arXiv:2602.00946v1 [cs.CV] 1 Feb 2026

ConsensusDrop：融合视觉与跨模态显著性，实现高效视觉语言模型

Dhruv Parikh[1] Haoyang Fan[1] Rajgopal Kannan[2] Viktor Prasanna[1]

[1]南加州大学{dhruvash, haoyangf, prasanna}@usc.edu[2]DEVCOM陆军研究办公室 rajgopal.kannan.civ@army.mil

摘要

视觉–语言模型（*VLM*）计算开销高昂，因为大语言模型需要处理数百个大多冗余的视觉词元。现有词元缩减方法通常利用视觉编码器显著性（覆盖面广但与查询无关）或大语言模型交叉注意力（能感知查询但稀疏且计算开销大）。我们证明，单独使用任一信号都不足够：与单模态视觉词元选择（排名）相比，将两者融合能稳定提升性能。然而，要让这种融合切实可行并非易事：跨模态显著性通常只有在大语言模型内部才能获取（此时已太迟，无法高效地在大语言模型处理前进行剪枝），而且这两种信号本质上并不对称，因此简单融合无法充分发挥它们互补的优势。我们提出***ConsensusDrop***，一种无需训练的框架，通过协调视觉编码器显著性与感知查询的交叉注意力，得出共识排名，保留信息量最大的词元，同时通过编码器引导的词元合并来压缩其余词元。在 *LLaVA-1.5/NeXT*、*Video-LLaVA* 及其他开源视觉语言模型上，在相同词元预算下，*ConsensusDrop* 的表现始终优于先前的剪枝方法，并实现了更优的准确率–效率帕累托前沿——即使在大幅缩减词元的情况下，仍能保持接近基线的准确率，同时降低首令牌生成时间和键值缓存占用。我们的代码将开源。

1. 引言

大语言模型（LLM）的最新进展涵盖了封闭式 [3, 58] 和开源范式， [1, 56, 57, 92], 推动了新一代视觉语言模型（VLM）的发展。这些模[55, 58] 型通过原生方式 [2, 43]或架构扩展实现了复杂的多模态推理，并将能力拓展至文档理解、图表解读和长视频理解等专门领域 [13, 15, 34, 35, 61, 63, 85]。标准视觉语言模型通常采用预训练视觉编码器[64, 87] 将图像映射为图像块词元，随后通过多模态投影器 [42, 43, 46] 将这些嵌入与大语言模型对齐。然而，这种范式会带来显著的计算开销。单张图像通常会被编码为数百甚至数千个词元（例如，LLaVA-1.5 [43], 2880中为 576 个，在 LLaVA-NeXT [46] 中为 ），远多于少于一百个词元的文本输入。这种冗余会导致高得难以接受的预填充延迟，以及解码期间过高的键值缓存消耗，从而阻碍其在机器人学和自动驾驶等对延迟敏感的应用中的规模化部署 [30, 62, 77]。

受视觉词元中大量冗余的启发，许多研究通过对图像块进行排名并丢弃信息量较低的图像块来提高视觉语言模型的效率。这些研究通常遵循两种范式：**纯跨模态剪枝** [11, 18, 23, 50, 67, 73, 81, 82, 86,90, 93] 利用大语言模型内部输入提示与图像令牌之间的文本-视觉注意力为图像块排名，而**纯视觉剪枝** [16, 29, 66, 70, 78, 79, 88, 91] 则根据视觉编码器侧的显著性筛选词元。尽管这两类方法在不同保留预算下均展现出显著的加速比和有竞争力的准确率 [16, 29, 66, 70, 78, 79, 88, 91], , 但它们仍存在不容忽视的局限，影响了实际效果。纯跨模态剪枝存在以下问题：(i) 旋转嵌入带来强烈的位置偏差，导致后索引图像块权重过高；(ii)稀疏且不稳定的跨模态分数可能遗漏视觉显著区域；(iii) 实际效率低下——提取层注意力会禁用 FlashAttention [14] ，并导致非均匀键值缓存，造成内存碎片。相反，纯视觉剪枝仍然与查询无关，会丢弃对问题至关重要的区域，尤其是小型文本条件区域（例如嵌入文本片段，或

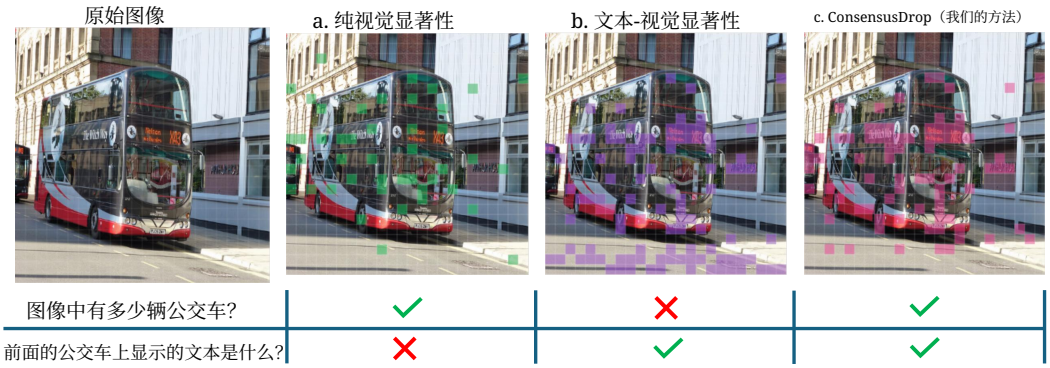


图 1. **动机：视觉词元稀疏化的互补性失效模式**。给定一张图像和两个需要不同视觉证据的问题（全局结构与局部细粒度文本），不同的词元选择标准呈现出系统性局限。仅视觉显著性能够保留广泛的空间覆盖范围和视觉显著区域，但可能会丢弃较小的文本关键区域。文本—视觉显著性能够捕捉查询对齐的局部细节，但通常较为稀疏，且偏向后索引图像块。**ConsensusDrop（我们的方法）**通过融合视觉与跨模态显著性信号，保留互补证据，从而留存回答不同问题所需的信息。

图表元素），这在文档或光学字符识别类任务中尤为明显。此外，视觉编码器表现出注意力汇行为，导致背景得分虚高，代价是损失细粒度的查询相关细节。另外，两类方法中的若干方法都需要额外的视觉语言模型训练 [7, 10, 25,37, 39, 66, 89], ，这会增加训练开销。

我们假设，视觉显著性与跨模态显著性互为补充，且两者对于稳健的视觉令牌选择都至关重要（图 1）。我们的对照实验证实了这一假设：与单模态剪枝相比，采用**多模态共识**能得到更可靠的重要性估计，并持续带来更优性能。然而，利用这种共识也带来了两个关键挑战。首先，跨模态显著性通常在语言模型内部计算，这迫使剪枝只能在推理流程后期进行，因而产生额外开销，并可能导致执行效率低下和键值缓存碎片化 [78]。其次，视觉信号与跨模态信号本质上具有不对称性——前者提供广泛的视觉覆盖范围，后者则较为稀疏但针对特定查询——因此，简单融合难以奏效。这引出了一个根本性问题：我们如何协调这些互补但不对称的信号，从而为词元选择得出有原则的共识？

基于这些发现并针对相关局限性，我们提出了**ConsensusDrop**，一种用于高效视觉令牌缩减的无需训练框架。具体而言，我们首先引入**静态交叉注意力探针（SCAP）**，这是一种轻量且无需训练的模块，可在文本–视觉注意力分数进入语言模型之前提取这些分数，从而无需访问内部的大语言模型注意力图。利用 SCAP 和视觉编码器的注意力分数，我们随后设计了一种新颖的**融合器模块**，执行“多模态”分数融合，以识别并保留最具信息量的前-K 个视觉词元。随后，其余词元通过新颖的**编码器引导的令牌合并（EGTM）**模块，被压缩为一个密集的视觉词元子集。实验证明，**ConsensusDrop** 在多种多模态基准测试中实现了最先进的准确率–效率权衡。

通过在大语言模型之前进行剪枝，它保留了 FlashAttention [14]并降低了键值缓存成本。无需训练的设计自然支持向高分辨率和基于视频的模型（例如 LLaVA-NeXT 和 Video-LLaVA）泛化。这些结果表明，多模态共识是进行视觉令牌选择的有效方法。我们的主要贡献如下：

- 我们证明，**多模态共识**——结合视觉显著性与跨模态线索——始终优于单模态视觉词元剪枝。
- 我们提出了**静态交叉注意力探针（SCAP）**，这是一种轻量级的预大语言模型模块，可高效提取查询感知的跨模态显著性。
- 为弥合视觉线索与跨模态线索之间的不对称性，我们提出了**融合器模块**，以生成统一且稳健的视觉词元重要性排名。
- 我们提出了**编码器引导的令牌合并（EGTM）**，在视觉编码器的引导下，于投影器空间中紧凑地合并被丢弃的词元。
- **ConsensusDrop** 无需训练、即插即用，并在多个开源视觉语言模型上实现了最先进的准确率–效率权衡。

2. 相关工作

纯视觉剪枝。 这一范式利用视觉编码器侧的显著性信号为词元打分，在进入大语言模型之前去除冗余。代表性方法包括 [66, 70], [16, 29, 79, 91]。VisionZip [78] 和 Vis-Pruner [88] 通过基于编码器注意力为视觉图像块打分，

并通过上下文合并或基于相似度的去重来压缩被丢弃的词元。这些方法虽然有效且无需训练，但本质上仍与查询无关，因为重要性完全由视觉模态决定。

纯跨模态剪枝。这类方法利用从大语言模型内部提取的文本-视觉交叉注意力对图像块进行排名和剪枝。相关方法包括早期的层级排名 [11, 90]和多阶段或协同聚合 [23, 50, 73, 81]，以及在大语言模型各层之间渐进式移除和动态跳过 [18, 67, 76, 82, 86,93]。尽管这类方法具备查询感知能力，但在大语言模型内部提取信号会禁用 FlashAttention 优化、造成键值缓存碎片化，并产生稀疏且受提示偏置的分数，常常忽略视觉显著区域。

ConsensusDrop 通过在进入大语言模型之前融合查询感知跨模态信号与视觉侧显著性，克服了这些局限，从而实现无需训练、即插即用的词元减少，同时保留推理效率。

3. 动机

3.1. 我们需要视觉和跨模态线索两者吗?

我们假设，单靠视觉编码器或语言模型，都无法可靠地估计视觉词元的重要性。为研究这一点，我们设计了一种受控设置，让两种模态分别对相同的N 图像块进行评分。我们首先评估一致性，即视觉分数与跨模态分数得出的排名前K 个词元之间的重叠程度。为考察一种模态能否纠正另一种模态，我们执行一个简单的恢复过程：丢弃一种模态（学生）排名最低的词元，并用另一种模态（教师）排名最高的候选词元替换。该受控设置将视觉显著性与跨模态显著性的贡献区分开来，并揭示每种模态捕获另一种模态所遗漏信息的频率，从而为我们的方法提供动机。

受控设置表明（第3.3节）：（i）视觉显著性与跨模态显著性产生了不可忽略的排名分歧；（ii）这些信号具有不对称性：视觉显著性提供了更强的基础排序，而跨模态线索主要用于有针对性的纠正。这促使我们采用带有明确视觉偏置的多模态融合方案作为默认设计（第4.2节）。

3.2. 问题形式化

视觉语言模型的形式化定义。形式上，视觉语言模型由一个冻结的视觉编码器 $\mathcal{E}_v$、一个投影器 $\mathcal{P}$ 和一个冻结的大语言模型 $\mathcal{L}$组成。给定一张图像 $\mathcal{I}$ 和查询$\mathbf{T} \in \mathbb{R}^{M\times d}$，图像会被投影为视觉令牌 $\tilde{\mathbf{V}} \in \mathbb{R}^{N\times d}$。模型处理拼接后的序列 $\mathbf{X} = [\mathbf{X}^{\text{sys}}, \mathbf{V}, \mathbf{T}]$。由于 $N \gg M$，我们的目标是在大语言模型推理**之前** 将$\mathbf{V}$ 压缩为一个紧凑的子集。

分数与选择。视觉侧分数 $t_v \in \mathbb{R}^N$ 通过基于CLS的注意力从视觉编码器的倒数第二层自注意力层中提取。跨模态分数$t_c \in \mathbb{R}^N$ 则从大语言模型的指定Transformer层中提取，方法是汇总所有查询token对每个视觉词元的注意力。这遵循了 FastV [11] 中视觉词元剪枝的设置。对于这项受控研究，为确保预算固定，我们定义保留率 ρ 并设定选择数量 $K = \rho N$。对于每个分数向量，我们选取排名前 K 的图像块索引，记为 $I_v(K)$ 和 $I_c(K)$。

一致性与不一致： 我们将一致性 定义为重叠一致性(K) $= |I_v \cap I_c|/K$，用于衡量两种模态在识别重要区域方面的一致程度；不一致率为 $1 -$agreement(K)。

恢复与纠正： 我们通过将 $I_v(K)$ 和 $I_c(K)$ 中的一组视为学- 生 I_s，另一组视为教师 I_t，来执行恢复（第 3.1 节）。对于恢复率r，我们从 $I_s(K)$ 中移除排名最低的 rK 个词元，并用 $I_t(K)$ 中排名最高、且尚未出现在学生保留词元中的 rK 个词元替换。设ℓ 为我们为找到 rK个此类不重叠词元而必须检查的教师排名最深位置；我们将纠正率定义为 $(K - \ell)/(K - rK)$，其取值范围为 $[0, 1]$，反映教师对学生排名的不认同程度。

3.3. 分析

多模态融合带来的准确率提升。我们评估了 LLaVA-1.5-7B 在六个基准测试（图 2）上的准确率。恢复始终能提升准确率，且在所有保留率下皆如此，这证实联合线索对词元重要性的估计优于任一单独信号。值得注意的是，视觉分数通常优于单独使用的跨模态分数。这表明存在一种非对称协同效应：视觉线索提供了基础，对跨模态线索的帮助大于反过来的帮助，这与此前关于纯视觉剪枝的发现一致 [29, 78, 88]。

要点。视觉显著性提供了强有力的基础排序，而跨模态线索主要用于纠正少数关键部分。这启发我们采用一种视觉偏置的共识机制（第4.2节）。

分歧—纠正—准确率机制。图3揭示了一条清晰的因果链：激进剪枝（低保留率）会加剧模态分歧，从而创造有效跨模态纠正的机会，并直接转化为准确率提升。从本质上说，分歧是一种信号，指出跨模态干预最有效的特定“盲点”。

要点。由于性能提升源于对特定分歧的纠正，因此，与其统一应用跨模态线索，不如有选择地将其应用于这些冲突区域（第 4.2节）。

纠正率衰减。在图 4中，纠正率起初

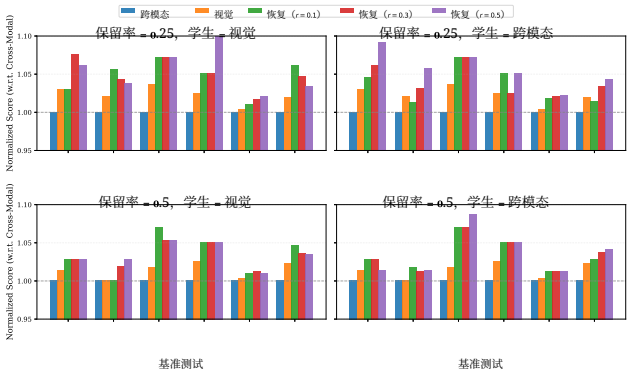


图 2. **跨模态恢复在各项基准测试中始终能提升准确率**。我们使用 LLaVA-1.5-7B，对比了基于 (i) 跨模态分数、(ii) 视觉分数，以及 (iii) 融合两者的恢复机制进行词元选择的效果。在 POPE、MME、MMBench、MMBench-CN、TextVQA 和 GQA 上，恢复机制在所有保留率（0.25 和 0.50）以及两种学生配置（视觉和跨模态）下均取得更高的准确率。这些结果验证了单一模态信号都不足以胜任，而将两者结合起来能够更可靠地估算令牌重要性。

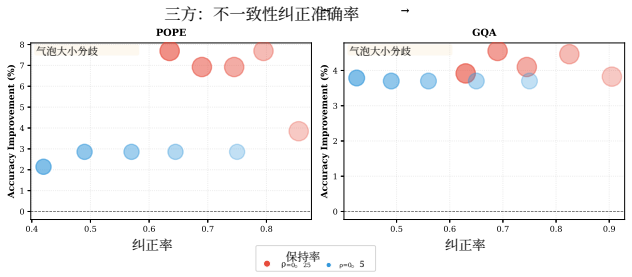


图 3. **模态分歧、纠正率与准确率提升之间的三方关系**。气泡大小表示视觉注意力与跨模态注意力之间的分歧；颜色区分保留率（$\rho = 0.25$ 与 $\rho = 0.5$）。分歧越大，纠正率始终越高，进而带来更大的准确率提升。低保留率（红色）表现出最强的效果，证实激进剪枝会放大可被共识恢复利用的有效模态分歧。

较高（$r = 0.1$时为 80–90%），表明早期恢复的目标确实是存在争议的词元。随着r 增加，纠正率急剧下降（至 40–50%），揭示了共识饱和效应：只有少数被丢弃的词元在不同模态之间存在显著分歧，超出这一范围进行恢复只会重新引入大量冗余词元。

要点。 由于进一步恢复带来的收益逐渐递减，剩余的低显著性词元应当合并，而非逐个保留，这启发了编码器引导的令牌合并（第 4.3节）。

一致性的含义。 在本文中，我们用一致性 表示视觉侧与跨模态显著性信号之间的协调一致，这种一致性是在调和两者的互补作用后得出的——

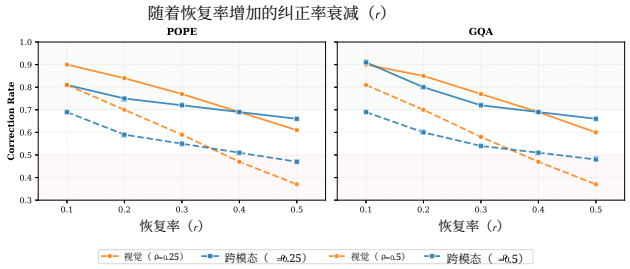


图 4. POPE 和 GQA 中，随着恢复率 r 提高，**纠正率衰减**的变化。较小的 r对应最高的纠正率，此时几乎所有恢复的词元都对应真正的多模态分歧。随着 r 增大，恢复带来的收益趋于饱和：重新引入的额外词元中，越来越多是两种模态原本已达成一致的词元。这揭示了质量与数量之间显著的权衡，并表明基于一致性的恢复会自然优先处理少量高价值纠正项。

视觉侧提供强有力的基础排序，而跨模态线索则引入有针对性、依赖于查询的修正。这并非简单地赋予两者相同权重，而是意味着以有原则的方式整合两种信号。

4. 方法

基于第3节的工作，我们提出*ConsensusDrop*，这是一种无需训练的视觉词元压缩框架，通过三个无需训练的组件融合视觉显著性与跨模态显著性：静态交叉注意力探针（SCAP）、多模态融合器和编码器引导的令牌合并（EGTM）。如图5所示，SCAP 首先提取 LLM 之前的交叉模态注意力。随后，融合器将其与视觉侧显著性结合，以选出前K 个词元；EGTM 则利用编码器的特征几何结构，在投影器空间中紧凑地合并剩余的低显著性词元，从而生成简化但保留信息的序列（算法2）。

4.1. 静态交叉注意力探针（SCAP）

SCAP 是一种轻量级、无需训练的探针，可在主大语言模型前向传播之前 估计每个视觉词元的查询条件显著性。具体而言，SCAP 会复现大语言模型首个解码器层中的输入层归一化和多头自注意力模块，并将其应用一次˜于完整的多模态输入序列 $\mathbf{X} = [\mathbf{X}^{\text{sys}}; \mathbf{V}; \mathbf{T}]$。令 $S = |\mathbf{X}^{\text{sys}}| + N + L$ 表示序列总长度。由此得到的头平均注意力矩阵 $\tilde{\mathbf{A}} \in \mathbb{R}^{S\times S}$ 为

$$\tilde{\mathbf{A}} = \frac{1}{H}\sum_{h=1}^{H} \text{Softmax}\left(\frac{\mathbf{Q}^{(h)}(\mathbf{K}^{(h)})^{\top}}{\sqrt{d_h}}\right), \qquad (1)$$

其中，Softmax 按行应用，并使用与大语言模型相同的因果掩码。

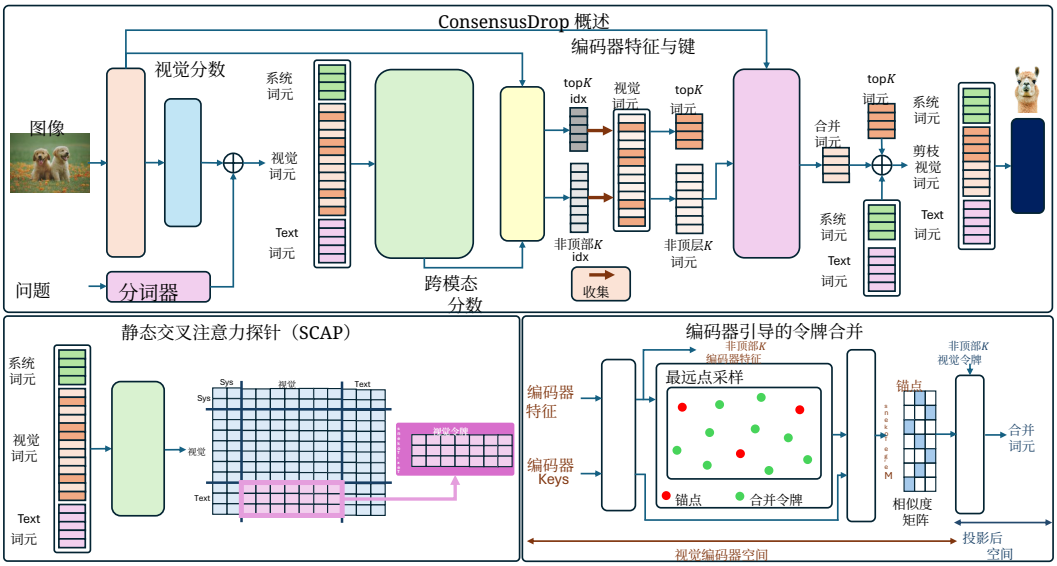


图 5. **ConsensusDrop 示意图**。给定一个图像—问题对，冻结的视觉编码器和投影器会生成视觉词元，而 SCAP 则提供查询感知的交叉注意力分数。轻量级融合器结合视觉侧显著性和跨模态显著性，选出排名靠前的 K 个词元；EGTM 则将其余词元合并为紧凑表示，从而为大语言模型生成压缩视觉序列。

我们提取文本到视觉块 $\mathbf{A} = \mathbf{A}[\mathcal{I}_{\text{text}}, \mathcal{I}_{\text{vis}}] \in \mathbb{R}^{L \times N}$，其中 $\mathbf{A}_{i,j}$ 表示从文本令牌 t_i 到视觉词元 $\tilde{v}_j$ 的注意力。我们将 $\mathbf{A}$ 聚合为跨模态显著性分数 $\mathbf{s}^{(c)} \in \mathbb{R}^N$，默认采用全词元策略：$\mathbf{s}^{(c)} = \frac{1}{L}\sum_{i=1}^{L} \text{Norm}(\mathbf{A}_{i,:})$（其他变体见 **附录 D.1**）。

4.2. 融合器模块

给定一张输入图像，我们为 N 个视觉词元获取两个互补的显著性向量：视觉侧分数和跨模态分数。我们从冻结的视觉编码器中提取视觉侧显著性 $\mathbf{s}^{(v)} \in \mathbb{R}^N$，方法是使用以下任一方式：(i) CLS 到图像块注意力，或 (ii) 所有其他图像块对每个图像块产生的聚合注意力，两者均基于视觉编码器的倒数第二层自注意力层计算（对各注意力头的注意力分数进行聚合）。SCAP（第 4.1 节）生成查询条件化的跨模态显著性 $\mathbf{s}^{(c)} \in \mathbb{R}^N$。

温度缩放。融合前，使用温度缩放将两个显著性向量转换为归一化重要性分布。对于模态 $m \in \{v, c\}$，我们定义

$$\hat{\mathbf{s}}_j^{(m)} = \frac{\left(\mathbf{s}_j^{(m)}\right)^{1/\tau_m}}{\sum_{k=1}^{N}\left(\mathbf{s}_k^{(m)}\right)^{1/\tau_m}}, \qquad j = 1, \ldots, N, \tag{2}$$

其中，$\tau_m > 0$ 控制分布锐度。较大的温度会产生更平坦的分布，而较小的温度则会突出高评分词元（分布更尖锐）。

凸融合器。我们的默认 Fuser 通过凸组合将两个归一化分布融合：

$$\hat{\mathbf{s}} = \alpha\, \hat{\mathbf{s}}^{(v)} + (1 - \alpha)\, \hat{\mathbf{s}}^{(c)}, \tag{3}$$

其中 $\alpha \in [0, 1]$ 控制视觉—跨模态平衡。受第 3节观察到的不对称行为启发，我们让融合偏向视觉显著性，并使用 $\alpha > 0.5$(默认 $\alpha = 0.7$)，除非另有说明。根据 $\hat{\mathbf{s}}$ 对词元进行排名；保留排名前 K 的索引 $\mathcal{I}_{\text{top}}$，其余 $\mathcal{I}_{\text{non}} = [N] \setminus \mathcal{I}_{\text{top}}$ 则传递给 EGTM 进行压缩。融合分布 $\hat{\mathbf{s}}$ 仅为此凸融合器定义。（替代融合机制请参见 **附录 D.2** 。）

4.3. 编码器引导的令牌合并（EGTM）

Fuser 选出排名前 K 个索引 $\mathcal{I}_{\text{top}}$ 后，我们将剩余的 $R = N - K$ 个视觉词元 $\mathcal{I}_{\text{non}} = [N] \setminus \mathcal{I}_{\text{top}}$ 压缩为 M 个合并词元，生成长度为 $K + M$ 的最终视觉序列。EGTM 采用编码器引导方式：结构决策（锚点选择和 token 分配）基于冻结的视觉编码器提取的特征计算，而特征聚合则在投影器后空间中进行，以确保与大语言模型兼容。

通过 FPS 选择锚点。令 $\mathbf{V} \in \mathbb{R}^{N \times d_v}$ 表示倒数第二层视觉编码器的图像块特征（投影前），并定义 $\mathbf{V}_{\text{non}} = \mathbf{V}[\mathcal{I}_{\text{non}}] \in \mathbb{R}^{R \times d_v}$。我们对 $\ell_2\mathbf{V}_{\text{non}}$ 进行归一化，并使用最远点采样（FPS）[60], 选择多样化的锚点词元 M，得到锚点索引集合 $\mathcal{A} \subseteq \mathcal{I}_{\text{non}}$，其中 $|\mathcal{A}| = M$。其余索引记为 $\mathcal{U} = \mathcal{I}_{\text{non}} \setminus \mathcal{A}$。

通过编码器键进行硬分配。令 $\mathbf{K} \in \mathbb{R}^{H_v \times N \times d_k}$ 表示来自同一视觉编码器倒数第二个块的键张量（每个注意力头一个）。我们为每个图像块构造一个经过头部平均且经 ℓ_2归一化的键嵌入，得到 $\mathbf{G} \in \mathbb{R}^{N \times d_k}$。限制为非前 K 个词元后，得到 $\mathbf{G}_{\text{non}} = \mathbf{G}[\mathcal{I}_{\text{non}}] \in \mathbb{R}^{R \times d_k}$。每个非锚点（合并）词元 $u \in \mathcal{U}$ 都根据硬最近锚点规则，分配给该键空间中与其最相似的锚点：$a^\star(u) = \arg\max_{a \in \mathcal{A}} \langle \mathbf{g}_u, \mathbf{g}_a \rangle$，其中 $\mathbf{g}_u$ 和 $\mathbf{g}_a$ 是 $\mathbf{G}_{\text{non}}$的行。该点积仅用于分配，从而围绕选定锚点对非前 K 个词元进行离散划分。

在投影器空间中合并。令 $\tilde{\mathbf{V}} \in \mathbb{R}^{N \times d}$ 表示输入大语言模型的投影视觉令牌。对于每个锚点$a \in \mathcal{A}$，我们通过对其自身的投影词元与分配给它的所有投影词元取平均，形成一个合并后的词元：

$$\tilde{\mathbf{v}}_a^{\text{merge}} = \frac{1}{|\mathcal{C}(a)|} \sum_{j \in \mathcal{C}(a)} \tilde{\mathbf{v}}_j, \tag{4}$$

其中 $\mathcal{C}(a) = \{a\} \cup \{u \in \mathcal{U} : a^\star(u) = a\}$. 最后，EGTM 输出缩减后的视觉序列

$$\tilde{\mathbf{V}}_{\text{final}} = \text{Concat}\big(\tilde{\mathbf{V}}[\mathcal{I}_{\text{top}}], \{\tilde{\mathbf{v}}_a^{\text{merge}}\}_{a \in \mathcal{A}}\big) \in \mathbb{R}^{(K+M) \times d}. \tag{5}$$

多模态设计原理。 EGTM 明确地将结构推断的位置 与压缩应用的位置 解耦：视觉编码器提供具有语义意义的度量空间，用于选择锚点和分配词元，而最终的合并则在与大语言模型对齐的投影器空间中进行。这种设计能够大幅压缩视觉词元，同时保留语言模型所使用的多模态表征。

5. 实验

5.1. 实验设置

数据集。 我们在三个不同类别的 12 个基准上评估 ConsensusDrop：（1）**通用视觉问答**：VQAv2 [21], GQA [27], ScienceQA-IMG [52],和 TextVQA [68]; （2）**多模态理解**：POPE [38], MME [19], MMBench/CN [48], 和 MM-Vet [84]; 以及（3）**视频问答**：TGIF-QA [28], MSVD-QA [74], 和 MSRVTT-QA [74]。所有实验均遵循官方评估协议。

模型架构。 我们将 ConsensusDrop 应用于有代表性的视觉语言模型：**(1) LLaVA 系列** （主要实验）：LLaVA-1.5 [44], LLaVA-NeXT [45] （高分辨率）和 Video-LLaVA [40]（视频理解）； 以及 **(2)其他开源视觉语言模型**: Qwen-VL [4], InternVL [12], 和 CogVLM [71]。

评估设置。 我们遵循每个模型的官方推理超参数和解码设置。对于 ConsensusDrop，我们默认使用凸融合器，采用 $\tau_c = \tau_m = 1$ 和 $\alpha = 0.7$，并启用 EGTM。

对比方法。 作为剪枝基线，我们考察了 ToMe [6], FastV [11], SparseVLM [90], LLaVA-PruMerge [65], VisionZip [78], 和 VisPruner [88]。这些方法涵盖了应用于视觉语言模型不同阶段的多种剪枝策略，提供了全面且公平的比较。

5.2. 主要结果

表1 展示了**ConsensusDrop** 在 LLaVA-1.5-7B 上、四种保留预算下（576 个视觉词元中保留 192、128、64 和 32 个）的性能。ConsensusDrop 在所有预算下均实现了最佳的准确率—效率权衡，分别保留了基线性能的**99.7%**、**99.2%**、**95.2%**和**90.5%**。随着压缩力度加大，ConsensusDrop 的优势愈发显著。在**32 个词元** （↓94.4%）时，它在所有基准测试中均取得最佳性能，大幅超过仅视觉和仅跨模态方法，同时保留了**90.5%**的基线准确率。这一结果凸显了在极限令牌缩减情况下多模态共识的重要性；此时，选择恰当的视觉词元至关重要。从效率角度看，ConsensusDrop 在首令牌生成时间和键值缓存占用方面持续优于 FastV 等仅跨模态方法 [11]和 SparseVLM[90] ，同时保持了与仅视觉方法相当的表现。VisionZip [78]在各种设置下延迟最低；不过，在令牌预算相近时，ConsensusDrop 的准确率显著更高，因此总体上实现了更优的准确率—延迟权衡。最后，尽管 SparseVLM [90]的理论键值缓存大小最小，但其大语言模型内部逐阶段剪枝会导致各层的键值缓存大小不均匀，进而可能造成表1未能反映的内存碎片化。相比之下，ConsensusDrop 在大语言模型推理之前进行剪枝，从而得到紧凑且统一的键值缓存。

5.3. 更高分辨率输入下的 ConsensusDrop

在 LLaVA-NeXT-7B 的高分辨率输入中 [46]，当视觉序列扩展至**2880** 个词元时，ConsensusDrop 在相同令牌预算下始终达到或超过最强剪枝基线的表现（表2）。在**640 个词元**（↓ 77.8%）下，它保留了完整模型性能的**99.0%**，略优于 VisPruner，并在 GQA、POPE 和 MME 上保持较高准确率。进一步激进剪枝至**320** （↓ 88.9%）和**160 个词元**（↓ 94.4%）时，ConsensusDrop 分别保留了**93.9%**和**87.1%** 的相对性能，明显优于 VisionZip、VisPruner 以及仅采用跨模态方法的方法。这些结果表明，即使在密集令牌化条件下，多模态共识仍然有效，从而支持激

方法	视觉问答 v2	GQA	SQA^{IMG}	视觉问答^{Text}	POPE	MME	MMB	MMBench^{CN}	MMVet	准确率	Rel	首令牌生成时间（毫秒）	每个输出令牌时间（毫秒）	键值缓存（MB）
上界：*576* 个词元（*100%*）														
LLaVA-1.5-7B	78.5	62.0	66.8	58.2	85.9	1510.7	64.3	58.3	31.1	64.5	100%	106.4	27.7	321
保留 *192* 个词元（↓ 66.7%）														
ToMe [6]	68.0	54.3	65.2	52.1	72.4	1148.8	60.5	50.2	27.8	56.4	87.4%	—	—	—
FastV [11]	67.1	52.6	**69.1**	52.5	64.8	1252.1	61.0	52.4	28.5	56.7	87.9%	78.0 (1.36×)	26.0 (1.07×)	144（↓55%）
SparseVLM [90]	77.0	59.5	68.7	**57.8**	85.3	1395.4	**64.1**	52.1	30.2	62.7	97.2%	86.6 (1.23×)	38.3 (0.72×)	**119**（↓**63%**）
VisionZip [78]	76.8	59.3	68.9	57.3	85.3	1443.6	63.0	56.9	31.7	63.5	98.4%	**69.5 (1.53×)**	**25.5 (1.09×)**	129 (↓60%)
VisPruner [88]	76.9	59.6	68.7	57.6	**86.0**	1468.4	63.7	57.4	**33.3**	64.1	99.4%	73.0 (1.46×)	26.2 (1.06×)	129 (↓60%)
ConsensusDrop	**77.1**	**60.4**	69.0	57.5	85.5	**1476.2**	63.8	**58.0**	33.1	**64.3**	**99.7%**	80.4 (1.32×)	26.2 (1.06×)	129 (↓60%)
保留 *128* 个词元（↓ 77.8%）														
ToMe [6]	63.0	52.4	59.6	49.1	62.8	1088.4	53.3	48.8	27.2	52.3	81.1%	—	—	—
FastV [11]	61.8	49.6	60.2	50.6	59.6	1208.9	56.1	51.4	28.1	53.1	82.3%	72.1 (1.48×)	25.8 (1.07×)	117 (↓64%)
SparseVLM [90]	73.8	56.0	67.1	54.9	80.5	1376.2	60.0	51.1	30.0	60.2	93.3%	83.0 (1.28×)	37.7 (0.74×)	**87 (↓73%)**
PruMerge+ [65]	74.7	57.8	67.6	54.3	81.5	1420.5	61.3	54.7	28.7	61.3	95.0%	—	—	—
VisionZip [78]	75.6	57.6	68.9	56.8	83.2	1432.4	62.0	56.7	32.6	62.8	97.4%	**60.2 (1.77×)**	**25.3 (1.10×)**	97 (↓70%)
VisPruner [88]	75.8	58.2	69.1	57.0	84.6	1461.4	62.7	57.3	33.7	63.5	98.4%	72.5 (1.47×)	26.2 (1.06×)	97 (↓70%)
ConsensusDrop	**76.1**	**60.2**	**69.3**	**57.1**	**85.1**	**1467.7**	**63.2**	**58.1**	**33.8**	**64.0**	**99.2%**	70.6 (1.51×)	25.8 (1.07×)	97 (↓70%)
保留 *64* 个词元（↓ 88.9%）														
ToMe [6]	57.1	48.6	50.0	45.3	52.5	922.3	43.7	38.9	24.1	45.1	69.9%	—	—	—
FastV [11]	55.0	46.1	51.1	47.8	48.0	1019.6	48.0	42.7	25.8	46.2	71.6%	71.4 (1.50×)	25.9 (1.07×)	96 (↓70%)
SparseVLM [90]	68.2	52.7	62.2	51.8	75.1	1221.1	56.2	46.1	23.3	55.2	85.6%	84.0 (1.27×)	38.1 (0.73×)	**55 (↓83%)**
PruMerge+ [65]	67.4	54.9	68.6	53.0	77.4	1198.2	59.3	51.0	25.9	57.5	89.1%	—	—	—
VisionZip [78]	72.4	55.1	69.0	55.5	77.0	1365.6	60.1	55.4	31.7	60.5	93.8%	**58.4 (1.82×)**	25.5 (1.09×)	65 (↓80%)
VisPruner [88]	72.7	55.4	69.1	55.8	**80.4**	1369.9	61.3	55.1	**32.3**	61.2	94.9%	72.0 (1.48×)	25.7 (1.08×)	65 (↓80%)
ConsensusDrop	**73.1**	**56.2**	**70.2**	**56.4**	79.8	**1372.9**	**61.7**	**55.8**	31.2	**61.4**	**95.2%**	64.7 (1.64×)	**25.2 (1.10×)**	65 (↓80%)
保留 *32* 个词元（↓ 94.4%）														
ToMe [6]	46.8	43.6	41.4	38.3	39.0	828.4	31.6	28.1	17.3	36.4	56.4%	—	—	—
FastV [11]	43.4	41.5	42.6	42.5	32.5	884.6	37.8	33.2	20.7	37.6	58.3%	66 (1.61×)	26.3 (1.05×)	81 (↓75%)
SparseVLM [90]	58.6	48.3	57.3	46.1	67.9	1046.7	51.4	40.6	18.6	49.0	76.0%	77.8 (1.37×)	36.7 (0.75×)	**48 (↓85%)**
PruMerge+ [65]	54.9	51.1	68.5	50.6	70.9	940.8	56.8	47.0	21.4	52.0	80.6%	—	—	—
VisionZip [78]	67.1	51.8	68.8	53.1	68.7	1247.4	57.7	50.3	25.5	56.2	87.1%	**54.3 (1.96×)**	26.1 (1.06×)	49 (↓85%)
VisPruner [88]	67.7	52.2	69.2	53.9	72.7	1271.0	58.4	52.7	28.8	57.7	89.5%	71.5 (1.49×)	26.4 (1.05×)	49 (↓85%)
ConsensusDrop	**67.8**	**53.2**	**70.1**	**54.7**	**73.1**	**1275.8**	**59.0**	**54.1**	**30.1**	**58.4**	**90.5%**	60.0 (1.77×)	**25.1 (1.10×)**	49 (↓85%)

表1. LLaVA-1.5-7B 在若干保留设置下的性能比较。Acc 表示平均准确率；Rel 表示相对于基线的性能。TTFT = 首个令牌生成时间，括号内为加速比；TPOT = 每个输出令牌时间及其加速比；KV = 键值缓存内存及其减少百分比。**粗体** 表示最佳；下划线表示次佳。

进压缩，同时保留与大语言模型对齐的视觉证据。

5.4. ConsensusDrop 与视频理解

表。3评估了 ConsensusDrop 在 Video-LLaVA 上的表现，涵盖三个视频问答基准；该设置中，连续帧（8 帧，分辨率为 224，**2048** 词元）存在严重的时间冗余。在相同的减少比例下，ConsensusDrop 的表现始终与以往方法相当或更优。在**455 个词元**(↓ 77.8%) 时，其平均准确率/得分最佳（48.6/3.32），与完整模型相当，且略优于 VisPruner 和 FastV。进一步激进剪枝至 **227** (↓ 88.9%) 和 **114 个词元**(↓ 94.4%) 后，ConsensusDrop 仍保持最强的平均性能，保留了约**92%**的完整模型准确率，并优于其他基线。

5.5. 在其他视觉语言模型架构上应用 ConsensusDrop

我们进一步在三种具有不同视觉—语言接口的代表性视觉语言模型骨干网络上评估 ConsensusDrop：Qwen-VL-7B [4], InternVL-Chat-13B [12], 以及 CogVLM-chat-1.1-17B [71] (表4）。在所有令牌预算下，ConsensusDrop 的表现始终优于仅采用跨模态剪枝的方法

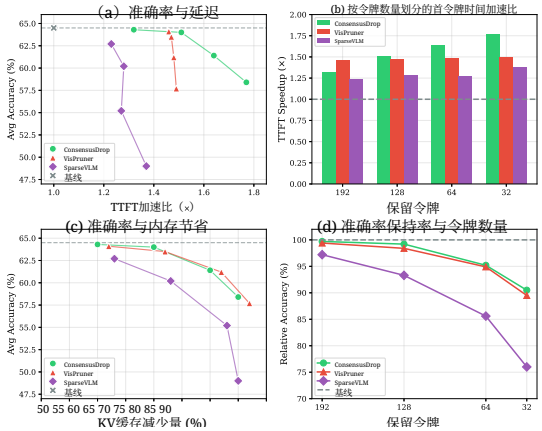


图6. **LLaVA-1.5-70亿参数的整体准确率—效率权衡**。 与仅视觉剪枝（VisPruner）和仅跨模态剪枝（SparseVLM）相比，ConsensusDrop 在不同令牌预算下始终位于帕累托前沿上或其上方 $\{192, 128, 64, 32\}$。

剪枝基线 FastV [11]，并且在平均准确率上比 VisPruner 略有但稳定的提升。对于 Qwen-VL，在 128/256 个词元下（↓ 50%），ConsensusDrop 保留了 **98.2%**的

方法	视觉问答^{V2}	GQA	视觉问答^{Text}	POPE	MME	准确率	Rel.
上界：全部 *2880* 个视觉令牌(100%)							
LLaVA-NeXT-7(亿参数	81.2	62.9	59.6	86.3	1513.8	73.1	100.0%
保留 *640* 个词元(↓ 77.8%)							
FastV [11]	78.9	60.4	58.4	83.1	1477.3	70.9	97.0%
SparseVLM [90]	78.2	59.1	56.2	80.9	1456.3	69.4	94.9%
VisionZip [78]	79.2	60.1	58.5	82.2	1468.4	70.7	96.7%
VisPruner [88]	79.8	61.6	59.3	85.9	1480.7	72.1	98.6%
ConsensusDrop	79.9	62.1	59.7	86.1	1483.9	72.4	99.0%
保留 *320* 个词元(↓ 88.9%)							
FastV [11]	71.9	55.9	55.7	71.7	1282.9	63.9	87.4%
SparseVLM [90]	71.4	56.5	52.4	73.5	1342.7	64.2	87.8%
VisionZip [78]	74.2	58.1	55.3	75.0	1348.8	66.0	90.3%
VisPruner [88]	75.7	58.4	57.6	80.4	1370.1	68.1	93.2%
ConsensusDrop	76.1	59.1	58.0	81.5	1372.8	68.7	93.9%
保留 *160* 词元(↓ 94.4%)							
FastV [11]	61.8	49.8	51.9	51.7	1079.5	53.8	73.6%
SparseVLM [90]	62.2	50.2	45.1	54.6	1167.1	54.1	74.0%
VisionZip [78]	67.3	54.3	54.7	59.4	1239.7	59.5	81.4%
VisPruner [88]	70.6	54.7	56.0	72.9	1226.0	63.1	86.3%
ConsensusDrop	71.0	55.3	57.2	73.2	1232.1	63.7	87.1%

表 2。LLaVA- NeXT-7B 上剪枝方法的性能比较。“准确率”表示跨基准测试的聚合得分，“相对值”表示在相应缩减比例下，相对于完整令牌 LLaVA- NeXT-7B 模型所保留性能的百分比。

方法	TGIF-QA 准确率	TGIF-QA 得分	MSVD-QA 准确率	MSVD-QA 得分	MSRVTT-QA Acc.	MSRVTT-QA 得分	平均值 准确率	平均值 得分
上界：全部 *2048* 个视觉令牌(100%)								
Video-LLaVA	19.8	2.53	70.5	3.93	57.5	3.50	49.3	3.32
保留 *455* 个词元（↓ 77.8%）								
FastV [11]	19.2	2.50	69.1	3.91	54.4	3.42	47.6	3.28
VisPruner [88]	18.4	2.49	70.2	3.95	56.7	3.50	48.4	3.31
ConsensusDrop	18.6	2.51	70.3	3.94	56.8	3.52	48.6	3.32
保留 *227* 个词元（↓ 88.9%）								
FastV [11]	14.3	2.42	68.9	3.90	53.0	3.40	45.4	3.24
VisPruner [88]	15.9	2.41	69.3	3.92	55.6	3.45	46.9	3.26
ConsensusDrop	16.1	2.43	69.5	3.94	55.8	3.43	47.1	3.27
保留 *114* 个词元(↓ 94.4%)								
FastV [11]	10.6	2.29	64.1	3.78	52.4	3.39	42.4	3.15
VisPruner [88]	14.1	2.35	65.4	3.79	54.1	3.41	44.5	3.18
ConsensusDrop	14.4	2.38	66.7	3.82	54.4	3.45	45.2	3.22

表 3. Video-LLaVA 在三个视频问答基准上的视频理解性能。评估使用每个基准的前 1,000 个样本，`gpt-3.5-turbo` 用于评分，与 [11]相同。

完整性能（准确率为 66.1，对比 67.3），优于 Vis- Pruner（96.9 %）。在 InternVL 的 144/576 个词元下（↓ 75%），它再次实现了最佳性能保留率（98.5%）和最高准确率（65.4）。在最极端的压缩

在 CogVLM 上压缩（123/1225 个词元，↓ 90%）时，Consensus-Drop 保留了原始准确率的 **91.6%**，优于 FastV（79.5%）和 VisPruner（90.6%）。

方法	视觉问答^{V2}	GQA	SQA^{IMG}	视觉问答^{Text}	准确率	Rel.
上界：全部 *256* 个视觉令牌(100%)						
Qwen-VL-7B	78.8	59.3	67.1	63.8	67.3	100.0%
保留 *128* 个词元 (↓ 50%)						
FastV [11]	76.5	56.9	65.3	58.2	64.2	95.4%
VisPruner [88]	77.4	57.8	65.9	59.6	65.2	96.9%
ConsensusDrop	78.1	58.9	66.4	61.1	66.1	98.2%
上界：全部 *576* 个视觉令牌(100%)						
InternVL-Chat-13B	79.3	62.9	66.3	57.0	66.4	100.0%
保留 *144* 词元 (↓ 75%)						
FastV [11]	74.1	58.2	66.6	55.6	63.6	95.8%
VisPruner [88]	76.7	60.2	67.3	55.3	64.9	97.7%
ConsensusDrop	76.9	60.5	67.9	56.1	65.4	98.5%
上界：全部 *1225* 个视觉令牌(100%)						
CogVLM-chat-1.1-17B	80.9	58.2	68.4	69.2	69.2	100.0%
保留 *123* 个词元 (↓ 90%)						
FastV [11]	74.2	40.3	63.5	41.9	55.0	79.5%
VisPruner [88]	74.6	48.4	68.2	59.6	62.7	90.6%
ConsensusDrop	75.1	49.3	69.6	59.5	63.4	91.6%

表 4. 不同视觉语言模型架构上的剪枝方法性能。准确率表示跨基准测试的平均准确率，相关性表示保留的性能百分比。

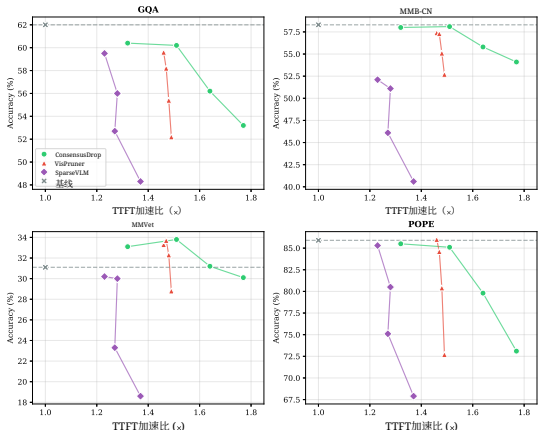


图 7. **各数据集的准确率与首令牌生成时间权衡**。在各个数据集上，ConsensusDrop 在相当或更高的首令牌生成时间加速比下保持了更高的准确率，表明其效率提升具有一致性。

5.6. 效率分析

我们在 LLaVA-1.5-7B 上评估词元剪枝的准确率–效率权衡，重点关注**TTFT**（预填充延迟）、**每个输出令牌时间**（解码延迟）以及**键值缓存**内存。**延迟**。随着令牌预算降低，ConsensusDrop 在保持较高准确率保留率的同时，实现了显著的 TTFT 加速比。在激进压缩（32 个词元，↓94.4%）下，它仍保留了**90.5%** 的基线准确率，且 **1.77**× 的 TTFT 加速比优于两种基线方法。**内存**。通过在大语言模型推理前剪枝词元，ConsensusDrop 可生成统一的键值缓存布局并实现稳定的内存节省，在极端压缩下最多可减少**85%** 的键值缓存用量。**帕累托最优性**。图 6显示，ConsensusDrop 始终实现了更有利的

准确率–延迟–内存权衡，优于 VisPruner 和 SparseVLM；在令牌预算较低、词元选择质量至关重要时，这一优势尤为明显。**跨数据集**。图7 证实，这些优势可推广至多个基准测试；在所有评估的数据集上，ConsensusDrop 的表现都更接近帕累托前沿。

6. 结论

ConsensusDrop 是一种无需训练的即插即用框架，利用多模态共识保留视觉语言模型中信息量最大的视觉词元。通过融合视觉显著性和跨模态显著性，该框架在激进的令牌缩减条件下始终优于以往的剪枝方法。

参考文献

[1] 阿里云。Qwen 3-next，2025。开放权重混合专家大语言模型，具有较大的上下文窗口，支持多语言。1, 13

[2] 阿里云。Qwen-omni：统一的多模态基础模型。技术报告，阿里云，2025。开放源代码模型，支持文本、图像、音频和视频，采用 Apache 2.0 许可。。Anthropic。Claude sonnet 4.5。https://www.anthropic.com/news/claude-sonnet-4-5，2025。Anthropic 的前沿模型。1, 13

[4] Jinze Bai、Shuai Bai、Shusheng Yang、Shijie Wang、Sinan Tan、Peng Wang、Junyang Lin、Chang Zhou 和 Jingren Zhou。Qwen-VL：能力多样的前沿大型视觉语言模型。arXiv 预印本 arXiv:2308.12966，2023。6, 7, 17

[5] Jinze Bai、Shuai Bai、Shusheng Yang、Shijie Wang、Sinan Tan、Peng Wang、Junyang Lin、Chang Zhou 和 Jingren Zhou。Qwen-VL：一种用于理解、定位、文本阅读等任务的多功能视觉语言模型，2023。23

[6] Daniel Bolya、Cheng-Yang Fu、Xiaoliang Dai、Peizhao Zhang、Christoph Feichtenhofer 和 Judy Hoffman。令牌合并：让你的视觉 Transformer 更快。arXiv 预印本 arXiv:2210.09461，2022。6, 7

[7] Mu Cai、Jianwei Yang、Jianfeng Gao 和 Yong Jae Lee。套娃式多模态模型。arXiv 预印本 arXiv:2405.17430，2024。2, 13

[8] Liwei Che、Tony Qingze Liu、Jing Jia、Weiyi Qin、Ruixiang Tang 和 Vladimir Pavlovic。幻觉图像令牌：一种无需训练的简易方法，用于检测和缓解大型视觉语言模型中的物体幻觉。发表于 IEEE/CVF国际计算机视觉会议，页码 21635–21644，2025。13

[9] David Chen 和 William B Dolan。收集高度平行的数据用于释义评估。发表于第49届计算语言学协会年会：人类语言技术会议论文集，页码 190–200，2011。16

[10] Jieneng Chen、Luoxin Ye、Ju He、Zhao-Yang Wang、Daniel Khashabi 和 Alan Yuille。通过视觉上下文压缩实现高效大型多模态模型。神经信息处理系统进展，37：73986–74007，2024。2, 13

[11] Liang Chen、Haozhe Zhao、Tianyu Liu、Shuai Bai、Junyang Lin、Chang Zhou 和 Baobao Chang。《第 2 层之后，一张图像只值 1/2 个词元：大型视觉语言模型的即插即用推理加速》。载于《欧洲计算机视觉会议》，第 19–35 页。Springer，2024。1、3、6、7、8、13、16、17

[12] Zhe Chen、Jiannan Wu、Wenhai Wang、Weijie Su、Guo Chen、Sen Xing、Muyan Zhong、Qinglong Zhang、Xizhou Zhu、Lewei Lu 等。《InternVL：扩展视觉基础模型并使其适用于通用视觉语言任务》。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 24185–24198 页，2024。6、7

[13] Zhenyu Chen 等。《InternVL 2.5》。arXiv:2412.05271，2024。1、13

[14] Tri Dao。《FlashAttention-2：通过更好的并行化和工作划分实现更快的注意力》，2023。1, 2。

[15] DeepSeek AI。《DeepSeek-OCR：上下文光学压缩》。https://arxiv.org/abs/2510.18234，2025。1、13

[16] Mohamed Dhouib、Davide Buscaldi、Sonia Vanier 和 Aymen Shabou。《PACT：基于剪枝与聚类的词元缩减方法，可加速视觉语言模型》，2025。1, 2, 13。

[17] Alexey Dosovitskiy。《一张图像胜过 X 个词：用于大规模图像识别的 Transformer》。arXiv 预印本 arXiv:2010.11929，2020。16

[18] Mark Endo、Xiaohan Wang 和 Serena Yeung-Levy。《轻踩油门：重新审视用于视觉语言模型加速的视觉词元剪枝》。载于《IEEE/CVF国际计算机视觉会议论文集》，第 22826–22835 页，2025。1、3、13

[19] Chaoyou Fu、Peixian Chen、Yunhang Shen、Yulei Qin、Mengdan Zhang、Xu Lin、Jinrui Yang、Xiawu Zheng、Ke Li、Xing Sun、Yunsheng Wu 和 Rongrong Ji。《MME：多模态大语言模型的综合评测基准》，2024。6、15

[20] Chaoyou Fu、Peixian Chen、Yunhang Shen、Yulei Qin、Mengdan Zhang、Xu Lin、Jinrui Yang、Xiawu Zheng、Ke Li、Xing Sun、Yunsheng Wu、Rongrong Ji、Caifeng Shan 和 Ran He。《MME：多模态大语言模型的综合评测基准》。载于第三十九届神经信息处理系统大会数据集与基准赛道，2025。13、17

[21] Yash Goyal、Tejas Khot、Douglas Summers-Stay、Dhruv Batra 和 Devi Parikh。《让 VQA 中的 V 发挥作用：提升图像理解在视觉问答中的作用》。载于《IEEE计算机视觉与模式识别会议论文集》，第 6904–6913 页，2017。6、15、22

[22] Zonghao Guo、Ruyi Xu、Yuan Yao、Junbo Cui、Zanlin Ni、Chunjiang Ge、Tat-Seng Chua、Zhiyuan Liu 和 Gao Huang。《LLaVA-UHD：能够感知任意宽高比和高分辨率图像的大语言模型》。载于《欧洲计算机视觉会议》，第 390–406 页。Springer，2024。13

[23] Yefei He、Feng Chen、Jing Liu、Wenqi Shao、Hong Zhou、Kaipeng Zhang 和 Bohan Zhuang。《ZipVL：采用动态词元稀疏化和 KV 缓存压缩的高效大型视觉语言模型》。arXiv 预印本 arXiv:2410.08584，2024。1、3、13

[24] Anwen Hu 等。《mplug-docowl2：用于无 OCR 多页文档理解的高分辨率压缩》。载于 ACL 2025，2025。13[25] Wenbo Hu、Zi-Yi Dou、Liunian Li、Amita Kamath、Nanyun Peng 和 Kai-Wei Chang。《面向大型视觉语言模型的套娃式查询变换器》。神经信息处理系统大会论文集，37：50168–50188，2024。2、13[26] Ding-An Huang 等。《Frag：面向长视频和长文档理解的帧选择增强生成》。arXiv:2504.17447，2025。13[27] Drew A Hudson 和 Christopher D Manning。《GQA：面向真实世界视觉推理和组合式问答的新数据集》。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 6700–6709 页，2019。6、13、15[28] Yunseok Jang、Yale Song、Youngjae Yu、Youngjin Kim 和 Gunhee Kim。《TGIF-QA：探索视觉问答中的时空推理》。载于《IEEE计算机视觉与模式识别会议论文集》，第 2758–2766 页，2017。6、16[29] Yitong Jiang、Jinwei Gu、Tianfan Xue、Ka Chun Cheung、Pavlo Molchanov、Hongxu Yin 和 Sifei Liu。《高效词元视觉语言模型：通过动态区域提议实现高分辨率图像理解》。载于《IEEE/CVF国际计算机视觉会议》（ICCV）论文集，第 24147–24158 页，2025。1、2、3、13[30] Moo Jin Kim、Karl Pertsch、Siddharth Karamcheti、Ted Xiao、Ashwin Balakrishna、Suraj Nair、Rafael Rafailov、Ethan Foster、Grace Lam、Pannag Sanketi 等。《OpenVLA：开源视觉-语言-动作模型》。arXiv 预印本 arXiv:2406.09246，2024。1、13[31] Ivan Krasin、Tom Duerig、Neil Alldrin、Vittorio Ferrari、Sami Abu-El-Haija、Alina Kuznetsova、Hassan Rom、Jasper Uijlings、Stefan Popov、Andreas Veit 等。《OpenImages：用于大规模多标签和多类别图像分类的公共数据集》。数据集地址：https://github.com/openimages，2(3):18，2017。15[32] Ranjay Krishna、Yuke Zhu、Oliver Groth、Justin Johnson、Kenji Hata、Joshua Kravitz、Stephanie Chen、Yannis Kalan-t idis、Li-Jia Li、David A Shamma 等。《Visual Genome：利用众包密集图像标注连接语言与视觉》。International Journal of Computer Vision，123(1):32–73，2017。15[33] Bohao Li、Rui Wang、Guangzhi Wang、Yuying Ge、Yixiao Ge 和 Ying Shan。《SEED-Bench：通过生成式理解对多模态大语言模型进行基准测试》。arXiv 预印本 arXiv:2307.16125，2023。16 、17[34] Bo Li 等。《LLaVA-OneVision：轻松实现视觉任务迁移》。arXiv:2408.03326，2024。1[35] Bo Li 等。《LLaVA-OneVision-1.5：用于通用多模态理解的完全开放框架》。https://arxiv.org/abs/2509.23661，2025。1[36] Wenyan Li、Raphael Tang、Chengzu Li、Caiqi Zhang、Ivan Vuli´c 和 Anders Søgaard。《迷失于嵌入：视觉语言模型中的信息损失》。arXiv 预印本 arXiv:2509.11986，2025。13[37] Wentong Li、Yuqian Yuan、Jian Liu、Dongqi Tang、Song Wang、Jie Qin、Jianke Zhu 和 Lei Zhang。《TokenPacker：高效视觉投影器，用于多模态大语言模型。International Journal of Computer Vision，第 1–19 页，2025 年。2, 13[38] Yifan Li、Yifan Du、Kun Zhou、Jinpeng Wang、Wayne Xin Zhao 和 Ji-Rong Wen。评估大型视觉语言模型中的物体幻觉。arXiv 预印本 arXiv:2305.10355，2023 年。6, 13, 15, 17[39] Yanwei Li、Chengyao Wang 和 Jiaya Jia。Llama-vid：在大型语言模型中，一张图像等于 2 个词元。载于《欧洲计算机视觉会议》，第 323–340 页。Springer，2024 年。2, 13[40] Bin Lin、Yang Ye、Bin Zhu、Jiaxi Cui、Munan Ning、Peng Jin 和 Li Yuan。Video-LLaVA：通过先对齐再投影学习统一的视觉表征。arXiv 预印本 arXiv:2311.10122，2023 年。6, 15, 16, 23[41] Tsung-Yi Lin、Michael Maire、Serge Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Doll´ar 和 C Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。载于《欧洲计算机视觉会议》，第 740–755 页。Springer，2014 年。15[42] Haotian Liu、Chunyuan Li、Yuheng Li 和 Yong Jae Lee。通过视觉指令微调改进基线，2023 年。1, 13, 15, 17[43] Haotian Liu、Chunyuan Li、Qingyang Wu 和 Yong Jae Lee。视觉指令微调。《神经信息处理系统进展》，36:34892–34916，2023 年。1, 13, 15, 17, 23[44] Haotian Liu、Chunyuan Li、Yuheng Li 和 Yong Jae Lee。通过视觉指令微调改进基线。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 26296–26306 页，2024 年。6, 16[45] Haotian Liu、Chunyuan Li、Yuheng Li、Bo Li、Yuanhan Zhang、Sheng Shen 和 Yong Jae Lee。LLaVA-NeXT：改进推理、光学字符识别和世界知识，2024 年。6, 16[46] Haotian Liu、Chunyuan Li、Yuheng Li、Bo Li、Yuanhan Zhang、Sheng Shen 和 Yong Jae Lee。LLaVA-NeXT：改进推理、光学字符识别和世界知识，2024 年。1, 6, 13[47] Yuan Liu、Haodong Duan、Yuanhan Zhang、Bo Li、Songyang Zhang、Wangbo Zhao、Yike Yuan、Jiaqi Wang、Conghui He、Ziwei Liu 等。MMBench：你的多模态模型是全能选手吗？载于《欧洲计算机视觉会议》，第 216–233 页。Springer，2024 年。13[48] Yuan Liu、Haodong Duan、Yuanhan Zhang、Bo Li、Songyang Zhang、Wangbo Zhao、Yike Yuan、Jiaqi Wang、Conghui He、Ziwei Liu 等。MMBench：你的多模态模型是全能选手吗？载于《欧洲计算机视觉会议》，第 216–233 页。Springer，2025 年。6, 13, 16[49] Yudong Liu、Jingwei Sun、Yueqian Lin、Jingyang Zhang、Ming Yin、Qinsi Wang、Jianyi Zhang、Hai Li 和 Yiran Chen。面向关键帧的视觉令牌剪枝：提升大型视觉语言模型在长视频处理中的效率，2025 年。13[50] Yuchen Liu、Yaoming Wang、Bowen Shi、Xiaopeng Zhang、Wenrui Dai、Chenglin Li、Hongkai Xiong 和 Qi Tian。Meteor：用于高效视觉语言模型的多编码器协同令牌剪枝，2025 年。1, 3, 13

[51] Doyen Lu 等。通过高效视觉词元压缩推进并加速 InternVL 系列。https:// arxiv.org/abs/2503.21307，2025 年。13[52] Pan Lu、Swaroop Mishra、Tanglin Xia、Liang Qiu、Kai-Wei Chang、Song-Chun Zhu、Oyvind Tafjord、Peter Clark 和 Ashwin Kalyan。学会解释：通过思维链进行多模态推理，以实现科学问答。《神经信息处理系统进展》，35 2521，2022 2507 6, 15, 17：–。[53] Gen Luo、Yiyi Zhou、Yuxin Zhang、Xiawu Zheng、Xi-aoshuai Sun 和 Rongrong Ji。大饱眼福：面向多模态大语言模型的分辨率混合自适应。arXiv 预印本 arXiv:2403.03003，2024 年。13[54] Muhammad Maaz、Hanoona Rasheed、Salman Khan 和 Fahad Khan。Video-ChatGPT：通过大型视觉语言模型实现详细的视频理解。载于《计算语言学协会第 62 届年会论文集》（第 1 卷：长篇论文），第 12585–12602 页，2024 年。16[55] Meta AI。Llama 4（Scout 和 Maverick），2025 年。开放权重多模态大语言模型，混合专家架构。1, 13[56] 月之暗面。Kimi K2：开放智能体智能。技术报告，月之暗面，2025 年。混合专家开放权重大语言模型，拥有 1 万亿参数，在智能体和编码任务中表现出色。1[57] OpenAI。gpt-oss-120b（及 20b）开放权重语言模型，

2025 2 0 1 。基

[58] 5

于 Apache 的开放权重发布。。OpenAI。gpt-系统卡。技术报告，OpenAI，2025 年。技术报告。1, 13[59] Wujian Peng、Sicheng Xie、Zuyao You、Shiyi Lan 和 Zuxuan Wu。综合、诊断与优化：迈向细粒度视觉-语言理解。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 13279–13288 页，2024 年。13[60] Charles R Qi、Li Yi、Hao Su 和 Leonidas J Guibas。Point-net++：度量空间中点集的深层次特征学习。载于 NeurIPS，

2017 5 。Bowen

[61]

Qu 等。ChartMoE：用于图表理解的多样化对齐专家混合。载于 ICLR 2025，2025 年。1[62] Guanqiao Qu、Qiyuan Chen、Wei Wei、Zheng Lin、Xianhao Chen 和 Kaibin Huang。面向大语言模型的移动边缘智能：当代综述。《IEEE Communications Surveys & Tutorials》，2025 年。1, 13[63] Qwen团队。Qwen2.5-VL 技术报告。技术报告，2025 年。1, 13[64]Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark、Gretchen Krueger 和 Ilya Sutskever。从自然语言监督中学习可迁移的视觉模型，2021 年。1, 13[65] Yuzhang Shang、Mu Cai、Bingxin Xu、Yong Jae Lee 和 Yan Yan。LLaVA-PruMerge：面向高效大型多模态模型的自适应词元缩减。arXiv 预印本 arXiv:2403 15388，2024 年 6, 7。。[66] Yuzhang Shang、Mu Cai、Bingxin Xu、Yong Jae Lee 和 Yan Yan。LLaVA-PruMerge：面向高效大型多模态模型的自适应词元缩减。载于《IEEE/CVF国际计算机视觉会议论文集》，第 22857–22867 页，2025 年。1, 2, 13[67] Zhenwei Shao、Mingyang Wang、Zhou Yu、Wenwen Pan、Yan Yang、Tao Wei、Hongyuan Zhang、Ning Mao、Wei Chen 和 Jun Yu。Growing a twig to accelerate large vision- language models。arXiv 预印本 arXiv:2503.14075，2025。1、3、13[68] Amanpreet Singh、Vivek Natarajan、Meet Shah、Yu Jiang、Xinlei Chen、Dhruv Batra、Devi Parikh 和 Marcus Rohrbach。Towards vqa models that can read。在《IEEE/CVF计算机视觉与模式识别会议论文集》中，第8317–8326页，2019。6、13、15、17[69] Shengbang Tong、Zhuang Liu、Yuexiang Zhai、Yi Ma、Yann LeCun 和 Saining Xie。Eyes wide shut? exploring the visual shortcomings of multimodal llms。在《IEEE/CVF计算机视觉与模式识别会议论文集》中，第9568–9578页，2024。13[70] Ao Wang、Fengyuan Sun、Hui Chen、Zijia Lin、Jungong Han 和 Guiguang Ding。[cls] token tells everything needed for training-free efficient mllms。arXiv 预印本 arXiv:2412.05819，2024。1、2、13[71] Weihan Wang、Qingsong Lv、Wenmeng Yu、Wenyi Hong、Ji Qi、Yan Wang、Junhui Ji、Zhuoyi Yang、Lei Zhao、Song XiX- uan 等。Cogvlm: Visual expert for pretrained language models。在《神经信息处理系统进展》中，第37卷：121475–121499，2025。6、7[72] xAI。Grok 4。https://x.ai/news/grok-4，2025。xAI推出的旗舰模型。13[73] Long Xing、Qidong Huang、Xiaoyi Dong、Jiajie Lu、Pan Zhang、Yuhang Zang、Yuhang Cao、Conghui He、Jiaqi Wang、Feng Wu 等。Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy re- duction。arXiv 预印本 arXiv:2410.17247，2024。1、3、13、17[74] Dejing Xu、Zhou Zhao、Jun Xiao、Fei Wu、Hanwang Zhang、Xiangnan He 和 Yueting Zhuang。Video question answer- ing via gradually refined attention over appearance and mo- tion。在《第25届ACM多媒体国际会议论文集》中，第1645–1653页，2017。6、16[75] Jun Xu、Tao Mei、Ting Yao 和 Yong Rui。Msr-vtt: A large video description dataset for bridging video and language。在《IEEE计算机视觉与模式识别会议论文集》中，第5288–5296页，2016。16[76] Cheng Yang、Yang Sui、Jinqi Xiao、Lingyi Huang、Yu Gong、Chendi Li、Jinghua Yan、Yu Bai、Ponnuswamy Sadayappan、Xia Hu 和 Bo Yuan。Topv: Compatible token pruning with inference time optimization for fast and low-memory multi- modal vision language model，

2025 3, 13 。Senqiao Yang、

[77]

Zhuotao Tian、Li Jiang 和 Jiaya Jia。Unified language-driven zero-shot domain adaptation。在《IEEE/CVF计算机视觉与模式识别会议论文集》中，第23407–23415页，2024。1[78] Senqiao Yang、Yukang Chen、Zhuotao Tian、Chengyao Wang、Jingyao Li、Bei Yu 和 Jiaya Jia。Visionzip: Longer is better but not necessary in vision language models。在《计算机视觉与模式识别会议论文集》中，第19792–19802页，2025。1、2、3、6、7、8、13、15、17、20[79] Sihan Yang、Runsen Xu、Chenhang Cui、Tai Wang、Dahua Lin 和 Jiangmiao Pang。Vflowopt: A token pruning frame-

work for lmms with visual information flow-guided opti- mization。在《IEEE/CVF国际计算机视觉会议论文集》中，第23924–23934页，2025。1、2、13[80] Yuhuai Yao 等。Minicpm-v: A gpt-4v level mllm on your phone。arXiv:2408.01800，2024。13[81] Weihao Ye、Qiong Wu、Wenhao Lin 和 Yiyi Zhou。Fit and prune: Fast and training-free visual token pruning for multi- modal large language models。在《AAAI人工智能会议论文集》中，第22128 22136页，2025 1、3、13 – 。[82] Xubing Ye、Yukang Gan、Yixiao Ge、Xiao-Ping Zhang 和 Yansong Tang。Atp-llava: Adaptive token pruning for large vision language models。在《计算机视觉与模式识别会议论文集》中，第24972–24982页，2025。1、3、13[83] Weihao Yu、Zhengyuan Yang、Linjie Li、Jianfeng Wang、Kevin Lin、Zicheng Liu、Xinchao Wang 和 Lijuan Wang。Mm-vet: Evaluating large multimodal models for integrated capabilities。arXiv 预印本 arXiv:2308.02490，2023。13[84] Weihao Yu、Zhengyuan Yang、Linjie Li、Jianfeng Wang、Kevin Lin、Zicheng Liu、Xinchao Wang 和 Lijuan Wang。Mm-vet: Evaluating large multimodal models for integrated capabilities。arXiv 预印本 arXiv:2308.02490，2023。6、16[85] Xi Yue 等。Mmmu: A massive multi-discipline multimodal understanding benchmark。arXiv:2311.16502，2023。1[86] Weili Zeng、Ziyuan Huang、Kaixiang Ji 和 Yichao Yan。Skip-vision: Efficient and scalable acceleration of vision- language models via adaptive token skipping。arXiv 预印本 arXiv:2503.21817，2025。1、3、13[87] Xiaohua Zhai、Basil Mustafa、Alexander Kolesnikov 和 Lucas Beyer。Sigmoid loss for language image pre-training，2023。1、13[88] Qizhe Zhang、Aosong Cheng、Ming Lu、Renrui Zhang、Zhiy- ong Zhuo、Jiajun Cao、Shaobo Guo、Qi She 和 Shanghang Zhang。Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms。在《IEEE/CVF国际计算机视觉会议论文集》中，第20857–20867页，2025。1、2、3、6、7、8、13、15、17、20、23[89] Shaolei Zhang、Qingkai Fang、Zhe Yang 和 Yang Feng。Llava-mini: Efficient image and video large mul- timodal models with one vision token。arXiv 预印本 arXiv:2501.03895，2025。2、13[90] Yuan Zhang、Chun-Kai Fan、Junpeng Ma、Wenzhao Zheng、Tao Huang、Kuan Cheng、Denis Gudovskiy、Tomoyuki Okuno、Yohei Nakata、Kurt Keutzer 等。Sparsevlm: Vi- sual token sparsification for efficient vision-language model inference。arXiv 预印本 arXiv:2410.04417，2024。1、3、6、7、8、13、15、17、20[91] Wangbo Zhao、Yizeng Han、Jiasheng Tang、Zhikai Li、Yib- ing Song、Kai Wang、Zhangyang Wang 和 Yang You。A stitch in time saves nine: Small vlm is a precise guidance for accelerating large vlms。在《计算机视觉与模式识别会议论文集》中，第19814– 19824页，2025。1、2、13[92] 智谱AI。Glm 4.5，2025。面向智能体工作流、在中国发布的开源大语言模型。1、13

[93] Yiwu Zhong、Zhuoming Liu、Yin Li 和 Liwei Wang。Aim：通过令牌合并与剪枝实现多模态大语言模型的自适应推理。发表于 *IEEE/CVF*国际计算机视觉会议论文集，第 20180–20192 页，2025 年。1，3，13

A. 相关工作

视觉语言模型。大语言模型（LLM） [1, 3, 55–58, 72, 92] 推动了多模态推理的快速发展，并促使人们开发用于图像和视频理解的视觉语言模型（VLM） [20, 43, 47, 83]。标准的 VLM [42, 43,46]将 CLIP 或 SigLIP 等预训练视觉编码器 [64, 87] 与轻量级投影器 [43]配对，将视觉图像块映射到大语言模型的嵌入空间。尽管现代 VLM 在各种任务中表现出色 [13, 63, 80], 但在细粒度感知任务上仍有困难 [59] 且容易产生视觉幻觉 [8,36, 69]。为提升视觉定位能力，人们尝试采用更高的输入分辨率 [22, 53] 和多编码器设计 [13, 51]。与此同时，视频理解和 OCR/文档任务需要处理大量帧或超高分辨率输入 [15, 24, 26, 49], , 进一步增加了视觉词元的数量。因此，VLM 每次输入都要处理数百到数千个视觉词元。视觉词元占据了预填充计算的主要部分，其计算量随令牌数量呈二次方增长，同时还会增加 KV 缓存内存和解码阶段浮点运算量，后两者均随令牌数量线性增长。因此，减少冗余视觉词元已成为实现实用且可扩展的视觉语言模型部署的关键 [30, 62]。

纯视觉剪枝。一类高效视觉语言模型研究会利用视觉编码器侧显著性信号，直接从视觉编码器中剪除视觉令牌。这些方法通常根据自注意力或相关视觉线索为图像块评分，并在词元进入语言模型之前移除不重要的词元。代表性方法包括 [66, 70], [16, 29, 79, 91]。VisionZip [78]和 Vis-Pruner [88] 通过编码器注意力为视觉图像块评分，并通过上下文合并或基于相似性的去重来压缩被丢弃的词元，从而取得了最先进的性能。这些方法虽然有效且无需训练，但本质上仍与查询无关，因为重要性完全由视觉模态决定。

纯跨模态剪枝。另一类方法利用大语言模型内部的文本-视觉交叉注意力来剪除视觉令牌。FastV [11] 利用早期层对文本查询的交叉注意力为图像令牌排名，而 SparseVLM [90] 则通过稀疏化注意力模式拓展了这一思路。后续研究探索了更深层或多阶段的交叉注意力信号，包括基于金字塔的缩减 [73], 、通过注意力聚合实现的无需训练剪枝 [23, 81], , 以及多编码器协同剪枝 [50]。其他方法则将基于注意力的重要性评估用于动态跳过或在大语言模型层中逐步移除词元 [18, 67, 76, 82, 86, 93]。尽管这些方法有效，但它们需要在大语言模型内部提取交叉注意力，从而禁用 FlashAttention、使键值缓存碎片化，并且只能产生依赖提示的稀疏性，因此经常忽略显著视觉区域。

基于训练的令牌剪枝。另一类独立的研究工作将令牌剪枝整合到视觉语言模型训练中，其中剪枝模块或压缩适配器与模型 [7, 10, 25, 37, 39, 89] 联合优化。这些方法能够在激进压缩下保持稳健性，但需要额外训练，并非即插即用。

ConsensusDrop 通过在大语言模型之前提取查询感知跨模态信号，并将其与视觉侧显著性融合，从而实现无需训练、即插即用的令牌缩减；在保持推理效率的同时，保留全局视觉上下文和查询关键图像块词元。

B. 假设评估设置

本附录介绍第 3 节受控研究的实现细节。我们的目标是计算视觉侧与跨模态显著性信号之间的一致性、不一致性和恢复行为。

B.1. 实验设置

设置。所有假设实验均使用 LLaVA-1.5-70亿参数 [42, 43] 和 CLIP-ViT-L/14，在 336×336下生成$N = 576$ 个视觉令牌（24×24 网格）。实验使用单块 NVIDIA RTX 6000 Ada GPU（48 GB）运行。

通过 FastV 进行视觉令牌剪枝。遵循 FastV [11],, 我们在大语言模型解码器内执行早期视觉令牌剪枝。令 K 表示进行令牌剪枝之前的从 *1* 开始编号的层。注意力分数从第 K层提取，剪枝后的令牌集被传递至第 $K+1$层。在所有实验中，我们设置 $K = 2$，因此注意力在第二个解码器层计算，并在第三层之前生效。

保留率与基准测试。我们在 POPE [38],MME [20], MMBench [48], MMBench-CN[48], TextVQA [68], 和 GQA [27] 上测试 $\rho \in \{0.25, 0.50\}$（保留 $K_{\text{tok}} = \lfloor \rho N \rfloor \in \{144, 288\}$ 个词元）。我们评估恢复率 $r \in \{0.1, 0.3, 0.5\}$。对于给定的K_{tok} 视觉词元保留预算，我们将学生模态选出的排名最低的 $\lfloor rK_{\text{tok}} \rfloor$个词元替换为从教师模态排名前$K_{\text{tok}}$ 的集合中抽取的词元。交换的词元不得与保留的学生词元重叠，并且在替换过程中优先使用教师模态中排名更高的词元。

B.2. 学生–教师恢复机制

我们形式化了假设评估中使用的学生–教师恢复机制，该机制也与第 4.2 节中的恢复融合器相匹配。

显著性信号。对于每张图像，我们会为 N 视觉令牌计算两个显著性向量：(i) 从视觉编码器中提取的视觉侧分数$\mathbf{s}^{(v)} \in \mathbb{R}^N$，以及 (ii) 从大语言模型解码器中提取的跨模态分数$\mathbf{s}^{(c)} \in \mathbb{R}^N$。视觉侧分数。默认情况下，我们使用视觉编码器倒数第二层自注意力层中的 CLS 到图像块注意力（t_v^{cls}），

从视觉编码器中提取。跨模态分数。跨模态显著性从解码器层 K（从 1 开始计数）中提取，即应用令牌剪枝前的最后一层。默认情况下，我们使用 $t_{\text{cross}}^{\text{last}}$，对应于从最后一个文本令牌到图像令牌的注意力。

学生/教师分配。 我们将一个模态指定为学生，另一个指定为教师：$\mathbf{s}^{(S)} \in \{\mathbf{s}^{(c)}, \mathbf{s}^{(v)}\}$，另一个是$\mathbf{s}^{(T)}$。

通过交换进行恢复。 给定保留率 ρ和交换率r，定义

$$K_{\text{tok}} = \lfloor \rho N \rfloor, \qquad M_{\text{tok}} = \lfloor r K_{\text{tok}} \rfloor.$$

恢复会保留学生排名前（$K_{\text{tok}} - M_{\text{tok}}$）的词元，然后加入 M_{tok} 个排名靠前且未被学生选中的教师词元。这实现了最小干预：教师仅在其与学生保留的集合不一致的地方提供贡献。

算法 1学生–教师恢复

要求： 学生分数 $\mathbf{s}^{(S)} \in \mathbb{R}^N$，教师分数 $\mathbf{s}^{(T)} \in \mathbb{R}^N$，保留率 ρ，交换率 r

$K_{\text{tok}} \leftarrow \lfloor \rho N \rfloor$，$M_{\text{tok}} \leftarrow \lfloor r K_{\text{tok}} \rfloor$
$\mathcal{I}_S \leftarrow \text{ARGSORTDESC}(\mathbf{s}^{(S)})[1 : K_{\text{tok}}]$ ▷ 学生顶部-K_{tok}
$\mathcal{I}_T \leftarrow \text{ARGSORTDESC}(\mathbf{s}^{(T)})[1 : K_{\text{tok}}]$ ▷ 教师前-K_{tok}
$\mathcal{I}_1 \leftarrow \mathcal{I}_S[1 : K_{\text{tok}} - M_{\text{tok}}]$ ▷保留学生前列
$\mathcal{I}_2 \leftarrow [\,]$
对于 $i = 1$ 到 K_{tok} **执行**
  如果 $|\mathcal{I}_2| = M_{\text{tok}}$ **则跳出**
  结束条件判断
  如果 $\mathcal{I}_T[i] \notin \mathcal{I}_1$ **则**
    将 $\mathcal{I}_T[i]$ 追加到 $\mathcal{I}_2$
  结束条件判断
结束循环
$\mathcal{I}_{\text{final}} \leftarrow$ 拼接$(\mathcal{I}_1, \mathcal{I}_2)$ ▷ $|\mathcal{I}_{\text{final}}| = K_{\text{tok}}$
返回 $\mathcal{I}_{\text{final}}$

一致性/不一致。 设 $\mathcal{I}_S$ 和 $\mathcal{I}_T$ 分别为学生/教师的前K_{tok}个词元集合。我们计算：

$$\text{Agreement} = \frac{|\mathcal{I}_S \cap \mathcal{I}_T|}{K_{\text{tok}}}, \qquad \text{Disagreement} = \frac{|\mathcal{I}_S \setminus \mathcal{I}_T|}{K_{\text{tok}}}.$$

校正因子。 设 c 为满足以下条件的最小前缀长度：教师前缀包含 M_{tok} 个不在 $\mathcal{I}_1$中的词元，即$|\{\mathcal{I}_T[1{:}c] \setminus \mathcal{I}_1\}| = M_{\text{tok}}$。我们将校正因子定义为

$$\text{CF} = \frac{K_{\text{tok}} - c}{K_{\text{tok}} - M_{\text{tok}}}. \tag{6}$$

直观而言，当 c较小时，排名较高的教师词元会与学生词元集合中的词元交换。这意味着教师判定为重要的词元被学生遗漏了，因此需要通过恢复进行“更强”或“更高程度”的纠正。

C. ConsensusDrop 和 EGTM 算法

讨论。 算法2 和3 概述了完整的 ConsensusDrop 推理流程及其编码器引导的令牌合并组件。ConsensusDrop 首先通过多模态共识识别出一组紧凑的显著视觉词元，然后通过 EGTM 压缩其余视觉信息，在固定词元预算下保留上下文覆盖范围。重要的是，这两个组件都是模块化的：共识阶段与具体的显著性估计器（例如 SCAP）无关，而 EGTM 只在编码器空间中运行，同时在 LLM 投影器空间中进行合并。正文重点介绍这一设计带来的实证表现和效率提升；此处提供算法，以确保内容完整且可复现。

算法 2 ConsensusDrop：用于视觉词元缩减

要求： 图像 $\mathcal{I}$、查询 $\mathcal{Q}$、视觉编码器 $\mathcal{E}_v$、投影器 $\mathcal{P}$、大语言模型 $\mathcal{L}$、Top-K 预算 K、合并预算 M

$\mathbf{s}^{(v)}$、$\mathbf{V}$，$\mathbf{K} \leftarrow \mathcal{E}_v(\mathcal{I})$ ▷ 视觉分数、词元和键
$\tilde{\mathbf{V}} \leftarrow \mathcal{P}(\mathbf{V})$ ▷ 投影到文本空间
$\mathbf{标} \leftarrow$ 标记化$(\mathcal{Q})$ ▷大语言模型分词器
$\mathbf{s}^{(c)} \leftarrow$SCAP$(\mathbf{T}, \tilde{\mathbf{V}})$ ▷ 跨模态分数
$\mathbf{s}$ $\mathcal{I}_{\text{top}} \leftarrow$ Fuser$(\mathbf{s}^{(v)}, \mathbf{s}^{(c)}, K)$ ▷ 融合分数和前K 项索引
$\mathcal{I}_{\text{non}} \leftarrow [1, \ldots, N] \setminus \mathcal{I}_{\text{top}}$ ▷非前K 项索引
$\tilde{\mathbf{V}}_{\text{top}} \leftarrow \tilde{\mathbf{V}}[\mathcal{I}_{\text{top}}]$
$\tilde{\mathbf{V}}_{\text{merge}} \leftarrow$EGTM$(\tilde{\mathbf{V}}, \mathbf{V}, \mathbf{K}\ \mathcal{I}_{\text{non}}, M)$ ▷ 合并非前K项
$\mathbf{V}_{\text{final}} \leftarrow$ 拼接$(\mathbf{V}_{\text{top}}, \mathbf{V}_{\text{merge}})$
$y \leftarrow \mathcal{L}(\mathbf{V}_{\text{final}}, \mathbf{T})$ ▷ 大语言模型推理
返回 y ▷ 最终输出

D. 方法与实现细节

D.1. SCAP聚合策略

在正文（第4.1节）中，我们采用**全词元**聚合作为默认策略，将原始文本到视觉注意力块$\mathbf{A} \in \mathbb{R}^{L \times N}$ 转换为最终的跨模态显著性向量$\mathbf{s}^{(c)} \in \mathbb{R}^N$。在此，我们正式定义消融研究中探讨的三种变体。回顾一下，$\mathbf{A}_{i,j}$ 表示从第 i个文本令牌到第 j个视觉词元的注意力权重，而 $\text{Norm}(\mathbf{z}) = \mathbf{z}/(\sum_k z_k + \epsilon)$ 表示逐行归一化。

全词元聚合（默认）。我们对所有 L 查询token 的归一化注意力分数取平均值。该策略能够捕捉视觉特征与整个文本指令之间的全局语义对齐：

$$\mathbf{s}_{\text{all}}^{(c)} = \frac{1}{L} \sum_{i=1}^{L} \text{Norm}(\mathbf{A}_{i,:}). \tag{7}$$

算法 3 编码器引导的令牌合并（EGTM）

要求： 投影视觉令牌 $\tilde{\mathbf{V}} \in \mathbb{R}^{N\times d}$、编码器图像块特征 $\mathbf{V} \in \mathbb{R}^{N\times d_v}$、编码器键 $\mathbf{K} \in \mathbb{R}^{H_v\times N\times d_k}$、前-$K$ 个索引 $\mathcal{I}_{\text{top}}$，合并预算 M

$\mathcal{I}_{\text{non}} \leftarrow [N] \setminus \mathcal{I}_{\text{top}}$
$\mathbf{V}_{\text{non}} \leftarrow$ 归一化$(\mathbf{V}[\mathcal{I}_{\text{non}}])$
$\mathcal{A},\ \mathcal{U} \leftarrow \text{FPS}(\mathbf{V}_{\text{non}},\ M)$ ▷ $|\mathcal{A}| = M,\ \mathcal{U} = \mathcal{I}_{\text{non}} \setminus \mathcal{A}$
$\mathbf{G} \leftarrow$ 归一化$\left(\frac{1}{H_v}\sum_{h=1}^{H_v}\mathbf{K}^{(h)}\right)$
$\mathbf{G}_{\text{non}} \leftarrow \mathbf{G}[\mathcal{I}_{\text{non}}]$
对每个 $u \in \mathcal{U}$ **执行**
  $a^\star(u) \leftarrow \arg\max_{a\in\mathcal{A}} \langle \mathbf{g}_u, \mathbf{g}_a \rangle$
结束循环
对每个 $a \in \mathcal{A}$ **执行**
  $\mathcal{C}(a) \leftarrow \{a\} \cup \{u \in \mathcal{U} : a^\star(u) = a\}$
  $\tilde{\mathbf{v}}_a^{\text{合并}} \leftarrow \frac{1}{|\mathcal{C}(a)|}\sum_{j\in\mathcal{C}(a)} \tilde{\mathbf{v}}_j$
结束循环
$\tilde{\mathbf{V}}_{\text{final}} \leftarrow$ 拼接$(\tilde{\mathbf{V}}[\mathcal{I}_{\text{top}}],\ \{\tilde{\mathbf{v}}_a^{\text{merge}}\}_{a\in\mathcal{A}})$
返回 $\tilde{\mathbf{V}}_{\text{final}} \in \mathbb{R}^{(K+M)\times d}$

末词元聚合。该策略仅依赖最后一个文本令牌（通常是指令终止符或分隔符），其假设因果注意力机制会将完整上下文聚合到最后一个位置：

$$\mathbf{s}_{\text{last}}^{(c)} = \text{Norm}(\mathbf{A}_{L,:}). \tag{8}$$

最大词元聚合。该策略会选择视觉词元从任何文本令牌接收到的最大注意力分数。它优先考虑峰值对齐信号（例如，某个特定关键词强烈关注某个图像块），而不考虑其他词元：

$$\mathbf{s}_{\text{max}}^{(c)} = \text{Norm}\left(\max_{i=1}^{L} \mathbf{A}_{i,:}\right). \tag{9}$$

D.2. 融合器策略

在第4.2节中，我们介绍了凸融合器，作为我们的主要方法。在此，我们将介绍另一种**恢复融合器**，并对两种设计进行可视化。图8展示了这两种融合策略。

恢复融合器。此外，我们还考虑一种受第 3 节受控分析启发、基于恢复的融合器。一种模态被视为学生，另一种则被视为教师。从学生排名前K 个词元开始，我们丢弃排名最低的 rK 个词元，并用教师中排名最高且不重叠的词元替换，其中$r \in [0, 1]$ 表示恢复率。该机制直接作用于排序索引，而非融合分布。除非另有说明，我们默认使用凸融合器，并主要通过恢复融合来验证第3节的分析。

E. 实验设置与其他细节

E.1. 数据集

我们在总计 13 个广泛使用的基准测试上评估 ConsensusDrop，其中包括 10 个基于图像的基准测试和 3 个基于视频的基准测试。这些数据集涵盖广泛的视觉-语言能力，包括视觉问答、多模态推理、幻觉检测和视频理解。除非另有说明，所有推理协议和评估指标均遵循 LLaVA-1.5 [42, 43] 和 Video-LLaVA [40]采用的标准设置。

E.1.1. 图像基准

我们在 10 个常用于评估大型视觉语言模型的图像基准测试上进行实验。其中包括四个视觉问答数据集，以及六个多模态推理与鲁棒性基准测试。

视觉问答2 [21]。 VQAv2 评估开放式视觉问答，重点在于减少语言偏差。该数据集由 MSCOCO [41],中的图像组成，每个问题都与多张对应不同答案的图像配对。我们采用标准评估协议，在包含 107,394 个图像—问题对的 test-dev 划分上报告准确率。

GQA [27]。 GQA 专注于组合推理和结构化视觉理解。问题根据 Visual Genome [32], 中的场景图生成，确保每个问题都对应一条明确的语义推理路径。我们在包含 12,578 个图像—问题对的 test-dev 划分上评估准确率。

ScienceQA（SQA-IMG） [52]。 ScienceQA 使用涵盖自然科学、社会科学和语言主题的多项选择题评估科学推理。我们采用基于图像的子集（SQA-IMG），其中包含来自测试集的 2,017 个图像—问题对，并遵循先前的研究[78, 88, 90]。

TextVQA [68]。 TextVQA 用于衡量模型对图像中嵌入的文本内容（如标志和产品标签）进行推理的能力。图像主要来源于 Open Images [31]。我们报告验证集上的准确率，该验证集包含 5,000 个图像—问题对。

POPE [38]。 POPE通过有关物体是否存在的二元问题，评估视觉语言模型中的物体幻觉。图像取自 MSCOCO [41]。我们按照标准协议，报告三个评测划分上的平均 F1 分数，共计 8,910 个图像—问题对。

MME [19]。 MME 对涵盖 14 个子任务的多模态感知与认知进行全面评估，包括光学字符识别、物体识别和细粒度视觉理解。所有问题均为二元判断任务。我们按照先前工作的做法 [88], ，报告感知分数，共计 2,374 个图像—问题对。

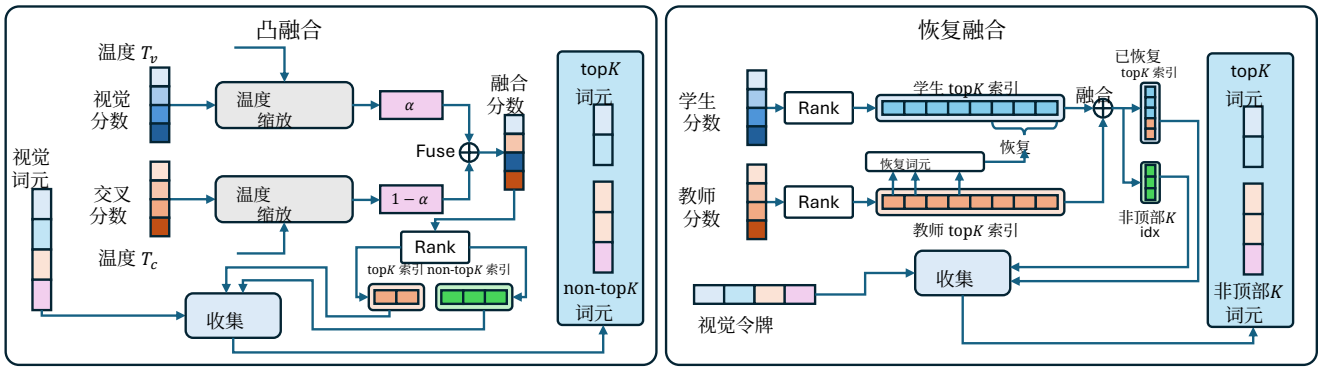


图8。**融合器模块设计**。融合器模块有两种变体：**凸**和**恢复**。左图：凸融合器对视觉分数和跨模态分数进行温度归一化，并以权重 α 形成凸组合，从而获得统一重要性分布。右图：恢复融合器将跨模态分数视为学生，将视觉分数视为教师，从学生中选取前K 个词元，并根据恢复率恢复一小组排名靠前的教师词元。

MMBench 和 MMBench-CN [48]。 MMBench 通过按分层能力等级编排的多项选择题，评估多模态感知与推理。我们评测了英文版（MMBench）和中文版（MMBench-CN），两者分别包含 4,377 和 4,329 个图像—问题对。

MM-Vet [84]。 MM-Vet 侧重于综合视觉—语言能力，例如光学字符识别、空间推理、知识锚定和数学推理。该基准测试包含 218 个经过精心整理的图像—问题对，并使用基于GPT的评估器为模型输出评分。

SEEDBench [33]。 SEEDBench 是一个大规模多模态基准测试，旨在评估模型对图像和视频的细粒度视觉理解与推理能力。它包含约 1.9 万道由人工评估者标注的多项选择题，涵盖 12 个不同的评估方面，包括物体识别、空间理解、时间推理和跨模态理解。在本文中，我们仅将 SEEDBench 用于消融研究，以便在受控设置下分析不同词元选择和融合组件的行为。

E.1.2. 视频基准测试

为评估模型在视觉冗余度较高时的表现，我们还在 Video-LLaVA [40]使用的三个视频问答基准测试上进行了评估。评估遵循 Video-ChatGPT 协议 [54], ，并使用 GPT-3.5-TURBO 作为评分助手。我们遵循 FastV [11]，并在每个基准测试的前 1,000 个样本上进行评估。

TGIF-QA [28]。 TGIF-QA 将视觉问答扩展到动画 GIF，要求模型同时对空间和时间信息进行推理。我们在帧问答任务子集上评估模型表现。

MSVD-QA [74]. MSVD-QA 基于 Microsoft Research Video Description 语料库 [9], 构建，其问题–答案对源自视频字幕。该数据集包含 1,970 个视频片段以及超过 5 万个问答对。

MSRVTT-QA [74]。 MSRVTT-QA 基于 MSRVTT 数据集 [75], ，该数据集包含比 MSVD [9]更多样、更复杂的视频场景。该基准测试包含 10K 个视频片段和 243K 个问答对。

E.2. 模型架构

我们在涵盖图像与视频理解的代表性先进视觉语言模型上评估 ConsensusDrop。所有模型均采用其公开发布的配置，且未进行架构修改。

LLaVA-1.5 [44]。 LLaVA-1.5 是一种广泛采用的开源视觉—语言模型，结合了基于 CLIP 的视觉编码器与 Vicuna 语言模型。视觉编码器提取的视觉特征通过线性投影层投射到语言嵌入空间，使大语言模型能够直接处理视觉词元。与原始 LLaVA 相比，LLaVA-1.5 将图像输入分辨率从 224×224 提高到 336×336，并加入了更多指令微调数据，因此在多模态基准测试中取得了显著提升。

LLaVA-NeXT (LLaVA-1.6) [45]。 LLaVA-NeXT 在 LLaVA-1.5 的基础上，支持动态高分辨率图像输入。模型不再采用固定分辨率，而是自适应地选择宽高比，并将高分辨率图像划分为多个子图像，分别由视觉编码器独立处理。随后，将得到的视觉表征拼接起来，再传递给语言模型。这种设计提升了模型在需要细粒度感知的任务（如光学字符识别和空间推理）上的表现，同时保留了底层架构。

Video-LLaVA [40]. Video-LLaVA将LLaVA架构扩展到视频理解，使用相同的视觉编码器对各个视频帧进行编码并对齐-

其表示，再投影到语言模型中。帧级特征经拼接形成统一的视觉词元序列，从而能够对空间与时间信息进行联合推理。经过多模态对齐训练后，Video-LLaVA在单一统一框架内支持两者，即基于图像和视频的问答。

Qwen-VL [4]。 Qwen-VL 是一种开源视觉—语言模型，将 OpenCLIP 视觉编码器与 Qwen 语言模型结合在一起。与 LLaVA 的线性投影不同，Qwen-VL 采用基于交叉注意力的视觉—语言适配器，将视觉特征映射为固定长度的词元序列。该模型经过多个阶段的训练，以对齐视觉和文本表征；Qwen-VL-Chat 则进一步针对对话式多模态任务进行了微调。

E.3. 实现细节

静态交叉注意力探针（SCAP）。 SCAP 被实例化为首个解码器层注意力模块的轻量级副本，并在预填充阶段执行一次。它能够提取查询感知的文本—视觉显著性，无需维护键/值缓存，也不会修改大语言模型的前向传播。因此，SCAP 不会干扰后续解码过程中使用的优化注意力内核（例如 FlashAttention）。

无需训练的部署。 ConsensusDrop 不会引入额外的可训练参数，且完全在推理时应用。SCAP 参数从基础模型深拷贝而来，并在所有实验中保持冻结，确保性能提升完全源于词元选择的改进，而非针对特定任务的适配。

E.4. 效率分析设置

所有效率测量（第 5.6 节）均在单台 NVIDIA RTX 6000 Ada GPU（48 GB）上进行，使用 CUDA 12.1、PyTorch 2.1.2 和 Hugging Face Transformers 库。除非另有说明，我们使用启用了缩放点积注意力（SDPA）的 LLaVA-1.5-7B，并在各方法间采用相同的生成设置进行评估。

首个令牌时间（TTFT）。 TTFT 衡量预填充延迟，定义为从提交输入到生成首个输出词元所经过的时间。该指标反映处理完整多模态提示并初始化键值缓存的开销。我们采用 GPU 同步的墙钟计时来报告 TTFT，并在预热后对多次运行结果取平均值。

每个输出令牌时间（TPOT）。 TPOT 衡量每个生成令牌的稳态解码延迟。我们通过启用贪心解码和键值缓存，生成固定长度的输出序列来计算 TPOT，并报告排除预填充阶段后的平均每令牌延迟。该指标反映初始提示处理完毕后的解码成本。

键值缓存占用。 键值缓存内存占用根据剪枝后保留的视觉词元数量进行解析估算。对于 LLaVA-1.5-7B（以 LLaMA-7B 为骨干网络），缓存大小计算为 KV (MB) $= \frac{2\times L\times H\times S\times D\times B}{2^{20}}$，其中 $L = 32$ 是 Transformer 层数，$H = 32$ 是键值头数，$D = 128$ 是头维度，S 是序列长度（视觉和文本令牌），$B = 2$ 是半精度浮点数所占的字节数。我们报告预填充后立即测得的键值缓存大小，以隔离视觉词元缩减的影响。所有报告的键值缓存大小均对应于跨所有 Transformer 层汇总的净 内存占用。对于在大语言模型内部剪枝词元的方法，其剪枝比例会随阶段变化（[73, 90]）；此估算未考虑潜在的键值缓存内存碎片，而这在实际应用中可能会增加有效内存用量。

评估一致性。 所有方法（FastV [11], Sparse-VLM [90], VisionZip [78], VisPruner [88], 和 ConsensusDrop）均使用各自的官方代码库，在相同条件下进行评估，包括输入图像、提示、保留预算、解码策略和硬件。基线结果对应于 LLaVA-1.5-7B 的完整视觉词元序列（576 个词元）。

F. 消融研究

我们在 LLaVA-1.5-7B 上进行了大量消融研究[42, 43]，以分析 ConsensusDrop 中各组件的作用。所有实验均保留 128 个视觉词元（总计 576 个），并在五个基准测试上进行评估：MME[20], POPE [38], ScienceQA [52], SEED-Bench [33], 和 TextVQA [68]。除非另有说明，我们使用默认配置：凸融合器，其中 $\alpha = 0.7$，$\tau_v = \tau_c = 1.0$，以及包含 $K = 108$ 个所选词元和 $M = 20$ 个锚点词元的 EGTM。默认的视觉侧显著性为 CLS 到图像块的注意力分数（从视觉编码器的倒数第二层提取）；默认的跨模态显著性为从 SCAP（第 4.1 节）提取的聚合查询到图像块分数。

F.1 融合权重（α）

凸融合权重 α 控制着融合分数中视觉侧显著性（$\mathbf{s}^{(v)}$）与跨模态显著性（$\mathbf{s}^{(c)}$）之间的平衡：$\mathbf{s} = \alpha \cdot \mathbf{s}^{(v)} + (1-\alpha) \cdot \mathbf{s}^{(c)}$。我们评估$\alpha \in \{0.3, 0.5, 0.7\}$，其中较高的取值会赋予视觉编码器的显著性更大的权重。

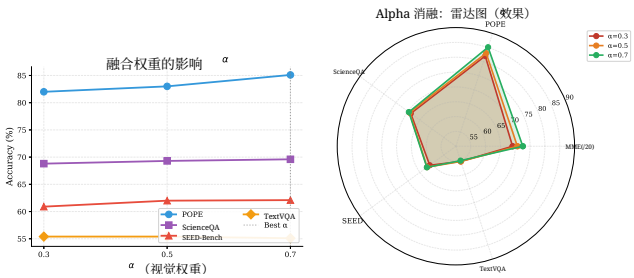
如表5 和图9所示，随着 α 从 0.3 增加到 0.7，性能大致呈单调提升。提升幅度最大的是 POPE（+3.1 F1值）和 MME（+68.8），这两项任务都要求准确的物体定位 [20, 38]。这表明，视觉编码器显著性能够捕捉空间连贯的区域，这些区域对幻觉敏感任务至关重要；而跨模态注意力则提供查询特定细化，补足这一点。

α	MME	POPE	ScienceQA	SEED	TextVQA
0.3	1382.3	82.0	68.8	60.9	55.4
0.5	1414.5	83.0	69.3	62.0	55.4
0.7	**1451.1**	**85.1**	**69.6**	**62.1**	55.1

α α =

表 5. 融合权重 0.7

的影响。较高的视觉权重（ ）始终带来更强的性能，这支持了以下假设：视觉编码器显著性提供了可靠的基础信号，而跨模态注意力则对其进行细化。



（a）准确率与融合权重 α的关系。 （b）所有基准测试的雷达图标记。

图9. 不同基准测试上的融合权重 α 的影响。提高视觉权重能够稳定提升性能，其中 $\alpha = 0.7$ 实现了最佳权衡。

τ_v	τ_c	MME	POPE	ScienceQA	SEED	TextVQA
0.7	0.7	1394.2	82.0	69.3	60.6	55.1
0.7	1.3	1418.3	82.3	68.7	61.3	55.3
1.3	0.7	1409.4	82.6	69.4	61.2	55.4
1.3	1.3	**1416.9**	**83.8**	69.1	**61.9**	55.3

表6. 温度缩放的影响。在所有配置中，性能变化幅度为 <2%，表明模型对这一超参数具有鲁棒性。

补充查询特定细化。性能单调提升验证了我们的设计直觉：视觉编码器的自注意力已编码语义分组（物体、纹理、边界），因此是一种可靠的“基础信号”，跨模态注意力应在此基础上进行细化，而不是取而代之。

F.2. 温度缩放（τ_v, τ_c）

温度缩放会在融合前使显著性分布变得更尖锐或更平滑。我们测试 τ_v， $\tau_c \in \{0.7, 1.3\}$，其中较低的温度会产生更尖锐（峰值更突出）的分布，较高的温度则会产生更平滑的分布。这项消融实验探究 ConsensusDrop 是否能从放大高置信度词元（低 τ）或保留分布熵（高 τ）中获益。

表。6和图。10 表明 ConsensusDrop 对温度缩放基本不敏感：在 POPE 上的最大变化为 1.8%，在 SEED-Bench 上为 2.1%。这

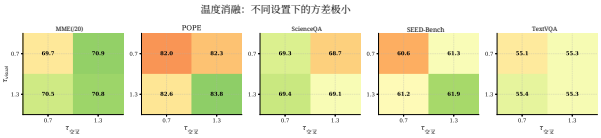


图10. 所有基准测试上的温度消融热图。方差极小（<2%）表明模型对温度缩放具有鲁棒性；为便于视觉比较，MME 数值显示为 $\times 1/20$。

学生	γ	MME	POPE	ScienceQA	SEED	TextVQA
视觉	0.1	1402.9	**83.8**	68.7	**61.3**	**55.3**
	0.3	1404.8	82.9	69.1	60.7	55.2
	0.5	1390.7	81.7	**69.2**	59.9	55.2
交叉	0.1	1320.4	80.4	69.0	58.9	54.1
	0.3	1398.2	81.3	68.9	59.3	54.9
	0.5	**1412.0**	82.2	68.6	60.1	55.1

表7. 恢复融合器消融实验。将视觉作为学生并采用极低的恢复量（$\gamma = 0.1$），可在对幻觉敏感的指标（POPE、SEED-Bench）上取得最佳表现；而将跨模态作为学生，则需要更高的恢复率来弥补其较弱的基础信号。

种稳健性对实际部署十分理想，因为它消除了数据集特定调优的需要。在 POPE 和 SEED-Bench 上略微偏好较高温度（$\tau_v = \tau_c = 1.3$），这表明保留分布熵对幻觉敏感任务略有帮助，可能是因为它避免了概率过度集中于少数占主导地位的词元。为简便起见，我们采用 $\tau_v = \tau_c = 1.0$ 作为默认值，因为它位于稳健的工作区间内。

F.3. 恢复融合器：学生模态与恢复率

恢复融合器实现了一种学生–教师校正机制（第4.2节），其中学生模态中排名最低的词元会被教师的高排名词元替换。我们消融了（i）哪种模态作为学生（视觉或跨模态），以及（ii）恢复率$\gamma \in \{0.1, 0.3, 0.5\}$，它控制学生排名最低的词元中有多大比例与教师排名最高的词元进行交换。

表7和图11揭示了明显的不对称性：视觉作为学生始终优于跨模态作为学生，在所有恢复率下均如此，且差距在低恢复率时最为显著（POPE：83.8 对 80.4，位于 $\gamma = 0.1$）。这与我们的假设一致，即视觉编码器提供稳定的基础信号，而跨模态注意力应作为校正性教师，而非主要选择器。

有趣的是，当视觉作为学生时，较高的恢复率（$\gamma = 0.5$）会降低 POPE 和 SEED-Bench 的表现，这表明过度纠正会破坏编码器连贯的空间信号。相反，跨模态作为学生 *im-*

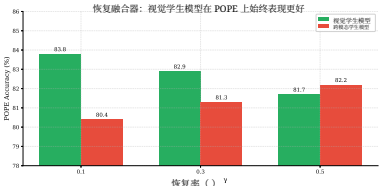


图11。恢复融合器：POPE F1 与恢复率 γ 的关系。视觉作为学生（绿色）始终优于跨模态作为学生（红色），且恢复率越低，差距越大。

K	M	MME	POPE	ScienceQA	SEED	TextVQA
128	0	1410.2	81.3	69.4	61.5	55.0
118	10	1415.8	82.8	69.6	61.8	55.1
108	20	**1441.2**	**84.9**	69.5	**63.2**	**57.0**
98	30	1430.0	83.6	69.1	62.0	55.0
88	40	1420.6	84.3	69.0	62.2	55.4
78	50	1418.7	84.7	**69.3**	62.1	55.5

表8。EGTM锚点令牌消融实验。 $M = 20$ 实现了最优平衡，相比纯选择（$M = 0$），POPE 的 F1 值提升了 +3.6，TextVQA 提升了 +2.0%。

*proves*在恢复率较高时，表现有所提升，这表明仅靠交叉模态显著性并不足够，还需要视觉教师进行大量纠正。这种不对称性为我们选择以视觉为权重的融合设计提供了机制层面的证据。

F.4. 锚点词元（EGTM）

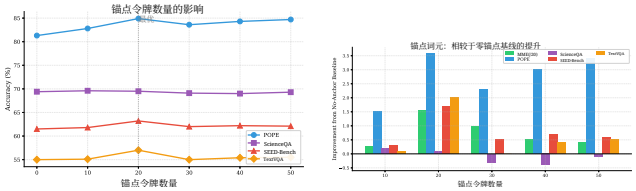
EGTM 将令牌预算划分为 K 个直接选择的令牌和 M 个合并锚点令牌，使得 $K + M$ =128。我们对 $M \in \{0, 10, 20, 30, 40, 50\}$ 进行扫描（相应地，$K \in \{128, 118, 108, 98, 88, 78\}$），以研究合并在选择保真度与上下文保留之间的权衡。

表 8和图 12表明，与纯 top-K 选择相比，EGTM 始终能带来提升。最佳点出现在 $M = 20$：与 $M = 0$（不合并）相比，此配置使 POPE 提升了 +3.6 F1值，SEED-Bench 提升了+1.7%，TextVQA 提升了 +2.0%。当 $M > 20$ 时，性能略有下降，因为过度合并会稀释锚点词元中保留的信息——合并后的表示会变成过度平滑的平均表示，而非聚焦摘要。

这验证了我们的 EGTM 设计：适度的合并预算可以从被丢弃的词元（背景、周边物体）中恢复上下文信息，同时不牺牲针对显著前景区域进行 top-K 选择的精度。

F.5. 分数类型

我们对两种模态的显著性提取方式进行消融研究：(i)视觉分数类型 $\in$ {词元、类别}，其中“词元”计算块间注意力的平均值，而“类



(a) 准确率与锚点数量 M。 (b) 相较于 $M = 0$ 基准线的改进。

图 12. EGTM 锚点词元权衡。(a) 所有基准测试的峰值都接近 $M = 20$。(b) 合并相比纯选择最多可提升 +3.6%，验证了保留上下文的价值。

视觉	交叉	MME	POPE	ScienceQA	SEED	TextVQA	平均值
词元	all	1431.1	83.9	68.8	61.5	55.5	67.4
类别	all	**1467.3**	83.7	69.1	**63.4**	**56.2**	**68.1**
词元	last	1443.5	82.3	68.0	61.8	55.8	67.0
类别	last	1420.5	**83.7**	**69.2**	61.9	55.2	67.5
词元	max	1436.5	83.6	69.7	61.0	55.6	67.5
类别	max	1424.1	83.5	69.1	62.0	55.2	67.5

表9. 分数类型消融实验。“class+all”配置的平均准确率最高（比次佳配置高+0.7%），且在 MME、SEED-Bench 和 TextVQA 上得分最优。

别”使用视觉编码器中的 CLS 到图像块注意力；以及 (ii) 跨模态分数类型 $\in$ {全部、最后、最大值}，其中“全部”对所有文本令牌到每个视觉词元的注意力取平均值，“最后”仅使用最后一个文本令牌的注意力，而“最大值”则对文本令牌取逐元素最大值。

表 9 和图13 表明，“class+all”配置的总体表现最佳。CLS 到图像块注意力提供了全局感知的显著性信号，优于局部块间注意力，尤其是在 MME（+36.2，高于 tokens+all）和 SEED-Bench（+1.9%）上。这是意料之中的：CLS 令牌在视觉编码过程中汇聚全局上下文，因此相比局部图像块交互，其注意力模式更能体现“图像级重要性” [17]。

对所有文本令牌的跨模态注意力取平均（“all”），效果优于“last”和“max”，这可能是因为它能捕捉查询意图的整体信息。仅使用最后一个令牌（“last”）可能会过度看重位置上的靠后程度，而“max”则容易受到个别文本令牌上异常注意力峰值的影响。

F.6. 总结

图15 和图14 总结了每项消融研究的最佳配置。在所有实验中，默认的 ConsensusDrop 配置（$\alpha = 0.7$，$\tau_v = \tau_c = 1.0$，采用 $\gamma = 0.1$的视觉学生模型，$K = 108$，$M = 20$，类别+全部评分）始终表现出色，在

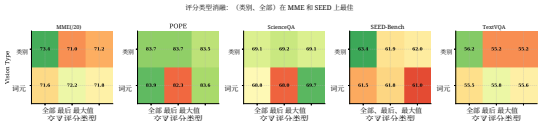


图13. 评分类型消融热力图。“类别+全部”配置（高亮显示）在大多数基准测试中表现最强，MME值如 $\times 1/20$所示。

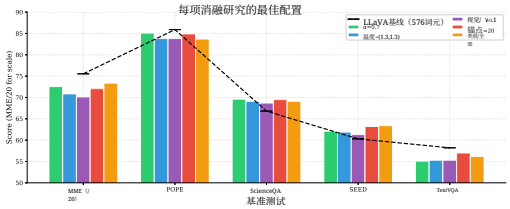


图14. 与LLaVA基线（虚线）的分组柱状图比较。所有最佳配置都以仅使用128个词元（减少78%）的代价，达到了接近LLaVA基线（576个词元）的表现。评分类型（类别+全部）和锚点词元（$M = 20$）取得了最高的绝对得分。

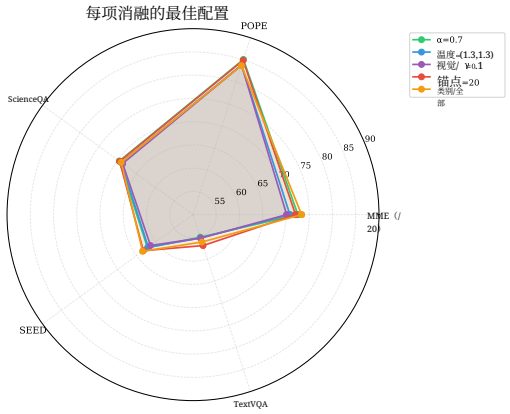


图15. 雷达图显示，所有最优配置都紧密聚类。

幻觉敏感基准测试上保留了超过 95% 的基线准确率，同时将视觉词元减少了 78%。主要发现包括：

• **视觉加权融合** ($\alpha = 0.7$)比均衡融合或交叉加权融合高出 +68.8 MME和 +3.1 POPE F1值，表明编码器显著性提供了可靠的基础信号，跨模态注意力应在此基础上进行优化。

• **温度缩放**的影响极小（所有基准测试中的变化幅度为<2%），这表明该方法具有稳健性，无需针对特定数据集进行调优。

•**视觉作为学生**在恢复融合器中的表现优于跨模态作为学生，在恢复率较低时 POPE F1 最多高出 +3.4，支持视觉（基础）与跨模态（细化）所发挥的不对称作用。

• **采用 $M = 20$锚点的 EGTM**相比纯粹的 Top-K 选择，POPE F1 提高了 +3.6，TextVQA 提高了 +2.0，这验证了通过令牌合并来保留上下文信息的价值——令牌合并。

• **CLS 到图像块注意力**（“类别”）与全文平均（“全部”）相结合，构成最佳得分提取策略，利用了两种模态的全局上下文。

G. 补充效率分析

本节全面对比 ConsensusDrop、VisionZip [78], VisPruner [88], 和 SparseVLM [90]在 LLaVA-1.5-7B 和 LLaVA-1.5-13B 模型上的效率基准。实验设置与第E.4节所述完全相同。所有方法均在匹配的词元保留预算下进行评估。

G.1. 效率指标汇总

我们报告第E.4节定义的指标，以及**GPU峰值显存**，即推理期间的最大内存消耗。对于每项指标，我们都计算绝对值以及相对于基线（576 个视觉词元）的提升幅度。

G.2. LLaVA-1.5-7B结果

表11 展示了 LLaVA-1.5-7B 在五种令牌缩减比例（25%–95%）下的详细效率指标。在中等缩减级别（50%）下，ConsensusDrop 实现了 **1.35×TTFT 加速比**，并将 **键值缓存减少了 46%**，与 VisionZip 相近 [78]，同时在生成延迟方面优于 SparseVLM。在激进缩减（90%）下，ConsensusDrop 实现了 **1.71×TTFT 加速比**，并实现**83% 的键值缓存减少量**。

值得注意的是，尽管 VisPruner [78] 在某些运行点的 TTFT 略快，但这是以迭代剪枝循环为代价的，该循环会产生额外的计算开销，表现为更高的浮点运算次数。SparseVLM 的 [90] 渐进式逐层剪枝策略会导致 GPU 峰值显存显著升高（在 50% 缩减时比 ConsensusDrop 高出 +3.7 GB），原因是其采用了多阶段键值缓存管理。

G.3. LLaVA-1.5-13B 结果

表 10 将分析扩展到更大的 LLaVA-1.5-13B 模型。视觉token剪枝带来的效率提升会随模型规模稳定地扩展。在减少量为 50% 时，ConsensusDrop 实现了 **1.56×TTFT加速比**和 **46% 的 KV缓存减少量**。与 SparseVLM 相比，更大的模型进一步放大了 ConsensusDrop 在内存方面的优势，后者在减少量为 50% 时需要额外 **+6.0 GB 的 GPU 内存**。

G.4. 效率与准确率的权衡

效率分析的关键发现是，**Consen- susDrop 实现了帕累托最优的准确率—效率权衡**。虽然 VisPruner 在

表10。LLaVA-1.5-13B（576个视觉词元基线）上的效率比较。较大的模型通过令牌剪枝获得的效率提升也相应更大。 † SparseVLM 使用渐进式逐层剪枝；参见表11 的说明文字。

方法	减少量	# 词元	首令牌生成时间（毫秒）	加速比	浮点运算次数（T）	键值缓存（MB）	GPU 内存（GB）	生成时间（毫秒）
基线	0%	576	156.5	1.00×	16.56	489	26.70	824.6
令牌缩减 25%								
ConsensusDrop（我们的方法）	25%	432	138.8	1.13×	12.84	377	26.66	800.4
VisionZip	25%	432	135.1	1.16×	12.84	377	26.47	792.5
VisPruner	25%	432	112.9	1.39×	13.24	377	25.87	762.7
SparseVLM†	25%	432	92.0	1.70×	10.61	334	32.40	960.7
令牌缩减 50%								
ConsensusDrop（我们的方法）	50%	288	106.7	1.47×	9.13	264	26.42	758.9
VisionZip	50%	288	103.5	1.51×	9.13	264	26.22	752.6
VisPruner	50%	288	84.7	1.85×	9.52	264	25.64	700.9
SparseVLM†	50%	288	93.1	1.68×	6.75	234	32.39	962.7
75% 令牌缩减								
ConsensusDrop（我们的方法）	75%	144	82.5	1.90×	5.45	152	26.40	729.1
VisionZip	75%	144	80.6	1.94×	5.45	152	26.20	724.4
VisPruner	75%	144	70.5	2.22×	5.83	152	25.41	639.0
SparseVLM†	75%	144	83.9	1.87×	4.19	155	32.30	952.8
令牌缩减 90%								
ConsensusDrop（我们的方法）	90%	57	66.5	2.35×	3.23	84	26.40	710.6
VisionZip	90%	57	62.4	2.51×	3.23	84	26.20	706.6
VisPruner	90%	57	64.4	2.43×	3.60	84	25.37	601.7
SparseVLM†	90%	57	77.3	2.02×	2.99	109	32.28	948.3
95% 令牌缩减								
ConsensusDrop（我们的方法）	95%	28	64.3	2.43×	2.49	61	26.40	706.1
VisionZip	95%	28	60.2	2.60×	2.49	61	26.20	702.2
VisPruner	95%	28	64.5	2.43×	2.86	61	25.37	589.2
SparseVLM†	95%	28	75.7	2.07×	2.61	93	32.31	944.4

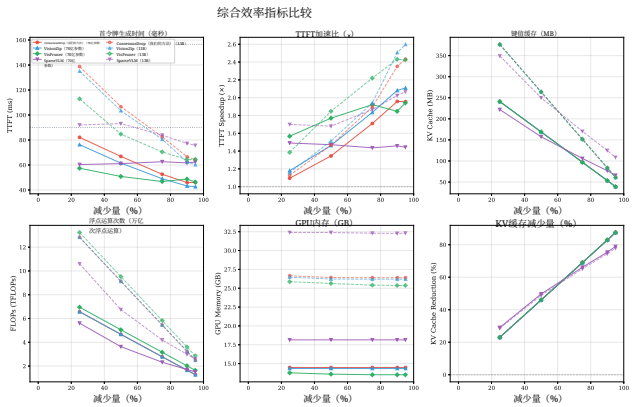


图16。不同令牌缩减比例下的综合效率指标比较。每个子图展示一种不同的指标：（**a**）TTFT延迟、（**b**）相对于基线的TTFT加速比、（**c**）KV缓存内存占用、（**d**）计算FLOPs、（**e**）GPU内存峰值，以及（**f**）KV缓存缩减百分比。实线表示LLaVA-1.5-7B；虚线表示LLaVA-1.5-13B。ConsensusDrop（红色）实现了与VisionZip相当的效率特征，同时避免了SparseVLM的内存开销。

某些运行点的首令牌生成时间更快，但代价是：**(i)** 由于其迭代剪枝循环，计算FLOPs更高；**(ii)** 在激进缩减比例下，准确率（以及推理性能）有所下降（第5节）。同样，虽然 SparseVLM 的渐进式剪枝能够实现有竞争力的首令牌生成时间，但它需要大量更多的 GPU 内存，且端到端生成时间更慢。

ConsensusDrop 的效率特征如下：

- **稳定的 TTFT 加速比**（1.35–2.0×），并随令牌缩减比例增加
- **线性减少键值缓存**，减少量与令牌缩减量成正比
- **内存开销极小**，相较于基线如此；而 SparseVLM 的内存开销为 +3–6 GB
- **生成延迟稳定**，不会因激进剪枝而下降

图17展示了这两种模型规模下的趋势，表明 ConsensusDrop 在保持准确率的同时，仍具备良好的效率特性（第 5节）。

图 18 提供了在固定缩减级别下各方法TTFT加速比的分组比较。在中等缩减（50%）时，所有方法均取得了有意义的

表11. LLaVA-1.5-7B上的效率对比（576个视觉令牌基线）。ConsensusDrop实现了具有竞争力的TTFT加速比，同时内存开销低于SparseVLM。 †SparseVLM采用渐进式逐层剪枝；所示令牌数量为目标预算（由于其多阶段调度，实际最终层令牌更少）。

方法	减少量	词元数量	TTFT（毫秒）	加速比	浮点运算次数（T）	键值缓存（MB）	GPU 内存（GB）	生成时间（毫秒）
基线	0%	576	90.0	1.00×	8.47	313	14.50	471.6
				令牌缩减 *25%*				
ConsensusDrop（本文方法）	25%	432	82.2	1.09×	6.57	241	14.48	458.0
VisionZip	25%	432	82.5	1.09×	6.57	241	14.48	464.7
VisPruner	25%	432	57.4	1.57×	6.95	241	13.78	436.2
SparseVLM†	25%	432	60.4	1.49×	5.61	212	18.15	626.7
				50% 令牌缩减				
ConsensusDrop（我们的方法）	50%	288	66.9	1.35×	4.67	169	14.48	446.7
VisionZip	50%	288	65.2	1.38×	4.67	169	14.34	440.7
VisPruner	50%	288	50.8	1.77×	5.05	169	13.60	400.9
SparseVLM†	50%	288	61.1	1.47×	3.63	148	18.15	625.9
				令牌缩减 *75%*				
ConsensusDrop（我们的方法）	75%	144	52.6	1.71×	2.78	97	14.48	430.7
VisionZip	75%	144	50.9	1.77×	2.78	97	14.36	437.1
VisPruner	75%	144	46.8	1.92×	3.16	97	13.52	365.5
SparseVLM†	75%	144	62.6	1.44×	2.32	96	18.15	614.1
				90% 令牌缩减				
ConsensusDrop（我们的方法）	90%	57	47.5	1.89×	1.64	54	14.44	430.5
VisionZip	90%	57	45.9	1.96×	1.64	54	14.36	434.3
VisPruner	90%	57	48.7	1.85×	2.02	54	13.52	344.1
SparseVLM†	90%	57	63.2	1.42×	1.62	56	18.15	619.6
				令牌缩减 *95%*				
ConsensusDrop（本文方法）	95%	28	46.0	1.96×	1.27	39	14.48	426.8
VisionZip	95%	28	42.7	2.11×	1.27	39	14.36	434.4
VisPruner	95%	28	46.3	1.94×	1.64	39	13.52	337.0
SparseVLM†	95%	28	45.2	1.99×	1.51	41	18.15	613.8

加速比（70亿参数为1.35–1.77×），且差异在更高压缩率下更为显著。重要的是，ConsensusDrop的加速表现与VisionZip高度一致，证实我们的多模态共识机制相对于纯视觉剪枝几乎不引入额外开销。

H. 定性可视化

我们展示了定性可视化结果，以说明共识融合的词元选择机制（示例来自 VQA2 [21] test-dev2015 划分，并以 LLaVA-1.5-7B 作为基础视觉语言模型）。我们提出的共识融合方法将基于视觉的显著性分数与跨模态注意力相结合，以识别同时具有视觉显著性和语义相关性的问题词元。

H.1. 词元选择对比

图19 和20 展示了各种词元选择策略的并排对比：**仅视觉**根据 CLIP 编码器的视觉显著性选择词元，**仅跨模态**使用文本到图像注意力的注意力得分，而**共识**通过我们的凸融合将两者结合（$\alpha = 0.7$）。

可视化结果表明，仅视觉选择通常会聚焦于全局显著区域（例如高对比度边缘、纹理），却忽略与问题相关的细节。受位置偏差（旋转位置嵌入 RoPE）影响，跨模态注意力往往会过度关注靠后的位置（图像底部的图像块）。我们的共识融合平衡了这些互补信号，保留了兼具视觉信息量且在语义上与问题相关的词元。

H.2. 高分歧示例

我们分析模态分歧最大的示例——即仅视觉注意力与交叉模态注意力选择的词元集合存在显著差异的情况（图 21 至30）。这些示例最能体现共识融合的价值。在较高压缩率下，谨慎进行词元选择变得更加关键。这些示例展示了 ConsensusDrop 在激进剪枝下的有效性。

H.3. 关键观察

• **互补信号：**基于视觉的分数捕捉全局显著性，而跨模态注意力则识别

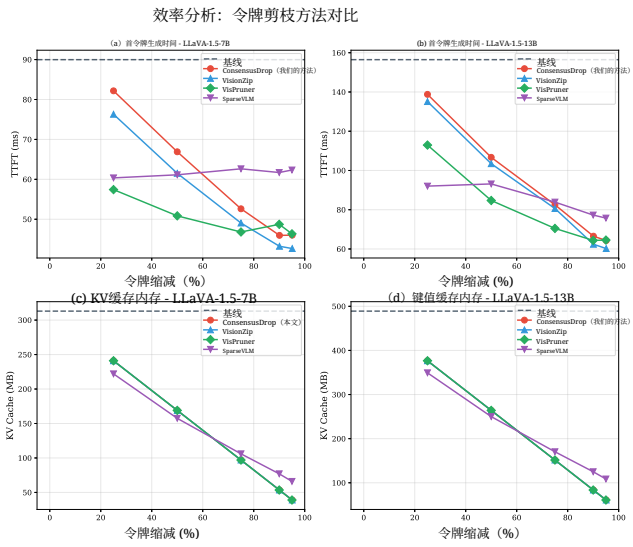


图 17。不同令牌缩减比例下的效率比较。（**a**、**b**）对于所有方法，首令牌生成时间都会随着令牌缩减而持续降低。（**c**、**d**）键值缓存内存与保留的词元数量呈线性关系。ConsensusDrop（绿色/红色/蓝色；彼此重合）在避免 SparseVLM（紫色）的内存开销和 VisPruner 的计算开销的同时，实现了具有竞争力的效率。

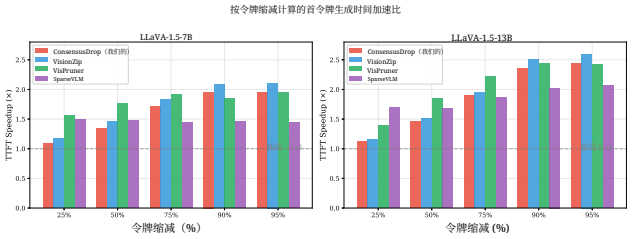


图 18。固定缩减级别下的首令牌生成时间加速比比较。ConsensusDrop 实现了与 VisionZip 相当的加速比，同时通过多模态共识提供了更高的准确率。SparseVLM 的加速比由于其渐进式剪枝开销而趋于饱和。

与问题相关的区域。单独依靠任何一种都不够。

• **跨模态中的位置偏差：**跨模态注意力会因 RoPE 而表现出位置偏差，有时会过度加权序列边界处的词元，而不是语义相关的词元；这是一个已知问题 [88]，存在于采用 RoPE 嵌入的语言模型骨干网络的视觉语言模型中。

• **共识优势：**我们的凸融合（$\alpha = 0.7$）成功结合了两种信号，保留了既 视觉上显著又 语义上对齐的词元。

•**跨 K 的稳健性：**共识机制在不同保留率下均能保持有效，从中等压缩（K=288）到激进压缩（K=128）皆是如此。

I. 局限性与讨论

ConsensusDrop 依赖于有意义的跨模态注意力信号；对于视觉定位极弱或模糊的提示词，基于共识的融合所带来的收益可能会减弱。尽管 SCAP 能够在大语言模型之前高效提取跨模态线索，但它仍只是对完整大语言模型跨模态注意力的一种轻量级近似，可能无法充分捕捉更高层的语义交互。我们的方法专注于减少视觉词元，并未处理文本令牌中的冗余，也未涉及大语言模型内部跨层的词元动态。最后，虽然编码器引导的令牌合并在实践中能够保留语义上下文，但在高度局部化的视觉推理任务中，激进压缩仍可能导致细粒度空间细节的丢失。

J. 可复现性

我们的实现基于 LLaVA [43], 、Video-LLaVA [40], 和 Qwen-VL [5]，的官方开源代码库，并保留了所有模型变体、预处理流程和评估协议，除非另有明确说明。我们将发布完整的 ConsensusDrop 代码库，包括推理脚本、配置文件和评估工具，以便复现所有报告结果。正文和附录详尽介绍了实验设置、词元保留预算、效率测量和消融配置，以确保透明度和可复现性。

K. 大语言模型的使用

作者在完成核心方法、实现和实验后，使用大语言模型作为辅助工具，协助进行代码重构、跨消融设置的实验编排，以及少量旨在提升清晰度和语法准确性的编辑工作。所有算法设计、实验决策、结果解读和最终写作均由作者全权负责。

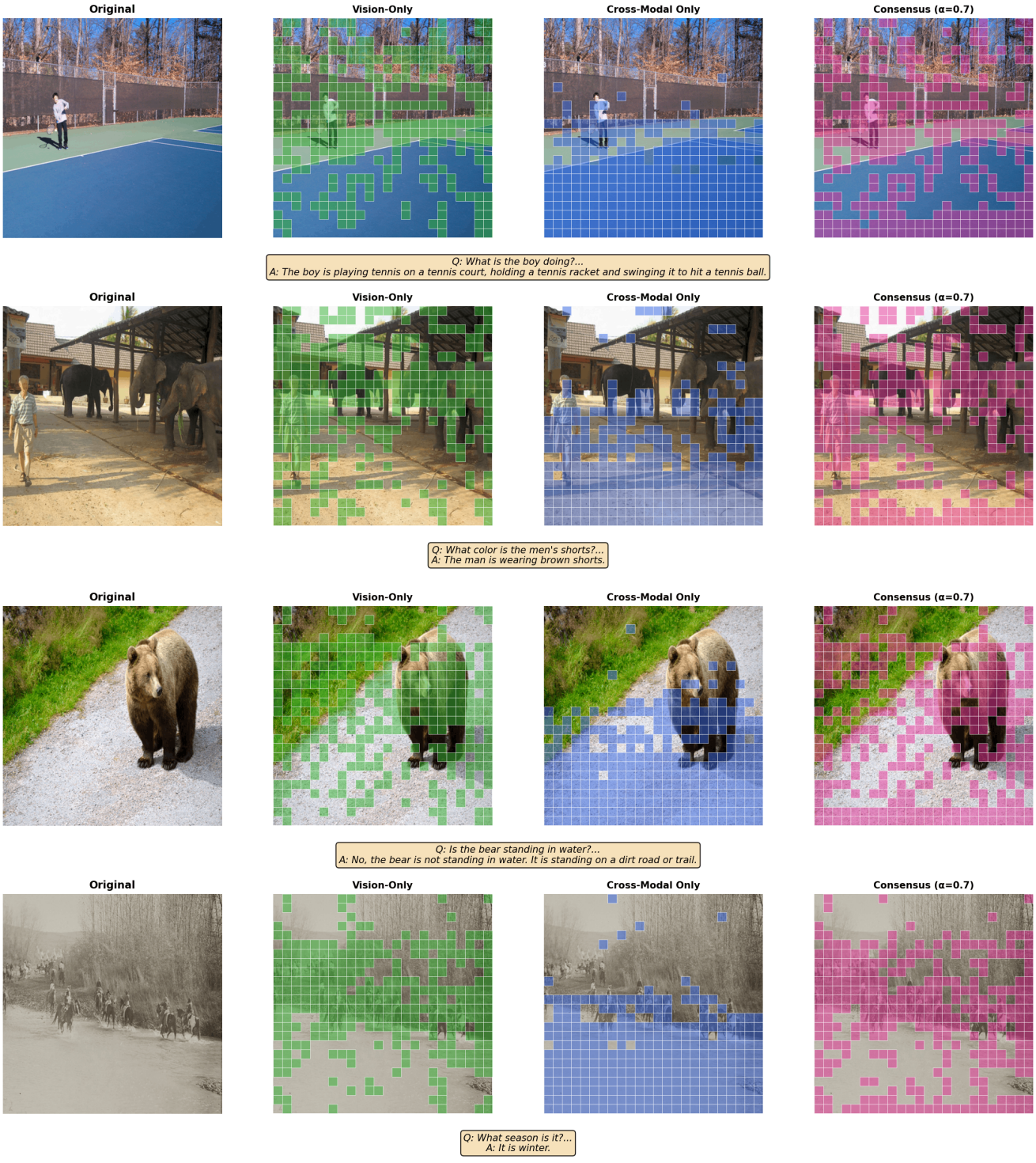


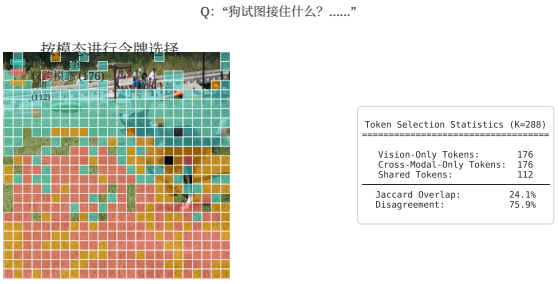
图 19. $K = 288$ 处的词元选择对比（减少量为 50%）。每个面板展示：原始图像、仅视觉选择（绿色）、仅跨模态选择（蓝色）和共识融合（洋红色）。每组对比下方展示问题和模型回答。



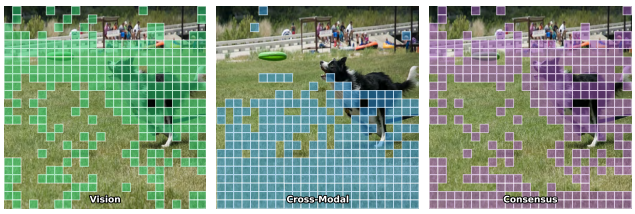
图 20. $K = 128$ 处的词元选择对比（减少量为 78%）。在更高的压缩率下，视觉信号与交叉模态信号之间的不一致会更加明显，因此共识融合变得更加关键。



（a）注意力热图



（b）词元集合重叠



（c）最小比较

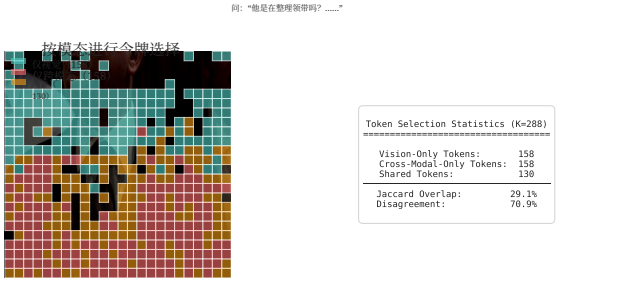


（d）不同保留率下

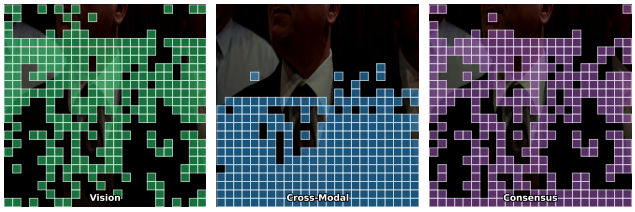
图21。**问：**“这只狗想要接住什么？”**答：**“这只狗想要接住一个飞盘……”（不一致：61.1%）



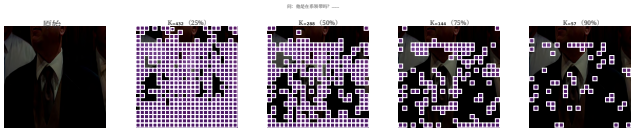
（a）注意力热图



（b）词元集合重叠



（c）最小比较



（d）不同保留率下

图22。**问：**“他正在整理领带吗？”**答：**“是的，图像中的男人正在整理领带，这是在……之前常见的活动。”（不一致：54.9%）



(a) 注意力热图

(b) 词元集合重叠



(c) 最小比较



(d) 不同保留率下

图23. **问：** “这个男人的手腕上戴着什么？”**答：** “这个男人的手腕上戴着一块手表.....”（不一致：54.9%）



(a) 注意力热图

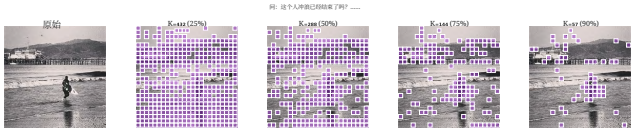
问："那家伙已经冲完浪了吗？....."



(b) 词元集合重叠



(c) 最小比较

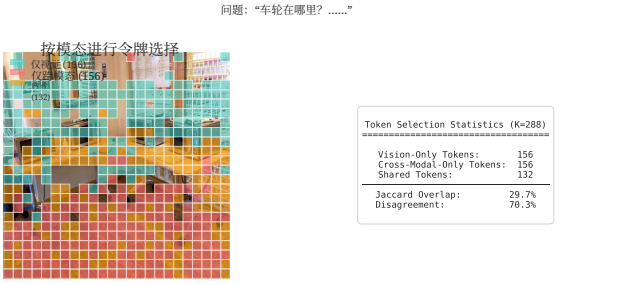


(d) 不同保留率下

图24. **问：** “这个人已经冲完浪了吗？”**答：** “是的，这个人已经冲完浪，正从水里走出来，他的.....”（不一致：54.2%）



(a) 注意力热图



(b) 词元集合重叠

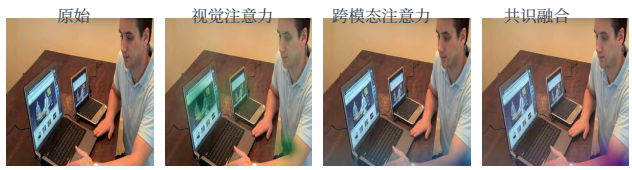


(c) 最小比较



(d) 不同保留率下

图25. **问：** "车轮在哪里？" **答：** "车轮在桌子上……。"（不一致：54.2%）



(a) 注意力热图



(b) 词元集合重叠



(c) 最小比较

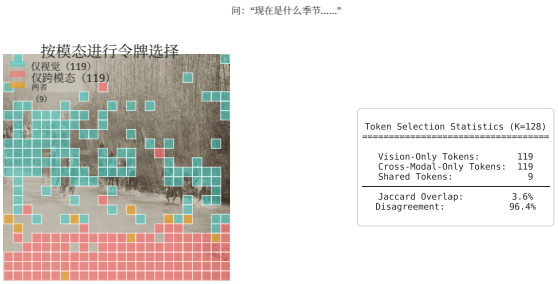


(d) 不同保留率下

图 26. **问：** "能看到几个人？" **答：** "图像中只能看到一个人……。"（不一致：93.0%）



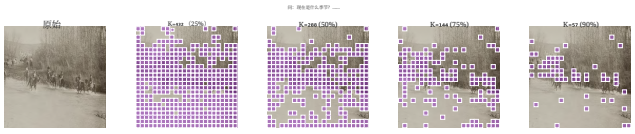
(a) 注意力热图



(b) 词元集合重叠



(c) 最小比较

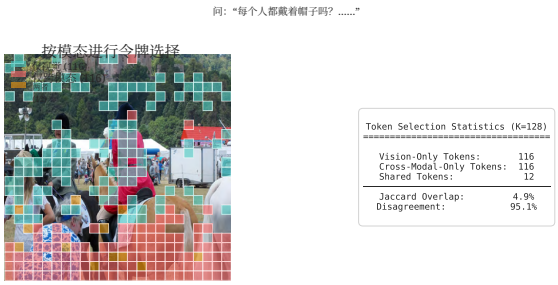


(d) 跨不同保留率

图27. **问：** "现在是什么季节？" **答：** "现在是冬季....."（不一致：93.0%）



（a）注意力热图



（b）词元集合重叠



（c）最小比较



（d）不同保留率下

图28. **问：** "大家都戴着帽子吗？" **答：** "是的，图像中的每个人都戴着帽子....."（不一致：90.6%）



(a) 注意力热图

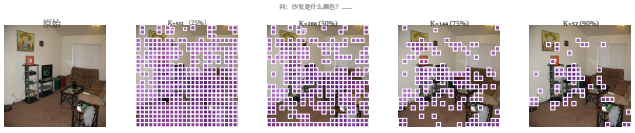
问：“沙发是什么颜色？ ……”



(b) 词元集合重叠



(c) 最小比较



(d) 不同保留率下

图29。**问：** “沙发是什么颜色的？ ” **答：** “沙发是棕色的……”（不一致：89.1%）



（a）注意力热图

问：“他们在做什么？ ……”



（b）词元集合重叠



（c）最小比较



（d）跨保留率

图30。**问：** “他们正在做什么？ ” **答：** “图像中的人正在积雪的斜坡上滑雪和进行单板滑雪……”（不一致：88.3%）