to-X 工具转换成包含 speech/image/music/video 等输出或输入的跨模态指令。例如把 LLaVA 的 Image-Text2Text 指令转换成 Image-Speech-Text2Text。

训练顺序

Omni 的数据不仅要构造，还要根据不同模块的架构选择考虑训练顺序。multi-branch Omni-MLLM 可以分别对每个 modality-specific projector 做单独 alignment；uni-branch 模型使用统一 projector 时，不同模态之间可能发生 alignment interference，因此有工作采用 progressive alignment；离散编码模型如 AnyGPT 则会混合不同模态的 alignment data 同时训练。

在 instruction fine-tuning 阶段，Omni 不仅混合不同模态的 uni-modal instruction data，还使用 cross-modal instruction data；一些工作会采用 multi-step fine-tuning，按照特定顺序引入单模态和跨模态数据，以逐步增强 uni-modal 和 cross-modal 能力。

数据集对Omni的重要性

- 数据决定 Omni 是否真的具备 Any-to-Any 能力。Omni 需要 cross-modal instruction data 和 cross-modal generation data 才能把"架构上的可能性"变成"行为上的能力"。
- 数据决定生成端是否能对齐。Omni 有 text-based、modality-token-based、representation-based 三类生成方式；其中 token-based 需要学习生成模态离散 token，representation-based 需要学习 signal token representation 到 decoder condition vector 的映射。因此输出对齐数据和相应 loss 设计会直接影响非文本模态生成质量。
- 数据混合会影响灾难性遗忘。当扩展新模态时，共享参数可能被新模态数据拉走，导致旧模态能力下降。综述提到，混合已经训练过的旧模态数据，或者只微调模态特定参数，可以部分缓解 catastrophic forgetting。
- 时序模态数据决定 temporal alignment 能力。对于视频、语音、音频这类序列模态，仅有全局 caption 是不够的；模型还需要保留音频-视频之间的时间对应关系。综述提到 interleaved modality-specific tokens 和 time-related special tokens 等做法，就是为了保留 temporal alignment 信息
- 低资源模态合成数据虽然有用，但可能引入偏差。合成方法可以缓解低资源模态缺少 text-paired data 和 instruction data 的问题，但如果缺少真实模态数据，模型对该模态的理解可能产生 bias。

一些经典Omni模型中的数据集选择和使用方式

NExT-GPT

NExT-GPT 的核心策略是冻结已有强模型：多模态 encoder、LLM、大部分 diffusion decoder 都尽量不动，只训练 input/output projection layers，并在最后 instruction tuning 阶段用 LoRA 微调一小部分 LLM 参数。论文claim只需要更新约 1% 参数，以避免从零训练的巨大成本。

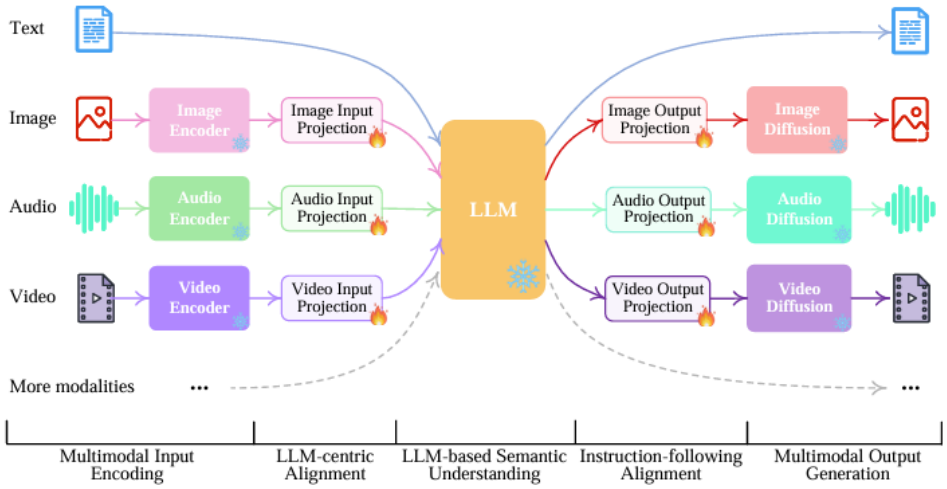


Figure 1. By connecting LLM with multimodal adaptors and diffusion decoders, NExT-GPT achieves universal multimodal understanding and any-to-any modality input and output. ❄and 🔥represent the frozen and trainable modules, respectively.

Encoding-side LLM-centric Alignment使用的是经典的X-caption pair data，采用 X-to-text generation task，训练时只更新 input projection layer，其他模块冻结；论文把这一步理解为“为冻结的 LLM 训练一个兼容的多模态 tokenizer”。

Decoding-side Instruction-following Alignment是为了让 LLM 输出的 modality signal tokens 能被后面的 image/video/audio diffusion decoder 理解。仍然使用 caption pair 数据，把 caption作为输入，并设计对应的loss来更新output projection layers。

最后是End-to-end Instruction Tuning，让模型真正学会 any-to-any instruction following：什么时候理解图像、什么时候输出音频、什么时候多轮对话里切换模态。这一阶段的数据就非常复杂了：

数据类型	数据集	作用
Text → Text	Cleaned-Alpaca	保持普通语言指令遵循能力
Text+Image → Text	LLaVA-150K	视觉理解、图文问答、图像对话
Text+Video → Text	VideoChat	视频理解、多轮视频对话
Text → Text+X	T2M dataset	让模型学会根据文本请求生成 image/video/audio
Modality-switching	MosIT	训练复杂多模态输入输出切换

比如这里的T2M 数据的构造方式：从已有 X-caption pair 中取 caption，例如 CC3M 图像 caption、WebVid 视频 caption、AudioCaps 音频 caption；用一些模板，把 caption 变成自然语言用户请求；使用 GPT-4 生成更丰富、多样化的 text instructions；最终得到 text-to-multimodal instruction data。普通 MLLM instruction data 多数只训练模型“看图/看视频后回答文字”，但 NExT-GPT 要学会“根据用户指令生成非文本内容”。T2M 就是补上这一类缺口：让模型知道当用户说“给我看一个……”“播放一个……”“生成一个……”时，应该输出 text + modality signal token，而不是只用文字回答。

另一个非常关键的构造的数据集是MosIT，全称为 Modality-switching Instruction Tuning dataset。论文中说，MosIT 的构造过程包括：

1. 设计一些 Human-Machine 对话模板。
2. 用 GPT-4 基于这些模板生成更多对话，覆盖 100+ topics / keywords。
3. 要求对话包含直接需求和隐含需求，例如 perception、reasoning、suggestion、planning。
4. 每个 conversation 包含 3–7 轮 QA。
5. 输入侧或输出侧都可以涉及多种模态，并且模态要交替切换。
6. 当对话需要图像、视频、音频内容时，从 YouTube、检索系统，或者 Stable-XL、Midjourney 等 AIGC 工具中找匹配内容。
7. 最后经过人工检查和过滤，得到 5K 高质量 dialogues。

AnyGPT

AnyGPT把 image / speech / music 都离散化成 token，扩展 LLaMA-2 词表，然后用普通 next-token prediction 训练。所以它的数据构造核心不是“projection alignment”，而是：把所有模态都变成离散 token 序列，再用文本作为桥梁做多模态对齐，最后用 AnyInstruct 做任意模态交互 SFT。

首先是Multimodal Alignment Pre-training：让 LLM 学会在 text、image token、speech token、music token 之间建立语义对应关系。由于任意模态两两对齐的数据非常稀缺，AnyGPT 采用了 text-centric bi-modal alignment dataset：让每种模态都先和 text 对齐，再通过 text 作为桥梁实现模态之间的间接对齐。这一阶段使用了很多的Image-Text and Speech-Text and Music-Text来。并且使用的时候，是双向instruction，包括：

- X-to-text：例如“请描述这张图片 / 请转写这段语音 / 请描述这段音乐”
- text-to-X：例如“请根据这段文本生成图像 / 语音 / 音乐”

所以本质是：把所有 X-text pair 都重写成 instruction-style next-token prediction 数据，同时学 “理解” 与 “生成”。

论文还提到，由于很多 image/music 数据来自 web，噪声较大，因此初始 pre-training 后，又选择更高质量的数据继续训练 4000 steps：

方向	**继续预训练数据**
text-to-image	JourneyDB、LAION-Aesthetics
image captioning	LAION-COCO
music	AnyInstruct-108k 中的 music 数据

之后是Instruction Tuning，使用的是作者构造的 AnyInstruct-108k。这是一个多轮、多模态、interleaved instruction dataset，总计 108k conversations。它包含 text、speech、image、music 的任意组合，目标是让模型学会真实对话中的模态切换，而不是只会单步 text-to-image 或 image-to-text。论文和官方 repo 都说明：base model 用来对齐四种模态，chat model 则是在 AnyInstruct 上训练，用于自由多模态对话。

AnyInstruct 是怎么构造的

第一阶段：先生成“纯文本版的多模态对话”。这一阶段还没有真正生成图片/音乐/语音，而是先让 GPT-4 写出包含多模态元素描述的对话：

1. 先设计 100 个 meta topics，覆盖大量日常 audiovisual scenarios；
2. 用 GPT-4 把这些 meta topics 扩展成 20,000 个具体 topics；
3. 根据 topic 生成 conversation scenarios；
4. 再让 GPT-4 生成多轮对话；
5. 在对话中，image / music 等非文本内容先以详细 textual description 的方式出现。

也就是说，第一步得到的是这样的逻辑：

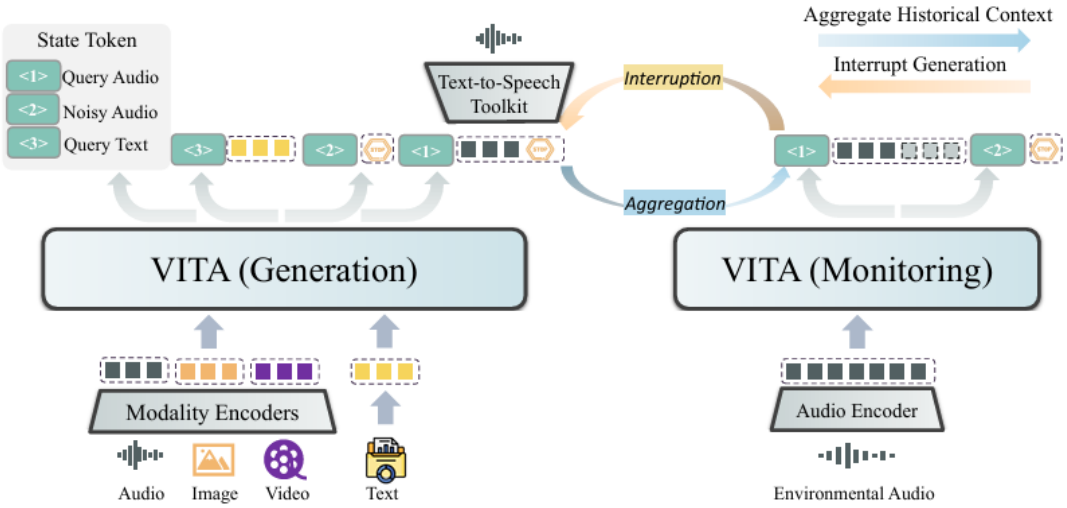
```
User: 我上传了一张“海边日落”的图片……
Assistant: 这张图看起来……
User: 能不能根据这个氛围生成一段舒缓的音乐？
Assistant: 好的，生成一段“slow, warm, ambient ocean sunset music”……
```

这里的图片和音乐还只是文字描述。第二阶段：Text-to-Multimodality Conversion。接下来，作者把这些文字描述转换成真实多模态内容：

目标模态	使用工具
Image	DALL-E-3
Music	MusicGen
Speech	Microsoft Azure TTS

论文说，经过过滤后，最终得到 108k 高质量多模态对话，包含约 205k images、503k voice recordings、113k music tracks。另外，作者还从已有 text-only instruction datasets 中抽取适合 spoken narration 的对话，用 TTS 扩展出 100k voice dialogues。

VITA



VITA 主要是把 vision + audio encoder 接到 Mixtral 8x7B 上，让模型能听、能看、能理解，并通过 TTS 形成语音回复。 VITA 的训练分成三大阶段：LLM Instruction Tuning → Multimodal Alignment → Multimodal Instruction Tuning。论文明确说整体 pipeline 包括这三步，最后阶段还专门加入 state token 来支持非唤醒交互和音频打断交互。

首先是LLM Instruction Tuning使用的是 纯文本 bilingual instruction data，扩充词表，然后让Mixtral模型的中英文能力。

之后是Multimodal Alignment，目标是把视觉、视频、音频 encoder 的输出对齐到 LLM 的文本空间，分成视觉对齐和音频对齐。Visual Alignment使用的数据类型：覆盖了很多的任务，致力于形成视觉理解较深的、对齐之后的模型。

数据类型	**代表数据**
图像描述 / caption	ShareGPT4V、Allava-Caption、ShareGPT4o-Image、中文 synthetic description
通用图像 QA	LLaVA-150K、LLaVA-Mixture-sample、Lvis-Instruct、ScienceQA、中文 synthetic QA
OCR / 图表理解	Anyword-3M、ICDAR2019-LSVT、ICDAR2017-RCTW、Open-Chart、synthetic OCR/diagram data
视频描述 / 视频 QA	ShareGemini、基于 Video-ChatGPT 和 VideoChat2 重新标注得到的视频 QA

Audio Alignment使用的数据是：

任务	**数据集**	**目的**
ASR	WenetSpeech、GigaSpeech	学会把语音内容映射到文本语义
Audio Captioning	WavCaps 的 AudioSet SL 子集	学会理解非语音声音事件

这里数据集的选择有巧思：因为VITA需要音频能力不是单纯“听懂人说话”，还要能处理交互场景中的环境音和非语音频。Audio captioning 数据集中有环境声音、事件声音等。

最后是Multimodal Instruction Tuning，模型真正学会：根据文本或语音指令，结合图像/视频内容完成问答，并且识别当前输入到底是不是一个需要回答的 query。数据源基本和 Stage 2 表 1 中的数据相同，但做了两个重要改造：

1. 把大约一半问题随机替换成语音版本，使用 GPT-SoVITS 等 TTS 工具生成 audio question；
2. 为 image / video / pure text 设置不同 system prompt，避免图像、视频、文本知识之间发生冲突。

因为 VITA 的交互目标是用户可以直接语音提问，而不是只能打字。 VITA 相比普通 MLLM 的特点还有：在真实人机交互中，麦克风会一直听到声音，但不是所有声音都需要模型回答。VITA 把这类输入称为 noisy audio。这一部分构造方法是：

1. 从已有 multimodal / unimodal QA 数据的答案中随机采样一些句子；
2. 这些句子被当作“不需要回复的非 query 内容”；

3. 让它们的长度分布和真实问题长度接近；
4. 用 TTS 把这些文本转成 audio；
5. 作为 negative audio samples 训练模型识别“这不是需要回答的问题”。

这样做因为 VITA 想实现 Non-awakening Interaction，即不需要用户每次说唤醒词，模型自己判断当前音频是不是有效问题。没有 noisy audio 负样本，模型就很可能听到任何语音都开始回答。

训练时，作者把对应 state token 插到答案开头，让模型学会区分不同输入类型。对于 noisy audio，论文说直接让模型输出 EOS 会损害训练，因此他们先把 noisy audio 对应文本送入 LLM，拿 LLM 输出作为训练 target；但在推理时，Noisy Audio token被当作特殊 EOS，直接终止输出。

Qwen2.5-Omni

Pre-training Stage 1：Encoder Alignment。使用的是：image-text pairs和audio-text pairs。论文说第一阶段会 freeze LLM parameters，只训练 vision encoder 和 audio encoder，使用大量 audio-text 和 image-text pairs，让 LLM 获得视觉-文本、音频-文本语义对齐能力。这和 NExT-GPT 的 encoding-side alignment 有点像，都是先解决“非文本模态如何进入 LLM”的问题；但 Qwen2.5-Omni 不是单纯训练 projector，而是围绕 Qwen2.5-VL / Whisper-large-v3 这类强 encoder 做适配和进一步训练。

Pre-training Stage 2：Full-parameter Multimodal Pre-training。会解冻all parameters，使用更大规模、更混合的多模态数据。使用的数据规模大致如下：

数据类型	**规模**
image + video related data	800B tokens
audio related data	300B tokens
video with audio related data	100B tokens
pure text data	用于保持和提升语言能力

整个预训练数据包括 image-text、video-text、video-audio、audio-text 和 text corpus；同时他们把层级标签替换成自然语言 prompt，以增强泛化能力和 instruction-following 能力。第二阶段引入更多混合多模态数据和更多任务，使模型更好地学习 audio、visual、textual information 之间的交互。

Pre-training Stage 3：Long-sequence Training。使用更长上下文的数据，把训练序列长度扩展到 32,768 tokens。论文说，前面阶段为了训练效率把最大 token 长度限制在 8192，随后加入 long audio 和 long video 数据，并把 text、audio、image、video 数据都扩展到 32,768 token 长度进行训练。

Post-training：Thinker 的 instruction tuning。使用 ChatML 格式的 instruction-following data，包括四类，是为了把预训练得到的多模态能力转成 assistant 行为：

数据类型	作用
pure text-based dialogue data	保持普通文本对话和指令遵循
visual-modality conversation data	图像/视频问答、视觉对话
audio-modality conversation data	语音/音频理解与对话
mixed-modality conversation data	多模态混合输入下的对话

Post-training：Talker 的三阶段训练。Qwen2.5-Omni不是简单用外部 TTS 把文本答案读出来，而是训练 Talker 直接根据 Thinker 的 hidden representations 和文本 token 生成 speech tokens。因此 Talker 有单独的三阶段训练。Talker 第一阶段使用包含 multimodal contexts and spoken responses 的大规模对话数据，做 speech continuation task，通过 next-token prediction 学习从上下文到语音的续写。它不仅使用和 Thinker 类似的文本监督，还额外进行 speech continuation。第二阶段使用 DPO。论文说，由于预训练数据覆盖大量说话人和场景，不可避免包含 label noise 和 pronunciation errors，可能导致 speech hallucination。为了解决这个问题，他们构造三元组数据：

```
(x, y_w, y_l)
```

符号	含义
x	输入文本序列
y_w	good generated speech sequence
y_l	bad generated speech sequence

这些样本根据 reward score 排序，reward 与 WER 和 punctuation pause error rate 有关。让 Talker 偏好低 WER、停顿更合理的语音序列，从而减少语音幻觉和发音错误。最后是第三阶段 speaker fine-tuning / multi-speaker instruction fine-tuning，让 Talker 能采用 specific voices，并提升语音自然度。

一些Benchmark

MME

MME是早期经典 MLLM 综合评测，全称是 MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models，它的目标不是测试某一个单独能力，而是系统评估 MLLM 的两大能力：感知能力和认知 / 推理能力，共 14 个子任务，并且 instruction-answer pairs 都是人工设计，以避免直接使用公开测试集标注带来的数据泄漏。

MME 的数据构造哲学有四点：

- 覆盖基础感知和高级认知。perception 包括 object existence、count、position、color 等基础识别，也包括 poster、celebrity、scene、landmark、artwork、OCR 等更细粒度识别；cognition 则包括 commonsense reasoning、numerical calculation、text translation、code reasoning。

- 尽量避免数据泄漏。MME 不直接使用公开数据集原有标注，而是手工设计 instruction-answer pairs。
- 使用极简指令，避免 prompt engineering 干扰。MME 的问题格式非常简单，通常是一个 yes/no 问题，后面加上 “Please answer yes or no.”。这样做的原因是：如果 prompt 太复杂，评测结果可能反映的是模型对 prompt 的适应能力，而不是图像理解能力。MME 希望所有模型在同一简单指令下被公平比较。
- 评测结果容易自动量化。MME 将输出限制为 yes/no，避免 GPT judge 或人工打分带来的主观性。

评测方式：Accuracy和Accuracy+。MME 的每张图像对应两个问题，一个答案是 yes，一个答案是 no。这样设计是为了防止模型总是回答 yes 或总是回答 no。如果模型能把同一张图的两个问题都答对，更能说明它真的理解了图像和问题，而不是随机猜测。MME 使用两个指标：Accuracy：按每个问题计算准确率；Accuracy+：按每张图像计算，只有该图对应的两个问题都答对才算正确。

Video-MME

MME 向视频方向的扩展，但不只是把图片换成视频。它的核心目标是评估 MLLM 是否真的能处理 sequential visual data，也就是连续视频中的时间、事件、动作、上下文和长程依赖。论文指出，当时 MLLM 主要集中在静态图像理解，而视频理解能力还缺少全面、高质量评测，因此提出 Video-MME。

Video-MME 的数据构造哲学可以概括为：Diversity：视频类型多样；Duration：覆盖短中长视频；Modality：视频帧之外还考虑字幕和音频；Quality：人工选择、人工标注、人工质检。

Video-MME 覆盖 6 个大类，这些大类下有 30 个细分类别。这样设计的目的是避免 benchmark 只偏向某一种视频，比如短视频、电影片段或教学视频。且视频长度分为三档，从11到一小时的都有。Video-MME 每个视频对应 3 个问题，每个问题是 4 选 1 multiple-choice QA。标注者需要完整观看视频，并反复回看视频内容来设计问题和答案。问题覆盖 perception、reasoning、information synopsis 等类型。论文还做了“question-only”检查，即只给问题不给视频时，强模型也很难答对，从而减少纯文本 shortcut。

Video-MME 的基本输入形式是：`video frames + optional subtitles/audio + multiple-choice question`。评测指标是 accuracy，随机猜测准确率是 25%，并且不会有LLM as judge 的potential bias。

Video-MME-v2

Video-MME-v2 是 Video-MME 的后续升级版。它的提出背景是：随着视频理解模型快速进步，原有 benchmark 开始出现“leaderboard 分数变高，但真实能力仍不足”的饱和问题，因此 Video-MME-v2 更强调评估模型在视频理解中的 robustness 和 faithfulness。

它相对 Video-MME 的核心升级有两个。第一，它设计了一个 progressive tri-level hierarchy，把视频理解难度分成逐步递进的三层：从 multi-point visual information aggregation，到 temporal dynamics modeling，再到 complex multimodal reasoning；也就是说，它不仅问模型“看到了什么”，还进一步问模型能否把多个视觉证据点聚合起来、理解时间动态，并在多模态信息上完成复杂推理。第二，它不再只依赖传统的 per-question accuracy，而是提出 group-based non-linear evaluation：把相关问题作为一组来评估，要求模型在一组相关问题中保持一致性和多步推理连贯性，避免模型靠局部猜对拿到高分。

数据质量方面，Video-MME-v2 采用更严格的人工标注流程：涉及 12 位标注者、50 位独立 reviewer、3300 human-hours，并经过最多 5 轮 quality assurance。实验发现，即使当前最强模型与人类专家之间仍有明显差距，而且模型在低层视觉信息聚合和时间动态建模中的错误会向上传播，限制更高层复杂推理；论文还指出，thinking-based reasoning 对文本线索依赖较强，有字幕时可能提升表现，但在纯视觉场景中有时反而会下降。

它的数据构造哲学大概可以拆成下面几层。

- 把 数据泄漏 / 预训练污染 当成第一问题来处理。它不是随便从经典电影、热门影视片段或老视频里抽样，而是强调 recency-oriented collection，也就是优先采集新近发布的视频。这样做的目的，是降低这些视频已经出现在当前 MLLM 预训练语料中的概率，让模型表现更接近真实理解能力，而不是训练记忆。
- Video-MME-v2 不是只测电影、运动、教学或短视频，而是用一个两层 taxonomy 来组织视频来源。它有四个一级领域，这些一级领域下面又细分成 31 个细粒度类别。
- 重点转向中长视频。视频平均时长约 10.4 分钟，99% 小于 20 分钟，53% 小于 10 分钟，让视频长度足以支持 multi-point aggregation → temporal dynamics → complex reasoning 这种分层任务设计
- Video-MME-v2 还用 view count 作为一种源头质量控制信号。论文中说，它们用播放量作为内容质量的 proxy：所选视频的平均播放量约 483 万，中位数约 35.5 万
- 从单题 QA 变成 group-based construction。传统benchmark都是简单的选择题或者是group选择题，但是Video-MME-v2 则是800 个视频，每个视频 4 个问题，每个问题 8 个选项。group-based design 有两种：Consistency group 主要测试模型在同一个能力维度上的表现是否稳定；Coherence group 更偏复杂推理。它不是只问最终结论，而是把一个复杂问题拆成若干中间步骤，让问题组模拟人类解决问题的逻辑过程。
- 选项构造方式有两点：控制选项长度，八个选项的平均词数被严格控制得比较一致，避免模型利用“正确答案通常更长/更短”这种统计 shortcut；每题至少有一个 adversarial distractor，这个干扰项会根据部分视觉或音频证据看起来很合理，但又与 ground truth 中一个关键细节矛盾。

评测指标为：如果是4个问题答对N个，则`(N / 4)^2`是得分。

OmniBench

> OmniBench: Towards The Future of Universal Omni-Language Models

它和 MME、Video-MME 的区别在于：OmniBench 的目标不是只测图像或视频，而是明确要求模型同时处理视觉、声学、文本三模态共同输入，论文将能处理这种三模态上下文的模型称为 omni-language models, OLMs。OmniBench 的目标是评估模型是否能够在 visual、acoustic、textual inputs 同时存在时进行 recognition、interpretation 和 reasoning。

OmniBench 最重要的数据构造原则是：每个正确答案必须依赖跨模态信息整合，而不是单看图像或单听音频就能完成。项目页明确说明，annotation scheme 的基本原则是：每个问题的正确答案必须需要 image 和 audio 两部分信息，从而确保 benchmark 真正评估模型跨模态分析能力，而不是单模态能力。质量控制包括人工检查和 MLLM 辅助自动检查。

OmniBench 包含 1142 个 question-answer pairs，任务输入是：`(image, audio, text question)`，输出是 text answer，通常设置为四选一。任务类型包括 8 类，这些类别从低层感知到高层推理都有覆盖。

OmniBench 的音频不是单一语音，而包括三类：speech：人声交流；sound events：非语音事件声，例如环境声、机械声、自然声；music：音乐片段。很多audio benchmark 主要测 ASR 或 speech understanding，但 OmniBench 更关注"声音作为场景证据"的作用。例如图像中看到一个场景，音频中听到掌声、哭声、警报声、音乐或脚步声，模型需要综合判断发生了什么。

OmniBench 作者还构造了 OmniInstruct，包含 84.5K instruction tuning samples，用于训练 OLM 适应三模态上下文。

MMMU

> MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI

MMMU为了测专家级、多学科、图文结合的理解与推理能力，核心思考是：模型能否像大学生 / 专业人士一样，结合图像、图表、公式、结构图、医学图、地图等视觉信息，完成多学科知识推理？MMMU 包含 11.5K 个精心收集的多模态问题，来自 college exams、quizzes 和 textbooks，覆盖 6 个核心学科、30 个 subjects、183 个 subfields，并包含 30 种高度异质的图像类型，如 charts、diagrams、maps、tables、music sheets、chemical structures 等。

MMMU 的数据构造哲学可以概括为：从"日常视觉问答"转向"专家知识 + 复杂图文推理"。强调的不是单纯视觉识别，而是 advanced perception + domain-specific knowledge + deliberate reasoning。论文也明确说，MMMU 不同于已有 benchmark，它聚焦高级感知和推理，需要领域知识，挑战模型执行类似专家面对的任务。任务形式依然是选择题。

思考和总结

从训练数据构造的角度看，MLLM 到 Omni 的核心变化，是从 X-to-Text 对齐 逐渐走向 Any-to-Any 行为塑造。早期 MLLM 更关注把图像、视频等模态对齐到语言空间，让模型能够"看图说话"；而 Omni 模型进一步要求模型能够在文本、图像、视频、语音、音乐等模态之间自由输入和输出。因此，NExT-GPT 的 MosIT、AnyGPT 的 AnyInstruct、VITA 的 noisy audio / state token 数据、Qwen2.5-Omni 的 Thinker-Talker 分阶段数据，都说明 Omni 能力是通过非常具体的数据设计而驯服出来的。

一个重要认识是：多模态数据集构造的本质，不只是收集更多数据，而是设计更合理的能力分布。粗粒度大规模数据适合建立基础对齐，高质量细粒度数据适合提升 grounding、减少幻觉；instruction data 决定模型是否像助手，preference / DPO / RLHF 数据决定模型是否更符合人类偏好；而跨模态、多轮、交错式数据则决定模型是否真的具有交互能力。如果某一模态数据不足、不平衡或质量较低，模型就容易产生 modality bias，表现为忽视该模态、过度依赖文本，甚至出现多模态幻觉。为了Omni最后有不同的性质或基本的能力，需要使用到对应设计地、有对应性质的数据。

Benchmark 的演化也反映了这一点。MME 关注基础感知与认知能力，Video-MME 开始强调长视频、时间动态和多模态线索，Video-MME-v2 进一步通过 group-based evaluation 测试一致性和推理链条，OmniBench 则要求答案必须依赖图像和音频共同信息，MMMU 则把评测推向专家知识和复杂图文推理。也就是说，benchmark 不再只是"测模型答对多少题"，而是在逐渐追问：模型是否真的利用了多模态证据？是否能跨时间、跨模态、跨知识领域进行稳定推理？

在 benchmarking 方面，我认为：评测集本身也在定义我们想要测评的多模态能力或方面是什么。一个好的 benchmark 不能只是堆更多题目，而要避免数据泄漏、文本 shortcut、单模态 shortcut 和 prompt engineering 带来的干扰。例如 Video-MME 会做 question-only 检查，OmniBench 要求答案必须同时依赖图像和音频，Video-MME-v2 通过 group-based evaluation 评估一致性和多步推理，这些设计都说明 benchmark 需要尽量确认模型是真的利用了多模态证据，而不是靠语言先验或训练记忆猜答案。同时，评测指标也不能只看单题 accuracy，还应关注模型在长上下文、时间动态、跨模态证据整合、幻觉控制和推理一致性上的表现。