CLIP4Clip：用于端到端视频片段检索的 CLIP 经验研究

Huaishao Luo[1*] , Lei Ji[2] , Ming Zhong[3] , Yang Chen[3] , Wen Lei[3] , Nan Duan[2] , Tianrui Li[1]

[1]西南交通大学，中国成都 huaishaoluo@gmail.com, trli@swjtu.edu.cn[2] 微软亚洲研究院，中国北京
[3]中国北京微软科技与创新中心
{leiji,minzhon,emchen,wen.lei,nanduan}@microsoft.com

摘要

视频-文本检索在多模态研究中发挥着重要作用，并已广泛应用于现实世界的许多网络应用中。CLIP（对比语言-图像预训练）是一种图像语言预训练模型，它展示了从网络收集的图像-文本数据集中学习视频概念的能力。在本文中，我们提出了 CLIP4Clip 模型，以端到端的方式将 CLIP 模型的知识优势转移到视频语言检索中。本文通过实证研究探讨了几个问题：1) 图像特征是否足以用于视频文本检索？2) 基于 CLIP 的大规模视频文本数据集的后预训练对性能有何影响？3) 视频帧间时间依赖性建模的实用机制是什么？4) 模型在视频文本检索任务中的超参数灵敏度。大量实验结果表明，从 CLIP 移植而来的 CLIP4Clip 模型可以在各种视频文本检索数据集（包括 MSR-VTT、MSVC、LSMDC、ActivityNet 和 DiDeMo）上实现 SOTA 结果。我们在 https://github.com/ArrowLuo/CLIP4Clip 上发布了我们的代码。

1 导言

随着上传至网络的视频数量与日俱增，视频文本检索正成为人们高效查找相关视频的新需求。除了实际的网络应用之外，视频文本检索还是多模态视觉和语言理解的基础研究任务。我们可以直接根据其输入：原始视频（像素级）或视频特征（特征级）来区分之前的工作。

通常，预训练模型（Zhu 和 Yang，2020 年；Luo 等人，2020 年；Li 等人，2020 年；Gabeur

*这项工作是第一作者在 MSR 亚洲实习期间完成的。

等人，2020；Patrick 等人，2021；Rouditchenko 等人，2020）是特征级的，因为它们是在一些大规模视频文本数据集上训练的，例如 Howto100M（Miech 等人，2019）。输入是通过现成的冷冻视频特征提取器生成的缓存视频特征。如果输入是原始视频，预训练就会变得非常缓慢或不可行。不过，得益于大规模数据集，预训练模型在视频-文本检索方面表现出了显著的性能提升。

像素级方法直接将原始视频作为输入来训练模型（Torabi 等人，2016；Kiros 等人，2014；Yu 等人，2016a；Kaufman 等人，2017；Yu 等人，2017，2018）。早期的文献几乎都属于这种方法。这种方法与配对文本共同学习视频特征提取器。相反，特征级方法高度依赖于合适的特征提取器。它无法将学习结果传播回固定的视频编码器。

最近的一些研究开始采用像素级方法对模型进行预训练，使预训练模型从原始视频中学习。如何减少密集视频输入的高计算量是一大挑战。通常情况下，ClipBERT（Lei 等人，2021 年）采用稀疏采样策略来实现端到端的预训练。具体来说，该模型在每个训练步骤中只对视频中的一个或几个短片段进行稀疏采样。结果表明，端到端训练有利于底层特征提取。稀疏采样的几个片段就足以解决视频文本检索任务。Frozen（Bain 等人，2021 年）将图像视为单帧视频，并设计了一个课程学习计划，在图像和视频数据集上训练模型。研究结果表明，课程学习计划从图像到多帧的渐进学习可以提高效率。我们的目标不是预先训练一个新的视频-文本检索模型。我们主要研究的是如何将知识从图像中转移到视频中。

本文将图像-文本预训练模型 CLIP（Radford 等人，2021 年）应用于视频-文本检索。

我们利用预训练的 CLIP，提出了一个名为 **CLIP4Clip（CLIP 用于视频片段检索）**的模型来解决视频文本检索问题。具体地说，CLIP4Clip 是在 CLIP 的基础上构建的，它设计了一个相似性计算器来研究三种相似性计算方法：无参数型、顺序型和紧密型。与我们的工作一样，Portillo-Quintero 等人（2021 年）的并行工作也是建立在 CLIP 的基础上，用于视频文本检索。不同的是，他们的工作直接利用 CLIP 进行零镜头预测，而没有考虑不同的相似性计算机制。然而，我们设计了一些相似性计算方法来提高性能，并以端到端的方式训练模型。我们工作的贡献在于1）我们研究了基于预训练 CLIP 的三种相似性计算机制；2）我们进一步在有噪声的大规模视频语言数据集上对 CLIP 进行后预训练，以学习更好的检索空间。大量实验表明，我们的模型在 MSR-VTT（Xu 等人，2016）、MSVC（Chen 和 Dolan，2011）、LSMDC（Rohrbach 等人，2015）、ActivityNet（Krishna 等人，2017a）和 DiDeMo（Hendricks 等人，2017）数据集上取得了新的 SOTA 结果。

此外，我们还可以从大量实验中总结出以下几点启示：

1) 对于用于视频文本检索的视频编码来说，单幅图像远远不够。

2) 需要在 CLIP4Clip 模型的大规模视频-文本数据集上进行后期预训练，这样可以提高性能，尤其是在零镜头预测方面，提高幅度很大。

3) 有了强大的预训练 CLIP，对于小数据集，最好不要引入新参数，而采用视频帧均值池机制。同时，对于大型数据集，最好引入更多参数（如自我注意层）来学习时间依赖性。

4) 我们仔细研究了超参数，并报告了最佳设置。

2 相关作品

视频编码器骨干 之前的工作主要集中在视频表示的 2D/3D 时空对话（Tran 等人，2015 年；Xie 等人，2018 年；Feichtenhofer 等人，2019 年）。最近，基于变换器的图像编码器 ViT（Dosovitskiy 等人，2021 年）引起了广泛关注。

基于变换器的视频编码器仍处于用于动作分类的早期阶段（Bertasius 等人，2021 年；Arnab 等人，2021 年）。我们主要研究基于变压器的视频骨干网在多模态视频-文本检索中的有效应用。

从文本监督中学习视觉表征 视觉表征学习是一项具有挑战性的任务，人们已经用监督或自我监督的方法对其进行了广泛研究。考虑到从大规模未编辑数据中进行语义监督，从文本表征中学习视觉表征（Radford 等人，2021 年；Miech 等人，2020 年；Lei 等人，2021 年）是一个新兴的研究课题，它得益于从互联网上收集的大规模视觉和语言对。CLIP（对比语言-图像预训练）（Radford 等人，2021 年）取得了显著的成功，证明了其通过对大规模图像和文本对进行预训练，从语言监督中学习 SOTA 图像代表的能力。预训练模型可以学习图像的细粒度视觉概念，并将这些知识用于检索任务。通常，MIL-NCE（Miech 等人，2020 年）主要研究如何利用有噪声的大规模 Howto100M（Miech 等人，2019 年）结构化视频，以端到端方式学习更好的视频编码器。此外，ClipBERT（Lei 等人，2021 年）通过稀疏采样提出了一种高效的端到端方法，并指出图像语言数据集的预训练有助于更好地初始化视频文本检索。与 ClipBERT 不同，我们采用了基于变换器视觉骨干的 CLIP，并将这种图像语言预训练模型扩展到视频语言预训练，用于视频文本检索。考虑到视频的时间序列，我们使用了二维/三维线性嵌入和视觉变换器上的相似性计算器来捕捉时间序列特征。

视频-文本检索 早期的视频-文本检索研究（Torabi等人，2016；Kiros等人，2014；Yu等人，2016a；Kaufman等人，2017；Yu等人，2017，2018）为跨模态学习设计了密集的融合机制。最近，预训练模型（Zhu 和 Yang，2020 年；Amrani 等人，2021 年；Luo 等人，2020 年；Li 等人，2020 年；Miech 等人，2020 年；Gabeur 等人，2020 年；Patrick 等人，2021 年；Lei 等人，2021 年；Dzabraev 等人，2021 年；Liu 等人，2021 年）在视频-文本再三维检索的排行榜上独占鳌头，并在零镜头检索方面取得了显著成果。

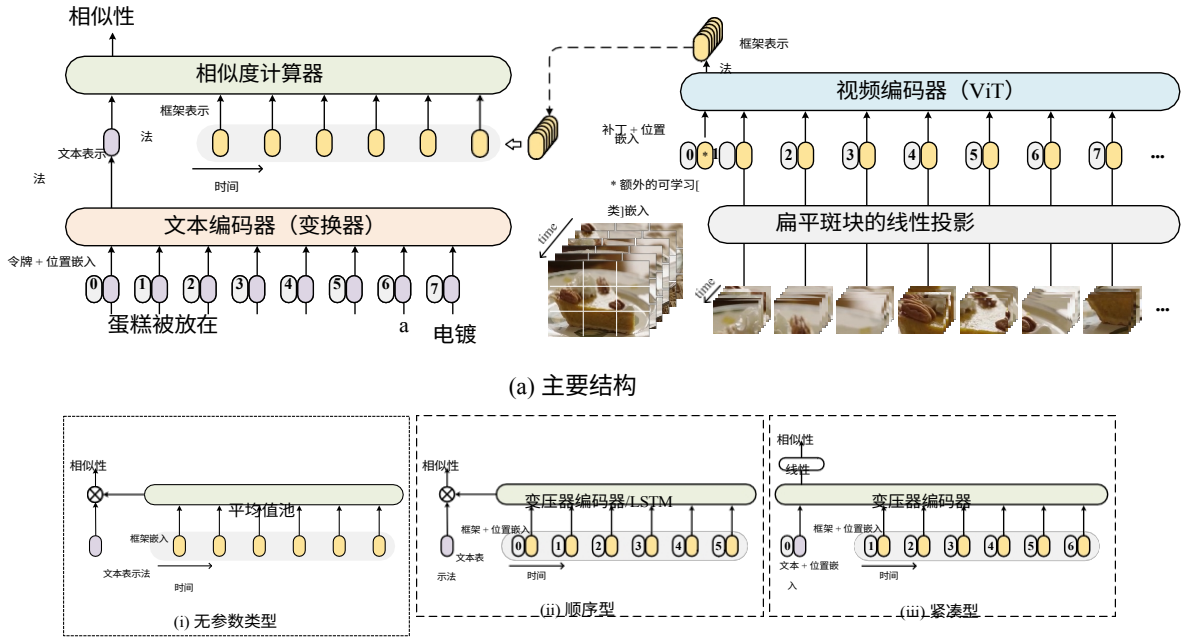


图 1：CLIP4Clip 的框架由三个部分组成，包括两个单模态编码器和一个相似性计算器。该模型将一对视频和文本作为输入。对于输入视频，我们首先将输入视频采样为顺序帧（图像）。然后，将这些图像帧重塑为一系列扁平化的二维斑块。这些斑块通过线性斑块嵌入层映射到一维嵌入序列，然后输入到图像编码器中进行表示，就像在 ViT（Dosovitskiy 等人，2021 年）中那样。最后，相似性计算器会预测这些帧的文本表示和表示序列之间的相似性得分。在这项工作中，我们研究了三种类型的相似性计算器，包括无参数型、连续型和紧密型。⊗ 表示余弦相似度。我们使用 CLIP (ViT-B/32)（Radford 等人，2021 年）初始化了两个单模态编码器。

和微调检索。与我们的工作同时，Portillo-Quintero 等人（2021 年）将 CLIP 应用于零镜头预测，而 Bain 等人（2021 年）则提出了一种基于变换器的视频骨干网。我们

我们建议直接移植预训练 CLIP 的强大知识，并在大规模视频语言数据集上继续预训练所设计的基于视频的 CLIP4Clip。实证研究证明了 CLIP4Clip 模型的有效性。

3 框架

给定一组视频（或视频片段）V 和一组字幕 T，我们的目标是学习一个函数 $s(v_i, t_j)$ 来计算视频（或视频片段）$v_i \in V$ 和字幕 $t_j \in T$ 之间的相似度。其目的是在文本到视频检索中根据相似度得分对查询标题给出的所有视频（或视频片段）进行排序，或在视频到文本检索任务中对查询视频（或视频片段）给出的所有标题进行排序。$s(v_i, t_j)$ 的目标是为相关的视频文本对计算高相似度，为不相关的视频文本对计算低相似度。

在本文中被视为帧（图像）序列。从形式上看，视频（或视频片段）v_i 是由$|v_i|$采样帧组成的，这样 $v_i = \{v_i^1, v_i^2, \ldots, v_i^{|v_i|}\}$。我们的模型是一个端到端

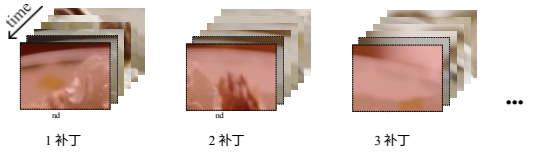
视频（或视频片段）$v_i \in V$ 表示

方式（E2E）通过将帧作为输入直接对像素进行训练。图 1 展示了我们的框架，主要包括文本编码器、视频编码器和相似度计算器。本节将详细介绍每个部分。

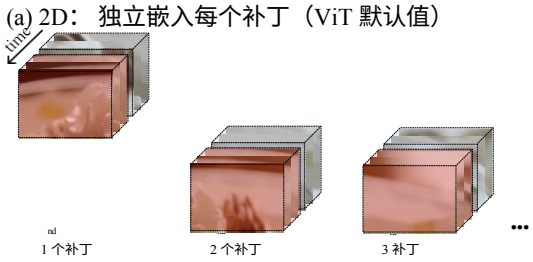
3.1 视频编码器

为了获得视频表示，我们首先从视频片段中提取帧，然后通过视频编码器对其进行编码，以获得特征序列。在本文中，我们采用有 12 层、补丁大小为 32 的 ViT-B/32（Dosovitskiy 等人，2021 年）作为视频编码器。具体来说，我们使用预先训练好的 CLIP (ViT-B/32)（Radford 等人，2021 年）作为骨干，主要考虑将图像表示转换为视频表示。预训练的 CLIP (ViT-B/32) 在本文的视频文本检索任务中非常有效。

ViT（Dosovitskiy 等人，2021 年）首先提取不重叠的图像斑块，然后对这些图像斑块进行 "渲染"。



(a) 2D： 独立嵌入每个补丁（ViT 默认值）



(b) 3D： 跨时间嵌入补丁

图 2：视频编码器中扁平斑块线性投影的不同视图。带颜色的虚线框为内核。

我们使用线性投影法将它们投射到一维标记中，并利用变换器架构来模拟输入图像的每个片段之间的相互作用，从而得到最终的表示。继 ViT 和 CLIP 之后，我们使用 [class] 标记的输出作为图像表示。对于输入视频 v 的帧序列 $_i = \{v^1, v^2, .., v^{|v_i|}_i\}$、$_i$ $_i$ 生成的表示可以表示为 $\mathbf{Z}_i = \{\mathbf{z}^1_i, \mathbf{z}^2_i, .., \mathbf{z}^{|v_i|}_i\}$。

我们研究了

a) ViT 的扁平斑块线性投影被视为二维线性投影，它对每个二维帧斑块进行独立投影。b) 因此，我们研究了一种三维线性投影，类似于（Arnab 等人，2021 年），以加强时间特征提取。图 2 显示了二维和三维的比较。三维线性嵌入了跨时间的斑块。具体来说，三维线性使用核为 $[t \times h \times w]$ *的*三维卷积作为线性，而不是二维线性中的核为 $[h \times w]$，其中 t、h 和 w 分别为时间维度、高度维度和宽度维度。

3.2 文本编码器

我们直接应用 CLIP 中的文本编码器来生成字幕表示。文本编码器是一个转换器（Vaswani et al.

Rad- ford 等人，2019）中描述的架构修改。它是一个 12 层 512 宽模型

3.3 相似度计算器

提取视频表示后 $\mathbf{Z}_i = \{\mathbf{z}^1_i, \mathbf{z}^2_i, .., \mathbf{z}^{|v_i|}_i\}$ 和标题表示 $\mathbf{w}$、$_j$

关键点在于相似性计算。由于我们的模型是基于预先训练好的图像-文本模型建立的，因此我们应该在相似性计算器模块中仔细添加新的可学习权重。

ule。没有权重初始化很难学习

因此，我们根据模块是否引入新的学习参数，将相似性计算器的机制分为三类。因此，我们根据模块是否引入新参数进行学习，将相似度计算器的机制分为三类。*无参数*方法，即 "意义池"（meaning pool-ing），无需新参数即可融合视频表示。此外，还有两种方法会引入新的权重进行学习，包括*顺序*型和*紧密*型方法，新权重的大小各不相同。图 1b 展示了这三种机制的详细结构。无参数类型和顺序类型相似度计算器属于松散类型，分别采用视频和文本表示的两个独立分支来计算余弦相似度。而紧密型相似度计算器则使用转换器模型进行多模态交互，并通过线性投影进一步计算相似度，这两种方法都包含新的权重学习。

无参数类型 根据 CLIP，通过对图像和文本对的大规模预训练，帧表示 $\mathbf{Z}_i$ 和标题表示 $\mathbf{w}_j$ 已被层归一化并线性地投射到多模态嵌入空间中。自然的思路是采 用 无参数类型，从视频角度直接计算与图像/帧的相似度。无参数类型首先使用均值池法汇总所有帧的特征，得到一个 "平均帧"，即 $\hat{\mathbf{z}}_i = \texttt{mean-pooling}(\mathbf{z}^1_i, \mathbf{z}^2_i, \ldots, \mathbf{z}^{|v_i|}_i)$. 那么相似度函数 $s(v_i, t_j)$ 定义为余弦相似性、

$$s(v_i, t_j) = \frac{\mathbf{w}_j^\top \hat{\mathbf{z}}_i}{\|\mathbf{w}_j\| \|\hat{\mathbf{z}}_i\|}. \quad (1)$$

有 8 个注意头。根据 CLIP，[EOS] 标记处转换器最高层的激活被视为标题的特征表示。对于标题 $t_j \in \mathsf{T}$，我们将其表示为 $\mathbf{w}_j$ 。

序列类型 平均池化操作会忽略帧间的序列信息。因此，我们探索了两种为序列类型相似性计算器建立序列特征模型的 方法。 一种是 LSTM（Hochreiter

和 Schmidhuber，1997；Gers 等人，2002），另一种是带有位置嵌入 **P** 的变换器编码器（Vaswani 等人，2017），这两种编码器都是序列特征的有效模型。我们将它们分别表述为 $\tilde{Z}_i = \texttt{LSTM}(\mathbf{Z}_i)$ 和 $\tilde{Z}_i = \texttt{Transformer-Enc}(\mathbf{Z}_i + \mathbf{P})$。通过编码，$\tilde{Z}_i$ 已经包含了时间信息。后续操作与无参数类型相似性计算器相同，相似性函数也是式 (1)，$\mathbf{Z}_i = \texttt{mean-pooling}(\tilde{Z}_i)$。

严密型 与上述无参数型和顺序型不同，严密型使用变形编码器（Vaswani 等人，2017 年）进行视频与字幕之间的多模态交互，类似于（Luo 等人，2020 年），并通过线性层预测相似性，引入最未初始化的权重。首先，变换器编码器将字幕表示 $\mathbf{w}_j$ 和帧表示 $\mathbf{Z}_i = \{\mathbf{z}_i^1, \mathbf{z}_i^2, \ldots, \mathbf{z}_i^{|v_i|}\}$ 作为融合特征 $\mathbf{U}_i$ 制定为：

$$\mathbf{U}_i = [\mathbf{w}_j, \mathbf{z}_i^1, \mathbf{z}_i^2, \ldots, \mathbf{z}_i^{|v_i|}], \quad (2)$$

$$\tilde{\mathbf{U}}_i = \texttt{Transformer-Enc}(\mathbf{U}_i + \mathbf{P} + \mathbf{T})。 \quad (3)$$

其中，[,]表示连接操作。**P** 是位置嵌入，**T** 是类型嵌入，类似于 BERT 中的段嵌入（Devlin 等人，2019 年）。**T** 包含两种类型的嵌入，一种是标题嵌入，另一种是视频帧嵌入。接下来，我们用两个线性投影层计算相似度得分，并对最后一层 $\tilde{\mathbf{U}}_i[0, :]$ 的第一个标记输出进行激活。具体来说，相似度函数 $s(v_i, t_j) = \texttt{FC}(\texttt{ReLU}(\texttt{FC}(\tilde{\mathbf{U}}_i[0, :])))$，其中 FC 是线性投影，ReLU 指 ReLU 激活函数（Agarap，2018）。

3.4 培训战略

损失函数 给定一批 B（视频、文本）或（视频片段、文本）对，模型需要生成并优化 $B \times B$ 相似性。我们在这些相似性得分上使用对称交叉熵损失来训练模型参数、

$$\mathcal{B}$$

损失 L 是视频到文本的损失 L_{v2t} 和文本到视频的损失 L_{t2v} 之和。

帧取样 由于我们的模型是通过将帧作为输入直接对像素进行训练的，因此提取帧是一项重要的策略。有效的采样策略需要考虑信息丰富度和计算复杂度（尤其是内存成本）之间的平衡。为了考虑视频（或视频片段）中的顺序信息，我们采用了均匀帧采样策略，而不是（Lei 等人，2021 年）中使用的随机稀疏采样策略。采样率为每秒 1 帧。此外，我们还在实验中研究了不同的帧长和不同的提取位置。

预训练 尽管 CLIP 可以有效地学习图像的视觉概念，但从视频中学习时间特征也是必不可少的。为了进一步将知识迁移到视频中，我们在 Howto100M 数据集（Miech 等人，2019 年）上对 CLIP4Clip 模型进行了后预训练。出于效率考虑，在视频-文本数据集上进行预训练极具挑战性。我们进行了初步探索，使用 "食品和娱乐 "类别（约 38 万个视频）作为预训练后数据集（本文其他部分称为 *Howto100M-380*k）。我们采用 MIL-NCE loss（Miech 等人，2020 年）来优化无参数类型中的 CLIP。优化器为 Adam（Kingma 和 Ba，2015 年），学习率为 1e-8。标记长度为 32，帧长为 12，批量大小为 48。训练在 8 台英伟达 Tesla V100 GPU 上进行。我们运行了 5 个历元，耗时约 2 周。在本文中，预训练后测试可视为对这一方向的初步研究，供今后工作参考。

4 实验

我们首先介绍了数据集和实施细节，然后展示了五个数据集的最新结果。然后，我们消减了模型的各种设置。最后，我们讨论了一

些有前景的方向。

$$L_{v2t} = -\frac{1}{B}\sum_{i} \log \frac{\exp(s(v_i, t))_i}{\sum_{j=1}^{B} \exp(s(v_{i,j} t)}, \quad (4)$$

$$L_{t2v} = -\frac{1}{B}\sum^{B} \log \frac{\exp(s(\gamma_t, t))}{\sum_{B} \exp(s(v, t)}, \quad (5)$$

$$i \qquad j=1 \qquad j\ i$$

$$L = L_{v2t} + L_{t2v} \circ \quad (6)$$

4.1 数据集

我们在五个数据集上验证了我们的模型：MSR-VTT、MSVC、LSMDC、ActivityNet 和 DiDeMo。

MSR-VTT 数据集（Xu 等人，2016 年）由以下部分组成

10,000 个视频，每个视频的长度从 10 秒到 32 秒不等，以及 200,000 个标题。我们

方法	TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
C+LSTM+SA[a]	M	C	4.2	12.9	19.9	55	-
VSE[b]	M	C	3.8	12.7	17.1	66	-
SNUVL[c]	M	C	3.5	15.9	23.8	44	-
考夫曼等人[d]	M	C	4.7	16.6	24.1	41	-
CT-SAN[e]	M	C	4.4	16.6	22.3	35	-
JSFusion[f]	M	C	10.2	31.2	43.2	13	-
HowTo100M[g]	H+M	C	14.9	40.2	52.8	9	-
ActBERT[h]	H+M		8.6	23.4	33.1	36	-
噪音E[i]	H+M		17.4	41.6	53.6	8	-
UniVL[j]	H+M		21.2	49.6	63.1	6	-
英雄[k]	H+M		16.8	43.4	57.7	-	-
ClipBERT[L]	C+G+M	C	22.0	46.8	59.9	6	-
(我们的）-平均值P	W+M	C	**42.1**	**71.9**	**81.4**	**2**	**15.7**
(我们的）-seqLSTM	W+M	C	41.7	68.8	78.7	**2**	16.6
(我们的）-seqTransf	W+M	C	42.0	68.6	78.7	**2**	16.2
(我们的）--严密转移	W+M	C	37.8	68.4	78.4	**2**	17.2

(a) 培训 7K

方法	TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
MIL-NCE H[m]		C	9.9	24.0	32.4	29.5	-
CLIP-straight[n]	WC		31.2	53.7	64.2	4	-

(b) 零射

方法	TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
CE[o]	M		20.9	48.8	62.4	6	28.2
MMT[p]	H+M		26.6	57.1	69.6	4	24.0
AVLnet[q]	H+M		27.1	55.6	66.6	4	-
SSB[r]	H+M		30.1	58.5	69.3	3	-
MDMMT[s]	MD+M		38.9	69.0	79.7	**2**	16.5
冷冻[t]	CW+M	C	31.0	59.5	70.5	3	-
HiT[u]	H+M		30.7	60.9	73.2	2.6	-
TT-CE+[v]	M		29.6	61.6	74.2	3	-
(我们的）-平均值P	W+M	C	43.1	70.4	80.8	**2**	16.2
(我们的）-seqLSTM	W+M	C	42.5	70.8	80.7	**2**	16.7
(我们的）-seqTransf	W+M	C	**44.5**	71.4	**81.6**	**2**	15.3
(我们的）--严密转移	W+M	C	40.2	**71.5**	80.5	**2**	**13.4**

(c) 培训培训-9K

表 1：MSR-VTT 数据集的文本到视频检索结果。表(a)和(c)列出了数据集不同分集的结果。Training-7K "沿用了（Miech 等人，2019 年）的数据拆分，"Training-9K "沿用了（Gabeur 等人，2020 年）的数据拆分。它们的测试集相同，但训练集不同。在每个表格中，"TrainD "列显示了用于预训练和训练的数据集，其中 M、H、W、C、G 表示 MSR-VTT、HowTo100M（Miech 等人，2019 年）、WIT（Radford 等人，2021 年）、COCO Captions（Chen 等人，2015 年）和 Visual Genome Captions（Krishna 等人，2017 年 b）。此外，MDMMT（Dzabraev et al.，2021）中使用的 MD 表示包括 MSR-VTT、LSMDC、HowTo100M 等多域数据集的组合，CW 表示 CC3M（Sharma et al.，2018）加上 WebVid-2M（Bain et al.，2021）。带 Cmeans 的 "E2E "一栏是指以端到端的方式从原始视频中进行训练。基线方法有：[a] C+LSTM+SA (Torabi et al., 2016)、[b] VSE (Kiros et al., 2014)、[c] SNUVL (Yu et al., 2016b)、[d] Kaufman et al. Kaufman et al. (2017)、[e] CT-SAN (Yu et al., 2017)、[f] JSFusion (Yu et al., 2018)、[g] HowTo100M (Miech et al.、2019),[h] ActBERT (Zhu and Yang, 2020),[i] NoiseE (Amrani et al., 2021),[j] UniVL (Luo et al., 2020),[k] HERO (Li et al., 2020),[l] ClipBERT (Lei et al.、2021),[m] MIL-NCE (Miech et al., 2020),[n] CLIP-straight (Portillo-Quintero et al., 2021),[o] CE (Liu et al., 2019),[p] MMT (Gabeur et al., 2020)、[q]AVLnet (Rouditchenko et al., 2020),[r] SSB (Patrick et al., 2021),[s] MDMMT (Dzabraev et al., 2021), Frozen[t] (Bain 等人，2021 年）、[u] HiT（Liu 等人，2021 年）、[v] TT-CE+（Croitoru 等人，2021 年）。

使用 "Training-7K "和 "Training-9K "两种数据分片与基线进行比较。训练-7K "沿用了（Miech 等人，2019 年）的数据拆分，而 "训练-9K "则沿用了（Gabeur 等人，2020 年）的数据拆分。这两个拆分中的测试数据都是 "test 1k-A"，其中包含 1,000 个剪辑-文本对，遵循 JSFusion（Yu 等人，2018 年）。如果没有额外注释，我们使用 "Training-9K "作为默认设置。

MSVD（Chen 和 Dolan，2011 年）包含 1,970 个视频，每个视频的长度从 1 秒到 62 秒不等。训练、验证和测试部分分别包含 1,200 个、100 个和 670 个视频。每个视频都有大约 40 个相关的英语句子。

LSMDC （Rohrbach 等人， 2015 年） 由 118,081 个视频组成，每个视频的长度从 2 秒到 30 秒不等。这些视频是从 202 部电影中提取的。验证集包含 7,408 个视频，测试集包含来自电影的 1,000 个视频。

独立于训练和验证分割。**ActivityNet**（克里希纳等人，2017a）由 20,000 个 YouTube 视频组成。我们按照以下设置
从（Zhang 等人，2018 年；Gabeur 等人，2020 年）到
将视频中的所有描述串联起来形成一个段落，并通过对 "val1 "分割进行视频-段落检索来评估模型。

DiDeMo （Hendricks 等人， 2017 年） 包含 10,000 个视频，注释了 40,000 个句子。我们按照（Liu 等人，2019 年；Lei 等人，2021 年；Bain 等人，2021 年）的方法对视频段落检索进行了评估，将视频的所有句子描述串联成一个查询。

我们使用标准检索指标：排名 K 的召回率（R@K，越高越好）、中位数排名（MdR，越低越好）和平均排名（MnR，越低越好）来评估我们模型的性能。R@K（*K 级召回率*）计算的是在查询样本的前 K *个*检索点中找到正确结果的测试样本的百分比。我们

方法	TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
多种线索[a]	M	C	20.3	47.8	61.1	6	-
CE[b]	M		19.8	49.0	63.8	6	-
SSB[c]	H+M		28.4	60.0	72.9	4	-
噪音E[d]	H+M		20.3	49.0	63.3	6	-
CLIP-straight[e]	W	C	37.0	64.1	73.8	3	-
冰冻[f]	CW+M	C	33.7	64.7	76.3	3	-
TT-CE+[g]	M		25.4	56.9	71.3	4	-
(我们的）-平均值P	W+M	C	**46.2**	**76.1**	**84.6**	**2**	**10.0**
(我们的）-seqLSTM	W+M	C	**46.2**	75.3	84.5	**2**	10.2
(我们的）-seqTransf	W+M	C	45.2	75.5	84.3	**2**	10.3
(我们的）--严密转移	W+M	C	40.0	71.5	82.1	**2**	13.3

表 2：在 MSVD 数据集上进行文本到视频检索的结果。在 "TrainD "一栏中，M、H 和 W 表示在 MSVD、HowTo100M（Miech 等人，2019 年）和 WIT（Radford 等人，2021 年）上进行的训练，CW 表示 CC3M（Sharma 等人，2018 年）加上 WebVid-2M（Bain 等人，2021 年）。带 Cmeans 的 "E2E "一栏指的是以端到端的方式从原始视频中进行训练。基线方法有：[a] Multi Cues (Mithun et al., 2018)、[b] CE (Liu et al., 2019)、[c] SSB (Patrick et al., 2021)、[d] NoiseE (Amrani et al., 2021)、[e] CLIP-straight (Portillo- Quintero et al., 2021)、[f] Frozen (Bain et al., 2021)、[g] TT- CE+ (Croitoru et al., 2021)。

方法	TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
CT-SAN[a]	L	C	5.1	16.3	25.2	46.0	-
JSFusion[b]	L	C	9.1	21.2	34.1	36.0	-
CE[c]	L		11.2	26.9	34.8	25.3	96.8
MMT[d]	H+L		12.9	29.9	40.1	19.3	75.0
噪音E[e]	H+L		6.4	19.8	28.4	39.0	-
CLIP-straight[f]	L	C	11.3	22.7	29.2	56.5	-
MDMMT[g]	MD+L		18.8	38.5	47.9	12.3	**58.0**
冷冻[h]	CW+L	C	15.0	30.8	39.8	20.0	-
HiT[i]	H+L		14.0	31.2	41.6	18.5	-
TT-CE+[j]	L		17.2	36.5	46.3	13.7	-
(我们的）-平均值P	W+L	C	20.7	38.9	47.2	13.0	65.3
(我们的）-seqLSTM	W+L	C	21.6	**41.8**	**49.8**	**11.0**	**58.0**
(我们的）-seqTransf	W+L	C	**22.6**	41.0	49.1	**11.0**	61.0
(我们的）--严密转移	W+L	C	18.9	37.8	46.7	13.0	61.6

表 3：LSMDC 数据集上的文本到视频检索结果。在 "TrainD "一栏中，L、H 和 W 表示在 LSMDC、HowTo100M（Miech 等人，2019 年）和 WIT（Radford 等人，2021 年）上进行的训练，MD 用于

方法	TrainD	E2E	R@1↑	R@5↑	R@50↑	MdR↓	MnR↓
FSE[a]	A		18.2	44.8	89.1	7.0	-
CE[b]	A		18.2	47.7	91.4	6.0	23.1
HSE[a]	A		20.5	49.3	-	-	-
MMT[c]	H+A		28.7	61.4	94.5	3.3	16.0
SSB[d]	H+A		29.2	61.6	94.7	3.0	-
HiT[e]	H+A		29.6	60.7	95.6	3.0	-
ClipBERT[f]	C+G+A	C	21.3	49.0	-	6.0	-
TT-CE+[g]	A		23.5	57.2	96.1	4.0	-
（我们的）-平均值P	W+A	C	**40.5**	**72.4**	98.1	**2.0**	7.4
（我们的）-seqLSTM	W+A	C	40.1	72.2	98.1	**2.0**	**7.3**
（我们的）-seqTransf	W+A	C	**40.5**	**72.4**	**98.2**	**2.0**	7.5
（我们的）--严密转移	W+A	C	19.5	47.6	93.1	6.0	17.3

表 4：Activi-tyNet 数据集上的文本到视频检索结果。在 "TrainD "一栏中，A、H、W、C 和 G 表示在 ActivityNet、HowTo100M（Miech 等人，2019 年）、WIT（Radford 等人，2021 年）、COCO Captions（Chen 等人，2015 年）和 Visual Genome Cap- tions（Krishna 等人，2017 年 b）上进行的训练。列 "E2E "与 Cmeans 以端到端的方式从原始视频中进行训练。 基线方法有[a] FSE,HSE（Zhang 等人，2018 年）、[b] CE（Liu 等人，2019 年）、[c] MMT（Gabeur 等人，2020 年）、[d] SSB（Patrick 等人，2021 年）、[e] HiT（Liu 等人，2021 年）、[f] ClipBERT（Lei 等人，2021 年）、[g] TT-CE+（Croitoru 等人，2021 年）。

方法	TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
S2VT[a]	D	C	11.9	33.6	-	13.0	-
FSE[b]	D		13.9	36.0	-	11.0	-
CE[c]	D		16.1	41.1	-	8.3	43.7
ClipBERT[d†]	C+G+D	C	20.4	48.0	60.8	6.0	-
冷冻[e†]	CW+D	C	34.6	65.0	74.7	3.0	-
TT-CE+[f]	D		21.6	48.6	62.9	6.0	-
(我们的）-平均值P	W+D	C	**43.4**	**70.2**	**80.6**	**2.0**	**17.5**
(我们的）-seqLSTM	W+D	C	**43.4**	69.9	80.2	**2.0**	**17.5**
(我们的）-seqTransf	W+D	C	42.8	68.5	79.2	**2.0**	18.9
(我们的）--严密转移	W+D	C	25.8	52.8	66.3	5.0	27.3

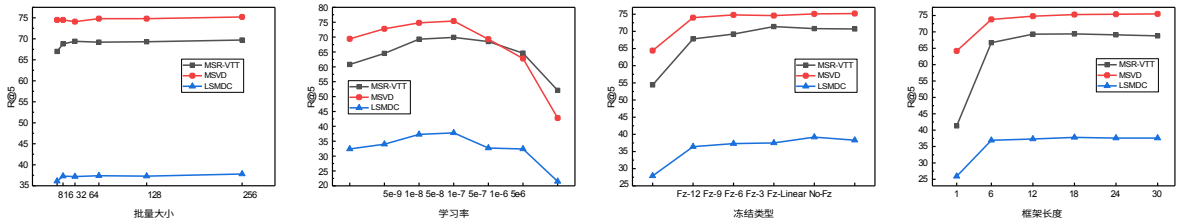
（Dz- abraev et al、2021）中使用的 MD 表示包含 MSR-VTT、LSMDC、HowTo100M 等的组合多域数据集，CW 表示 CC3M（Sharma 等，2018）加上 WebVid-2M（Bain 等，2021）。带 Cmeans 的 "E2E

"一栏是指以端到端的方式从原始视频中进行训练。基线方法有：[a] CT-SAN (Yu et al., 2017)、[b] JSFusion (Yu et al., 2018)、[c] CE (Liu et al., 2019)、[d] MMT (Gabeur et al., 2020)、[e] NoiseE (Amrani et al.、 2021)、[f] CLIP-straight (Portillo-Quintero et al., 2021),[g] MDMMT (Dzabraev et al., 2021), [h]Frozen (Bain et al., 2021),[i] HiT (Liu et al., 2021), TT-CE+ ([j]Croitoru et al., 2021)。

报告 R@1、R@5 和 R@10（或 ActivityNet 的 R@50）的结果。中位数排名计算

表 5：DiDeMo 数据集上的文本到视频检索结果。在 "TrainD "一栏中，D、H、W、C 和 G 表示在 DiDeMo、HowTo100M（Miech 等人，2019 年）、WIT（Radford 等人，2021 年）、COCO Captions（Chen 等人，2015 年）和 Visual Genome Captions（Kr- ishna 等人，2017 年b）上进行的训练，CW 表示 CC3M（Sharma 等人，2018 年）加上 WebVid-2M（Bain 等人，2021 年）。带 C 的 "E2E "一栏指的是以端到端的方式从原始视频中进行训练。† 表示候选视频是使用地面实况建议进行分类的。基线方法有[a] S2VT（Venugopalan 等人，2015 年）、[b] FSE（Zhang 等人，2018 年）、[c] CE（Liu 等人，2019 年）、[d] ClipBERT（Lei 等人，2021 年）、[e] Frozen（Bain 等人，2021 年）、[f] TT-CE+（Croitoru 等人，2021 年）。

中位数。同样，平均排名计算所有正确结果的平均排名。



(a) 批量大小 (b) 学习率 (c) 冻结层 (d) 框架长度

图 3：不同批次大小、帧长、冻结层和学习率下的检索结果。**批次大小**：冻结层为 6。 **学习率**：批次大小为 128，冻结层为 6，帧长为 12。**冻结层**：Fz-[NO.]表示冻结第[NO.]-层以下的各层（包括第[NO.]-层），Fz-Linear 表示只冻结底部的线性层，No-Fz 表示不冻结训练，批量大小为 128，帧长为 12，学习率为 5e-8。**帧长**：批量为 128，冻结层为 6，学习率为 5e-8。

4.2 实验细节

在本章中，我们使用 CLIP (ViT-B/32)（Radford 等人，2021 年）初始化文本编码器和视频编码器。实际问题是如何初始化相似性计算器中的参数，例如顺序类型的参数。我们的解决方案是重新使用 CLIP（ViT-B/32）中的类似参数。具体而言，对于顺序类型和紧密类型中的位置嵌入，我们通过重复 CLIP 文本编码器中的位置嵌入进行初始化。同样，变换器编码器也是通过预训练的 CLIP 图像编码器的相应层权重来初始化的。其余参数，如 LSTM 和线性投影，则随机初始化。第 3.1 节中三维线性和二维线性的时间维度 t、高度维度 h 和宽度维度 w 分别设置为 3、32、32。对于三维线性，我们在时间维度上将步长和填充设为 1。我们按照（Arnab 等人，2021 年）的方法，使用 "中心帧初始化 "策略从 CLIP 的预训练二维线性中初始化三维线性。具体来说，我们使用 CLIP 的二维权重 $\mathbf{E}_{2D}$ 中的 $[\mathbf{0}, \mathbf{E}_{2D}, \mathbf{0}]$。

我们使用 Adam 优化器（Kingma 和 Ba，2015 年）对模型进行微调。至于学习率，我们按照 CLIP（Radford 等人，2021 年）使用余弦计划（Loshchilov 和 Hutter，2017 年）进行衰减。如无特殊说明，文本编码器和视频编码器（包括线性投影）的初始学习率为 1e-7，新模块（如 LSTM）的初始学习率为 1e-4，标题标记长度为 32，帧长度为 12，批量大小为 128，运行 5 个 epochs。在我们的实验中，LSTM 的层数为 1，连续型和紧密型的变换编码器层数均为 4。所有微调实验均在 4 台英伟达™（NVIDIA®）显卡上进行。

Tesla V100 GPU。请注意，ActivityNet 和 DiDeMo 被视为视频段落的再三元组，因此我们将标题标记长度和帧长度设置为 64。对它们的实验是在 16 个英伟达 Tesla V100 GPU 上进行的。

4.3 与最新技术的比较

我们将基于预训练 CLIP 的各种类型的相似性计算器与最先进的计算器（SOTA）进行比较："-meanP"、"-seqLSTM"、"-seqTransf "和 "-tightTransf "是第 3.3 节中提到的无参数类型（即均值池）、LSTM 的顺序类型、前编码器（Trans- former Encoder）和紧密类型的简称。表 1-5 列出了我们的模型在 MSR-VTT、MSVC、LSMDC、ActivityNet 和 DiDeMo 上的文本到视频再评估结果。每个表的标题中都列出了每个数据集的基线，以便说明。与所有基线相比，我们在所有五个数据集上都以较大优势取得了 SOTA 结果。我们发现，通过我们的结果和同时进行的 CLIP-straight（Portillo- Quintero 等人，2021 年），检索性能的增长得益于预训练的 CLIP。此外，我们的端到端微调所带来的改进证明了图像-文本预训练模型在视频-文本三重检索方面的潜力。

在 MSR-VTT 数据集上，无参数类型（-meanP）模型在 "训练-7k "数据拆分中取得了最佳结果，而顺序类型（-seqTransf）模型在 "训练-9k "数据拆分中的表现优于其他方法。我们认为，在数据集较小的情况下，很难学习预训练参数之外的额外参数。如果数据集较大，它就有能力学习额外的参数。对于 LSMDC 数据集，顺序类型的模型比其他两种类型的模型更好。两种序列类型 -seqLSTM 和 -seqTransf，实现了

帧选择	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
			MSR-VTT		
负责人	42.3	**70.8**	**80.8**	2	**15.7**
尾巴	40.5	67.9	77.4	2	18.8
制服	**42.6**	70.4	80.1	2	16.3
			MSVD		
负责人	45.7	75.1	84.1	2	10.2
尾巴	45.6	**75.3**	84.3	2	10.2
制服	**46.0**	**75.3**	**84.5**	2	**10.1**
			LSMDC		
负责人	20.3	39.2	46.8	13	**63.3**
尾巴	**20.7**	38.2	46.4	14	63.6
制服	**20.7**	**39.0**	**47.1**	13	63.4

表 6：关于取样策略的研究。头部"、"尾部 "和 "均匀 "是从视频中选择帧的三种取样策略。批量大小为 128，冻结层为 0，帧长为 12，学习率为 5e-8 。

结果具有可比性。对于 MSVD 数据集，无参数类型的性能最好。我们注意到，MSVD 的训练数据比 MSR-VTT 和 MSVD 数据集至少小 2 倍，原因是额外的参数需要额外的大数据集来保持预训练权重的进步。视频段落检索在 ActivityNet 和 DiDeMo 上的表现进一步证明了利用预训练模型的无参数类型的优势。在五个数据集中，几乎所有紧缩型（-tightTransf）的结果都是所有计算器中最差的。我们认为，在没有足够数据集的情况下，紧密型仍难以学习跨模态交互。

4.4 超参数和学习策略

我们进行了大量实验，研究参数和学习策略，以寻找最佳设置。图 3 显示了实验结果图。在 3a 中，随着*批量大小*的增加，性能也随之增加，批量大小为 128 和 256 时，性能相当。在实验中，我们将批量大小设置为 128。关于 3d 中对*帧长的*研究，我们可以看到 1 到 6 帧之间的帧*长有*明显的增加，这表明视频模型实际上需要多帧序列，而不是单帧。我们在实验中采样了 12 帧，这样既高效又有效。我们还研究了是否应*冻结* CLIP 预训练的各层参数。从图 3c 中可以看出，最好以较小的学习率微调所有变换编码器层，并保留最底层的线性层。关于 3b 所示的*学习率*、

预培训	P-PT	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
			MSR-VTT			
ZS		30.6	54.4	64.3	4	41.8
ZS	C	32.0	57.0	66.9	4	34.0
FT		43.1	70.4	**80.8**	**2**	**16.2**
FT	C	**43.5**	**70.7**	80.5	**2**	16.3
			MSVD			
ZS		36.2	63.8	73.5	3	20.4
ZS	C	38.5	66.9	76.8	2	17.8
FT		46.2	**76.1**	84.6	**2**	10.0
FT	C	**46.6**	**76.1**	**84.8**	**2**	**9.9**
			LSMDC			
ZS		13.6	27.9	35.5	32	134.5
ZS	C	15.1	28.5	36.4	28	117.0
FT		20.7	38.9	47.2	13	65.3
FT	C	**21.7**	**39.5**	**49.1**	**11**	**61.2**

表 7：使用 HowTo100M-380k 数据集对（Ours）-meanP 模型的后预训练（P-PT）测试。ZS：归零，FT：微调。

2D/3D	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
		MSR-VTT			
2D	43.1	70.4	80.8	2	16.2
3D	41.6	69.9	79.5	2	17.3
		MSVD			
2D	46.2	76.1	84.6	2	10.0
3D	44.0	73.6	83.0	2	11.3
		LSMDC			
2D	20.7	38.9	47.2	13	65.3
3D	20.8	40.6	49.3	11	61.0

表 8：在（我们的）均值 P 上测试二维和三维斑块线性。

最佳学习率为 1e-7，不能太大也不能太小。过大的学习率会降低性能，更无法发挥预训练权重的优势。

4.5 视频数据集的后预训练

我们的模型建立在预训练的 CLIP 基础上，而 CLIP 是一个图像预训练模型。为了解决这种数据类型（图像与视频）的差异，我们在 Howto100M-380k 视频数据集上对模型的后预训练进行了初步探索，并报告了零拍摄和微调的结果。从表 7 中可以看出，在零拍摄和微调设置下，性能都有所提高。零点拍摄的性能提高幅度更大，这说明使用相同数据类型（视频）进行后预处理可以学习到常识并直接迁移到任务中。此外，对后预训练模型的微调也提高了在 LSMDC 和 MSVD 数据集上的性能，并在 MSR-VTT 上取得了近似结果。

数据集。在未来的工作中，我们将利用更大的数据集来探索预训练的能力。

4.6 抽样策略

我们对视频采用了三种不同的采样策略。头部 "是对视频开头的第一帧进行采样，"尾部 "是对视频结尾的最后一帧进行采样，而 "均匀 "是对视频的所有帧进行均匀采样。实验结果表明，"Uniform "是相对较好的选择，而 "Head "与之相当。而 "尾部 "采样策略则不太可能使用。

4.7 二维/三维补丁线性

我们对第 3.1 节中提到的二维和三维线性进行了比较。表 8 列出了它们的性能。与我们的预期相反，三维补丁线性可以提取帧间的时间信息并生成更好的判别特征和性能，但在 MSR-VTT 和 MSVD 上，三维线性生成的结果比二维线性差。我们认为，CLIP 是针对二维线性而非三维线性训练的，而三维线性初始化的偏差使其很难学习到时间信息。在未来的工作中，我们将在大型视频-文本数据集上进行预训练，以释放其潜力。

5 结论

在本文中，我们以预训练的 CLIP 为骨干，从帧级输入解决视频片段检索任务。我们采用无参数类型、顺序类型和紧密类型相似性计算器来获得最终结果。实验结果证明了我们模型的有效性，并在 MSR-VTT、MSVC、LSMDC、ActivityNet 和 DiDeMo 上取得了 SOTA 结果。此外，我们还从实证研究中获得了一些启示：1) 图像特征也能促进视频-文本的重新三值；2) 即使是出色的图像-文本预训练 CLIP，其后预训练也能进一步提高视频-文本检索的性能；3) 3D 补丁线耳投影和顺序类型相似性是检索任务中很有前途的方法；4) 用于视频-文本检索的 CLIP 具有学习率敏感性。

参考资料

Abien Fred Agarap.2018. 使用整流线性单元（relu）的深度学习。*ArXiv 预印本 arXiv:1803.08375*。

Elad Amrani、Rami Ben-Ari、Daniel Rotman 和 Alex Bronstein。2021.自监督多模态学习中的密度测定噪声估计。在 *AAAI*。

Anurag Arnab、Mostafa Dehghani、Georg Heigold、Chen Sun、Mario Luc˘ic´和Cordelia Schmid。2021.Vivit：*ArXiv preprint arXiv:2103.15691*.

Max Bain、Arsha Nagrani、Gül Varol 和 Andrew Zis- serman。2021.时间凝固：用于端到端检索的视频和图像联合编码器。*ArXiv 预印本 arXiv:2104.00650*.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani.2021.视频理解只需要时空注意力吗？*arXiv预印本arXiv:2102.05095*.

David Chen and William Dolan.2011.收集高度并行数据用于仿写评估。在 *ACL-HLT* 中，第 190-200 页。

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakr- ishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick.2015.Microsoft coco captions：*ArXiv preprint arXiv:1504.00325*.

Ioana Croitoru, Simion-Vlad Bogolin, Yang Liu, Samuel Albanie, Marius Leordeanu, Hailin Jin, and Andrew Zisserman.2021.Teachtext： *ArXiv preprint arXiv:2104.08271*.

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019.Bert：用于语言理解的深度双向变换器的预训练。在 *NAACL-HLT* 中，第 4171-4186 页。

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby.2021. 一幅图像胜过 16x16 个单词：大规模图像识别变换器。In *ICLR*.

Maksim Dzabraev、Maksim Kalashnikov、Stepan Komkov 和 Aleksandr Petiushko。2021.*ArXiv preprint arXiv:2103.10699*.

Christoph Feichtenhofer、Haoqi Fan、Jitendra Malik 和 Kaiming He。2019.用于视频识别的慢速网络。In *ICCV*, pages 6202-6211.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid.2020.用于视频检索的多模态变换器。*ECCV 第5 卷*。

Felix A Gers、Nicol N Schraudolph 和 Jürgen Schmidhuber。2002.用 lstm 循环网络学习精确计时。*机器学习研究期刊》*，3（8 月）：115-143。

Lisa Anne Hendricks、Oliver Wang、Eli Shechtman、Josef Sivic、Trevor Darrell 和 Bryan Russell。2017.用自然语言定位视频中的瞬间。In *ICCV*.

Sepp Hochreiter 和 Jürgen Schmidhuber。1997.Long short-term memory. *神经计算*，9（8）：1735-1780。

Dotan Kaufman、Gil Levi、Tal Hassner 和 Lior Wolf。2017.时态细分：视频分析的统一方法。*ICCV*，第 94-104 页。

Diederik P Kingma 和 Jimmy Ba.2015.亚当：一种随机优化方法。*ICLR*。

Ryan Kiros、Ruslan Salakhutdinov 和 Richard S Zemel。2014.用多模态神经语言模型统一视觉-语义嵌入。*arXiv preprint arXiv:1411.2539*.

Ranjay Krishna、Kenji Hata、Frederic Ren、Li Fei-Fei、Juan Carlos Niebles.2017a.视频中事件的密集字幕。In *ICCV*, pages 706-715.

Ranjay Krishna、Yuke Zhu、Oliver Groth、Justin John- son、Kenji Hata、Joshua Kravitz、Stephanie Chen、Yannis Kalantidis、Li-Jia Li、David A. Shamma、Michael S. Bernstein 和 Li Fei-Fei。2017b.视觉基因组：使用众包密集图像注释连接语言和视觉。*Int.J. Com- put.Vis.*，123（1）：32-73.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu.2021.少即是多：通过稀疏采样进行视频和语言学习的 Clipbert。In *CVPR*.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu.2020.HERO: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*.

Song Liu, Haoqi Fan, Shengsheng Qian, Yiru Chen, Wenkui Ding, and Zhongyuan Wang.2021.Hit: Hierarchical transformer with momentum contrast for video-text retrieval. *ArXiv preprint arXiv:2103.15049*.

Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman.2019.物尽其用：*ArXiv preprint arXiv:1907.13487*.

Ilya Loshchilov and Frank Hutter.2017.SGDR：暖重启的随机梯度下降。In *ICLR*.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou.2020.UniVL：用于多模态理解和生成的统一视频和语言预训练模型。*arXiv 预印本 arXiv:2002.06353*.

Antoine Miech、Jean-Baptiste Alayrac、Lucas Smaira、Ivan Laptev、Josef Sivic 和 Andrew Zisserman。2020.从未整理的教学视频中端到端学习视觉重现。In *CVPR*.

安托万-米奇、迪米特里-朱可夫、让-巴蒂斯特-阿拉拉克、马卡兰德-塔帕斯维、伊万-拉普捷夫和约瑟夫-西维奇。2019.Howto100m：通过观看上亿个叙述视频剪辑学习文本视频嵌入。*ICCV*.

Niluthpol Chowdhury Mithun、Juncheng Li、Florian Metze 和 Amit K Roy-Chowdhury。2018.Learn- ing joint embedding with multimodal cues for cross modal video-text retrieval.*2018 年 ACM多模态检索国际会议论文集》*，第19-27页。

Mandela Patrick、Po-Yao Huang、Yuki Asano、Florian Metze、Alexander G Hauptmann、Joao F. Henriques 和 Andrea Vedaldi。2021.视频-文本表征学习的支持集瓶颈。In *ICLR*.

Jesús Andrés Portillo-Quintero、José Carlos Ortiz-Bayliss 和 Hugo Terashima-Marín。2021.使用片段进行视频检索的直接框架》。*arXiv 预印本 arXiv:2102.12443*.

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。2021.从自然语言监督中学习可转移的视觉模型》。*ArXiv 预印本 arXiv:2103.00020*.

Alec Radford、Jeff Wu、Rewon Child、David Luan、Dario Amodei 和 Ilya Sutskever。2019.语言模型是无监督的多任务学习者。

Anna Rohrbach、Marcus Rohrbach 和 Bernt Schiele。2015.电影描述的长短故事。In *GCPR*, volume 9358, pages 209-221.Springer.

Andrew Rouditchenko, Angie Boggust, David Harwath, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, et al. Avl- net：*ArXiv preprint arXiv:2006.09199*.

Piyush Sharma、Nan Ding、Sebastian Goodman 和 Radu Soricut。2018.概念性标题：用于自动图像标题的经过清理的超文本图像alt-text数据集。In *ACL*.

Atousa Torabi、Niket Tandon 和 Leonid Sigal。2016.Learning language-visual embedding for movie un- derstanding with natural-language. *arXiv preprint arXiv:1609.08124*.

Du Tran、Lubomir Bourdev、Rob Fergus、Lorenzo Tor- resani 和 Manohar Paluri。2015.用 3d 卷积网络学习时空特征。In *ICCV*, pages 4489-4497.

Ashish Vaswani、 Noam Shazeer、 Niki Parmar、 Jakob Uszkoreit、Llion Jones、Aidan N Gomez、 Łukasz Kaiser 和 Illia Polosukhin.2017.注意力就是你所需要的一切。In *NeurIPS*, pages 5998-6008.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, and Kate Saenko.2015.使用深度递归神经网络将视频翻译成自然语言。在 *NAACL-HLT* 中，第 1494-1504 页。

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy.2018.重新思考空间特征学习：视频分类中的速度-精度权衡。In *ECCV*, pages 318-335.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui.2016.Msr-vtt：连接视频与语言的大型视频描述数据集。In *CVPR*, pages 5288-5296.

Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu.2016a.使用分层递归神经网络的视频段落字幕制作。在 *CVPR 上*，第 4584-4593 页。

Youngjae Yu, Jongseok Kim, and Gunhee Kim.2018.用于视频问题分析和检索的联合序列融合模型。In *ECCV*, pages 487-503.

Youngjae Yu、 Hyungjin Ko、 Jongwook Choi 和 Gunhee Kim。2016b.具有语义注意力的视频字幕和再三元模型。 In *EC- CVLSMDC2016 Workshop*.

Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gun- hee Kim.2017.用于视频字幕、检索和问题解答的端到端概念词检测。In *CVPR*, pages 3261-3269.

Bowen Zhang, Hexiang Hu, and Fei Sha.2018.视频和文本的跨模态和分层建模。*ECCV*, pages 385-401.

Linchao Zhu and Yi Yang.2020.Actbert：学习全局-局部视频-文本表征。In *CVPR*.

从视频到文本的检索

表 A1-A3 列出了 CLIP4Clip 在 MSR-VTT、LSMDC、MSVD、ActivityNet 和 DiDeMo 上的视频到文本检索结果。

方法	TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
零点射击							
CLIP-straight[a]	W	C	27.2	51.7	62.6	5	-
培训-7K							
HowTo100M[b]	H+M	C	16.8	41.7	55.1	8	-
培训-9K							
CE[c]	M		20.6	50.3	64.0	5.3	-
MMT[d]	H+M		27.0	57.5	69.7	3.7	-
AVLnet[e]	H+M		28.5	54.6	65.2	4	-
SSB[f]	H+M		28.5	58.6	71.6	3	-
HiT[g]	H+M		32.1	62.7	74.1	3	-
TT-CE+[h]	M		32.1	62.7	75.0	3	-
(我们的）-平均值 P	W+M	C	43.1	70.5	81.2	2	12.4
(我们的）- seqLSTM	W+M	C	42.8	71.0	80.4	2	12.3
(我们的）- seqTransf	W+M	C	42.7	70.9	80.6	2	11.6
(我们的）--严密转移	W+M	C	40.6	69.5	79.5	2	13.6
CLIP-straight[a]	W	C	59.9	85.2	90.7	1	-
TT-CE+[b]	W	C	27.1	55.5	67.1	4	-
(我们的）-平均值 P	W+M	C	56.6	79.8	84.3	1	7.6
(我们的）- seqLSTM	W+M	C	52.5	74.0	78.1	1	14.7
(我们的）- seqTransf	W+M	C	62.0	87.3	92.6	1	4.3
(我们的）--严密转移	W+M	C	54.3	85.3	91.0	1	6.0

表 A1：MSR- VTT 数据集上的视频到文本检索结果。"训练-7K "沿用了（Miech 等人，2019 年）的数据拆分，"训练-9K "沿用了（Gabeur 等人，2020 年）的数据拆分。它们的测试集相同，但训练集不同。"TrainD "列显示了用于预训练和训练的数据集，其中 M、H、W 分别表示 MSR-VTT、HowTo100M（Miech 等人，2019 年）和 WIT（Radford 等人，2021 年）。带 Cmeans 的 "E2E "一栏指的是以端到端的方式从原始视频中进行训练。基线方法有：[a] CLIP-straight (Portillo-Quintero et al., 2021)、[b] HowTo100M (Miech et al., 2019)、[c] CE (Liu et al., 2019)、[d] MMT (Gabeur et al.. 2020)、AVLnet、2020）、[e] AVLnet（Rouditchenko 等人，2020）、[f] SSB（Patrick 等人，2021）、[g] HiT（Liu 等人，2021）、[h] TT-CE+ （Croitoru 等人，2021）。

方法	TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
JSFusion[a]	L	C	12.3	28.6	38.9	20	-
CLIP-straight[b]	L	C	6.8	16.4	22.1	73	-
TT-CE+[c]	L		17.5	36.0	45.0	14.3	-
(我们的）-平均值 P	W+L	C	20.6	39.4	47.5	13	56.7
(我们的）- seqLSTM	W+L	C	20.9	40.7	49.1	11	53.9
(我们的）- seqTransf	W+L	C	20.8	39.0	48.6	12	54.2
(我们的）--严密转移	W+L	C	17.4	36.7	45.0	15	65.3

表 A2：MSVD 数据集上的视频到文本检索结果。在 "TrainD "一栏中，M 和 W 表示在 MSVD 和 WIT（Radford 等人，2021 年）上进行训练，CW 表示 CC3M（Sharma 等人，2018 年）加上 WebVid-2M（Bain 等人，2021 年）。带 Cmeans 的 "E2E "一栏指的是以端到端的方式从原始视频中进行训练。基线方法为[a] CLIP-straight（Portillo-Quintero 等人，2021 年），[b] TT-CE+ （Croitoru 等人，2021 年）。

方法	TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓

- 表 A3：LSMDC 上视频到文本的检索结果

数据集。在 "TrainD "一栏中，L 和 W 表示训练数据集。在 LSMDC 和 WIT（Radford 等人，2021 年）、MD Dzabraev et al.

多域数据集包含 MSR-VTT、LSMDC、HowTo100M 等，CW 指 CC3M（Sharma 等人，2018 年）加上 WebVid-2M（Bain 等人，2021 年）。带 Cmeans 的 "E2E "一栏指的是以端到端的方式从原始视频中进行训练。基线方法为[a] JSFusion（Yu 等人，2018 年）、[b] CLIP-straight（Portillo-Quintero 等人，2021 年）、[c] TT-CE+ （Croitoru 等人，2021 年）。

C

方法		TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
FSE[a]		A		16.7	43.1	-	7.0	-
CE[b]		A		17.7	46.6	-	6.0	24.4
HSE[a]		A		18.7	48.1	-	-	-
MMT[c]		H+A		28.9	61.1	-	4.0	17.1
SSB[d]		H+A		28.7	60.8	-	2.0	-
TT-CE+[e]		A		23.0	56.1	-	4.0	-
(我们的)	-平均值 P	W+A	C	42.5	74.1	85.8	2.0	6.6
(我们的)	- seqLSTM	W+A	C	42.6	73.4	85.6	2.0	6.7
(我们的)	- seqTransf	W+A	C	41.4	73.7	85.3	2.0	6.7
(我们的)	--严密转移	W+A	C	18.9	49.6	65.8	6.0	16.3

表 A4：Activity-iteNet 数据集上的视频到文本检索结果。在 "TrainD "一栏中，A、H 和 W 表示在 ActivityNet、HowTo100M（Miech 等人，2019 年）和 WIT（Radford 等人，2021 年）上进行的训练。带 Cmeans 的 "E2E "列表示以端到端方式从原始视频进行训练。基线方法为[a] FSE,HSE (Zhang et al., 2018)、[b] CE (Liu et al., 2019)、[c] MMT (Gabeur et al., 2020)、[d] SSB (Patrick et al., 2021)、[e] TT-CE+ (Croitoru et al., 2021)。

方法		TrainD	E2E	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
S2VT[a]		D	C	13.2	33.6	-	15.0	-
FSE[b]		D		13.1	33.9	-	12.0	-
CE[c]		D		15.6	40.9	-	8.2	42.4
TT-CE+[d]		D		21.1	47.3	61.1	6.3	-
(我们的) P	-平均值	W+D	C	42.5	70.6	80.2	2.0	11.6
(我们的) seqLSTM	-	W+D	C	42.4	69.2	79.2	2.0	11.8
(我们的) seqTransf	-	W+D	C	41.4	68.2	79.1	2.0	12.4
(我们的) 转移	--严密	W+D	C	21.5	51.1	64.8	5.0	22.4

表 A5：在 DiDeMo 数据集上进行视频到文本检索的结果。在 "TrainD "一栏中，D 和 W 表示在 DiDeMo 和 WIT（Radford 等人，2021 年）上进行的训练。E2E "列中的 C 表示以端到端的方式从原始视频中进行训练。† 表示使用地面实况建议对原始视频进行连接。基线方法有 [a] S2VT （Venugopalan 等，2015 年）、[b] FSE（Zhang 等，2018 年）、[c] CE（Liu 等，2019 年）、[d] ClipBERT（Lei 等，2021 年）、[e] Frozen（Bain 等，2021 年）、[d] TT-CE+ （Croitoru 等，2021 年）。