OmniEVIE：面向多模态检索的证据向量引导嵌入

Zifei Wang[1], *, Wei Wen[2], *†, Qiang Ji[1], Qian-Wen Zhang[1,2], Ruizhi Qiao[2], He Liu[1], Xing Sun[2], Qiang Lin[2]

[1]腾讯IMA产品中心 [2]腾讯优图实验室

arXiv:2611.32203v1 [cs.IR] 24 Nov 2026

如今，多模态检索需要匹配密集页面、细粒度实体、有序视频，以及智能体查找的工具和记忆。单向量会丢失决定这些匹配结果的局部证据。交叉编码器可以找回这些证据，但代价是必须重新读取每个召回的页面或视频片段。晚期交互将证据保留在索引中；然而，存储每个词元向量的成本过高，无法用于在线服务。OmniEVIE 是一个 4.5B 多向量嵌入模型，通过一个压缩索引和三阶段检索器覆盖文本、图像、视觉文档、视频和工具。每条记录最多包含 128 个维度为 64 的 int8 向量：每条记录占 9,472 字节，每百万条记录占 8.82 GiB。检索在该压缩索引上运行，返回的命中结果会标明页面区域或视频区间。训练、合并、索引构建和检索经过协同设计。每项任务都被写成一个检索样本。三级证据检查会剔除假负例，采样率则根据基线与教师模型之间的差距确定。训练期间使用 ST-HAC 对候选项进行压缩，采用与索引中存储形式相同的表示，因此训练得分和服务得分使用相同的向量。分阶段运行结果和专家分支按参数块合并。显著性门控和噪声底会重新调整查询得分。一个压缩记录提供均值向量、1 位符号和 int8 多向量，分别用于粗阶段、中间阶段和最终阶段。在公开数据集上训练后，完整序列在 $d = 2048$ 上的得分为 60.68，在 MMEB-V3（190 个任务）上也是如此；$K128 \times d64$ int8 索引的得分为 60.06。同一压缩索引在 MMEB-V2（78 个任务）上的得分为 77.96，在 ViDoRe V3（48 个任务）上的得分为 62.96。对于包含 100 万条和 500 万条记录的语料库，单线程查询分别耗时 54.2 毫秒和 120.2 毫秒；与对 4096 维单向量进行精确搜索相比，分别快 11.1× 倍和 25.3× 倍；与全量 int8 MaxSim 相比，则分别快236× 倍和 532× 倍。视频直接由帧编码，命中结果可回溯到以秒为单位的起止时间，因此索引构建和定位都不需要视频解析模型。在 QVHighlights 验证集的 1,083 个查询中，78.0% 的热度峰值落在标注片段内，而返回时间跨度与其相交的比例为

94.3%，而相同长度的随机时间跨度为 44.9%；该跨度平均覆盖视频的 18%。权重和代码见 https://huggingface.co/tencent/OmniEVIE-4B 和 https://github.com/Tencent/OmniEVIE。

📅 **日期：** 2026年11月19日**通信邮箱：**
✉ jawnrwen@tencent.com**代码：**
https://github.com/Tencent/OmniEVIE**模型：**
https://huggingface.co/tencent/OmniEVIE-4B

1 引言

1.1 从单向量到多向量检索

如今，检索查询涵盖多页报告、细粒度产品和地标、包含有序动作的长视频，以及由工具描述、界面状态和多步历史构成的智能体上下文

*贡献相同。
†通讯作者。联系方式：jawnrwen@腾讯.com。

[1, 2, 3]。相关性通常取决于少数几个不相邻的证据片段：财务表格中的头部和一个单元格、视频中两个动作的先后顺序，或一份冗长工具文档中的一个参数。

单个d维向量通过一次内积为查询和候选项打分。Weller 等人 [4] 表明，该分数只能将部分组合排在首位：当语料库规模超过随d多项式增长的阈值后，一个必须通过多个片段同时识别的文档无法被任何查询排在首位。增加数据或扩大骨干网络都无法消除这一限制。

晚期交互保留查询令牌集$Q = \{q_1, \ldots\ldots, q_{L_q}\}$以及候选集$D = \{v_1, \ldots\ldots, v_K\}$，并通过候选项 [5, 6] 中的最佳匹配为每个查询词元打分：

$$\mathrm{MaxSim}(Q, D) = \sum_{i=1}^{L_q} w_i \max_{1 \le j \le K} \langle q_i, v_j \rangle, \tag{1}$$

其中$w_i > 0$。由于每个查询词元都可以落在不同的片段上，该分数不再受单个向量维度的限制。领先的视觉文档和文本到视频检索器都采用了这种形式 [6, 7, 8, 9]。

1.2 服务时的三个障碍

这种表征的服务成本很高，原因有三。高分辨率页面或几十秒的视频片段会产生数千个向量，因此百万条记录以浮点数存储时会占用数百 GB。查询随后必须与每个向量进行比较：在百万条记录的语料库上执行一次单线程查询，耗时约为十秒。文档、图像、视频和工具之间的故障模式也各不相同：加大某一模态的权重通常会降低其他模态的表现，而在完整词元序列上训练的模型，一旦这些词元为建立索引而被压缩，准确率就会下降。

1.3 OmniEVIE 的作用

通常的折中方案是使用一个向量进行检索，再通过交叉编码器进行重排序 [7, 10]。每个被召回的 k个页面或视频片段都会连同查询再次转发，因此系统必须在线读取原始媒体。未压缩的多向量索引则存在相反的问题：已有研究表明，MaxSim 只会选取约 1% 的存储向量 [11]。OmniEVIE 将宽度保持为 $d \le 128$，限制向量数量，并以低位宽存储这些向量。搜索在该压缩索引上运行（表 1）。

该模型将 EVIE 的证据检查和局部多向量索引从视觉文档扩展到通用多模态检索 [12]。挖掘出的候选项只有在缺少查询所需证据时，才会被视为负例。位置编码用于指导聚类，之后会被丢弃，因此索引存储的是语义质心。检索器用于过滤命中项、并为命中项命名的，都是同一条整数记录。此版本使用 57 个公开数据集进行训练。改动涉及骨干网络、添加模态的顺序，以及在写入索引的同一批聚类向量上进行训练这一决定。

命中项还会携带其证据所在的位置。页面或图像会返回命中簇的边界矩形；视频则会返回命中质心对应的起止时间（以秒为单位），这样下游生成器读取的就是相应区域或片段，而不是整页或整段视频。视频端直接对采样帧进行编码，并根据帧索引获取时间，因此索引构建和定位都不需要经过解析模型将视频转录为文本。定位结果见第 9.5 节。图1 展示了由此获得的准确率和延迟，表 1 对比了四种服务设计。

报告的其余部分将沿着该记录，从训练讲到搜索。第3 节介绍检索、分类、

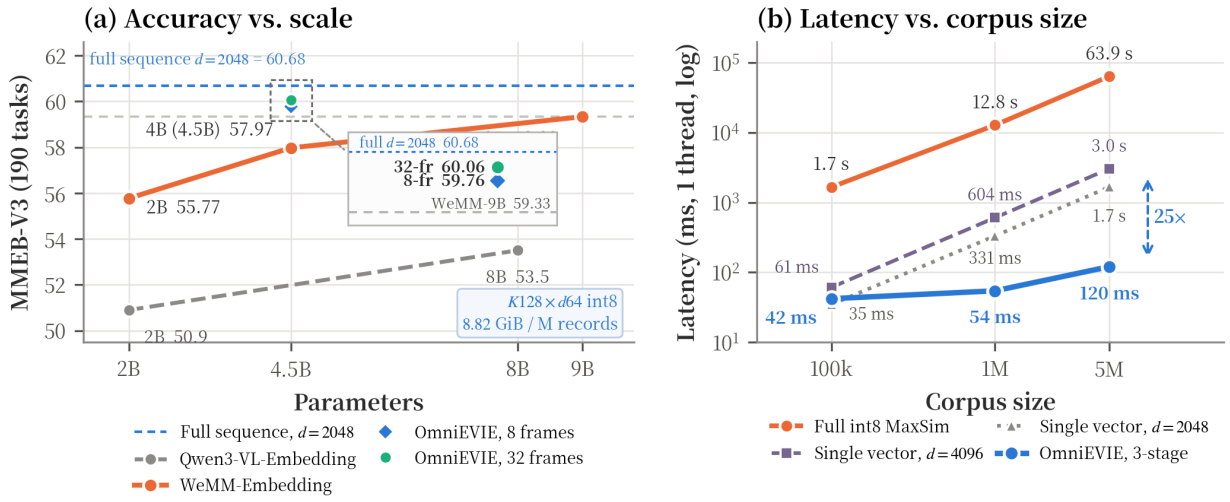

图 1。 OmniEVIE 的准确率和延迟。（a）MMEB-V3 在 190 个任务上的均值与参数规模的关系。蓝线表示 $d = 2048$处的完整序列：相同的权重保留每个词元向量，不使用 ST-HAC 和 Int8量化，分数为 60.68。圆形和菱形分别表示已发布的 $K128 \times d64$ int8索引，其每百万条记录占用 8.82 GiB；在 32 帧和 8 帧下，分数分别为 60.06 和 59.76。WeMM-Embedding-4B 和 OmniEVIE 共用一个 4.5B 骨干网络。（b）全量 int8 MaxSim、$d = 2048$和 $d = 4096$处的精确单向量搜索，以及三阶段检索器在不同语料库规模下的单线程延迟；其中，三阶段检索器的第一阶段使用 $d = 64$ 均值向量。

表 1。 四种多模态检索设计的服务对比。

	单向量	单向量 + 重排序器	完整多向量	OmniEVIE
分数	一次内积	粗粒度分数，然后进行交叉注意力	全词元 MaxSim	固定多向量，$K \leq 128$, $d = 64$
局部命中	None	无坐标	逐词命中	页面矩形框和视频秒；跨度命中 94.3%
百万级索引	几 GiB 的向量	重排序必须读取原始媒体-ing	数百 GiB	7.87–8.82 GiB
在线前向传递	对查询进行一次编码	进行一次查询编码，加上 k 个媒体前向传递	仅编码一次查询	仅编码一次查询
瓶颈	单个向量的维度	重排序尾延迟	索引大小与全扫描	整数和位运算
宽度	一个训练好的宽度，或一个单独的 MRL 头	与检索器分开维护的重排序器检索器	固定高维度	一个 Prefix-MRL 头，六个宽度，训练时的 ST-HAC time

问答、定位和工具使用，将它们作为一个样本，评估挖掘负样本中的证据，并加入仅在顺序上有所不同的视频负样本。第4节训练一个带有原生视频词元的双向骨干网络，以及一个覆盖 64 至 2048 宽度的 Prefix-MRL 头；训练期间的聚类和查询侧分数均在该节中介绍。第5 节按参数块添加分阶段检查点和专用检查点。第6节为每种模态使用一次时空聚类；其中给出了一个记录中的字节数。第7节从一个记录中推导出三种表征，并缩小候选窗口。第7.5 节在不使用视频解析模型的情况下，将命中结果还原到页面矩形框或视频中的具体秒数。相关数字见第9.5节。

2 相关工作

2.1 通用多模态嵌入和多向量表示

如今，人们期待一个通用嵌入模型能够处理文本、图像、文档和视频。VLM2Vec 和 MMEB 套件将检索、分类、问答和定位纳入同一项评估

[1, 2, 3]。Esfandiarpoor 等人 [13] 衡量了当查询需要多条证据时，这种相似度分数所付出的代价。Qwen3-VL-Embedding、WeMM-Embedding、Ovis-Embedding 和 U-MARVEL 通过多阶段对比学习、困难负样本、重排序器监督和跨尺度蒸馏来训练单个向量 [10, 14, 15, 16]。相比之下，jina-embeddings-v4 和 v5-omni 同时保留单向量和多向量输出，并冻结视觉塔，以保持几何结构对齐 [17, 18]。

ColBERT 和 ColBERTv2 [5, 19] 为文本确立了晚期交互，ColPali 则将其扩展到整页视觉文档，用页面截图取代 OCR 流程 [6]。ViDoRe V3 将这一评估扩展至多语言页面 [20]。Jina-ColBERT-v2 涵盖多语言文本 [21]。ModernVBERT 和 Nemotron-ColEmbed-V2 探索了用于文档晚期交互的双向注意力和模型合并[22, 7]。Video-ColBERT 和 CLAMR 将晚期交互扩展到视频 [8, 9]。EVIE 将证据核验、Prefix-MRL 头和布局感知 HAC 应用于视觉文档 [12]。套娃式表征学习和 MatFormer 在一次前向传递中嵌套多个宽度 [23, 24]。MetaEmbed 可在测试时改变元标记的数量 [25]。OmniEVIE 将这一设计扩展到图像、视频、文本和智能体任务，并新增共享的时空压缩、分块合并、查询侧评分，以及百万级三级检索。

2.2 压缩与快速多向量搜索

压缩多向量索引，关键在于选择保留匹配的哪一部分。MUVERA 将 MaxSim 替换为一个固定维度的向量，从而可以复用现有的最大内积索引 [26]。PLAID 保留多向量分数，但通过全局码本和残差码进行过滤 [27],；查询感知令牌预算则将原始令牌的选择推迟到最终阶段 [28]。第三种方法是减少存储向量的数量，要么在编码器内部合并词元 [29]，要么对冻结的索引进行聚类和剪枝，例如注意力引导聚类 [11], MarginMerge [30], 和先剪枝后合并 [31]。ST-HAC 属于第三种方法，但有一点不同：训练时和写入索引时都使用相同的聚类。随后，均值、1 位符号和 int8 多向量便成为对同一条记录的三种读取方式，且使用固定的 int8 缩放。

2.3 假负例、排序蒸馏与模型合并

对比式嵌入的质量取决于哪些负例是假负例，以及蒸馏的是谁的排序。FALCON 研究了大规模图文预训练中的假负例 [32]。CORE 使用 Rank-KL 将重排序器的排序蒸馏到嵌入中 [33]。在参数方面，任务算术通过加减任务向量进行操作 [34],，而 TIES-Merging 会丢弃较小的增量参数，并解决符号冲突 [35]。Conan-embedding-v3、PlaM 和免训练缓解分别研究了模态特定融合、选择性层注入，以及多模态微调后语言能力的恢复 [36, 37, 38]。OmniEVIE 中的合并为每个参数块分配独立的系数，因此合并器和投影头可以采取不同于语言层的更新步幅。

3 训练数据

OmniEVIE 将每项任务都纳入一个检索样本中。指令说明任务类型，证据等级说明挖掘负例是否确实为负例，而动态采样规则则调整每个样本被重复访问的频率。

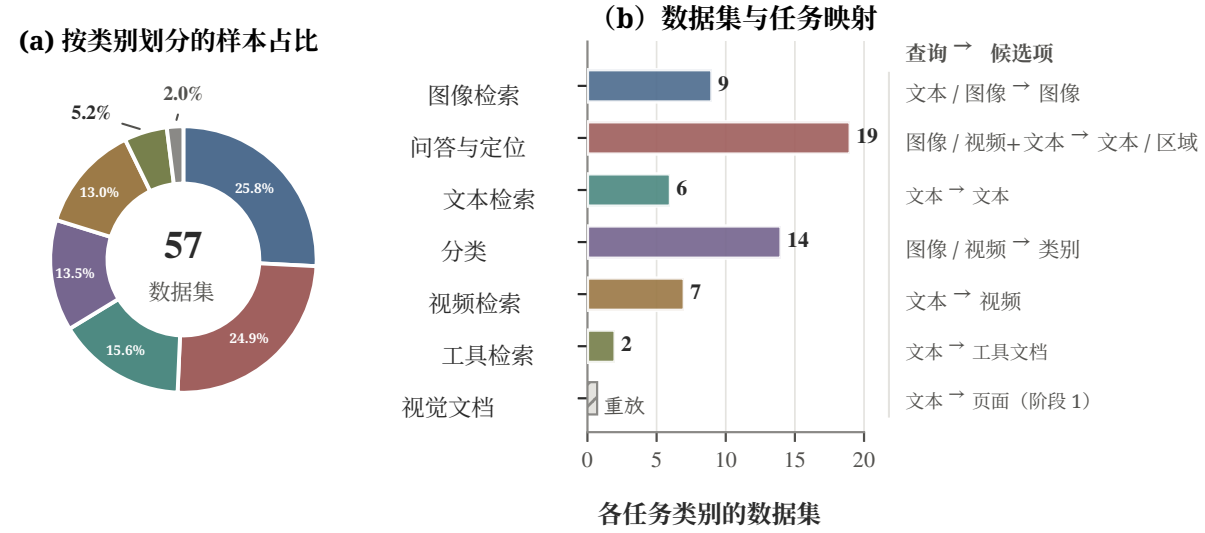

图 2。 57 个公开数据集中的训练混合情况（约 2M 个样本）。（a）按任务类别划分的样本占比：图像检索 25.8%、问答和定位 24.9%、文本检索 15.6%、分类 13.5%、视频检索 13.0%、工具检索 5.2% 以及视觉文档 2.0%。（b）各任务类别的数据集数量及查询到候选项的映射；视觉文档从阶段 1 重放。

3.1 单个样本格式

每个示例都是

$$z_i = \left(I_i,\, q_i,\, d_i^{+},\, \mathcal{N}_i,\, y_i\right), \tag{2}$$

其中，I_i 是场景指令，q_i 是文本、图像、图像-文本对或视频，$d^+{}_i$ 是正例，$\mathcal{N}_i$ 是一组明确的困难负样本，y_i 是教师分数或分布。检索和定位使用页面、图像、裁剪图或视频片段作为正例。分类使用类别名称作为候选项，并优先选择来自同一数据集且与目标类别相近的类别，按类别频率进行平衡。开放式问答将问题和图像与参考答案配对。多项选择将正确选项作为正例，将其他选项作为困难负样本。工具检索使用工具文档或技能描述作为候选项。界面和记忆任务留作零样本评估。

一个批次包含一个任务，因此每个批内负例都是该任务的另一个候选项。训练混合数据包含来自 57 个公开数据集的约 200 万个样本（图 2）。

3.2 场景指令

图像到图像搜索将指令、图像和一个空文本片段作为查询。组合式搜索会将实际问题置于图像之后。文本查询会将指令置于问题之前，其余零样本任务共用一个任务前缀（表 2）。过薄或过小的图像会按短边调整大小并进行填充，以使宽高比保持在限定范围内。

3.3 证据等级与时间硬负例

高分挖掘出的候选项通常是困难负样本的来源，也是漏标正例藏身之处。如果将回答查询的页面视为负例，梯度就会将该页面推远。

表 2. 场景指令和查询布局。

场景	查询	指令	候选项
图像到图像	指令 + 图像，空文本	查找具有相同物体或实体的图像	图像
图文组合	指令 + 图像 + 问题	查找能够回答有关以下内容的问题的图像给定的图像	图像
文本到图像或页面	指令 + 查询文本	检索请求中提到的图像或页面	图像或页面
文本生成视频	指令 + 查询文本	检索请求所指明的视频；指令置于前面	原生视频
其他零样本任务	任务指令 + 查询	一个共享任务前缀	文本、图像或视频

表3。 对视觉文档池进行三级证据检查。计数是在应用等级之前的候选负例数量。

等级	含义	处理方式
完整证据	该候选答案满足查询，被提升为正例。在 775,635 个查询中，有 175,137 个获得了一个额外的正例。	
部分证据	一条有用线索，而非完整答案	在分母中被屏蔽。与完整证据一起，223,877/1,551,270 (14.4%) 的候选项负例已被屏蔽。
没有证据	无关或相互矛盾	保留。仍有 1,327,393 个困难负样本。

来自查询。多模态评判器为每个挖掘负例分配三个等级之一（见表3）。具有完整证据的候选项会被提升为正例（当查询有$n^+{}_i$个正例时，每个正例的权重为 $\omega_i = 1/n^+{}_i$），具有部分证据的候选项会在损失分母中被屏蔽，而没有证据的候选项仍为难负例。

静态背景通常足以识别一个视频片段，因此两种困难负样本都让损失聚焦于时间顺序。反向负样本将一个正向片段（正例）与同一片段的反向版本配对，而查询则将正向描述与反向描述配对。窗口外负样本保留目标片段作为正例，并在原始查询下，将同一部长视频中的另一个片段作为负例。

3.4 从基线和多个教师模型进行动态采样

并非每个样本都需要相同的更新。有些样本基线模型已经能够解决，而对于某些任务，所有可用的教师模型都表现不佳。教师模型的选择取决于答案的形式。开放式答案和细粒度分类采用多模态嵌入教师模型及其批内相似度分布。排序任务（包括文本到图像、视频定位和工具检索）采用交叉编码器，并依据候选项之间的差距进行处理。多项选择则采用判别式教师模型及其在各选项上的分布。

随后，根据基线模型和教师模型是否将正例排在首位，为每个样本进行标注（表4）。两种模型均以较大幅度正确排序的样本会被下采样。教师模型排序正确而基线模型未能正确排序的样本会被加大权重。当基线模型胜过教师模型时，将关闭蒸馏，并使用金标准标签计算损失。如果基线模型在某项任务上表现优于教师模型的样本数多于反向情况，则会针对该任务关闭蒸馏。

4 模型与训练

表 4. 五路样本策略。“First”表示正例排名第一。仅当教师模型预测正确时才进行蒸馏。

类别	基线	教师模型	保留规则	困难负样本
1，简单一致	首先，大间隔	正确	固定下采样；类别任务保持每类下限	基线选择加上原始集合
1b，临界样本	首先，小间隔正确		全部保留	与类别1相同
2，教师模型填补空缺	不是第一	正确	加大权重	与类别1相同
3，基线领先	第一	错误	全部保留，仅金标准标签	仅原始集合
4，困难	不是第一	错误	全部保留，仅保留金标准标签	仅保留原始集合

4.1 骨干网络、双向注意力与原生视频

OmniEVIE 构建于 Qwen3.5-VL 混合注意力骨干网络之上，由视觉编码器、跨模态融合器、语言层和 Prefix-MRL 头组成。发布的模型包含 45 亿个参数，隐藏宽度为$d_{\text{hid}} = 2560$。语言层交替采用 GatedDeltaNet 块和全注意力。文档阶段将一个具有$d_{\text{hid}} = 4096$ 的 80 亿参数双向 EVIE 教师模型蒸馏到这个 45 亿参数模型中 [12]。

文本由分词器嵌入。图像和页面经过视觉编码器与空间合并后，转化为图像块标记。因果掩码会使视觉标记无法看到其后出现的指令。全注意力层采用双向注意力，掩码则标记填充部分。线性注意力模块保持骨干网络中的原样。这样，每个词元都可以关注完整序列 [7, 12, 22]。

OmniEVIE 将每个视频片段作为一个序列输入，因此相邻帧会在编码器内部相遇。逐帧编码则对每帧分别进行一次前向处理，然后再对向量进行池化，因此帧只有在经过骨干网络后才会相遇；反向播放的视频片段会产生相同的帧集合，而超出标记预算时则必须丢弃帧。相邻帧以时间步长 $\Delta t_{\text{patch}} = 2$ 打包后输入视觉编码器；帧数为奇数时，最后一帧会重复。每个时间组前面都会加上以秒为单位的时间戳。超出标记预算时，所有帧都会按相同比例缩小，并保留时间样本。时间戳随位置递增，因此反向播放的视频片段会使内容与不同的时间戳配对，而第 3 节中的反向负样本具有非零梯度。

4.2 Prefix-MRL

蒸馏需要宽向量，而索引需要窄向量。一个共享矩阵$W \in \mathbb{R}^{2048\times 2560}$ 位于骨干网络之上。对于最终隐藏状态$h \in \mathbb{R}^{2560}$和宽度$d \in \mathcal{D} = \{64, 128, 256, 512, 1024, 2048\}$，

$$e^{(d)}(h) = \mathrm{L2}(W_{:d}\, h) = \frac{W_{:d}\, h}{\|W_{:d}\, h\|_2} \in \mathbb{S}^{d-1}. \tag{3}$$

一次前向传播即可生成所有宽度，而低宽度损失只更新前d 行。服务时始终存储$d = 64$，而$d = 2048$ 分支则为自蒸馏提供高宽度参考。

4.3 四个训练阶段

在单一阶段从头开始训练所有模态，会让频繁出现的简单任务挤占较难的对齐任务；而之后添加模态，则可能抹去已学到的文档行为。OmniEVIE 分四个阶段添加模态（图 3，表5），并按固定比例重放早期数据。每个批次仍然只包含单一任务。

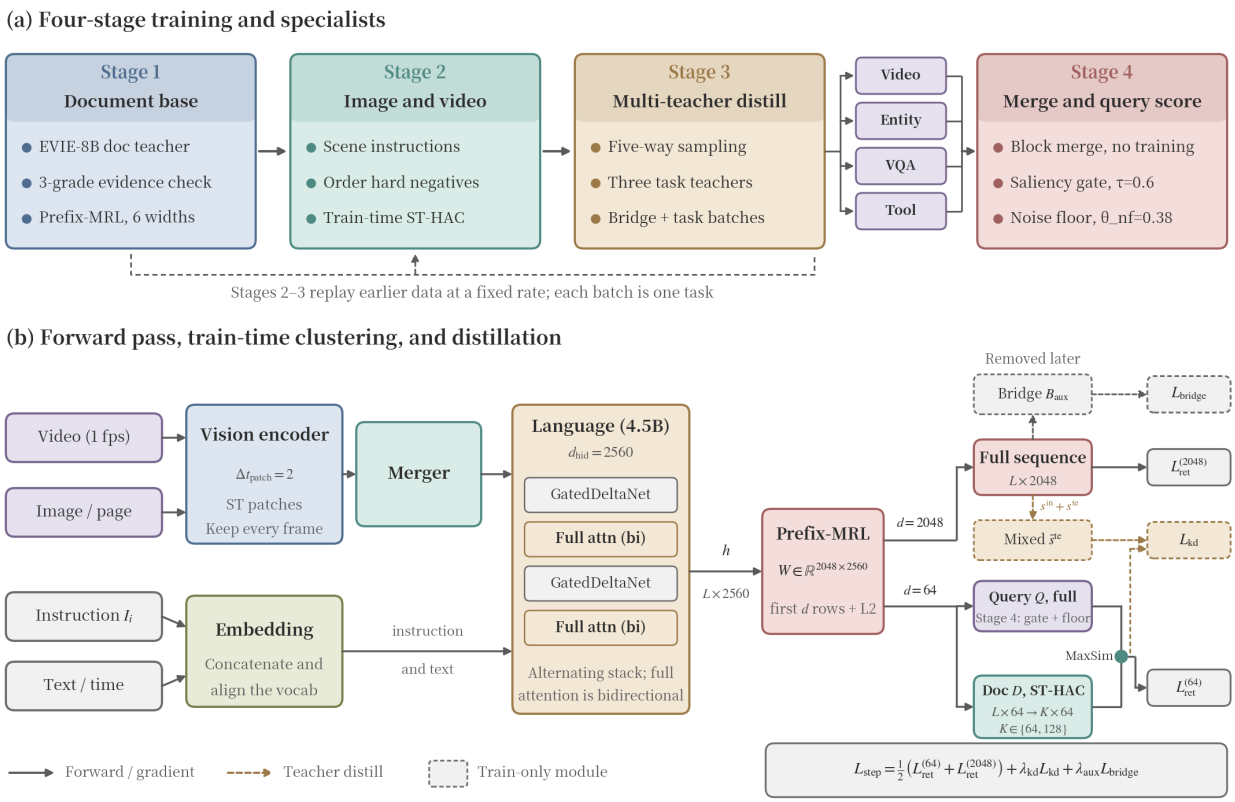

图3。 四阶段调度与前向图。（a）课程以及四个专家分支在阶段3之后分离出来。（b）视觉、指令和文本经过骨干网络与 Prefix-MRL 头，然后分为$d = 2048$参考分支和$d = 64$ 服务分支。查询保留其完整序列，阶段4附加门控和噪声底。候选项在训练期间使用 ST-HAC 进行聚类。$d = 2048$ 分支和外部教师对$d = 64$分支进行蒸馏。

表5。 四个训练阶段及每个阶段重放的内容。

阶段	输入	监督	重放
1，文档	双向文档向量和 六种 Prefix-MRL 宽度	来自 EVIE-8B 的列表式蒸馏，三 等级检查，六种宽度的联合训练	后续阶段的文档池
2，图像和视频	图像到图像、组合图像- 文本、文本到图像、原生视频	场景指令、时间困难负例、一个 单向排序约束、在其上的 ST-HAC 批次中的候选项	文档和较早的数据，以 固定速率；一个批次是一项任务
3，所有任务	分类、问答、 定位、长文本、工具、接口、 记忆	五路采样器，每个任务配备三位教师， 通往高宽度教师模型的投影桥接	保留重放配额；批次仍为 单任务
4，合并与查询得分	重新加入专家模型；查询端 分数经过训练	分块合并，以及门控与噪声 底	将增量参数添加到它们所 训练过的块上

4.4 训练期间的损失、蒸馏与聚类

宽分支和窄分支都会接受训练。窄分支上的候选项会被聚类，因为它们将用于索引构建时；分数则基于这些质心计算。

查询的候选池由批次中的其他文档及其自身的困难负样本组成。三类候选项会从负样本集合中排除（同一查询的其他正例仍为正例，来自另一来源的样本不参与此损失，而经教师模型评分至少与正例一样高的候选项则不再作为负例），候选池中剩余的每个候选项都是负例。设 π_i 为正例的 Softmax 概率。聚焦权重会突出尚未被分离的查询。

$$\mathcal{L}_{\text{focal}} = \mathbb{E}_{\text{batch}}\big[a_i\,(-\log \pi_i)\big], \quad a_i = \frac{\text{sg}\big((1-\pi_i)^\gamma\big)}{\mathbb{E}_{\text{batch}}\big[\,\text{sg}\big((1-\pi_j)^\gamma\big)\big]}, \quad \gamma = 1. \tag{4}$$

sg 会通过 a_i 阻断梯度，因此该权重相对于参数而言是一个常数。除以批次均值可使平均权重保持为 1，因此损失的尺度不会随批次的难度而变化。双向样本以权重 $\beta = 0.5$ 加入反向交叉熵 $\mathcal{L}_{d\to q}$。有序对（包括正向片段与其反向片段之间的有序对，以及已分级的负例）会加入成对项 $\mathcal{L}_{\text{ord}}$。较高分数 s_0 的梯度会被阻断，因此只有较低分数 s_1 会降低。一个分支上的检索损失为

$$\mathcal{L}_{\text{ret}} = \mathcal{L}_{\text{focal}} + \beta\,\mathcal{L}_{d\to q} + \lambda_{\text{ord}}\,\mathcal{L}_{\text{ord}}. \tag{5}$$

蒸馏目标将模型内部的$d = 2048$ 分数s^{in} 与外部教师得分s^{te}混合。两者均在有效候选项上进行标准化。使用混合权重w_e后，目标会重新映射到s^{in}的尺度上：

$$\tilde{s}^{\text{te}} = \mu_{\text{in}} + \sigma_{\text{in}}\big((1-w_e)\hat{z}(s^{\text{in}}) + w_e\hat{z}(s^{\text{te}})\big).$$

蒸馏 损失是教师与学生排序之间的 KL散度，再加上一个间隔项 ，

$$\mathcal{L}_{\text{kd}} = c_i\Big[\,\text{KL}\big(p^{\text{te}} \,\|\, p^{\text{stu}}\big) + \lambda_{\text{margin}}\,\mathcal{L}_{\text{margin}}\Big], \tag{6}$$

其中，当教师分布的熵较低时，c_i 的值较大。

外部教师的维度可能为$d_T = 4096$，因此无法直接与维度为 2048 的学生向量进行评分。一个无偏置线性映射$B_{\text{aux}} \in \mathbb{R}^{d_T\times 2048}$经正交初始化后，将学生均值向量映射到教师空间。点项为 1 减去余弦相似度。批次项是查询—文档余弦矩阵之间经过平滑处理的 ℓ_1 损失。训练结束后会丢弃B_{aux}，推理时使用 Prefix-MRL 头。

每一步中，窄分支以 0.3 的概率使用预算 $K = 64$ 对候选项进行聚类，以 0.7 的概率使用预算 $K = 128$进行聚类。簇分配保持固定。簇内的均值及其后续归一化会接收梯度。簇中的每个词元均获得质心梯度的相同比例 $1/|C_k|$。步损失为

$$\mathcal{L}_{\text{step}} = \frac{1}{2}\big(\mathcal{L}_{\text{ret}}^{(64)} + \mathcal{L}_{\text{ret}}^{(2048)}\big) + \lambda_{\text{kd}}\,\mathcal{L}_{\text{kd}} + \lambda_{\text{aux}}\,\mathcal{L}_{\text{bridge}}. \tag{7}$$

$d = 64$ 分支对质心进行评分。$d = 2048$ 分支对完整序列进行评分。

4.5 查询侧显著性门控和噪声底

对于多模态查询，均匀MaxSim 在每个非填充词元上使用 $w_i = 1$，但存在两个问题。视觉图块和背景词元会主导总和，而且每个查询词元都能在任意候选项上找到中等的内积。这些中等匹配累积起来，会稀释真正的命中。

门控从均匀权重 $w(0)\,i$ 开始。其输入是每个词元的十个领域无关特征 ϕ_i：归一化前范数、与查询均值方向的偏差、与相邻词元的差异、查询内部的唯一性、模态和位置标记，以及查询长度。一个两层 SiLU网络生成分数 g_i。词元方向上的线性项带来 ℓ_2 惩罚。随后，权重将被

$$m_i = \exp\big(\tau_{\text{gate}}\tanh(g_i)\big), \qquad w_i = \frac{L_q\,w_i^{(0)}m_i}{\sum_{j=1}^{L_q} w_j^{(0)}m_j}. \tag{8}$$

门控输出层初始化为零，因此训练从均匀MaxSim开始。在 $\tau_{\text{gate}} = 0.6$时，每个词元的尺度会在约 0.55 到 1.8 之间重新调整，并且权重的总和仍为L_q。门控包含 257 个参数。

令$r_i = \max_{v \in D} \langle \hat{q}_i, v \rangle$。噪声底分数会放大$r_i$高于 θ_{nf} 的部分：

$$s_{\text{tok}}(q_i, D) = r_i + \beta_{\text{nf}} \operatorname{ReLU}(r_i - \theta_{\text{nf}}), \qquad s(Q, D) = \sum_{i=1}^{L_q} w_i \, s_{\text{tok}}(q_i, D). \tag{9}$$

低于噪声底的匹配保持不变。高于噪声底的匹配仅超出部分会乘以 $1 + \beta_{\text{nf}}$。由此可得公式 $\beta_{\text{nf}} = 0$，恢复公式 1。 s_{tok} 关于 r_i单调递增。argmax 聚类与标准 MaxSim 相同，第 7.5 节通过该聚类进行定位。

在d 维单位球面上，查询词元与随机质心的内积均值为 0，方差为 $1/d$。对K 个质心取最大值，其期望至多为

$$\mathbb{E}\Big[\max_{1 \le j \le K} \langle \hat{q}_i, v_j \rangle\Big] \;\le\; \sqrt{\frac{2 \ln K}{d}} \;=\; \sqrt{\frac{2 \ln 128}{64}} \;\approx\; 0.389. \tag{10}$$

因此，无关记录对每个查询词元的贡献约为 0.389。噪声底设为 $\theta_{\text{nf}} = 0.38$。在 ViDoRe V3 上，测得 r_i 在无关页面上的均值为 0.393，在相关页面上的均值为 0.694；相关页面上有 92% 的词元超过 0.38。取 $\beta_{\text{nf}} = 2.5$ 时，相关页面均值处的词元得分约为无关页面均值的 3.8 倍，而接近噪声底的匹配几乎不变。噪声底的计算见 S3（第 7 节）。

5 按块模型合并

5.1 什么是合并

这些权重是对不同阶段训练出的模型进行合并的结果。第一层会对共享同一起点的全数据运行结果取平均，并对相邻阶段取平均。在阶段 3 之后，第二层会针对视频、细粒度实体、视觉问答和工具使用训练专家模型，然后将它们的增量参数加回主干模型（图 4）。每种能力都用各自的数据训练，并在参数空间中添加 [34]。

参数被划分为五个块 $b \in$ {视觉、合并器、词汇表、LLM、头部}：视觉编码器、合并器、标记嵌入、语言层和 Prefix-MRL 头。令 P 表示主干模型， $\theta^{(k)}$ 表示第 k 个模型。增量参数为 $\Delta_k = \theta^{(k)} - P$，而 $\Delta_{k,b}$ 表示其在块 b上的限制。合并后的块为

$$\theta_b = P_b + \sum_{k=1}^{M} c_{k,b} \, \mathcal{T}_\rho(\Delta_{k,b}), \qquad c_{k,b} \in [0, 1]. \tag{11}$$

均匀平均是一种特殊情况，其中每个块共享同一个系数。文本或工具专家在视觉编码器和合并器上的系数为 0，在语言层上的系数为正数。图 4 中，条形的长度表示该系数。文本或工具专家的视觉条形为空，因为系数为零。

$\mathcal{T}_\rho$ 保留每个张量中幅值最大的 ρ 部分，并将其余部分置零 [35]：

$$\mathcal{T}_\rho(\Delta)_j = \begin{cases} \Delta_j, & |\Delta_j| \ge \tau_\rho, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

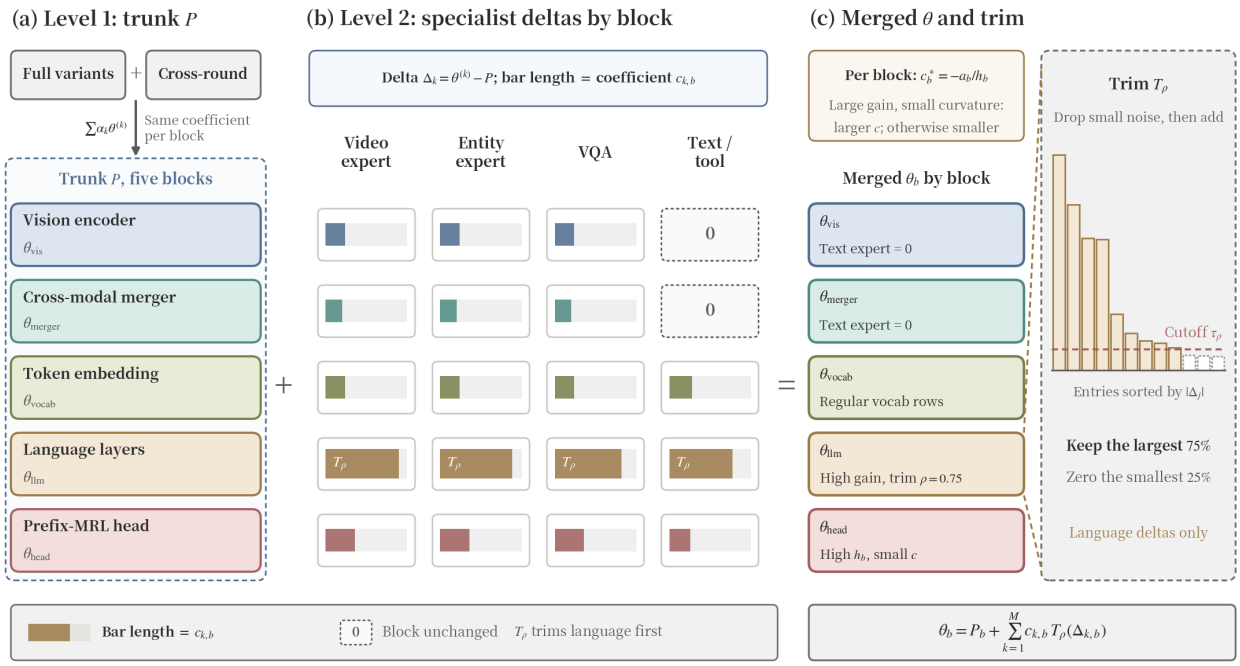

图 4. 双层分块合并。(a) 对来自 onestart 的全数据变体以及相邻轮次的权重，按每块一个系数进行平均，合并到主干模型 P 中。(b) 四个专门模块添加 $\Delta_k = \theta(k) - P$，并按块加权；条形长度为 $c_{k,b}$。文本和工具专门模型将视觉和合并器系数设为 0，并按 $\mathcal{T}_\rho$ 修剪语言层增量。(c) 合并后的逐块策略。

其中，τ_ρ 是 $|\Delta|$ 的 $(1-\rho)$ 分位数。专家增量中的大多数条目都很小，且接近随机值。它们对目标性贡献甚微，还会干扰其他专家模型。语言层增量使用 $\rho = 0.75$。视觉、合并器和头部增量使用 $\rho = 1$。

5.2 逐块系数

二阶展开将所有块共用一个系数时产生的额外损失表示出来。考虑一个模型，其修剪后的块增量为 δ_b，系数为 c_b。设 g 和 H 分别为模型在P处的损失梯度和海森矩阵，并忽略跨块二阶导数。合并后的损失近似为

$$\mathcal{L}(\theta) \approx \mathcal{L}(P) + \sum_b \left(a_b\, c_b + \frac{1}{2} h_b\, c_b^2 \right), \qquad a_b = g_b^\top \delta_b, \quad h_b = \delta_b^\top H_{bb}\, \delta_b. \tag{13}$$

$a_b < 0$ 表示该增量会降低损失，而 $h_b > 0$ 表示曲率代价。每个块对应一个关于c_b的二次函数，其顶点为

$$c_b^* = -\frac{a_b}{h_b}, \tag{14}$$

使用的系数是将该值截断至 $[0, 1]$ 后得到的值。如果强制每个块共用一个系数 c，则最佳共享值是逐块顶点的曲率加权平均值，额外损失为

$$\Delta\mathcal{L} = \frac{1}{2} \sum_b h_b \left(\bar{c} - c_b^* \right)^2 \;\geq\; 0. \tag{15}$$

随着各块的最优值逐渐分歧，这种差距也随之扩大。用于写入存储向量的投影头曲率较大，因此其最优系数较小。语言层承载了大部分任务能力，因此倾向于使用较大的系数。文本或工具专家对视觉编码器的 依赖 $_b \geq 0$ 为零，因此该系数为 0。修剪会移除那些几乎不改变 向量$_b$ 、却会增加 干$_b$ 扰其他专家模型的条目。各合并步骤的测量效果见表 12。

6 ST-HAC

6.1 凝聚聚类

高分辨率页面或 32 帧片段会产生数百至数千个向量，而且这些向量的分布并不均匀。第 1.3 节计为未选中的向量，大多是共线背景和静态帧 [11]。决定检索结果的证据——一个单元格、小徽标或短时动作——是远离背景的孤立点。压缩会将共线背景和静态帧合并为少数几个簇，并使孤立的证据片段单独成簇。Ward 聚类会合并使簇内平方和增幅最小的一对簇：邻近的背景词元首先合并，而孤立点则保留到最后。ST-HAC 在时空特征空间中执行这种聚类。表 13 对比了该方法与均匀池化和随机剪枝。

6.2 适用于每种模态的统一流程

经过$d = 64$投影和逐词元 ℓ_2 归一化后，任意输入都是一个序列 $\{\hat{v}_i\}_{i=1}^{N} \subset \mathbb{S}^{d-1}$。ST-HAC 将该序列替换为至多$K \le 128$ 个维度为 64 的向量。图5 展示了具体步骤。如果聚类仅依据语义，那么报告中两列里相似的数字，或视频中相距很远的相似背景，都会被归入同一个簇。因此，每个词元都会获得归一化坐标$(x_i, y_i, t_i) \in [0,1]^3$。对于以步幅 Δt{v31>打包为F 组的视频，

$$t_i = \frac{f_i}{\max(F-1,\,1)}. \tag{16}$$

对于文本序列，$t_i = (i-1)/\max(N-1, 1)$。聚类特征将语义向量与正弦空间与时间编码混合，并重新归一化：

$$z_i = \mathrm{L2}\Big(w_v\,\hat{v}_i + w_s\,p_{xy}(x_i,y_i) + w_t\,p_t(t_i)\Big), \qquad w_v = 1 - w_s - w_t. \tag{17}$$

视频使用 $(w_s, w_t) = (0.07,0.03)$。图像和页面使用 $(0.10,0)$。文本使用 $(0,0.10)$。缺失的轴权重为零。代码和输出格式是共用的。

长输入首先进行分段。视频按 60 秒切分，文本按 1,024 个词元切分。每个片段都进行聚类，然后再次对分段质心进行聚类，直至达到 K，并赋予权重为 $w_t = 0.03$ 的分段时间编码。随后，Ward 距离矩阵始终保留在片段内部。对 $\{z_i\}$ 进行 Ward 聚类，直到得到 K 个簇为止。如果 $N \le K$，则保留完整序列。位置权重远小于语义权重，因此划分仍由语义主导。

6.3 单调时间码

空间编码 p^{xy} 是标准的多频正弦函数。相同的频率用于时间时并不安全。高于 π 的频率会发生回绕，因此相距很远的两个镜头可能获得相似的编码。ST-HAC 从 $[\pi/4, \pi]$ 中均匀选取 $M = d/2$ 个频率

$$p_t(t) = \frac{1}{\sqrt{M}}\big(\sin\omega_1 t,\ \cos\omega_1 t,\ \ldots,\ \sin\omega_M t,\ \cos\omega_M t\big)^{\top}. \tag{18}$$

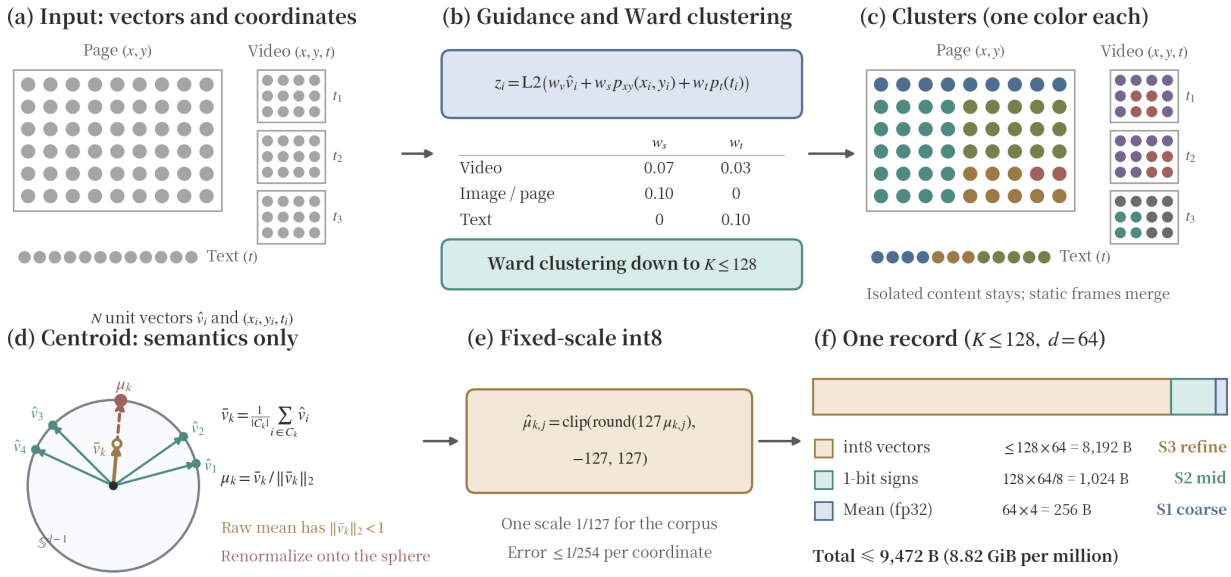

图 5。 ST-HAC 离线压缩。(a)(b) 对语义向量和时空编码进行 Ward 聚类。(c) 孤立区域保持为独立的簇，相邻的静态帧则合并。(d)质心仅由语义向量的重新归一化均值构成。(e) 定标 int8。(f) 一条记录中的字节数。

前置因子为 $1/\sqrt{M}$。每个正弦–余弦对对平方范数的贡献为 $1/M$，而 M 对贡献为 $\|p_t(t)\|_2^2 = 1$。间隔为 Δt 的两个时间点的内积为

$$K_t(\Delta t) = \langle p_t(t),\, p_t(t+\Delta t)\rangle = \frac{1}{M}\sum_{m=1}^{M}\cos(\omega_m \Delta t). \tag{19}$$

每个 $\omega_m \Delta t$ 对于 $\Delta t \in [0, 1]$ 都属于 $[0, \pi]$，其中余弦相似度会降低，因此 K_t 会从 $K_t(0) = 1$逐渐降低。相邻的静态帧会先合并，距离较远的镜头则彼此分开。时间坐标会归一化为 $[0, 1]$,，不受帧率和时长影响。

6.4 语义质心

位置特征用于确定划分，随后会从索引中移除。否则，与查询的内积会包含与内容无关的偏置。给定簇 $\mathcal{C} = \{C_1, \ldots, C_K\}$，质心仅使用原始语义向量 $\bar{v}_k = \frac{1}{|C_k|}\sum_{i\in C_k} \hat{v}_i$，并进行重新归一化：

$$\mu_k = \frac{\bar{v}_k}{\|\bar{v}_k\|_2} \in \mathbb{S}^{d-1}. \tag{20}$$

由于每个成员都是单位向量，

$$\frac{1}{|C_k|}\sum_{i\in C_k}\|\hat{v}_i - \bar{v}_k\|_2^2 = 1 - \|\bar{v}_k\|_2^2. \tag{21}$$

分散程度更高的簇具有更短的均值。使用 $\bar{v}_k$ 进行评分会惩罚覆盖更多内容的簇。公式 20 之后，内积为 $\cos\theta$，这与训练期间使用的分数一致。表 13对二者进行了比较。

表 6. 三个阶段的候选表示、算子和字节数。K' 是存储的质心数量，最多为 128。

阶段	表征	算子	窗口	字节
S1	$\mu_{\text{avg}} = \text{L2}(\frac{1}{K'}\sum_k \mu_k)$	精确内积或近似最近邻	$N_{\text{db}} \to N_1$	256
S2	符号(μ_k)	汉明 MaxSim、异或和人口计数	$N_1 \to N_2$	1,024
S3	定标 int8 $\hat{\mu}_k$	Int8 MaxSim，以及公式 9	$N_2 \to$前k	8,192

6.5 定标 int8 与单条记录字节数

的每个坐标 μ_k 都位于 $[-1, 1]$中，并且语料库采用 1/127 的尺度。索引构建会在该尺度上进行舍入，$\hat{}\mu_{k,j} = \text{clip}(\text{round}(127\,\mu_{k,j}),\ -127,\ 127)$。每个坐标的误差最多为 1/254。分数通过整数运算累加，并直接追加新记录。表 17 比较了这些尺度。在 $K \le 128$ 和$d = 64$的情况下，int8 多向量最多占用 8,192 字节，fp32 均值占用 256 字节，可选的 1 位符号占用 1,024 字节。两级索引最多占用 8,448 字节（每百万条记录为 7.87 GiB）。加入符号位后，三阶段索引最多占用 9,472 字节（每百万条记录为 8.82 GiB）。

7 三级检索

7.1 单条记录的三种视图

两阶段流水线通常会在粗检索之后读取全精度侧索引，而在百万级规模下，该侧索引的大小达到数百 GB。OmniEVIE 从第 6 节的记录中读取全部三个阶段6，表 6）。

查询仅编码一次，其均值和符号位均从同一输出中读取 .

7.2 延迟与窗口权衡

对于语料库规模为 N_{db}、包含 L_q 个查询词元、最多使用 K 个质心且维度为 d 的情况，检索时间为

$$T_{\text{total}} = c_{\text{s1}}\, N_{\text{db}}\, d + c_{\text{pop}}\, N_1\, L_q\, K + c_{\text{int8}}\, N_2\, L_q\, K. \tag{22}$$

第一项是 S1 扫描，第二项是汉明 S2，第三项是 int8 S3。当 $d = 64$ 时，一个质心的符号模式为一个 64 位字，通过一次异或和一次人口计数进行评分，而 int8 点积执行 64 次乘加运算。在查询长度中位数和 $K = 128$的情况下，int8 评分每个候选项的开销约为 20.0 μ秒，汉明评分约为 4.0 μ秒，因此 $r = c_{\text{int8}}/c_{\text{pop}} = 5$。在保持 S2 与 S3 之和不变的情况下，减少$\Delta N_2$ 个 int8 候选项可以换取

$$\Delta N_1 = r\, \Delta N_2 \tag{23}$$

S1 窗口中的额外候选项。直接将 S1 窗口送入 S3 时，窗口会保持较小，因为 int8 分数占据了主要的时间开销。在更大的语料库中，该固定窗口所占记录的比例更小，而均值向量会在 S3 看到真实命中项之前将其筛掉。S2 会将 int8 窗口缩减至约一千个候选项，并且根据公式 23，以更低的总时间换取 S1 中约 r 倍的候选项。

S2 在两侧都存储符号，因此一对只需进行一次异或和一次人口计数。浮点查询针对 1 位文档的计算成本仍大致相当于一次 int8 点积；该行是表 18的最后一行。对称 1 位是中间阶段，而对称 int8 是最终阶段。

表7. 三种多向量检索设计的部署属性。

	PLAID	MUVERA	OmniEVIE
索引	全局码本和残差倒排表	一个高维 FDE 向量	记录上的均值、1 位符号和 int8 质心记录
更新	语料库发生变化时重建码本	固定随机超平面	追加一个独立编码的记录
过滤器	倒排列表中的标量过滤器	单向量引擎的过滤器	限制 S1候选集
局部命中	质心映射，残差必须进行解码	被投影丢弃	保留自 S3 最大值参数
引擎	倒排索引和残差解码	高维最大内积搜索	内积、人口计数、int8 点积-uct

7.3 符号阶段的顺序保持

S2 将相关候选项保留在前 N_2 个之内。S3 生成最终排序。符号保留每个坐标所在的正交象限。对于单位向量 u、v 和随机旋转 R [39]，

$$\mathbb{E}\left[\frac{1}{d}\langle \mathrm{sign}(Ru),\, \mathrm{sign}(Rv)\rangle\right] = 1 - \frac{2\theta}{\pi}, \qquad \theta = \arccos\langle u, v\rangle. \tag{24}$$

汉明相似度与余弦相似度呈单调关系，因此中间阶段保持这一顺序。OmniEVIE 在原始坐标中取符号。全扫描前十名的保留情况见表 18。

7.4 PLAID 和 MUVERA

PLAID 将语料库聚类为全局码本，并将残差存储在倒排索引中 [27]。MUVERA 将一条记录的多向量投影为一个固定维度编码，并复用单向量索引 [26]。表7 比较了这三种需要更新的系统。PLAID 适用于可以重新聚类的语料库。MUVERA 适用于已有单向量搜索能力的技术栈。OmniEVIE 适用于持续接收不同模态记录的语料库。

7.5 命中定位

每个质心对应一组图像块或一个时间区间。边界框和时间跨度是源文档的元数据，而 S3 最大值参数则标示命中位置。

对于页面或图像，μ_k 的元数据存储其成员图像块的最小矩形及页面索引。对于高权重查询词元，$\hat{k} = \arg\max_k \langle \hat{q}_i, \hat{\mu}_k \rangle$ 会突出显示该矩形。argmax 是 MaxSim 的一个中间结果。定位会保留矩形覆盖页面面积不超过 5% 的簇。较大的矩形意味着空白区域和背景分散在页面各处。对于较长的视频，片段采用相同的记录格式按固定窗口编码，每个质心存储 $[t_{\min}, t_{\max}]$。同一视频的各个窗口根据视频标识符排列在同一时间轴上，其时间跨度即为公式 25 中的热度。

一个动作词命中的时间区间覆盖片段的一部分，而完整片段则来自将每个查询词元置于同一时间轴上。在 QVHighlights 验证集上，60.7% 的相关片段跨越了窗口边界。窗口分别进行编码和聚类，且质心始终位于其窗口内，因此每个词元会在每个窗口中选取其最佳质心，而高于噪声底的证据会进入边界两侧的热度中。令 $r_{(j)\,i}$ 表示词元 i 在窗口 j 内的最大相似度，以及 $\hat{k}_{(j)\,i}$

对应的质心。组g 的热度为

$$h(g) = \sum_{j=1}^{N_{\text{win}}} \sum_{i:\, g \in [t_{\min}(\hat{k}_i^{(j)}),\, t_{\max}(\hat{k}_i^{(j)})]} w_i \,\text{ReLU}\big(r_i^{(j)} - \theta_{\text{nf}}\big). \tag{25}$$

每个词元都会将公式 9 中高于噪声底的部分乘以公式 8的门控权重，然后累加到其质心对应的时间组中，权重越大，贡献越大。热度最高的组即为定位点，而热度不低于峰值 80% 的连通线段即为返回时间跨度。排序分数仍取公式 9中的全局最大值。图9 展示了示例、测量结果，以及传递给下游生成器的上下文。

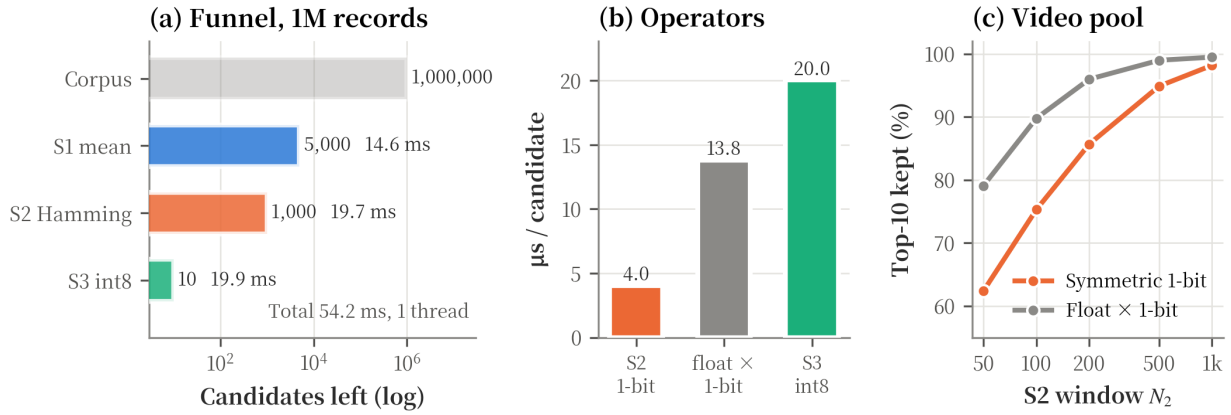

图 6. 三级检索。(a) 百万条记录上的候选窗口及各阶段耗时。(b) 在查询长度中位数和 $K = 128$ 条件下，三个算子的每个候选项耗时。(c) 混合视频池上的 S2 窗口 N_2 及前十保留率。

8 实验

8.1 实验方案

除非某一行另有说明，所有 OmniEVIE 数值均采用 $K \le 128$、$d = 64$以及固定 int8 缩放。任务定义遵循 MMEB [1, 2, 3]；附录B 列出了各套件。WeMM-Embedding 和 Qwen3-VL-Embedding 在 MMEB-V2 上的数值为官方排行榜结果。Qwen3-VL-Embedding 在 MMEB-V3 上的文本和智能体数值取自 Zhou 等人 [14]。表 10 中，Qwen3-VL-Embedding-8B 的平均值为已发布的 ViDoRe V3 数值 [20]；各领域数值是我们按该设置重新运行所得，重跑后的平均值为 58.37。WeMM-Embedding-9B 在 MMEB-V3 上的数值为排行榜值 59.33。

8.2 MMEB-V2 和 MMEB-V3

表8 细分了 MMEB-V3（190 个任务）中的文本子组和代理子组。文本子组包括推理检索（T-RR）、指令遵循（T-IF）、长上下文（T-LC）、多条件（T-MC）和通用检索（T-GR）。代理子组包括工具、图形用户界面、记忆和 MCMR。表9 将 MMEB-V2（78 个任务）归为四个类别：全部、图像、视频和视觉文档。图 1 中的蓝线表示全序列行。1。

表 8。 MMEB-V3 在 190 个任务上的表现。附录 B 中的 All 为未加权平均值。Text 报告 nDCG@5；其他组报告 Hit@1。8 帧行仅改变视频帧上限。粗体表示该列中的最佳值。

模型	All	文本	T-RR	T-IF	T-LC	T-MC	T-GR	智能体	工具	图形用户界面	记忆	MCMR
任务	190	53	20	9	6	5	13	47	35	8	4	1
Qwen3-VL-Emb-2B	50.9	39.2	16.6	40.6	53.1	61.2	58.0	39.3	42.6	30.4	28.4	42.0
Qwen3-VL-Emb-8B	53.5	42.5	18.2	44.8	58.0	61.2	61.2	38.4	41.3	33.5	22.8	38.0
WeMM-Emb-2B	55.77	45.34	26.29	49.30	61.90	59.72	58.74	44.05	46.23	38.87	35.38	42.51
WeMM-Emb-4B	57.97	47.88	30.01	52.02	62.43	60.14	61.09	48.06	50.58	41.44	**39.19**	41.91
WeMM-Emb-9B	59.33	48.76	**31.82**	52.78	64.48	58.28	61.12	50.10	52.95	**43.33**	38.65	49.34
OmniEVIE-4B	60.06	51.77	26.36	69.65	70.71	67.75	63.58	54.03	**58.84**	41.34	37.35	47.39
OmniEVIE-4B，8 帧	59.76	51.77	26.36	69.65	70.71	67.75	63.58	54.03	58.84	41.34	37.35	47.39
OmniEVIE-4B，完整序列 $d = 2048$	**60.68**	**52.97**	27.46	**70.81**	**73.04**	**68.55**	**64.60**	**54.14**	58.81	42.00	37.60	**49.74**

表9。 MMEB-V2 在78项任务上的表现。所有数值均为未加权平均值。图像和视频报告 Hit@1；视觉文档报告 nDCG@5。WeMM-Emb-4B 使用与 OmniEVIE 相同的 45亿参数骨干网络。粗体表示该列中的最佳值。

模型	所有图像		视频	VisDoc
任务	78	36	18	24
Qwen3-VL-Emb-2B	73.25	74.96	61.87	79.22
Qwen3-VL-Emb-8B	77.82	80.12	67.15	82.36
WeMM-Emb-2B	77.94	79.63	70.82	80.75
WeMM-Emb-4B	79.19	80.84	72.11	82.04
WeMM-Emb-9B	**80.58**	**81.87**	**74.31**	83.34
OmniEVIE-4B	77.96	78.09	69.62	84.02
OmniEVIE-4B，8 帧	77.24	78.09	66.49	84.02
OmniEVIE-4B，完整序列 $d = 2048$	78.56	78.70	70.08	**84.70**

表 10。 ViDoRe V3 在八个领域和六种语言上的分级 nDCG@10 宏平均值。压缩行的存储空间以每百万页 GiB 为单位。粗体表示该列中的最佳值。

模型	索引	平均	压缩能耗	金融-英语	金融-法语	人力资源	工业	医学	物理学
Nemotron-ColEmbed-V2-8B	完整多向量	63.76	79.35	70.05	67.72	51.50	**66.92**	56.81	67.39	50.31
Qwen3-VL-Emb-8B	单向量	59.00	75.86	64.47	60.88	44.70	58.88	47.78	66.05	48.31
OmniEVIE-4B，$d = 2048$	Prefix-MRL，完整序列	**64.34**	**79.89**	**71.25**	**67.76**	**53.45**	66.33	**57.03**	**68.11**	**50.87**
OmniEVIE-4B，$K128 \times d64$	7.87 GiB/M	62.96	78.73	69.94	66.01	52.52	64.68	55.08	66.47	50.28
OmniEVIE-4B，$K32 \times d64$	2.15 GiB/M	61.16	78.13	68.61	64.52	49.43	60.90	51.73	65.73	50.22

OmniEVIE 在 MMEB-V3 上达到 60.06。将视频限制为官方规定的 8 帧后，同一组权重得分为 59.76，高于 WeMM-Embedding-9B 的 59.33。MMEB-V2 得分为 77.96，整体比 WeMM-Embedding-9B 低 2.62，图像项低 3.78，视频项低 4.69。图像和视频方面的差距体现在细粒度分类和视频检索上。MMEB-V3 的提升体现在长文本上：文本总体得分为 51.77，其中指令遵循为 69.65，长上下文为 70.71，多条件检索为 67.75。代理总体得分为 54.03，工具检索得分为 58.84。晚期交互会将每种查询条件与该文本中的不同片段相匹配。发布索引在视觉文档上的得分为 84.02。表 9 该列中的最高项是全序列，其得分为 9 的$d = 2048$，84.70。

8.3 视觉文档、细粒度图像和视频

表 10 和表 11 报告视觉文档、细粒度图像和视频。

在表 10 中，$K128 \times d64$ int8索引在 ViDoRe V3 上的分数为 62.96，而未压缩的$d = 2048$ 序列为 64.34。每百万页面的压缩索引占用 7.87 GiB，约为 fp16完整序列的 1/486，损失为 1.38 分。将K 进一步降至 32，即每百万页面占用 2.15 GiB 时，得分为 61.16。计算和能源是得分最高的领域（78.73 和 69.94）；法国金融和物理学的得分最低

表11. 细粒度 图像 Top-1、视频nDCG@10 和正向与反向准确率。四数据集平均值 s MSR-VTT , MSVD, DiDeMo 和 VATEX。粗体标示该列中的最佳值。

模型	FORB	S2S	MFashion	MSR	MSVD	DiDeMo	VATEX	Vid-4	排序
Qwen3-VL-Emb-8B	**92.80**	62.90	65.10	71.95	77.84	74.54	**76.60**	75.23	51.63
帧无关	–	–	–	63.76	73.11	78.30	69.37	71.13	51.17
OmniEVIE-4B	89.94	**72.16**	**75.98**	**72.38**	**78.89**	**87.54**	76.14	**78.74**	**75.63**

表12。 MMEB-V3 上的阶段、查询侧和合并消融实验。M 单元格按之前 → 之后的格式书写。T1 视频使用 8 帧。

ID	变体	V3	图像	视频	VisDoc	Text	智能体
T1	第2阶段结束	44.38	39.33	45.17	83.59	45.74	36.73
T2	第3阶段结束	58.14	75.67	67.25	82.71	50.77	50.85
T3	最终模型	60.06	78.09	69.62	84.02	51.77	54.03
Q1	仅门控	59.85	78.02	69.65	83.87	51.71	53.35
Q2	均匀MaxSim	59.71	78.04	69.37	83.80	51.64	52.99
M1	单一变体 → 线性平均	58.37→58.80	76.34→76.97	67.90→67.81	82.93→83.25	50.96→51.27	50.65→51.41
M2	单轮 → 跨轮	59.16→59.29	77.15→77.59	68.74→68.76	83.69→83.83	51.24→51.30	52.25→52.31
M3	主干模型 → 四个专家模型	59.44→59.67	77.84→78.11	68.95→69.31	83.98→83.66	51.48→51.67	52.34→52.83
M4	共享系数 → 每块	59.60→59.66	77.95→77.98	69.24→69.36	83.84→83.62	51.56→51.65	52.72→52.94

（52.52 和 50.28）。

在表 11 中，FORB、Street2Shop 和 MovingFashion 上的 Top-1 准确率分别为 89.94、72.16 和 75.98。Street2Shop 和 MovingFashion 将在不同场景中拍摄的产品照片配对，因此匹配的是产品在不同场景中的局部部分。OmniEVIE 在四个视频数据集上的平均准确率为 78.74。将原生视频标记替换为对每一帧进行独立编码后，该平均准确率降至 71.13，正向与反向准确率也从 75.63% 降至 51.17%。此时，正向和反向片段会产生相同的一组向量，准确率降至随机水平。

8.4 消融实验

表12的 M 行中，每个单元格分别列出合并前和合并后的分数。T1 中的视频编号使用 8 帧。

阶段 3 是图像、文本和智能体发生变化的阶段。加入分类、问答和工具后，图像的分数从 39.33 提升至 75.67，文本从 45.74 提升至 50.77，智能体从 36.73 提升至 50.85，而视觉文档则保持在 83 左右。与均匀 MaxSim 相比，门控使智能体的分数提升 0.36，视频提升 0.28；随后，噪声底又使智能体提升 0.68，视觉文档提升 0.15。工具和记忆候选项是长文档，少数查询词元决定了分数；放大高于噪声底的部分，能提高这些词元所占的比重。四个合并步骤均通过参数相加实现。对从同一起点开始的全数据运行取平均，使智能体提升 0.76，图像提升 0.63。跨轮次合并使图像提升 0.44。四个专家模型使智能体提升 0.49，视频提升 0.36，图像提升 0.27，而视觉文档的变化为 -0.32。与使用一个共享系数相比，使用逐块系数使智能体提升 0.22，视频提升 0.12，正如方程 15 所预测的那样，而视觉文档的变化为 -0.22。

8.5 相同预算下的压缩方法

在相同的 128 个 64 维 int8 向量预算下，ST-HAC 得分为 62.96，完整序列得分为 63.09，均匀池化得分为 56.50，随机标记保留得分为 60.41。两项消融实验与 ST-HAC 的差距均在 0.06 以内：$w_s = 0$ 从聚类特征中移除空间项，仅根据语义和时间进行分组

表 13。 在 $K = 128$、$d = 64$ 下的压缩，按 ViDoRe V3 nDCG@10 评分。索引时间为单线程每页毫秒数。

方法	训练	nDCG@10	毫秒/记录
完整序列	–	63.09	–
均匀网格池	在索引构建时	56.50	2.3
保留 K 随机词元	在索引构建时	60.41	0.1
ST-HAC	在训练时	62.96	20.0
ST-HAC，$w_s = 0$	经过训练	62.90	20.1
ST-HAC，原始均值	经过训练	63.00	20.0

表14。 商业场景检索，Top-1，在每个场景内取均值。场景名称已泛化。粗体表示该行中的最佳值。

场景	Qwen3-VL-Emb-8B	WeMM-Emb-9B	OmniEVIE-4B
文本到图像（均值）	54.6	61.2	**64.2**
图像到图像（均值）	93.4	94.1	**96.5**
图像-文本（均值）	81.2	84.1	**90.4**
金融，文本到图像	**76.3**	66.4	**76.3**
工业，文本到图像	61.8	70.9	**81.8**
博物馆，文本到图像	53.2	67.5	**79.2**
地标，图像到图像	78.6	81.4	**92.9**
汽车，图像-文本	86.4	90.9	**100.0**

（62.90），而原始均值变体跳过了公式 20 中的质心重新归一化，并保留簇均值的原始范数（63.00）。构建索引约需单线程每页 20.0 毫秒。记录会在进程之间拆分：一亿个页面约需 2×10^6 单线程秒数，即在 64 个进程上约需 9 小时，并且会直接追加新记录。

8.6 业务场景

除公开基准测试外，我们还在真实业务场景（金融、工业、博物馆、地标和汽车）中进行评估，涵盖文本到图像、图像到图像以及图像-文本联合检索；场景名称经过泛化处理，并不对应特定数据源。表 14 报告了两种强大的业务基线——Qwen3-VL-Embedding-8B 和 WeMM-Embedding-9B——在 Top-1 指标上的表现。

在类别均值方面，OmniEVIE 在三项任务中均居首：文本到图像为 64.2，对比 61.2；图像到图像为 96.5，对比 94.1；图像-文本为 90.4，对比 84.1。在具有代表性的场景中，文本到图像任务里的金融、工业和博物馆分别达到 76.3、81.8 和 79.2，较 9B基线分别高出 9.9、10.9 和 11.7 个百分点；图像到图像任务里的地标达到 92.9，高出 11.5 个百分点；图像-文本任务里的汽车达到 100.0，高出 9.1 个百分点。当匹配需要依赖布局细节或局部产品特征时，提升幅度最大，这与上述公开数据集结果一致。

9 分析

9.1 字节预算中的两个自由度

在字节数固定的情况下，索引可以容纳更多向量K，或者采用更高的维度d。表15 对这两种选择进行了比较。前几行展示了各个宽度下完整的 Prefix-MRL序列。最后一组比较保持

表15. Prefix-MRL ViDoRe V3 上的宽度和 ST-HAC 存储预算。字节数包括 ST-HAC 存储的均值向量 s 每百万条记录的存储量。记录。

设置	(K, d)	字节/条记录。	GiB / M	ViDoRe V3
Prefix-MRL，完整序列	$\times 2048$	4,106,184	3,824	64.34
Prefix-MRL，完整	序列。$\times 512$	1,026,546	956	64.12
Prefix-MRL，完整	序列。$\times 128$	256,636	239	63.55
Prefix-MRL，完整	序列。$\times 64$	128,318	120	63.09
ST-HAC	$\leq 128 \times 64$	8,448	7.87	62.96
ST-HAC	$\leq 64 \times 64$	4,352	4.05	62.47
ST-HAC	$\leq 32 \times 128$	4,608	4.29	61.82
ST-HAC	$\leq 32 \times 64$	2,304	2.15	61.16

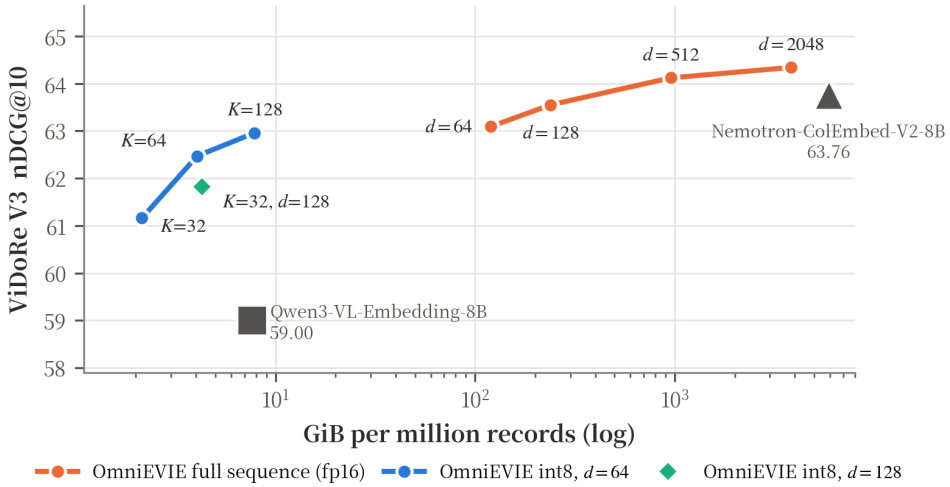

图 7. 在对数坐标下，每百万条记录的存储开销与 ViDoRe V3 准确率的关系，比较了 Prefix-MRL 宽度和 ST-HAC 预算。灰色方块和三角形分别代表单向量基线 Qwen3-VL-Embedding-8B 和多向量基线 Nemotron-ColEmbed-V2-8B，其位置对应各自的索引存储开销（后者采用 4096 维度，每页约 773 个词元；以 fp16 存储时，每百万条记录约占 5,898 GiB）；标记大小反映了约 8B 的模型尺度，高于 OmniEVIE 的 4.5B。

多向量载荷为 4,096 字节，并考察 $K = 64, d = 64$ 是否优于 $K = 32, d = 128$。第 1 节指出，数量决定一次可以匹配多少个局部片段，而维度决定单个向量区分相似项的能力。证据越分散，数量就越重要。

如图 7 和表 15所示，将完整序列的维度从 2048 降至 512、128 和 64，得分分别为 64.34、64.12、63.55 和 63.09。 32× 存储开销减少的同时，得分仅损失 1.25 分，因此即使向量窄得多，大部分完整序列得分仍得以保留。载荷为 4,096 字节时，$K = 64, d = 64$ 得分为 62.47，而$K = 32, d = 128$ 得分为 61.82。将字节用于增加向量数量，比用于提高维度更划算。

在 MMEB 上，相同的比较使用完整$d = 2048$ 序列，即图1中的蓝线，作为附录B （表16）中的190任务均值。发布索引在 V3 上的分数为 60.06，比 60.68 的上限低 0.62；在 V2 上的分数为 77.96，比 78.56 低 0.60。截取为$d = 64$ 的完整序列在 V3 上的分数为 59.02，低于发布索引。

9.2 Int8缩放因子

表17 比较了第6 节中采用的固定尺度在 ViDoRe V3 上的表现。

表 16. F　完整序列上限和 MMEB 上的发布索引。V3 和 V2 使用附录中的均值B　.

表征	V3	V2 图像	视频	VisDoc		文本	智能体
完整序列，$d = 2048$	**60.68**	**78.56**	**78.70**	**70.08**	**84.70**	**52.97**	**54.14**
完整序列，$d = 64$	59.02	77.89	78.06	69.52	83.90	52.50	49.17
$K128 \times d64$ int8	59.71	77.81	78.04	69.37	83.80	51.64	52.99
发布索引	60.06	77.96	78.09	69.62	84.02	51.77	54.03

表 17. ViDoRe V3 上的 Int8量化，分级 nDCG@10。

方案	额外字节数／向量	比较	追加	nDCG@10
fp16质心	+64 B	浮点数	直接	62.99
int8 + fp16 尺度	+2 B	浮点数	直接	62.96
int8，省略缩放因子	0	整数	直接	57.82
语料库最大值/127	0	整数	重新扫描	62.97
固定缩放比例 1/127	0	整数	直接	62.96

表18. 单线程检索延迟，单位为毫秒。单元格中的数值为总计。 Δ 是文档池和视频池上的 nDCG@10 变化。

流水线	N_1	N_2	100k	1M	5M	Δ 文档 / 视频
全量 int8 MaxSim	all	–	1,657	12,781	63,892	0 / 0
单向量 $d = 2048$	all	–	34.6	331.0	1,666	–
单向量 $d = 4096$	all	–	60.9	604.0	3,039	–
S1 +S3	2,000	–	41.8	55.2	128.5	−0.60 / −0.66
S1 +S3	5,000	–	103.9	115.1	186.5	−0.13 / −0.10
三阶段	5,000	200	24.6	38.4	110.2	−0.20 / −3.09
三阶段	5,000	500	30.6	44.2	116.6	−0.13 / −0.84
三阶段	5,000	1,000	41.9	54.2	120.2	−0.13 / −0.26
三阶段	10,000	1,000	62.3	75.7	127.7	0.00 / −0.35
浮点数 × 1 位	5,000	100	72.7	85.8	156.7	−0.21 / −1.05

固定尺度的分数为 62.96，与 fp16质心（62.99）、逐向量fp16缩放因子（62.96）和全语料库缩放因子（62.97）相差不超过 0.03。去掉尺度后，分数降至 57.82。

9.3 延迟

表 18 报告了三种语料库规模下的单线程CPU检索时间。工作点为 $N_1 = 5{,}000$ 和 $N_2 = 1{,}000$。增量参数和前十保留率是相对于 ViDoRe V3 语料库中包含 19,252 个页面的全量 int8 MaxSim，以及包含 13,932 个片段的视频池进行测量的。表 19 展示了百万级语料库中查询长度的变化。

工作点（$N_1 = 5{,}000$，$N_2 = 1{,}000$）的前十保留率在文档上为 99.5%，在视频上为 98.2%。在 $N_2 = 200$ 处，相同窗口的保留率分别为 98.5% 和 85.7%。在 $N_2 = 500$ 处，保留率分别为 99.4% 和 94.9%。在 $N_1 = 10{,}000$ 处，保留率分别为 99.9% 和 97.7%。$N_1 = 2{,}000$ 和 $N_1 = 5{,}000$ 处的两阶段行分别保留 97.8%/97.8% 和 99.5%/99.6%。非对称浮点数×1位行的保留率为 98.6% 和 89.8%。

在百万条记录上，工作点耗时 54.2 毫秒，其中 S1、S2 和 S3 分别耗时 14.6、19.7 和 19.9 毫秒。完整的 int8MaxSim 耗时 12,781 毫秒，因此三阶段路径快了 236× 倍；文档上的 nDCG@10 变化了 −0.13，视频上的变化了 −0.26。在 10 万条记录上，相同阶段的耗时分别为 1.5 / 19.6 / 20.8 毫秒。在 500 万条记录上，S1 增至 80.5 毫秒，因为它会扫描整个语料库，总耗时为 120.2 毫秒。

S1 是一次 64 维扫描，因此当语料库规模较大时，同样的比较也表明 OmniEVIE 优于精确的高维单向量方案。在 10 万条记录时，$d = 2048$ 和 $d = 4096$耗时分别为 34.6 毫秒和

表19。 在100万条记录上，$N_1 = 5{,}000$ 和 $N_2 = 1{,}000$ 时，按查询长度计的延迟（单位：毫秒）。加速比以全量 int8 MaxSim 为基准。

查询	L_q	S1	S2	S3	总计
短文本	20–40	14.8	14.6	17.3	46.7 (274×)
图像或图像-文本	90–270	14.8	91.6	49.8	156.2 (82×)
长上下文或整页	150–500+	14.8	321.8	103.3	439.9 (29×)

60.9 毫秒，接近三阶段总耗时。在 100 万条记录时，两者耗时分别为 331.0 毫秒和 604.0 毫秒，而三阶段方案为 54.2 毫秒（6.1× 和 11.1×）。在 500 万条记录时，两者耗时分别为 1,666 毫秒和 3,039 毫秒，而三阶段方案为 120.2 毫秒（13.9× 和 25.3×）。延迟相近的两阶段流水线方案 $N_1 = 2{,}000$ 耗时 55.2 毫秒，但归一化折损累计增益降低了 0.60 和 0.66。准确率相近的两阶段流水线方案 $N_1 = 5{,}000$ 耗时 115.1 毫秒。S1 耗时不受查询长度影响。S2 和 S3 则会随查询长度增加。长度约为 400 个词元的查询耗时 439.9 毫秒，仍比全量 int8 MaxSim 快 29× 。

9.4 词元对齐与门控权重

单向量会返回一个标量。多向量索引会显示每个词元命中了候选项的哪一部分。图 8 使用了一个来自 ViDoRe V3 的医学查询：“FDA 的《2018年战略政策路线图》是否取消了生物类似药审批流程中的临床试验要求？”词元 Strategic、Policy 和 Road 都命中了粗体标题中匹配词所在的簇。这些簇包含 3 到 6 个图像块，最大相似度高于 0.94。门控将这三个词元的权重提升至均匀权重之上，并将最后 10 个扩展词元的权重降至均匀权重之下。在相关页面上，每个词元都高于噪声底，均值为 0.82。在同一领域的无关页面上，均值为 0.43，约有一半词元低于噪声底。两个页面上都出现的词，例如 FDA，仍然会匹配。公式 (9) 会放大高于 0.38 的部分，而低于噪声底的词元则保持原值。

9.5 证据定位与视频时间定位

同一个 argmax 会将命中定位到页面区域或视频时间（第 7.5 节）。在图9(a)(b)(c) 中，三个 ViDoRe V3 查询都在 S3 排名第一的页面上命中了答案：Denmark 命中 DK行，contracts 命中列标题；(b) 中的页面为法语页面，percentage 命中“3 %”，photovoltaic 则在同一句中命中“photovoltaïque totale”；(c) 中，2014 命中柱状图的 2014 列，Therapy 命中图表标题。

图 9(d) 展示了一段 150 s 的 QVHighlights 视频 [40], , 编码为 4 个窗口。66–90 s 的标注片段跨越了 75 s 处的窗口边界。Couch 在边界两侧的分数分别为 0.77 和 0.69，hosts 的分数分别为 0.74 和 0.57，均高于噪声底，因此公式 25 会将这些分数相加；news 则在 9–38 s 命中另一个镜头。热度峰值位于 77 s，返回时间跨度为 75–94 s，与标注重叠，并超出其范围 4 s。在全部 1,083 个查询中，峰值有 78.0% 落在标注范围内，时间跨度有 94.3% 与标注相交。时间跨度平均为 27.7 s，占视频的 18%；相比之下，同等长度的随机时间跨度占 44.9%，随机点占 24.0%（图 9(e)）。

全局向量返回整个视频或整个页面，因此，即使问题询问的是哪一秒或页面的哪一部分，仍会将整个视频或页面发送给下游生成器。OmniEVIE 的命中结果携带其位置：对于页面或图像，是一个边界矩形；对于视频，则是公式 25 所返回的时间跨度。下游生成器只读取该矩形或该时间跨度；图 9（a）传递 DK 所在的行，图 9（d）传递 75–94 秒。视频侧同样不需要视频解析模型。常见的长视频处理流程首先运行视频理解模型，将每个片段转录为带时间戳的字幕，然后在文本上进行检索。OmniEVIE 直接对采样帧进行编码，并获取每个质心的时间区间的

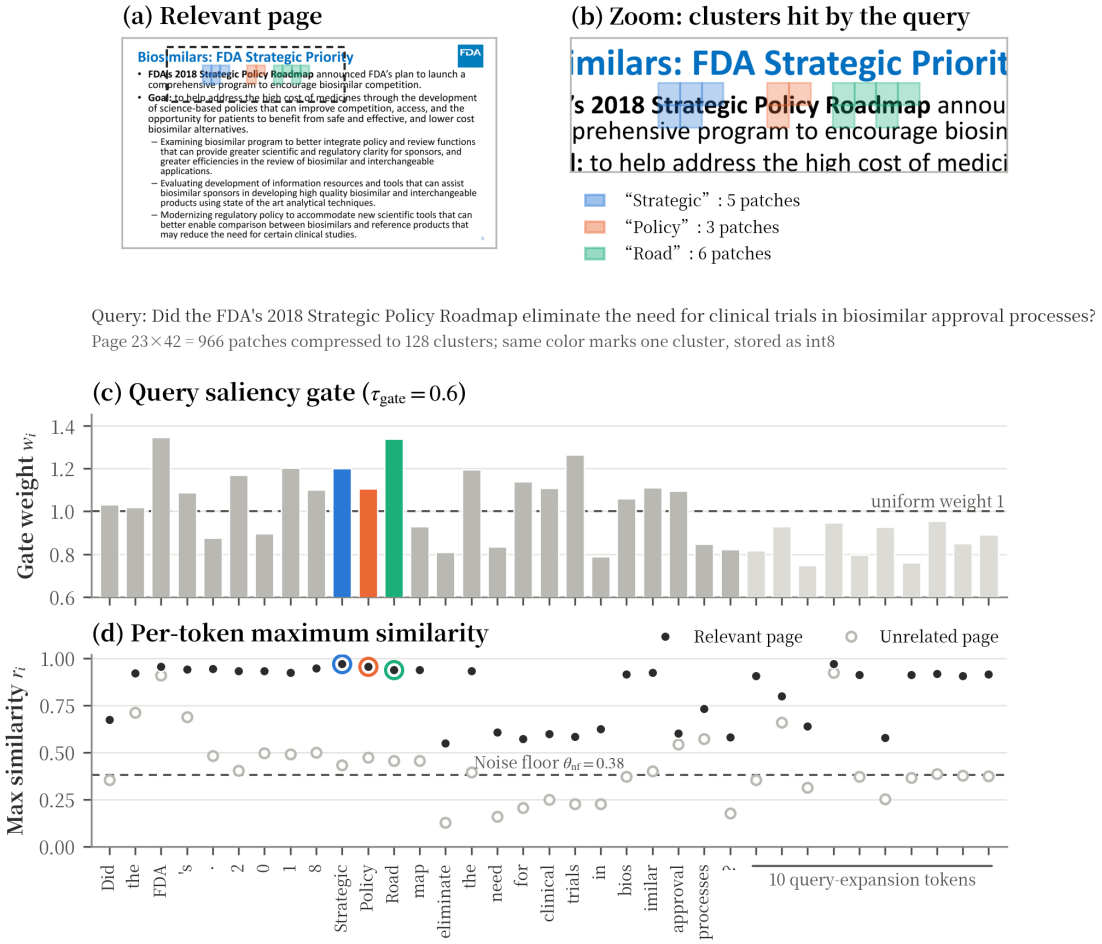

图8。 逐词命中与查询侧门控权重。(a)(b) ViDoRe V3 中医疗文档页面上被三个查询词命中的簇的成员图像块，每个簇分别用不同颜色标示。(c) 显著性门控给出的逐词权重，虚线表示均匀权重。(d) 相关页面与无关的同领域页面上的逐词最大相似度，虚线标示噪声底 $\theta_{\text{nf}} = 0.38$。

帧索引，因此索引构建和定位都无需经过转录；这是一条原生的视频检索路径。长视频以固定窗口存储，每个窗口对应一条记录，公式 25返回匹配片段在完整视频时间线上的起点和终点。

图 10 展示了三个 QVHighlights 查询，每行中间的帧取自返回时间跨度，侧边帧则取自跨度之外。(a) 合上装满物品的行李箱，标注时间为 24–40 秒，时间跨度为 23–38 秒，峰值位于 26 秒，此时她正把衣服压进箱子；跨度之外的画面包括包的特写、她面对镜头的画面、她的脸部特写，以及车内两人的画面。(b) 展示柜中的玩偶，标注时间和时间跨度均为 84–98 秒，峰值位于 87 秒；陈列柜占据整个时间跨度，侧边画面则包括宴会厅、堆叠的物品、圆形地面图案的俯视图，以及女子的自拍照。(c) 一名身穿蓝色衬衫的男子驾驶黄色汽车，标注时间为 118–136 秒，时间跨度为 117–136 秒，峰值位于 120 秒；跨度内的画面从后座拍摄驾驶员，侧边画面则是同一名女子的自拍照。

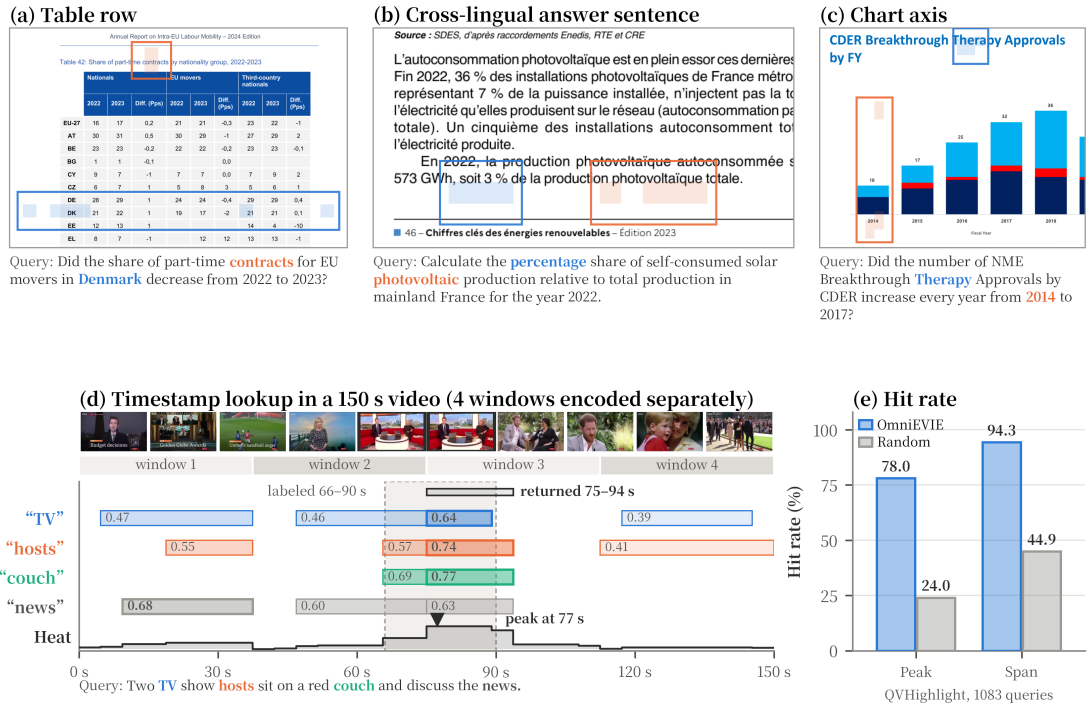

图9。局部化证据。（a）（b）（c）在排名最高的页面上命中的簇：一种颜色表示一个簇，框为其边界矩形，粗体查询词与这些框相匹配。（d）150 秒视频中的时间戳查找：条形表示每个窗口中每个查询词的最佳质心，其阴影深浅表示相对于噪声底的裕度；曲线表示公式 25 的热度，色带表示标注片段，带轮廓的条形表示返回时间跨度。（e）峰值和返回时间跨度的命中率，与相同长度的随机时间跨度进行对比。

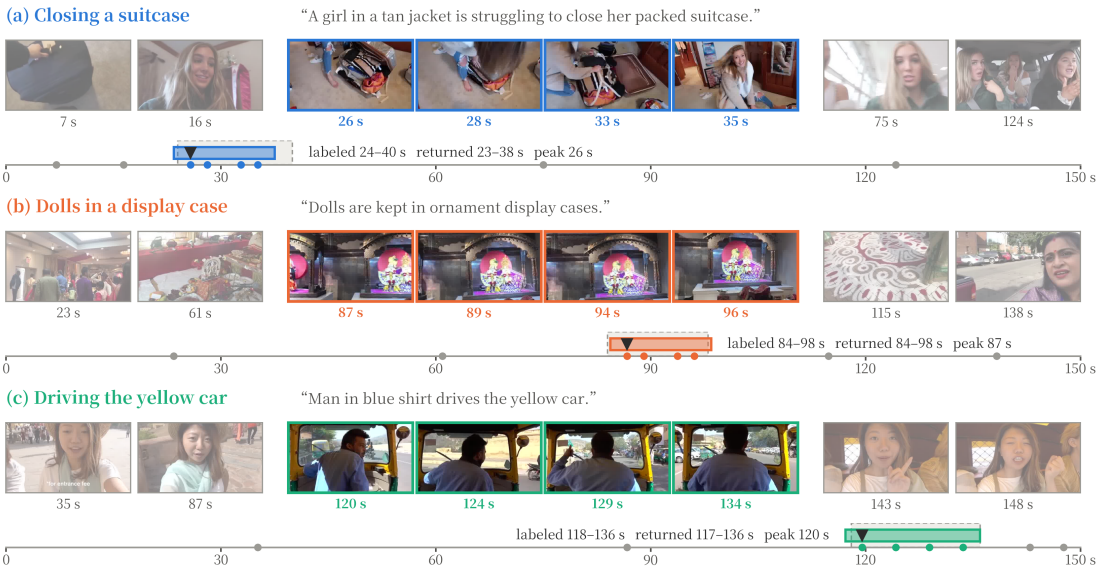

图10。三个 QVHighlights 查询的返回时间跨度。带轮廓的中间帧来自返回时间跨度，灰色侧边帧来自其外部；帧下方的数字表示其时间，点标记这些帧。条形表示返回时间跨度，虚线框表示标注片段，三角形表示热度峰值。（a）关闭手提箱。（b）展示柜中的玩偶。（c）驾驶黄色汽车。

9.6 局限性

一个只有几秒、以低帧率采样的视频片段，在步幅为 2 的情况下会被压缩成少数几个时间组，因此动作信息所剩无几。帧率可以根据视频片段的长度来选择。推理检索、图形用户界面和记忆仍然是文本和智能体子组中表现最差的类别；它们需要多步推理检索数据和界面截图。门控和噪声底在 S3 内部计算，因此其开销会随 L_{η} 线性增长。

10 结论

OmniEVIE 通过一个 4.5B 模型和一个索引，处理文本、图像、文档、视频和工具。MMEB-V3 得分为 60.06，比相同 190 个任务上完整 $d = 2048$ 序列的 60.68 低 0.62。每条记录最多占 9,472 字节，每百万条记录约占 8.8 GiB。对于一百万条和五百万条记录，单线程耗时分别为 54.2 毫秒和 120.2 毫秒，比对 4096 维向量进行精确搜索快 11.1× 倍和 25.3× 倍。搜索使用压缩索引。命中结果指向一个页面区域或一个视频区间。

训练和搜索使用相同的分数。证据等级决定哪些负例属于假负例。训练期间会将候选项聚类为最终存储的向量。单独训练的专家模型按参数块合并，每个参数块都有自己的系数。在线运行时，分数采用内积、人口计数或 int8 点积计算。

在一项任务中需要多次检索的智能体，如果每次命中都已标明对应证据，就能从中受益：这样一来，生成器便可读取该区域或时间段（图 9）。本次发布在公开数据上测试了架构、计划和压缩索引：共 57 个数据集和约 200 万个样本。下一阶段将加入领域数据，将同一记录扩展至音频和长篇混合文档，并把定位到的区域或时间段传递给生成器。在更大规模下，均值向量本身也可以量化或置于图索引中。

权重地址：https://huggingface.co/tencent/OmniEVIE-4B。推理、索引构建和三阶段检索器位于 https://github.com/Tencent/OmniEVIE。语料库包括公开基准测试，以及获得许可的文档、图像和视频集合。证据检查会过滤敏感材料和个人身份信息。超参数和评测套件列于附录 A和附录 B。

参考文献

[1] Ziyan Jiang、Rui Meng、Xinyi Yang、Semih Yavuz、Yingbo Zhou 和 Wenhu Chen。VL M2Vec：为大规模多模态嵌入任务训练视觉语言模型。*arXiv:2410.05160*，2024年。[2] Rui Meng、Ziyan Jiang、Ye Liu、Mingyi Su、Xinyi Yang、Yuepeng Fu、Can Qin、Zeyuan Chen、Ran Xu、Caiming Xiong、Yingbo Zhou、Wenhu Chen 和 Semih Yavuz。VLM2Vec-V2：推进视频、图像和视觉文档的多模态嵌入。*arXiv:2507.04590*，2025年。[3] Haohang Huang、Xuan Lu、Mingyi Su、Xuan Zhang、Ziyan Jiang、Ping Nie、Kai Zou、Tomas Pfister、Wenhu Chen、Wei Zhang、Xiaoyu Shen 和 Rui Meng。MMEB-V3：衡量全模态嵌入模型的性能差距。*arXiv:2604.23321*，2026年。[4] Orion Weller、Michael Boratko、Iftekhar Naim 和 Jinhyuk Lee。论基于嵌入的检索的理论局限性。载于国际学习表征会议，2026年。[5] Omar Khattab 和 Matei Zaharia。Colbert：通过 BERT 上的上下文化晚期交互实现高效且有效的段落检索。载于第 *43* 届国际 *ACM SIGIR* 信息检索研究与发展会议论文集，第 39–48 页，2020年。

[6] Manuel Faysse、Hugues Sibille、Tony Wu、Bilel Omrani、Gautier Viaud、Céline Hudelot 和 Pierre Colombo。ColPali：利用视觉语言模型实现高效的文档检索。载于国际学习表征会议，2025年。[7] Gabriel de Souza P. Moreira、Ronay Ak、Mengyao Xu、Oliver Holworthy、Benedikt Schifferer、Zhiding Yu、Yauhen Babakhin、Radek Osmulski、Jiarui Cai、Ryan Chesler、Bo Liu 和 Even Oldridge。Nemotron ColEmbed V2：用于视觉文档检索的顶尖晚交互嵌入模型。*arXiv:2602.03992*，2026年。[8] Arun Reddy、Alexander Martin、Eugene Yang、Andrew Yates、Kate Sanders、Kenton Murray、Reno Kriz、Celso M. de Melo、Benjamin Van Durme 和 Rama Chellappa。Video-ColBERT：用于文本到视频检索的上下文化晚期交互。*arXiv:2503.19009*，2025年。[9] David Wan、Han Wang、Elias Stengel-Eskin、Jaemin Cho 和 Mohit Bansal。CLAMR：用于多模态内容检索的上下文化晚期交互。*arXiv:2506.06144*，2025年。[10] Mingxin Li、Yanzhao Zhang、Dingkun Long、Keqin Chen、Sibo Song、Shuai Bai、Zhibo Yang、Pengjun Xie、An Yang、Dayiheng Liu、Jingren Zhou 和 Junyang Lin。Qwen3-VL-Embedding 和 Qwen3-VL-Reranker：用于先进多模态检索与排序的统一框架。*arXiv:2601.04720*，2026年。[11] Hanxiang Qin、Alexander Martin、Rohan Jha、Chunsheng Zuo、Reno Kriz 和 Benjamin Van Durme。任意模态下的多向量索引压缩。*arXiv:2602.21202*，2026年。[12] Zifei Wang、Wei Wen、Qiang Ji、Qian-Wen Zhang、Ruizhi Qiao 和 Xing Sun。EVIE：用于视觉文档检索的证据向量知情嵌入。*arXiv:2610.11553*，2026年。[13] Reza Esfandiarpoor、Radek Osmulski、Yauhen Babakhin、Gabriel de Souza P. Moreira、Oliver Holworthy、Jie He、Ronay Ak、Jiarui Cai、Ryan Chesler、Bo Liu 和 Even Oldridge。超越语义相似度：复杂任务中智能体检索的性能与成本。*arXiv:2610.05750*，2026年。[14] Junjie Zhou、Ke Mei、Lei Li、Tianyi Wang、Fengyun Rao 和 Jing Lyu。WeMM-Embedding：微信多模态嵌入技术报告。*arXiv:2608.24053*，2026年。[15] Ovis-Embedding 团队。Ovis-Embedding：推动通用全模态嵌入的前沿。*arXiv:2609.25165*，2026年。[16] Xiaojie Li、Chu Li、Shi-Zhe Chen 和 Xi Chen。U-MARVEL：通过利用多模态大语言模型进行嵌入学习，揭示通用多模态检索的关键因素。载于国际学习表征会议，2026年。[17] Michael Günther、Saba Sturua、Mohammad Kalim Akram、Isabelle Mohr、Andrei Ungureanu、Bo Wang、Sedigheh Eslami、Scott Martens、Maximilian Werk、Nan Wang 和 Han Xiao。jina-embeddings-v4：用于多模态多语言检索的通用嵌入。*arXiv:2506.18902*，2025年。[18] Florian Hönicke、Michael Günther、Andreas Koukounas、Mohammad Kalim Akram、Scott Martens、Saba Sturua 和 Han Xiao。jina-embeddings-v5-omni：通过锁定对齐塔实现保持几何结构的嵌入。*arXiv:2605.08384*，2026年。[19] Keshav Santhanam、Omar Khattab、Jon Saad-Falcon、Christopher Potts 和 Matei Zaharia。ColBERTv2：通过轻量级晚交互实现高效有效的检索。载于*2022*年美国计算语言学协会北美分会：人类语言技术会议论文集，第 3715–3734 页，2022年。[20] António Loison、Quentin Macé、Antoine Edy、Victor Xing、Tom Balough、Gabriel Moreira、Bo Liu、Manuel Faysse、Céline Hudelot 和 Gautier Viaud。ViDoRe V3：对复杂现实场景中的检索增强生成进行全面评估。*arXiv:2601.08620*，2026年。[21] Rohan Jha、Bo Wang、Michael Günther、Georgios Mastrapas、Saba Sturua、Isabelle Mohr、Andreas Koukounas、Mohammad Kalim Akram、Nan Wang 和 Han Xiao。Jina-ColBERT-v2：通用多语言晚交互检索器。*arXiv:2408.16672*，2024年。[22] Paul Teiletche、Quentin Macé、Max Conti、António Loison、Gautier Viaud、Pierre Colombo 和 Manuel Faysse。ModernVBERT：迈向更小型的视觉文档检索器。*arXiv:2510.01149*，2025年。[23] Aditya Kusupati、Gantavya Bhatt、Aniket Rege、Matthew Wallingford、Aditya Sinha、Vivek Ramanujan、William Howard-Snyder、Kaifeng Chen、Sham Kakade、Prateek Jain 和 Ali Farhadi。套娃式表征学习。载于神经信息处理系统进展，第 35 卷，第 30233–30249 页，2022年。

[24] Devvrit、Sneha Kudugunta、Aditya Kusupati、Tim Dettmers、Kaifeng Chen、Inderjit Dhillon、Yulia Tsvetkov、Hannaneh Hajishirzi、Sham Kakade、Ali Farhadi 和 Prateek Jain。MatFormer：用于弹性推理的嵌套式 Transformer。*arXiv:2310.07707*，2023年。[25] Zilin Xiao、Qi Ma、Mengting Gu、Chun-cheng Jason Chen、Xintao Chen、Vicente Ordonez 和 Vijai Mohan。MetaEmbed：利用灵活的后期交互在测试时扩展多模态检索。发表于国际学习表征会议，2026年。[26] Laxman Dhulipala、Majid Hadian、Rajesh Jayaram、Jason Lee 和 Vahab Mirrokni。MUVERA：通过固定维度编码实现多向量检索。*arXiv:2405.19504*，2024年。[27] Keshav Santhanam、Omar Khattab、Christopher Potts 和 Matei Zaharia。PLAID：用于后期交互检索的高效引擎。*arXiv:2205.09707*，2022年。[28] PS Rishi、Rajeev Ranjan Dwivedi 和 Vinod K. Kurmi。面向查询的词元预算分配：实现高效的后期交互视觉文档检索。*arXiv:2609.07262*，2026年。[29] Daniel Bolya、Cheng-Yang Fu、Xiaoliang Dai、Peizhao Zhang、Christoph Feichtenhofer 和 Judy Hoffman。令牌合并：让你的视觉Transformer运行得更快。发表于 国际学习表征会议，2023年。[30] Ailar Mahdizadeh、Aria Salari、Sohail Rajabi、Shahriar Mirabbasi、Panos Nasiopoulos 和 Alireza Morsali。覆盖范围很重要：用于压缩多向量视觉文档检索器的 MarginMerge。*arXiv:2608.02969*，2026年。[31] Yibo Yan、Mingdong Ou、Yi Cao、Xin Zou、Jiahao Huo、Shuliang Liu、James Kwok 和 Xuming Hu。雕刻向量空间：通过“先剪枝后合并”框架实现高效的多向量视觉文档检索。*arXiv:2602.19549*，2026年。[32] Myunsoo Kim、Seongwoong Shim 和 Byung-Jun Lee。FALCON：在视觉语言对齐中进行假阴性感知的对比负样本学习。*arXiv:2505.11192*，2025年。[33] Tingyu Song、Mingxin Li、Yanzhao Zhang、Dingkun Long、Chu Liu、Pengjun Xie、Yilun Zhao 和 Shu Wu。CORE：通过重排序器蒸馏改进多模态大语言模型嵌入中的组合推理。*arXiv:2609.04083*，2026年。[34] Gabriel Ilharco、Marco T. Ribeiro、Mitchell Wortsman、Suchin Gururangan、Ludwig Schmidt、Hannaneh Hajishirzi 和 Ali Farhadi。利用任务算术编辑模型。发表于国际学习表征会议，2023年。[35] Prateek Yadav、Derek Tam、Leshem Choshen、Colin Raffel 和 Mohit Bansal。TIES-Merging：解决模型合并中的干扰问题。发表于神经信息处理系统进展，2023年。[36] Shiyu Li、Zhiyuan Hu、Yifan Wang、Peiming Li、Zheng Wei 和 Yang Tang。Conan-embedding-v3：融合特定模态模型，实现全模态嵌入。*arXiv:2606.09331*，2026年。[37] Zijing Wang、Yongkang Liu、Mingyang Wang、Ercong Nie、Deyuan Chen、Zhengjie Zhao、Shi Feng、Daling Wang、Xiaocui Yang、Yifei Zhang 和 Hinrich Schütze。PlaM：用于提升多模态大语言模型视觉定位能力的免训练平台期引导模型合并。*arXiv:2601.07645*，2026年。[38] Neale Ratzlaff、Man Luo、Xin Su、Vasudev Lal 和 Phillip Howard。多模态指令微调后语言推理能力退化的免训练缓解方法。*arXiv:2412.03467*，2024年。[39] Moses S. Charikar。基于舍入算法的相似度估计技术。发表于第*34*届*ACM*年度计算理论研讨会论文集，第380–388页，2002年。[40] Jie Lei、Tamara L Berg 和 Mohit Bansal。通过自然语言查询检测视频中的时刻与精彩片段。发表于神经信息处理系统进展（*NeurIPS*），2021年。

A 超参数

表20. 发布模型所用的训练和索引设置。

组	设置
骨干网络	采用双向全注意力的 Qwen3.5-VL；$d_{\text{hid}} = 2560$；$W \in \mathbb{R}^{2048\times 2560}$；宽度 $\{64, 128, 256, 512, 1024, 2048\}$。
视频	$\Delta t_{\text{patch}} = 2$。训练时会将不足 12 帧的视频片段填充至 12 帧，上限为 3,600 个词元。评估使用 32 帧和 14,400 个词元，官方比较使用 8 帧和 3,600 个词元。
优化	AdamW，权重衰减为 0.02。LoRA 秩为 64 或 128，$\alpha = 4r$，丢弃率为 0.05。LoRA 的学习率为 2×10^{-5}，以及 4×10^{-6} 在合并器和头部上，余弦衰减至 0。1× 峰值，预热占步数的 5%。全局批次为 512。每个阶段训练一轮。
Loss	$\tau = 0.02, \gamma = 1, \beta = 0.5, \lambda_{\text{ord}} = 0.10$。$\lambda_{\text{kd}}$ 根据早期梯度范数比设定后即冻结。$w_e = 0.80$。
训练时 K	$\{64, 128\}$ 概率分别为 0.3 和 0.7。
查询侧	$\tau_{\text{gate}} = 0.6, \theta_{\text{nf}} = 0.38, \beta_{\text{nf}} = 2.5$.
合并	$c_{k,b} \in [0, 1]$ 每个块。语言层裁剪保留 $\rho = 0.75$；其他块使用 $\rho = 1$。
ST-HAC	视频 $(w_s, w_t) = (0.07, 0.03)$；图像和页面 $(0.10, 0)$；文本 $(0, 0.10)$。$K \leq 128, d = 64$。时间频率在 $[\pi/4, \pi]$.
量化	尺度为 1/127，$[-127, 127]$ 中的整数。记录上限分别为 8,448 字节（两级）和 9,472 字节（三阶段）。
检索	$N_1 = 5{,}000, N_2 = 1{,}000, k = 10$.

B 评测套件

表 21。 第8节中报告的套件。MMEB-V3 将 11 个音频任务的分数计为零，并将其纳入分母。智能体是工具 + 图形用户界面 + 记忆，其中 MCMR 在表8中单独报告。

套件	覆盖范围	任务指标
MMEB-V3	图像、视频、视觉文档、文本、工具、图形用户界面、记忆、多跳	190 Hit@1；文本上的 nDCG@5 和视觉文档
MMEB-V2	图像 (36)、视频 (18)、视觉文档 (24)	图像和视频上的 78 Hit@1
		视频上的 nDCG@5；sual 文档
ViDoRe V3	8 个领域，6 种语言	48 个分级 nDCG@10
细粒度	FORB、Street2Shop，MovingFashion；MSR-VTT，MSVD、DiDeMo、VATEX；ActivityNet 顺序	8 Top-1/ nDCG@10/ 胜场 rate

细粒度套件中的视频以每秒 1 帧编码，最多 32 帧，作为一个原生序列。