MuLTI：基于文本引导的多路采样器与多项选择建模的高效视频与语言理解

徐佳琪、刘博、陈云阔、程梦丽、石星

阿里巴巴集团，中国{zhoumo.xjq、xuanyuan.lb、chenyunkuo.cyk、mengli.cml、shubao.sx}@alibaba-inc.com

arXiv:2303.05707v2 [cs.CV] 1 Mar 2024

摘要

视频与语言理解在工业领域有着广泛的应用，例如视频问答、文本-视频检索和多标签分类。现有的视频与语言理解方法通常采用计算开销较大的多模态编码器和特征融合模块。尤其是，这些方法难以处理工业应用中常见的密集视频帧或长文本。

本文提出 MuLTI，这是一种准确且高效的视频与语言理解模型，能够实现高效有效的特征融合，并快速适应下游任务。具体而言，我们设计了一种基于自适应池化残差映射和自注意力模块的文本引导的多路采样器，用于对长序列进行采样并融合多模态特征，从而降低计算开销，并解决以往采样器导致的性能下降问题。因此，MuLTI 能够以有限的计算开销处理更长的序列。随后，为进一步提升模型性能，并弥补视频问答预训练任务的不足，我们提出了一种名为多项选择建模的新预训练任务。该任务弥合了预训练与下游任务之间的差距，并增强模型对齐视频和文本特征的能力。得益于高效的特征融合模块和新预训练任务，MuLTI 在多个数据集上取得了最先进的性能。实现代码和预训练模型将予以发布。

引言

视频与语言理解有广泛的应用，例如视频问答（*videoQA*）、文本-视频检索和多标签分类（Diba 等，2019）。现有方法在视频与语言理解方面取得了显著进展。然而，它们仍面临两项挑战：处理长序列时如何平衡计算效率与性能，以及预训练与下游任务之间的领域差距。

视频-文本模型通常由三个模块组成：文本编码器、视频编码器和特征融合模块。后两个模块通常会带来较高的计算成本。

特征融合模块面临效率与效果方面的挑战。以往研究（Fu 等，2021；Huang 等人，2022）将视频和文本编码器的输出拼接起来，交由 Transformer 编码器处理，其复杂度会随序列长度的平方增长。其他研究（Lei 等，2021b；Li 等，2021；Yang 等人，2022；Lei 等，2021a）在特征融合前通过均值池化或类别令牌凝缩视频特征，以减少计算量，但可能因此丢失关键细节。Flamingo（Alayrac 等，2022）采用采样器和随机查询来高效凝缩视频特征，但这种方法并非最优，且可能损害特征完整性。总之，在特征融合模块中平衡计算成本与模型准确率仍然具有挑战性。遵循（Miech 等，2019；Sun 等人，2019；Li 等人，2020；Zhu 和 Yang，2020；Miech 等，2020）的研究思路，我们探索了有选择地冻结编码器组件的策略，以降低视觉编码器的训练成本。

使预训练与下游任务保持一致颇具挑战。以往的预训练框架通常采用四种典型的预训练任务：用于优化视频编码器的掩码帧建模（MVM）任务（Lei 等，2021a；Ma、Lou 和 Ouyang，2021；Fu 等，2021；Huang 等，2022），用于优化文本编码器的掩码语言建模（MLM）任务（Devlin 等人，2018；Sun 等人，2019；Zhu 和 Yang，2020；Luo 等人，2020；Li 等人，2020；Lei 等，2021b），以及用于联合优化视频和文本编码器的视频文本匹配（VTM）和视频文本对比（VTC）任务（Li 等人，2020；Luo 等人，2020；Fu 等，2021；Li 等，2021）。尽管上述方法已证明能够有效学习视频和文本表征，但预训练与下游任务之间仍存在显著的领域差距，在视频问答中尤为明显。在上述预训练任务中，只有 VTC 任务与文本-视频检索一致。总之，缩小预训练与下游任务之间的领域差距仍然是一项挑战。

为应对这些挑战，我们提出了 MuLTI，其特点是采用文本引导的多路采样器进行序列压缩和多模态融合。现有方法通常使用可学习查询向量，通过自注意力模块对视频特征进行采样（Alayrac 等，2022）。随机初始化的查询向量可能会丢弃重要的原始特征信息，导致性能下降。我们在文本引导的多路采样器中设计了一种轻量级自适应池化方法，通过计算每个序列块的重要性来获取压缩特征。随后，我们将压缩特征添加到采样特征中，并使用

版权所有 © 2024，美国人工智能促进协会（www.aaai.org）。保留所有权利。

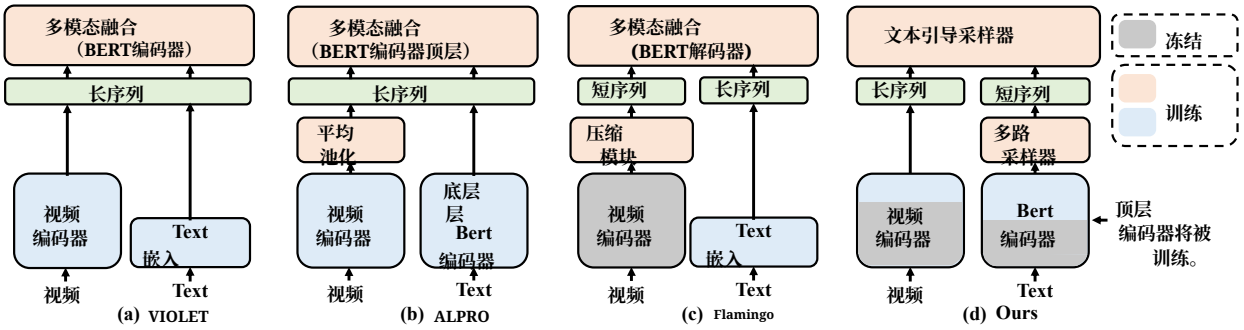

图 1：不同模型的比较。先前的工作（如 (a) 和 (b)）难以处理长序列。先前的工作（如 (c)）使用随机初始化的查询向量进行采样器采样，并浓缩视频特征，这并非最优方案。

较短的文本特征来采样并融合较长的视频特征。我们共享自注意力模块，并为采样器中的不同模态保留不同的前馈网络。

图 1 显示，先前的模型 (a)（Fu 等，2021；Huang 等，2022）和 (b)（Li 等，2021）通过冗长的拼接特征融合消耗了大量视频内存。(b) 和 (c)（Alayrac 等，2022）都会压缩视频特征；由于视频特征的长度通常远大于文本，这是一种常见做法。然而，视频特征蕴含丰富信息，过度压缩可能会损害性能。相比之下，我们设计了类似 (d) 的 MuLTI，并引入文本引导的多路采样器，以高效浓缩文本特征进行融合。由于文本更加简洁，我们利用精简的文本引导视频特征采样，从而提升性能。

在大规模视频文本数据集上进行预训练，可以显著提升视频文本模型的性能。然而，现有预训练任务与下游任务之间仍存在领域差距，尤其是在视频问答方面。将视频问答引入预训练任务的难点在于构建合适的问答对。为缩小视频问答预训练任务与下游任务之间的领域差距，我们提出了一种名为多项选择建模（MCM）的新预训练任务。MCM 通过在大规模视频文本数据集上构建多项选择问答任务，弥合预训练与下游任务之间的差距。该任务要求模型从随机构建的候选集合中找出与视频最匹配的文本描述，从而增强视频和文本编码器的表征能力，并提升视频特征与文本特征之间的对齐程度。

贡献总结如下：

(1) 我们提出了 MuLTI，这是一种准确率高且内存高效的视频与语言框架，通过特征采样和注意力模块实现高效且有效的特征融合。

(2) 我们提出一种文本引导的多路采样器，用于对长序列特征进行采样，促进视频特征与文本特征之间的交互，从而降低内存开销并提升性能。

(3) 我们设计了一种名为多项选择建模（MCM）的新预训练任务，以弥合预训练与下游任务之间的任务差距。实验结果表明，在七项英语任务和一项中文多标签分类任务上的实验结果证明了 MuLTI 的有效性。

尽管我们针对具有长序列的工业场景设计了 MuLTI，但 MuLTI 也能很好地处理短序列，并达到最先进的性能。

相关工作

视频与语言结构。 Clover（Huang 等，2022）和 VIOLET（Fu 等，2021）直接拼接视频特征与文本特征，并使用编码器处理二者复杂的交互，其复杂度与拼接后序列长度的平方成正比。ALPRO（Li 等，2021）同样使用编码器融合特征，但在拼接前对视频特征进行均值池化，可能导致关键信息丢失。AllInOne（Wang 等人，2022a）通过逐帧合并文本与图像特征来降低内存需求，但在 OCR 转录文本量较大时仍面临较高的计算开销。Flamingo（Alayrac 等，2022）尝试利用采样器和随机查询压缩视频特征以降低成本，但这种方法并不理想。为解决上述问题，我们基于自适应池化残差映射和自注意力模块设计了一种文本引导的多路采样器，以采样长序列特征并融合多模态特征。

视频与语言预训练。以往的预训练框架采用了四种典型的预训练任务：掩码帧建模（MVM）任务（Lei 等，2021a；Ma、Lou 和 Ouyang，2021；Fu 等，2021；Huang 等，2022）、掩码语言建模（MLM）任务（Devlin 等人，2018；Sun 等人，2019；Zhu 和 Yang，2020；Luo 等人，2020；Li 等人，2020；Lei 等，2021b；Fu 等，2021）、视频文本匹配（VTM）和视频文本比较（VTC）任务（Li 等人，2020；Luo 等人，2020；Fu 等，2021；Li 等，2021）。MVM 用于优化视频编码器，MLM 用于优化文本编码器，VTM 和 VTC 则用于联合优化视频和文本编码器。Ge 等人（2022）提出了多项选择题（MCQ），以学习细粒度视频和文本特征。然而，MCQ 通过对比损失进行训练，与视频问答的相关性不高。总而言之，预训练与下游任务之间仍存在下游任务差距，在视频问答中尤其如此。为解决这一问题，我们在 MuLTI 中加入 MCM，以衔接预训练任务与下游任务。

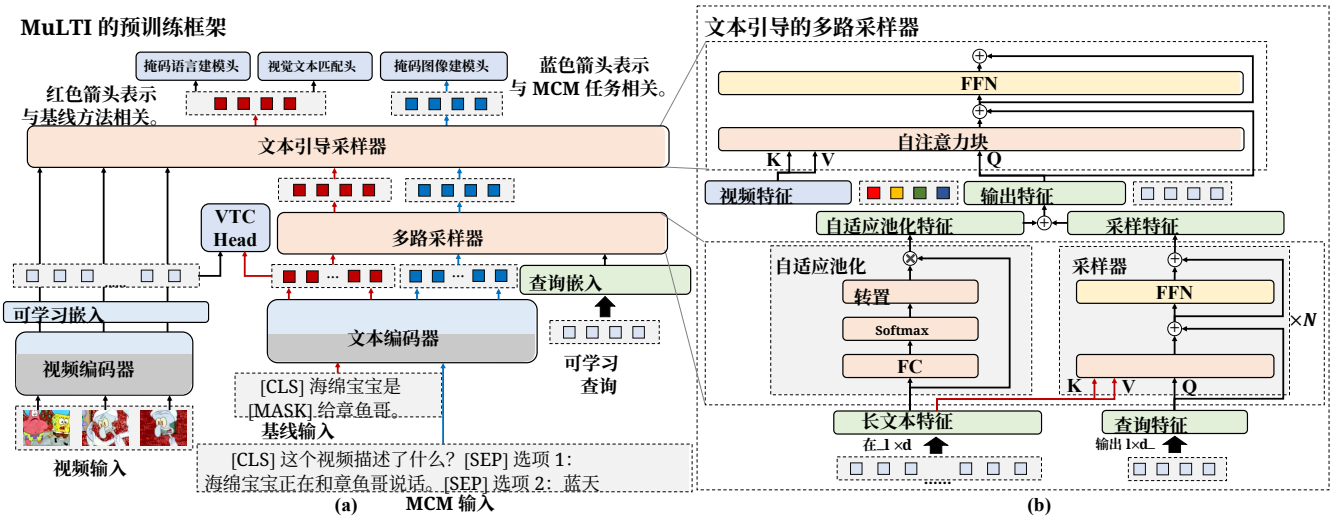

图 2：（a）展示了 MuLTI 的框架。MuLTI 包含一个视频编码器、一个文本编码器和一个文本引导的多路采样器。文本引导的多路采样器用于浓缩提取出的特征并进行特征融合。（b）展示了文本引导的多路采样器的框架。自适应池化特征提供了原始信息。我们在采样器中共享自注意力模块，并为不同模态保留不同的前馈网络，以适应各模态。

方法论

MuLTI 的架构

图 2（a）展示了 MuLTI 架构的概览。各组件的详细介绍如下。

视频和文本编码器： 除非另有说明，视频编码器采用 12 层的 VIT-B/16_{224} （Radford 等，2021）。我们从输入视频中稀疏采样 N_v 帧。VIT-B/16_{224} 模型将每帧划分为 K 个互不重叠的图像块。每个视频的特征为 $\vec{v} \in \mathbb{R}^{N_v \times K \times d}$，其中 d 是特征维度。视频编码器的输出是一个视频特征序列：$\{\vec{v}_1, ..., \vec{v}_{N_v}\}$，其中包含 $\vec{v}_i \in \mathbb{R}^{K \times d}$。实验表明，类别令牌并非必需，因此将其移除以节省计算量。除非另有说明，文本编码器采用 12 层 BERT（Devlin 等人，2018）。假设输入文本长度为 N_t，文本编码器的输出是一个文本特征序列 $\vec{t} \in \mathbb{R}^{N_t \times d}$：$\{\vec{t}_{\text{cls}}, \vec{t}_1, ..., \vec{t}_{N_t}\}$，其中包含 $\vec{t}_i \in \mathbb{R}^d$。$\vec{t}_{\text{cls}}$ 是文本 [CLS] 令牌的输出。遵循（Miech 等，2019；Sun 等人，2019；Li 等人，2020），我们探索了部分冻结编码器层的训练策略。

文本引导的多路采样器： 多模态融合的核心是文本引导的多路采样器，它由 Transformer 解码器改编而来，如图 2 (b) 所示。文本引导的多路采样器旨在浓缩文本特征，并高效融合不同模态的特征。遵循 Alayrac 等，2022 的方法，我们初始化一个随机的可学习查询，通过采样来浓缩特征。表达式 $Sampler(z,\ q)$ 表示通过采样器，使用查询向量 q 对特征 z 进行采样。

(i) 为什么需要自适应池化？ 可学习查询能够很好地压缩特征，但以随机向量作为起点可能会降低其效果。随机初始化可能会丢失原始特征中的关键信息，从而削弱模型捕捉并保留数据本质的能力。因此，我们设计了一种基于注意力的轻量级自适应池化方法，用于浓缩长序列特征。自适应池化结构如图 2(b) 左侧所示。公式如下，$AdaPool(z)$ 是自适应池化的输出，$W^{reduce} \in \mathbb{R}^{d \times N_s}$, d 是 Transformer 的隐藏维度，N_s 是浓缩后特征的长度，$*.T$ 是矩阵的转置。

$$AdaPool(z) = Softmax((W^{reduce} z).T) \times z \quad (1)$$

$Softmax((W^{reduce} z).T)$ 得到一个形状为 $[N_s, N_i]$ 的重要性权重矩阵，其中每个元素表示序列中相应块的相对重要性，而 N_i 表示输入特征的长度。自适应池化会选择性地突出关键输入片段，在浓缩特征的同时保留其关键属性。这种整合通过提炼后的信息丰富特征集，并确保充分利用数据，从而增强模型的能力和鲁棒性。

(ii)为什么要压缩文本特征？ 视频特征通常存在冗余，而文本特征则更稠密、含义也更丰富（He 等，2022）。语言引导是从视频中提炼有价值信息的关键。Li 等（2021）和 Alayrac 等，2022 都对视频特征进行了压缩。过度压缩会损害模型性能；使用压缩后的文本采样视频特征可以提升结果。在融合之前，可学习的时间嵌入能够增强图像特征，以进行时序建模。短文本特征用于采样视频特征，从而融合多模态特征。在我们的文本引导的多路采样器中，我们为每种模态使用共享的自注意力模块和不同的前馈网络，以高效处理多模态特征。融合后的特征如下所示，其中 q 是.....的查询嵌入

方法	浮点运算次数	参数量	FPS
MuLTI-S	99G	203M	20.74
MuLTI-B	346G	247M	10.13
VIOLET（Fu 等，2021）	249G	198M	9.05
ALPRO（Li 等，2021）	432G	235M	9.97
MuLTI-L	1509G	746M	3.12
FrozenBiLM（Yang 等人，2022）	1733G	1224M	2.54

表1：16帧模型的比较。文本
长度为512。每秒帧数基于1块 NVIDIA V100 16GB GPU。

文本特征， z_{out} 融合特征：

$$z_{out} = Sampler(\tilde{\vec{v}}, Sampler(\tilde{\vec{t}}, q) + AdaPool(\tilde{\vec{t}})) \quad (2)$$

与我们工作类似的是 Token Learner（Ryoo 等，2021年），它在模型中使用空间注意力，从图像中提取 8 或 16 个代表性向量。不同之处在于，我们使用自适应池化和自注意力来压缩特征，以进行多模态融合。采样器通过自注意力提取复杂信息，而自适应池化则通过残差映射快速生成简单特征。

(iii) **为什么文本引导的多路采样器效率高?** 简单分析表明，我们的特征融合模块在效率上优于基于展平的方法和 Transformer 编码器：假设采用 VIT-B/16_{224} 作为视频编码器，每帧将展平成长度为196的序列。设文本的查询数量为 N_t^q ，视频特征的长度为 $N_v \times 196$，文本特征的长度为 N_t。因此，展平方法的复杂度为 $O((N_t + N_v \times 196)^2)$。应用文本引导的多路采样器后，复杂度为 $O(N_t^q \times N_v \times 196 + N_t^q \times N_t)$。由于 N_t^q通常远小于 N_v和N_t，我们的方法比其他方法效率高得多。

适用于不同场景的 MuLTI。 在本节中，我们针对资源差异化场景构建了可扩展模型。我们将视频编码器从 VIT-B/16 替换为 VIT-L/14，并将文本编码器从 bert-base 替换为 bert-large，从而得到 MuLTI-L。此外，我们将视频编码器从 VIT-B/16 替换为 VIT-B/32，并将文本编码器从 12 层缩减为 6 层，从而得到 MuLTI-S。不同模型的浮点运算次数（FLOPs）、参数量（ Params）和每秒帧数（FPS）如表1所示。

MuLTI 的预训练

多项选择建模： 尽管 MLM 和 VTM 在学习视频文本表示方面取得了成功，但预训练与视频问答等下游任务之间仍存在显著差距。将视频问答引入预训练任务的难点在于构建合适的问答对。受多项选择视频问答的启发，我们发现，与视频配对的文本描述就是正确的自然答案。因此，我们引入了多项选择建模这一新的预训练任务，以弥合预训练与下游任务之间的差距。具体构造方式如下，这是一个四选一问题。

"[CLS]<问题> ？ [SEP] 选项 1： <答案1>。 [SEP] 选项 2： <答案 2>。 [SEP] 选项 3： <答案 3>。 [SEP] 选项 4： <答案 4>。 "

我们随机将正确的描述放入 <答案1>、 <答案 2>、 <答案 3>、 <答案 4> 中，并通过文本语料库获取正确描述以外的答案。 <问题> 也有多种选项，例如“这张图片描述了什么？”、“这段视频描述了什么？”等等。

如图 2 (a) 所示，典型的 MLM、VTM 和 VTC 任务对应图中的红色箭头和红色方块。MCM 对应图中的蓝色箭头和蓝色方块，且不会与其他预训练任务冲突。
MCM 与其他预训练任务无缝集成，无需额外的人工标注或数据预处理。它利用视频编码器提取视觉特征，并利用文本编码器生成文本表示，随后通过文本引导的多路采样器进行特征融合。掩码图像建模头会评估给定选项与视频的相关性，并使用交叉熵损失优化对齐。MCM 任务要求从选项中选出最佳描述，这与视频问答中的核心认知过程相似，能够增强模型的跨模态推理和对齐能力。MCM 可直接提升模型将文本与相应视频进行匹配的能力，从而提高其在文本视频检索任务中的表现。

预训练目标： 我们还采用 MLM、VTM 和 VTC，因其效果显著。MLM 以 15% 的概率随机遮蔽输入词元，并将其替换为 [MASK], ，再根据视频和文本预测这些词元。VTC 将匹配的视频文本对视为正样本对，并将批次中的其他视频文本对视为负样本对。VTM 与 VTC 略有不同，它会先融合多模态特征，再进行分类。MuLTI 的整体预训练目标为：

$$L = L_{mlm} + L_{vtc} + L_{vtm} + L_{mcm} \quad (3)$$

实验

实现细节

预训练数据集： 我们使用两个大型数据集对模型进行了预训练。其中一个是 WebVid-2M（Bain 等，2021），包含 250 万个视频-文本对。由于使用图像-文本对对视频-文本模型进行预训练也能提升模型性能（Lei 等，2021b），因此也使用 CC-3M（ Sharma 等，2018）作为预训练数据集，其中包含 300 万个图像-文本对。

我们使用 PyTorch（Paszke 等，2019）实现 MuLTI。具体而言，视频编码器使用 CLIP（Radford 等，2021）的预训练权重进行初始化。文本编码器使用 12 层 $\text{BERT}_{\text{base}}$ 模型（Devlin 等，2018）进行初始化。随后，使用 4 层文本引导多路采样器来浓缩文本特征并融合多模态特征。查询嵌入的长度设为 16。MuLTI 的预训练在 8 块 NVIDIA A100 GPU 上进行，共训练 10 个 epoch，批次大小为 256，总计 20 万次迭代。优化采用 AdamW，学习率为 $1e^{-4}$ ，权重衰减为 0.05，并使用预热调度器。我们为每个视频均匀采样 16 帧，并将其缩放至 224×224。

方法	#PT	MSRQ 准确率↑	MSVQ 准确率↑	TGIF。Act. 准确率↑	TGIF。Tran。准确率↑	TGIF Fra. 准确率↑	MSRVTT Ret R1 / R5 / R10 ↑	MSRVTT Ret G-M ↑	DiDeMo Ret R1 / R5 / R10 ↑	DiDeMo Ret G-M ↑
CLIP4CLIP	400M	-	-	-	-	-	43.1 / 70.4 / 80.8	62.6	43.4 / 70.2 / 80.6	62.6
QB-Norm	400M	-	-	-	-	-	47.2 / 73.0 / 83.0*	65.9*	43.3 / 71.4 / 80.8*	63.0*
CAMoE	400M	-	-	-	-	-	47.3 / 74.2 / 84.5*	66.7*	43.8 / 71.4 / 79.9*	63.0*
TS2-Net	400M	-	-	-	-	-	54.0 / 79.3 / 87.4*	72.1*	47.4 / 74.1 / 82.4*	66.1*
ALPRO	5.5M	42.1	45.9	-	-	-	33.9 / 60.7 / 73.2	53.2	35.9 / 67.5 / 78.8	57.6
VIOLET	185.5M	43.9	47.9	92.5	95.7	68.9	34.5 / 63.0 / 73.4	54.2	32.6 / 62.8 / 74.7	53.5
AllInOne	102.5M	44.3	47.9	92.7	94.3	64.2	37.9 / 68.1 / 77.1	58.4	32.7 / 61.4 / 73.5	52.8
Clover	5.5M	44.1	52.4	95.0	98.2	71.6	40.5 / 69.8 / 79.4	60.7	50.1 / 76.7 / 85.6	69.0
Flamingo	2139M	47.4	52.3	-	-	-	-	-	-	-
FrozenBiLM	10M	47.0	54.4	-	-	68.6	-	-	-	-
MuLTI-S	5.5M	45.6	50.0	97.3	98.9	71.2	41.3 / 70.6 / 79.7 45.8 / 73.5 / 82.0*	61.5 65.1*	42.6 / 71.4 / 80.0 47.9 / 73.0 / 82.6*	62.5 66.1*
MuLTI-B	5.5M	46.6	53.0	97.3	**99.1**	73.5	45.1 / 72.4 / 81.8 49.4 / 75.9 / 84.0*	64.4 68.0*	45.2 / 74.6 / 82.2 48.3 / 75.4 / 83.5*	65.2 67.2*
MuLTI-L	5.5M	**47.8**	**54.7**	**97.9**	99.0	**75.6**	48.7 / 75.9 / 83.9 **54.7/ 77.7/ 86.0***	67.7 **71.5***	50.5 / 78.5 / 86.2 **56.5 /80.2 /87.0***	69.9 **73.3***

表2：与现有方法的比较。#PT 表示预训练数据集数量。Acc. (%) 表示视频问答的性能。R@k 表示进行 k 次检索时的召回率 (%)。G-M 表示召回率@1、召回率@5 和召回率@10 的几何平均数。常用数据集包括 WebVid2M（Bain 等，2021）、WebVid10M（Bain 等，2021）、WIT（Radford 等，2021）、HowTo100M（Miech 等，2019）、YT-Temporal-180M（Zellers 等，2021）和 Conceptual Captions（Sharma 等，2018）。* 表示使用 DSL（Cheng 等人，2021）或 QB-Norm（Bogolin 等人，2021）进行后处理。

方法	#PT	OCR	多标签
VIOLET ‡	-	✘	55.22
ALPRO ‡	-	✘	58.53
MuLTI-S	-	✘	63.97
MuLTI-S	-	✔	**66.13**
MuLTI-B	-	✘	64.60
MuLTI-B	-	✔	**67.86**

表 3：多标签分类的平均精度均值（%）。‡ 表示这些方法是在我们的框架中复现的。

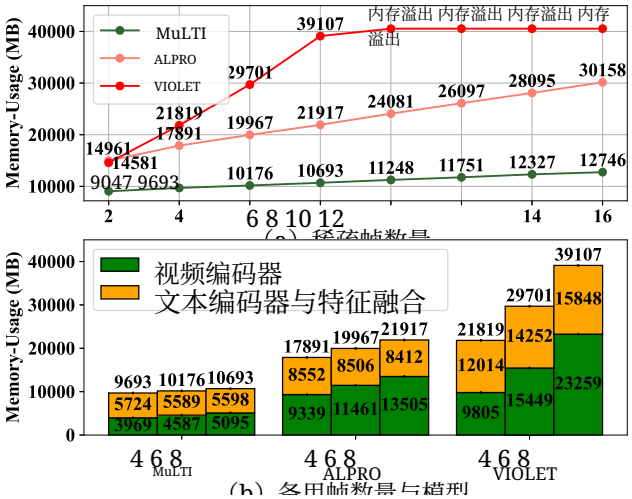

图 3：不同帧数下与现有方法在内存使用量方面的比较。文本长度为 512。

下游任务和数据集

视频问答。我们在五项广泛使用的视频问答任务上评估 MuLTI。**(1) MSRQ（MSRVTT- QA）**（Xu 等，2017、2016）是一项开放式视频问答任务，包含 1 万个视频和 24.3 万对问答。**(2) MSVQ（MSVD-QA）**（Xu 等，2017；Chen 和 Dolan，2011）是一项开放式视频问答任务，包含 1,970 个视频和 5 万对问答。**(3) TGIF-QA（Jang 等，2017）**包含三个数据集：TGIF-Action 和 TGIF-
用于多项选择视频问答任务的 Transition，以及用于开放式视频问答任务的 TGIF-Frame。

文本-视频检索。**（1）MSRR（MSRVTT-Ret）**包含 1 万个视频和 20 万条标注。遵循 Fu 等，2021 的做法，我们使用 9 千个视频进行训练，1 千个视频进行测试。**（2）DiDeMo（DiDeMo-Ret）**由 1 万个视频和 4 万条标注组成。遵循 Lei 等人 2021b 的做法，我们将视频的所有标注拼接成一个标题。

多标签分类。视频标签至关重要，因为在线广告中的排序模型[1]。我们从自己的应用中创建了一个短视频数据集，其中包含 48.6 万个带有字幕的视频和 21696 个标签。多位专业编辑对标签进行了交叉核查。我们采用了 ICDAR[2] 中的高性能文本检测器来获取 OCR 转录文本。OCR 转录文本截断至 512 个字符。多标签分类的示例见附录。

[1]https://algo.qq.com/index.html [2] https://rrc.cvc.uab.es/?ch=4&com=evaluation&task=4

方法	基础 TGMS	预训练基线		MCM	MSRQ MSVQ 准确率↑	准确率↑	MSRR R1↑	R5↑	R10↑	几何平均数 ↑
MuLTI-B	✔	✘	✘	✘	44.84	48.35	38.90	69.50	78.50	59.64
	✔	✔	✘	✘	45.54	49.86	38.80	70.30	80.10	60.22
	✔	✔	✔	✘	46.28	51.93	44.30	72.40	**81.90**	64.04
	✔	✔	✔	✔	**46.61**	**53.03**	**45.10**	**72.40**	81.80	**64.40**

表 4：所提方法的评估结果。TGMS：文本引导的多路采样器。PB（预训练基线）：采用掩码语言建模、视频文本匹配和视频文本对比的预训练模型。MCM：多项选择建模。准确率（%）用于衡量视频问答的性能。R@k 表示经过 k 次检索后的召回率（%）。G-M 表示召回率@1、召回率@5 和召回率@10 的几何平均数。

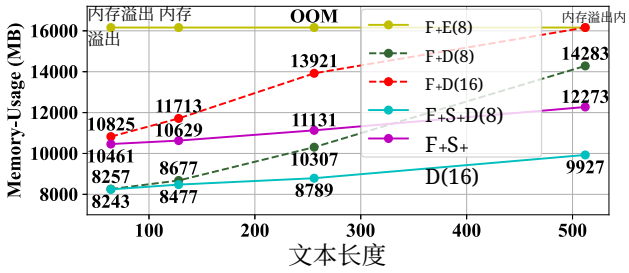

图 4：不同文本长度和帧数对内存使用量的影响比较。F 表示展平，D 表示解码器，E 表示编码器，S 表示采样器。括号中的数字表示帧数。

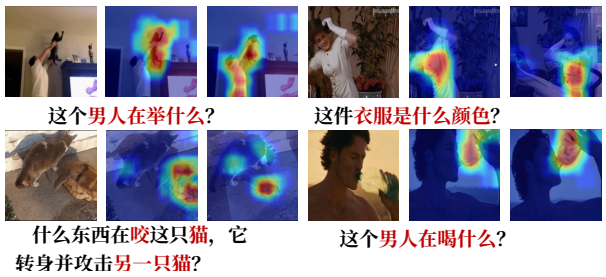

图 5：文本引导的多路采样器生成的交叉注意力图可视化。

方法	MSRQ	MSVQ	内存使用量
类别令牌	44.54	47.90	7081
均值池化	44.40	47.07	6941
最大池化	44.41	46.93	6963
展平 +编码器	44.84	48.35	15791
TGMS	**45.54**	**49.86**	10551

表 5：特征保留方法的消融研究。展平方法的稀疏帧数设为 6。TGMS 表示文本引导多路采样。

方法	CV	CT	SS	AP	MSRQ	MSVQ
展平解码器	✘	✘	✘	✘	45.13	49.19
	✔	✘	✘	✘	44.76	48.10
	✔	✘	✔	✔	45.14	48.92
	✔	✔	✘	✘	44.57	48.50
	✘	✔	✘	✘	45.08	49.38
	✘	✔	✔	✘	45.16	49.80
	✘	✔	✘	✔	45.48	49.54
	✘	✔	✔	✔	**45.54**	**49.86**

表 6：特征压缩方法的消融研究。CV 表示压缩视频，CT 表示浓缩文本，SS 表示共享采样器，AP 表示自适应池化。

所提方法的性能

表2将 MuLTI 与 CLIP4CLIP（Luo 等，2021）、QB-Norm（Bogolin 等人，2021）、CAMoE（Cheng 等人，2021）、TS2-Net（Liu 等人，2022）、ALPRO（Li 等，2021）、VIOLET（Fu 等，2021）、AllInOne（Wang 等人，2022a）、Clover（Huang 等，2022）、Flamingo（Alayrac 等，2022）和 FrozenBiLM（Yang 等人，2022）进行了比较。

在视频问答任务中，MuLTI 在 MSRQ、MSVQ、TGIF-Action、TGIF-Transition 和 TGIF-Frames 上的表现均优于所有基线模型。由于 MuLTI 不使用语音数据作为输入，因此将其与不使用语音数据的 FrozenBiLM（Yang 等人，2022）进行比较。总体而言，MuLTI 在各种问答任务上都取得了最先进的性能。

在文本视频检索任务中，我们使用 MSRVTT 和 DiDeMo 数据集对 MuLTI 进行微调。结果表明，MuLTI 在这两个基准测试中都极具竞争力，尤其是在 DiDeMo 数据集上。这些发现凸显了 MuLTI 在文本-视频检索中的有效性。

对于多标签分类，我们将 MuLTI 与 VIOLET 和 ALPRO 进行比较，但由于 FrozenBiLM 的规模过大，不适合工业部署，因此不纳入比较。VIOLET 和 ALPRO 不使用 OCR 转录文本，因为这会导致在 V100 GPU 上内存溢出。为进行公平比较，我们也在表 3 中报告了 MuLTI 的无OCR性能；MuLTI 显著优于 VIOLET 和 ALPRO。如图 3 所示，随着训练期间帧数增加，MuLTI 的视频显存开销仍不到 ALPRO 和 VIOLET 的一半，这是因为其高效的融合模块将显存开销的增长降至最低。

最后，我们在表 4 中评估了主要的技术贡献。与基线模型相比，我们的主要技术贡献提升了所有数据集上的性能。文本引导的多路采样器增强了 MuLTI 的多模态融合能力，精准定位冗余视频特征中的关键细节。MCM 提升了模型的对齐能力，并缩小了预训练与下游任务之间的差距。

PB	MVM	MCM	MSRQ	MSVQ	MSRR
✔	✘	✘	46.28	51.93	64.04
✔	✔	✘	45.87	50.16	63.41
✔	✔	✔	46.11	51.65	63.71
✔	✘	✔	**46.61**	**53.03**	**64.40**

表7：针对多项选择建模的消融研究。PB 表示预训练基线。

冻结参数 / 总参数		MSRQ	MSVQ	内存使用量
VE	TE			
12/12	12/12	44.06	46.83	6109
12/12	0/12	44.07	47.12	7439
6/12	0/12	45.10	47.57	18219
9/12	0/12	45.59	47.52	11541
9/12	3/12	45.50	49.63	11131
9/12	6/12	**45.54**	**49.86**	10551
9/12	9/12	45.04	49.14	10283

表 8：关于冻结层的消融研究。VE 指视频编码器，TE 指文本编码器。冻结层数/总层数指冻结层数与总层数。

适配器	ATT	MSRQ	MSVQ	MSRR	DiDeMo
✘	✘	45.54	49.86	60.22	51.68
✔	✘	45.61	50.48	60.54	52.08
✔	✔	**45.71**	**50.63**	**61.16**	**52.42**

表 9：关于注意力适配器的消融研究。ATT 表示注意力。

文本引导的多路采样器的重要性

为什么要凝缩文本特征? 我们在表 5 中比较了不同聚合方法（即类别令牌、均值池化、最大池化和展平）的性能。结果表明，展平优于其他聚合方法，但需要大量视频内存。上一节表明，对于长序列，解码器比编码器占用更少的内存，因此我们将其用于特征融合。解码器能够很好地处理 MSRQ 等数据集。然而，处理像我们的多标签数据集这样较长的文本和视频时，成本仍然很高。具体的内存开销如图 4 所示。遵循 Alayrac 等，2022 的做法，我们使用基于解码器的采样器进行特征凝缩。表 6 比较了不同的凝缩方法，显示文本压缩更具优势。如图 5 所示，与问题最相关的视觉部分获得了更高的权重。

共享采样器的重要性。 采样器和特征融合模块采用相同的解码器结构，因此可以共享权重而不影响性能，从而简化模型优化（Wang 等人，2022b）。我们共享采样器和解码器的自注意力，但为每种模态保留独立的前馈神经网络，从而减少参数量，同时保持性能。与展平方法相比，共享采样器在 MSRQ 和 MSVQ 上提升了准确率分别提高了 0.32% 和 1.45%。

自适应池化的重要性。 如表 6 所示，在浓缩文本和视频特征时，采样器会导致性能下降。采样器的随机查询向量存在丢失原始关键特征的风险；因此，我们设计了一种轻量级聚合模块——自适应池化，以保留原始特征。如表 6 所示，自适应池化提升了 MSRQ 和 MSVQ 上的准确率。此外，我们还探索了多种组合方法（即相加、拼接和相乘），并发现性能差异很小。在 MSRQ 上，采用拼接的准确率为 45.51%，采用相乘的准确率为 45.45%。

为验证这些技术的鲁棒性，我们将其用于浓缩视频特征，这也提升了模型性能。

T多选题建模的重要性

建模

MCM 旨在通过将视频问答融入预训练，弥合预训练与下游任务之间的差距，使模型更加关注视频和句子的主体，从而更好地进行多模态特征提取。

我们采用经典的掩码语言建模、视频文本匹配和视频文本对比任务对模型进行预训练，作为基线。由于 MVM 会造成视频内容损坏，MVM 任务与其他任务存在冲突（Lei 等，2021a）。在最初尝试将 MVM 纳入预训练时，我们观察到模型性能有所下降，如表7所示。因此，我们决定不使用 MVM 进行预训练。为确认 MCM 的鲁棒性，我们还在采用 MVM 的基础上加入了 MCM 进行预训练。结果表明，MCM 仍能显著提升模型性能。与使用基线预训练的模型相比，MCM 通过缩小预训练与下游任务之间的任务差距，显著提升了模型在视频问答任务上的表现。MCM 对多模态特征对齐的促进作用也提升了模型在检索任务上的表现。如表7所示，采用 MCM 预训练的模型在视频问答和检索任务上的表现均优于基线，证明了其有效性。

训练策略的消融实验

冻结层分析。 本节系统地评估冻结层数量的影响。表 8 展示了视频问答任务上的结果。结果表明，解冻视频和文本编码器的顶层可以提升两个数据集上的性能。

注意力适配器分析。 对冻结层的分析表明，解冻过多层会因参数调整过度而导致过拟合，从而降低准确率。遵循 Yang 等人（2022）的做法，我们在编码器的浅层中添加适配器。表 9 显示，适配器虽然表现良好，但其能力受限于基础前馈网络模块。通过集成轻量级注意力模块（Hu 等人，2017），模型能够更好地关注信息丰富的词元。

结论

我们提出 MuLTI，这是一种高性能视频语言框架，采用新颖的文本引导的多路采样器，以提高采样效率，并通过一项预训练任务更好地契合下游任务。MuLTI 在七个视频语言基准测试上取得了最先进的性能。

参考文献

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; Ring, R.; Rutherford, E.; Cabi, S.; Han, T.; Gong, Z.; Samangooei, S.; Monteiro, M.; Menick, J.; Borgeaud, S.; Brock, A.; Nematzadeh, A.; Sharifzadeh, S.; Binkowski, M.; Barreira, R.; Vinyals, O.; Zisserman, A.; and Simonyan, K. 2022. Flamingo：一种用于小样本学习的视觉语言模型。*arXiv*, abs/2204.14198. Bain, M.; Nagrani, A.; Varol, G.; and Zisserman, A. 2021. Frozen in Time：用于端到端检索的联合视频与图像编码器。*2021*年 *IEEE/CVF*国际计算机视觉会议（*ICCV*）, 1708–1718. Bogolin, S.-V.; Croitoru, I.; Jin, H.; Liu, Y.; and Albanie, S. 2021. 基于查询库归一化的跨模态检索。*2022 IEEE/CVF*计算机视觉与模式识别会议（*CVPR*）, 5184–5195. Chen, D.; and Dolan, W. B. 2011. 收集用于释义评估的高度平行数据。载于计算语言学协会第*49*届年会论文集：人类语言技术, 190–200. Cheng, X.; Lin, H.; Wu, X.; Yang, F.; and Shen, D. 2021. 通过多流语料对齐和双重Softmax损失改进视频文本检索。*arXiv*, abs/2109.04290. Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT：用于语言理解的深度双向Transformer预训练。*CoRR*, abs/1810.04805. Diba, A.; Fayyaz, M.; Sharma, V.; Paluri, M.; Gall, J.; Stiefelhagen, R.; and Gool, L. V. 2019. 大规模整体视频理解。载于欧洲计算机视觉会议。Fu, T.-J.; Li, L.; Gan, Z.; Lin, K.; Wang, W. Y.; Wang, L.; and Liu, Z. 2021. VIOLET：采用掩码视觉令牌建模的端到端视频语言Transformer。*arXiv*, abs/2111.12681. Ge, Y.; Ge, Y.; Liu, X.; Li, D.; Shan, Y.; Qie, X.; and Luo, P. 2022. BridgeFormer：通过多项选择题衔接视频文本检索。*arXiv*, abs/2201.04850. He, K.; Chen, X.; Xie, S.; Li, Y.; Doll'ar, P.; and Girshick, R. B. 2022. 掩码自编码器是可扩展视觉学习器。*2022 IEEE/CVF*计算机视觉与模式识别会议（*CVPR*）, 15979–15988. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; and Wu, E. 2017. 挤压-激励网络。*IEEE*模式分析与机器智能汇刊, 42: 2011–2023. Huang, J.; Li, Y.; Feng, J.; Sun, X.; and Ji, R. 2022. Clover：迈向统一视频语言对齐与融合模型。*arXiv*, abs/2207.07885. Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. TGIF-QA：迈向视觉问答中的时空推理。*2017 IEEE*计算机视觉与模式识别会议（*CVPR*）, 1359–1367. Lei, C.; Luo, S.; Liu, Y.; He, W.; Wang, J.; Wang, G.; Tang, H.; Miao, C.; and Li, H. 2021a. 通过对比式多模态预训练理解中文视频与语言。第*29*届*ACM*国际多媒体会议论文集。Lei, J.; Li, L.; Zhou, L.; Gan, Z.; Berg, T. L.; Bansal, M.; and Liu, J. 2021b. 少即是多：通过稀疏采样进行 Clipbert 视频与语言学习。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，7331–7341。Li, D.; Li, J.; Li, H.; Niebles, J. C.; and Hoi, S. C. H. 2021. Align and Prompt：使用实体提示进行视频与语言预训练。*arXiv*，abs/2112.09583。Li, L.; Chen, Y.-C.; Cheng, Y.; Gan, Z.; Yu, L.; and Liu, J. 2020. Hero：用于视频+语言全方位表征预训练的分层编码器。*arXiv*, abs/2005.00200。Liu, Y.; Xiong, P.; Xu, L.; Cao, S.; and Jin, Q. 2022. TS2-Net：用于文本-视频检索的令牌移位与选择变换器。载于欧洲计算机视觉会议论文集（*ECCV*）。Luo, H.; Ji, L.; Shi, B.; Huang, H.; Duan, N.; Li, T.; Li, J.; Bharti, T.; and Zhou, M. 2020. Univl：用于多模态理解与生成的统一视频与语言预训练模型。*arXiv*预印本 *arXiv:2002.06353*。Luo, H.; Ji, L.; Zhong, M.; Chen, Y.; Lei, W.; Duan, N.; and Li, T. 2021. CLIP4Clip：针对端到端视频片段检索的CLIP实证研究。*Neurocomputing*，508：293–304。Ma, Z.; Lou, M.; and Ouyang, X. 2021. QQ浏览器2021人工智能算法大赛赛道1冠军方案：多模态视频相似度。*arXiv*, abs/2111.01677。Miech, A.; Alayrac, J.-B.; Smaira, L.; Laptev, I.; Sivic, J.; and Zisserman, A. 2020. 从未经筛选的教学视频中端到端学习视觉表征。*2020 IEEE/CVF*计算机视觉与模式识别会议（*CVPR*），9876–9886。Miech, A.; Zhukov, D.; Alayrac, J.-B.; Tapaswi, M.; Laptev, I.; and Sivic, J. 2019. HowTo100M：通过观看数亿带旁白的视频片段学习文本-视频嵌入。*2019 IEEE/CVF*国际计算机视觉会议（*ICCV*），2630–2640。Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; 等。2019. PyTorch：一种命令式风格、高性能的深度学习库。神经信息处理系统进展，32：8026–8037。Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. 通过自然语言监督学习可迁移的视觉模型。载于*ICML*。Ryoo, M. S.; Piergiovanni, A. J.; Arnab, A.; Dehghani, M.; and Angelova, A. 2021. TokenLearner：8个学习得到的令牌能为图像和视频带来什么？*arXiv*，abs/2106.11297。Sharma, P.; Ding, N.; Goodman, S.; and Soricut, R. 2018. Conceptual Captions：一个经过清理并添加上位词的图像替代文本数据集，用于自动图像描述。载于*ACL*。Sun, C.; Myers, A.; Vondrick, C.; Murphy, K. P.; and Schmid, C. 2019. VideoBERT：用于视频与语言表征学习的联合模型。*2019 IEEE/CVF*国际计算机视觉会议（*ICCV*），7463–7472。

Wang, A.; Ge, Y.; Yan, R.; Ge, Y.; Lin, X.; Cai, G.; Wu, J.; Shan, Y.; Qie, X.; and Shou, M. Z. 2022a. All in One：探索统一视频-语言预训练。*arXiv*，abs/2203.07303。Wang, W.; Bao, H.; Dong, L.; Bjorck, J.; Peng, Z.; qiang liu; Aggarwal, K.; Mohammed, O. K.; Singhal, S.; Som, S.; and Wei, F. 2022b. 图像作为外语：面向所有视觉与视觉-语言任务的BEiT预训练。*arXiv*，abs/2208.10442。Xu, D.; Zhao, Z.; Xiao, J.; Wu, F.; Zhang, H.; He, X.; and Zhuang, Y. 2017. 通过对外观和运动进行逐步细化的注意力实现视频问答。载于*ACM*国际多媒体会议论文集，1645–1653。Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. MSR-VTT：一个用于衔接视频与语言的大型视频描述数据集。载于*IEEE*计算机视觉与模式识别会议论文集，5288–5296。Yang, A.; Miech, A.; Sivic, J.; Laptev, I.; and Schmid, C. 2022. 通过冻结的双向语言模型实现零样本视频问答。*arXiv*，abs/2206.08155。Zellers, R.; Lu, X.; Hessel, J.; Yu, Y.; Park, J. S.; Cao, J.; Farhadi, A.; and Choi, Y. 2021. MERLOT：多模态神经脚本知识模型。载于神经信息处理系统。Zhu, L.; and Yang, Y. 2020. ActBERT：学习全局-局部视频文本表示。*2020 IEEE/CVF*计算机视觉与模式识别会议（*CVPR*），8743–8752。