

清华 大学

综合 论文 训 练

题目：基于图文对预训练权重的小样本视频理解

系 别：自动化系

专 业：自动化

姓 名：王逸钦

指 导 教 师：周 杰 教 授

联合指导教师：唐彦嵩 助理教授

2023 年 6 月 15 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名: 王逸钦 导师签名: 刘洁 日 期: 2023-5-30

中文摘要

现存大多数的小样本视频分类方法在设计时并未考虑到预训练知识的迁移，在直接引入图文对预训练知识后精度非最优，且迁移效率偏低。而现存的采用监督学习的全样本视频分类方法，对每个样本独立编码，因此在小样本任务中无法联合编码支持集、查询集的全部样本。

为了将图文对预训练模型的场景建模能力有效迁移到视频领域，并在小样本视频分类任务上取得良好效果，本文提出了视频旁路聚合器（Video Bypass Aggregator, VBA）结构。通过跨层信息聚合，方法有效地迁移了预训练特征的多粒度场景建模能力。通过跨帧信息聚合，方法充分学习到针对视频的时序建模能力。通过跨视频信息聚合，方法联合利用支持集、查询集的所有样本信息，对每个样本获得更加鲁棒的特征表示。

在 6 个主流视频数据集共 7 种小样本划分下，本文方法相比同设定下的已有方法达到更高或可比精度，证明了方法的整体有效性。消融实验证明了本文各模块的作用。本文还论证了方法的训练高效性，展示了跨域理解能力，并给出了注意力图的定性可视化。

关键词：视频分类；小样本学习；图文预训练；迁移学习

ABSTRACT

Most existing few-shot video classification methods do not consider the transfer of pretrained knowledge in their design. The accuracy of these methods is sub-optimal when directly introducing pretrained knowledge, and the transfer efficiency is relatively poor. On the other hand, existing video classification methods for full-sample supervised learning always encode each sample independently, making it difficult to jointly encode all samples in the support and query sets in few-shot tasks.

To effectively transfer the scene modeling ability of pretrained models to the video domain and achieve fine accuracy in few-shot video classification tasks, this paper proposes the Video Bypass Aggregator (VBA) structure. Through cross-layer information aggregation, the method effectively transfers the multi-granularity scene modeling ability of pretrained features. Through cross-frame information aggregation, the method fully learns the temporal modeling ability for videos. Through cross-video information aggregation, the method uses all sample information in the support and query sets to obtain more robust feature representations for each sample.

The proposed method achieves higher or comparable accuracy compared to existing methods with the same settings on 6 mainstream video datasets and 7 few-shot splits, demonstrating the overall effectiveness of the method. The ablation experiments demonstrate the role of each module. The paper also demonstrates the training efficiency of the method, shows its cross-domain understanding ability, and provides qualitative visualizations of attention maps.

Keywords: video classification; few-shot learning; image-text pretraining; transfer learning

目 录

第 1 章 引言	1
1.1 研究背景	1
1.2 研究动机	2
1.3 方法贡献	3
1.4 论文结构	5
第 2 章 相关工作	7
2.1 小样本学习	7
2.2 视频理解	8
2.2.1 全样本视频分类	8
2.2.2 小样本视频分类	11
2.3 图文多模态预训练	12
2.4 迁移学习	13
2.5 本章小结	15
第 3 章 方法论述	16
3.1 学习范式	16
3.1.1 小样本学习	16
3.1.2 改进的孪生网络	17
3.2 骨干网络	19
3.3 视频旁路聚合器 VBA	24
3.3.1 跨帧的信息聚合	24
3.3.2 跨视频的特征增强	26
3.3.3 跨层的信息聚合	27
3.4 模型整体结构与本章小结	28
第 4 章 实验结果	30
4.1 数据集	30
4.2 实现细节	32

4.3 定量实验结果与分析	33
4.3.1 主实验结果及对比	33
4.3.2 各组件的有效性验证	36
4.3.3 跨域理解	37
4.3.4 训练开销	38
4.4 可视化展示	39
4.5 本章小结	39
第 5 章 结论	41
插图索引	43
表格索引	44
参考文献	45
致 谢	50
声 明	51
附录 A 外文资料的书面翻译	52
附录 B 数据集标签划分	67

主要符号表

N	类别总数
K	每类的支持集样本数量
M	查询集样本数量
\mathcal{N}	类别标签集
\mathcal{S}	支持集
\mathcal{Q}	查询集
l_i	第 i 个类别标签
l_i^q	第 i 个查询集样本的标签
$(x_{i \cdot K+j}^s, l_i)$	第 i 类的第 j 个支持集样本及其标签
(x_i^q, l_i^q)	第 i 个查询集样本及其标签
$\theta(\cdot, \cdot, \dots, \cdot)$	由多个样本到多个特征的网络映射
h_i	第 i 个支持集样本的特征
p_i	第 i 个查询集样本的特征
$L(i, j)$	第 i 个支持集样本和第 j 个查询集样本的损失
L	总体损失
$\text{sim}(\cdot, \cdot)$	两个特征的余弦相似度
$\text{score}(\cdot, \cdot)$	两个特征的相似度分数
$\text{avg_score}(l_i, x)$	样本 x 与支持集中第 i 类所有样本的平均相似度分数
$\ \cdot\ $	L2 范数
$\mathbb{I}(\cdot)$	示性函数
$\text{Softmax}(\cdot)$	Softmax 函数
$\text{Argmax}(\cdot)$	Argmax 函数
$\text{NLLLoss}(\cdot, \cdot)$	负对数相似度损失函数

第 1 章 引言

1.1 研究背景

图文对 (image-text pair) 预训练，指在互联网级的大规模图文对数据上进行训练，从而获得具有泛化性的图、文特征。典型的学习范式是 CLIP^[1]所采用的对比学习。由于图文对预训练数据规模大、多样性强，且文本数据贴近语义，使得视觉编码器抽取出的视觉特征语义区分度高，泛化性好。

小样本视频理解是小样本学习、视频理解的交叉任务，该任务考察模型在小样本场景下对视频动作或事件的理解能力。目前流行的方法在学习范式上采用度量学习 (metric learning) 方法，将视频映射为高维特征空间中的一个特征点，通过拉近同类视频的特征点、拉远非同类视频的特征点，从而不断优化映射函数。映射函数采用卷积神经网络 (CNN) 或视觉 Transformer(ViT)^[2]以保证的帧特征抽取能力，并设计时序模块融合不同帧的特征，以形成整个视频的特征。



图 1.1 5 类 1 样本任务示例^[3]

1.2 研究动机

小样本视频理解任务的难点在于数据短缺和视频建模困难：

- **推理阶段可用的带类别标签的样本数量少**

以 5 类 1 样本 (5-way, 1shot) 任务为例，模型需要利用共 5 类、每类 1 个带标签的样本（称为支持样本），来对查询样本归类。如图 1.1^[3]，每行表示一个任务，需要将右侧查询样本归类为左侧 5 类支持样本中的某类。

- **数据集总规模小，训练数据多样性不足**

以最为广泛使用的视频数据集 Kinetics^[4] 和 SSV2^[5] 为例，它们的小样本划分 mini-Kinetics 与 SSV2-small 中，每类仅包含 100 个样本^[6]。

- **视频建模困难**

视频数据需要模型同时对场景与时序进行良好建模，如图 1.2 展示了从一个选自 SSV2 数据集^[5] 的视频样本中顺序截取的 4 帧，该样本的类别标签为“拉近 xxx 与 xxx 的距离”。而 SSV2 数据集还存在“拉远 xxx 与 xxx 的距离”的标签，若对视频倒放则它的类别就会发生改变。可见视频数据对于帧的顺序是敏感的，视频理解方法除了建模视频内的场景信息，还需建模时序信息。这使得小样本任务在视频数据上尤为具有挑战性。

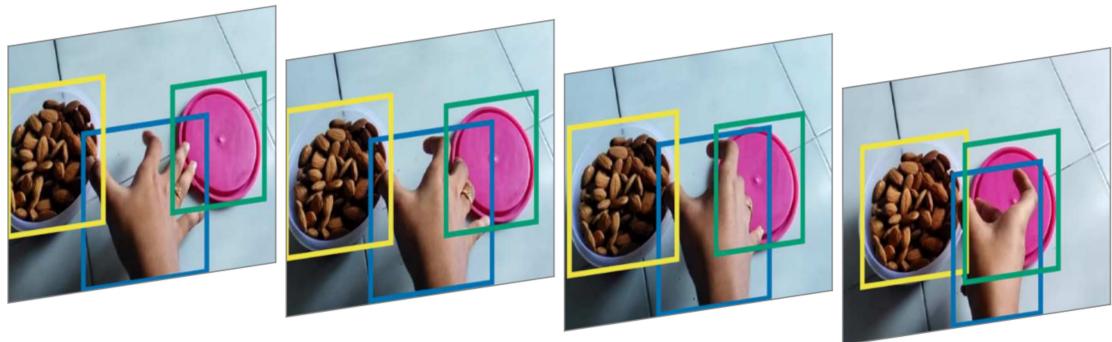


图 1.2 SSV2 数据集示例：“拉近 xxx 与 xxx 的距离”^[5]

随着图文多模态预训练^[1,7-10]的发展，以 CLIP^[1] 为代表的预训练视觉编码器已经具备强大的场景建模能力。此后，一系列工作^[11-15]尝试将图文预训练知识迁移到视频领域，在侧重于场景理解的视频数据集 Kinetics^[4] 上取得良好的效果。引入图文多模态预训练进行视频理解的最大优点在于，它预先提供了一个空间建模充分强大的基础，这可以让其它结构专注于图像预训练所不能建模的视频时序、细粒度动作等视频的专属特征。

然而，这些利用图文预训练知识的方法都是针对全样本视频分类任务设计的，

若直接应用到小样本视频分类任务则存在如下问题：

- 这些方法可训练参数量大，依赖较大规模的视频训练集。若套用到小样本任务中，不仅训练算力开销和显存开销较大，且极易引发过拟合。
- 这些方法对每个样本完全独立编码。若套用到小样本任务中，无法利用小样本支持集 (support set) 与查询集 (query set) 的全部样本信息。
- 这些方法得益于图文预训练，场景理解能力较强，但视频时序理解能力仍然不足。

此外，已有的小样本视频理解方法^[6,16-20]在设计时并未考虑从预训练知识中进行迁移学习的问题。实验部分将证明，这类不是为预训练迁移而设计的方法，若直接用图文预训练权重替换它们的骨干网络所达到的效果并非最优。

可见，如何设计一个网络结构，使之能在规模有限的小样本视频数据集上，高效地将图文多模态预训练知识迁移到视频数据上，是一个未被充分探索的问题。

1.3 方法贡献

为了让图文多模态预训练权重高效地迁移到小样本视频理解任务中，本文提出了视频旁路聚合器 (Video Bypass Aggregator, VBA) 整体架构。

本文方法在学习范式上采用改进的孪生网络，相比于常规的孪生网络增加了对于样本间的联合编码能力。联合编码的含义是让编码器同时接收所有支持集样本和所有查询集样本，对每个样本编码时考虑不同样本间的信息交互，用自注意力机制对样本进行“加权平均”，其权重源于独立编码样本特征的相似度。

以图 1.1的第一行为例讲解原理，右侧查询集中的“蘑菇”在语义上和左侧支持集中间的“蘑菇”相近，而与另外四个支持集样本远离，因此两个“蘑菇”样本的特征相互以较大权重彼此融合，而与另外四个样本支持集样本几乎不融合。由于两个蘑菇的背景场景不同，因此这种融合增加了“蘑菇”图片的语义丰富性，使最终每个“蘑菇”的特征都更鲁棒。

为便于直观理解本文方法，给出示意图如 1.3。图中以 5 类 1 样本任务为例，给出了本文所提出的视频旁路聚合器的整体结构。

本文的骨干网络选用 ViT-B/32^[2]，使用 CLIP^[1]权重进行初始化并冻结全部权重，向末 4 层插入可学习的时序融合模块。图 1.3 中灰色框表示骨干网络，其中蓝色的“帧编码器块”是冻结权重的 Transformer 编码器块，橙色的“跨帧时序融合”是可训练的时序融合模块。从图 3.13 中骨干网络的箭头变化可以看出，“帧编码器

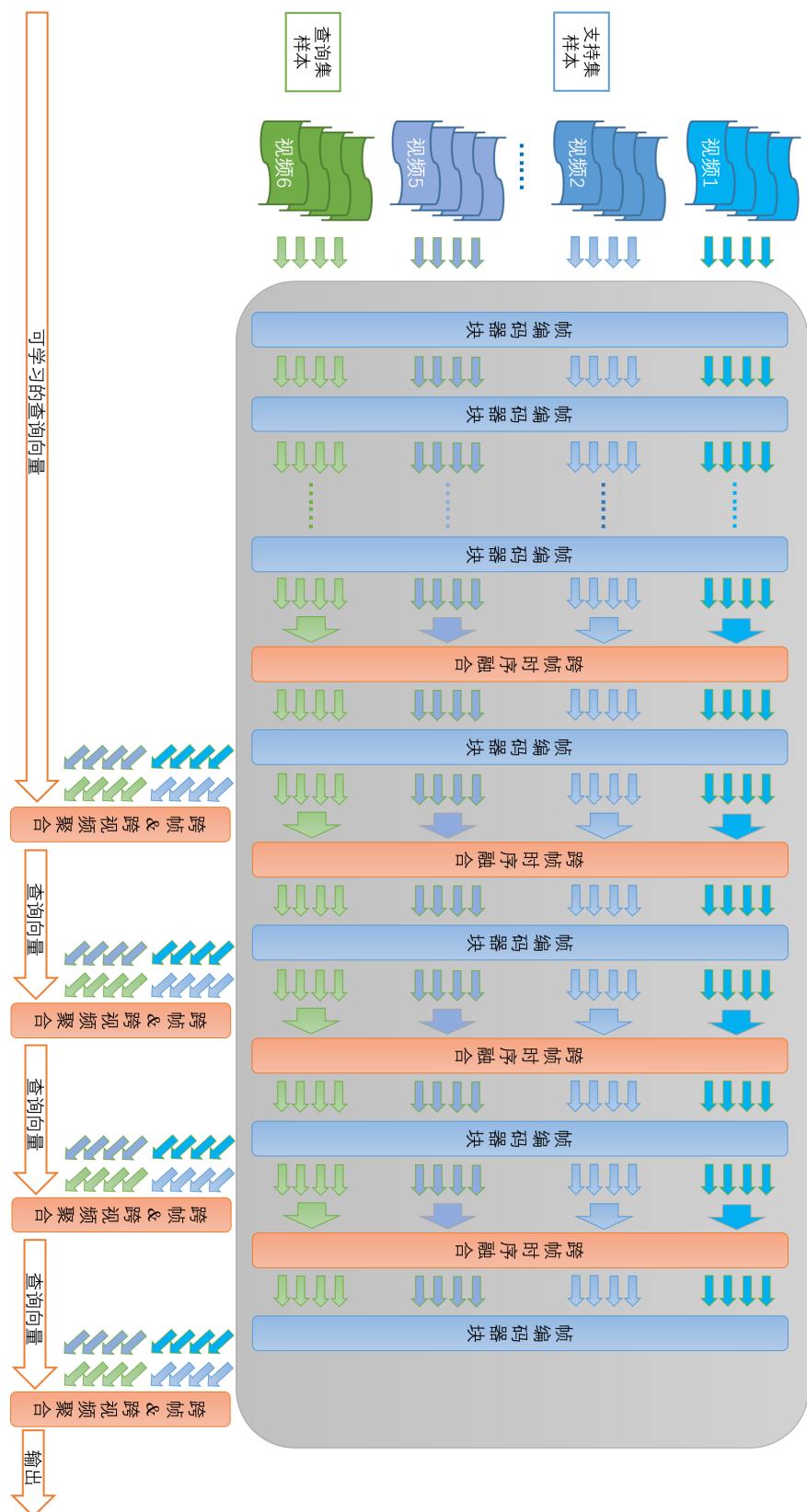


图 1.3 整体结构示意图

块”对每个视频的每帧进行独立编码，而“跨帧时序融合”则是对每个视频的所有帧进行联合编码。由此，骨干网络具备了初步的时序建模能力。

收集末4层编码器的帧特征输出，对每层的所有帧特征依次进行跨帧、跨视频的全面信息聚合。聚合过程通过交叉注意力机制与自注意力机制实现，将在第三章详细阐释。使用一个可学习的查询向量，将骨干网络末4层的输出进行逐层解码，并将查询向量层层传递，最终解码得到经过多帧、多视频、多层次全面聚合的支持集、查询集视频特征。

从上述过程不难发现，本文方法涉及到不同样本的特征融合与联合编码。这就是本节开始时所阐述的，本文方法对孪生网络的改进之处。

完成联合编码后，特征之间即可基于余弦相似度，算得查询集样本与所有支持集样本两两之间的相似度分数。训练阶段根据预测的相似度分数和二者真实标签是否一致，来计算得到损失，从而对模型的可学参数进行优化。推理阶段根据预测的相似度分数排序，直接给出预测的类别标签。

通过在旁路进行信息聚合，本文提出的VBA在不改变预训练骨干网络结构的条件下，实现了特征的跨帧聚合、跨视频聚合以及跨层聚合。第四章实验部分将用6个主流数据集共7个小样本划分，来定量证明：VBA在同样的任务设定下，在精度方面超过或基本追平所有现有方法，在训练效率方面大幅提升。

总的来说，本文提出的视频旁路聚合器（Video Bypass Aggregator, VBA）方法，能够让图文对预训练知识迁移到视频领域，并在小样本视频分类任务上取得良好效果。通过跨层信息聚合，方法有效地迁移了预训练特征的多粒度场景建模能力；通过跨帧信息聚合，方法充分学习到针对视频的时序建模能力；通过跨视频信息聚合，方法联合利用支持集、查询集的所有样本信息，对每个样本获得更加鲁棒的特征表示。

1.4 论文结构

第2章从小样本学习、视频理解、图文多模态预训练、迁移学习四个角度进行了文献综述。第3章首先明确小样本学习的任务设定；然后阐述本文采用的改进型孪生网络的学习范式；进而介绍本文所采用的骨干网络与预训练权重；最后从跨帧、跨视频、跨层三方面阐述本文的核心方法：视频旁路聚合器VBA。第4章介绍数据集与实现细节，然后展示了本文方法在六个小样本视频分类数据集共七种设定下的精度表现，与现有主流方法在同等设定下进行公平比较；进而通过消

融实验证明各组件的有效性；此外实验证明了方法的跨数据域的理解能力以及方法的运算高效性；最后展示可视化结果。

第 2 章 相关工作

2.1 小样本学习

随着深度学习的发展，特别是各类卷积神经网络 (Convolutional Neural Networks, CNNs)^[21-24]的出现，让图像分类等基础任务的准确率不断提升。然而标准的监督学习仍然依赖大量的带标签训练数据进行学习，才能有效泛化到新样本上；一旦训练样本过少，就容易导致标准的监督学习出现严重过拟合。此外，标准的监督学习只能给出那些在训练时见过的类别标签，无法对新类别进行分类。

与之形成对比的是，人类不仅可以通过少量的指导来高效学习，且可以举一反三地辨别新类。例如一个学龄前儿童只需要在图画书上看几张猫的图片，就可以在真实世界中分辨出哪些动物是猫。进一步给儿童几类他从未见过的动物图片各一张，如 1 号企鹅图片、2 号大象图片、3 号图片长颈鹿，当儿童在动物园里见到长颈鹿时，也可判断面前的动物和 3 号图片属于同类。

为了尝试让模型也能够在少量样本达成上述能力，学者提出了小样本学习任务，其中最为典型的任务是 N 类 K 样本 (N-way, K-shot) 分类任务。该任务要求模型根据支持集共 $N \times K$ 个样本及标签，判断出查询集样本的标签。

深度学习的发展为解决小样本问题带来了全新的思路。首先出现的是基于度量学习 (metric learning) 的方法。2015 年，Koch 等人提出了 Siamese Network^[25]，将两个网络共享权重，通过计算它们之间的相似度来实现提取特征和分类的任务。Siamese Network 在小样本学习领域中获得了良好的效果，是度量学习方法在小样本任务中应用的开山之作。Siamese Network 的思路对本文方法产生了重要启发。

此后，基于度量学习的方法又被进一步优化。2016 年，Vinyals 等人提出了 MatchingNet^[26]，如图 2.1 所示。该方法基于 Siamese Network，利用了一个附加的神经网络来进行分类。MatchingNet 中已经出现了“原型”的思想 (prototype)，将每个类别的样本表示为一个原型，并将测试样本分配给与其最接近的原型，在小样本分类任务中的效果进一步提升。2017 年，Snell 等人提出了 ProtoNet^[27]，该方法也是基于原型的方法，但是使用了更简单的原型表示方法。ProtoNet 将每个类别的样本表示为该类别样本的平均值，并通过计算测试样本与每个类别的平均值之间的距离来进行分类，计算复杂度得到进一步降低。

在 2017 年之后，出现了一些不基于度量学习的新方法。2017 年，Finn 等人提

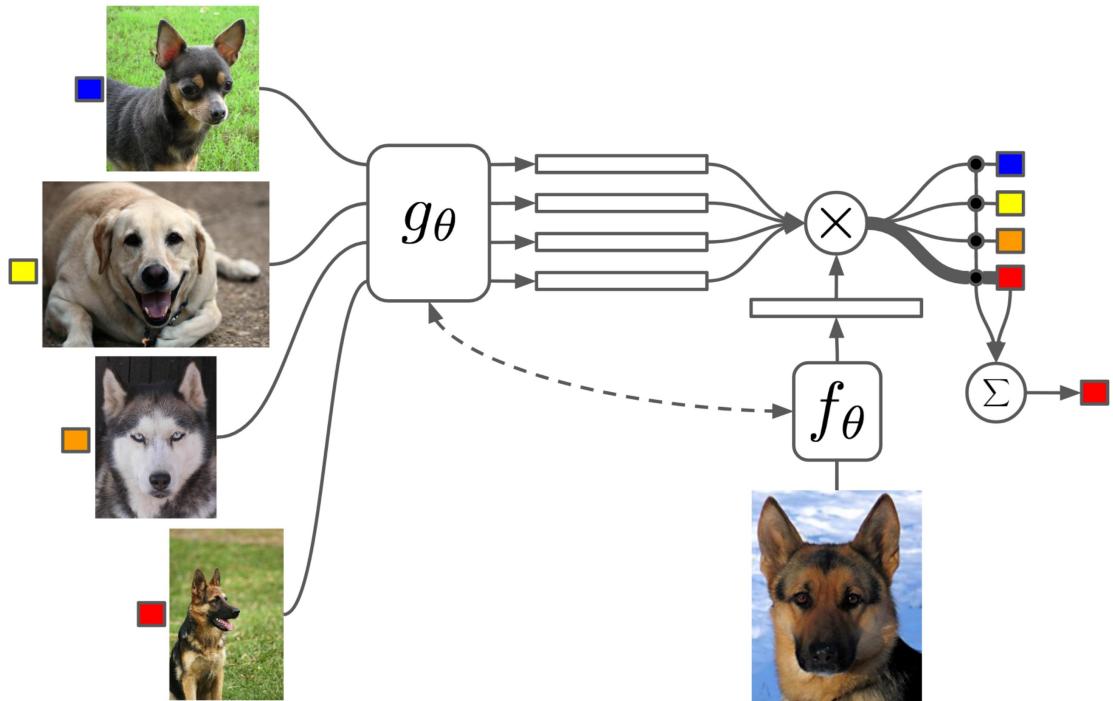


图 2.1 MatchingNet^[26]，一种典型的度量学习方法

出了模型无关的元学习（Model-Agnostic Meta-Learning, MAML）^[28]方法，在小样本学习领域中具有广泛的应用。MAML 通过学习一个通用的初始化参数，使得模型能够在少量的样本上快速适应新任务。MAML 将小样本学习与元学习方法相结合，为小样本学习领域带来了新的思路和方法。此外，近年来图神经网络（Graph Neural Networks, GNN）在小样本学习中的应用也受到了广泛关注。图神经网络可以处理图数据，这种数据形式可以用来描述物体之间的关系，在小样本学习中可以用来捕捉类别之间的关系从而提高分类精度。2018 年，Satorras 等人提出了基于图神经网络的小样本学习方法^[29]，该方法利用图神经网络来学习类别之间的关系，并通过这些关系来进行分类。该方法在小样本学习领域中取得了良好的效果。

2.2 视频理解

2.2.1 全样本视频分类

全样本视频分类是最常见的视频理解任务，是图像分类任务的视频版本。常用数据集包括偏场景理解的 UCF101^[30]、Kinetics^[4]等，以及偏时序理解的 Something-Something^[5]等。

视频分类任务一直伴随着计算机视觉的发展。自 2015 年起至 2019 年，基于

3D 卷积的方法是视频理解领域的主流方法。2015 年, Tran 等人提出了 C3D^[31], 将 3D 卷积运算应用于视频中, 能够捕捉到时间和空间信息, 对于视频分类和动作识别任务有较好的表现。2016 年, Wang 等人提出了 TSN^[32], 该网络利用了时空注意力机制来捕捉视频中的重要帧和重要时间段, 进一步提高了视频分类和动作识别的性能。2017 年, Carreira 等人提出了 I3D^[4], 该网络结合了 2D 和 3D 卷积, 使用“膨胀卷积”, 可以在 ImageNet 预训练模型的基础上进行微调, 能够在大规模视频数据上进行端到端的训练和分类。2018 年, Tran 等人提出了 R(2+1)D^[33], 该网络将 3D 卷积分解为一个 1x3x3 卷积和一个 3x1x1 卷积, 可减少参数量和计算量, 同时保持时间和空间信息的表达。同年, Wang 等人提出了 Non-local^[34], 该网络利用了非局部操作, 能够在视频中进行全局的交互和关联, 从而提高视频理解的性能。2019 年, Feichtenhofer 等人提出了 SlowFast^[35] 网络, 该网络使用了两个分支, 一个慢速分支用于捕捉空间信息, 一个快速分支用于捕捉时间信息, 能够在不牺牲精度的情况下减少计算量。

随着 2020 年 Vision Transformer (ViT)^[2] 的提出, 视频领域也迅速跟进使用 Transformer^[36] 结构。2021 年, Bertasius 等人提出了 TimeSformer^[37], 该网络基于 Transformer 结构, 可以在视频中分别建模全局的时、空关系, 从而提高视频理解的性能。同年, Zhang 等人提出了 VidTr^[38], 该网络使用了可分离的注意力机制, 能够在视频中捕捉时空信息, 且在保持同等性能下降低了 3.3 倍显存占用, 提高了视频理解的高效性。同年, Arnab 等人提出了 ViViT^[39], 该网络将 ViT 应用于视频中并将时空维度进行分解, 该结构可以利用预训练的图像模型, 从而能够在相对较小的数据集上进行训练。同年, Fan 等人提出了 MViT^[40], 该网络基于 ViT 并构建了多尺度特征金字塔, 网络浅层在高空间分辨率下操作, 用于建模简单的低级视觉信息; 网络深层在空间上更粗粒度, 但在更复杂的高维特征上运算。MViT 通过跨模态注意力机制来融合多个模态信息, 提高了视频理解的性能。同年, Liu 等人提出了 VideoSwin^[41], 该网络基于 Swin Transformer^[42] 结构, 通过横向连接和分层注意力机制来捕捉视频中的信息, 实现了对视频中对象的分类和定位。

随着图文多模态预训练^[1,7-10] 的发展, 以 CLIP^[1] 为代表的预训练视觉编码器已经具备强大的场景建模能力, 可以编码图片得到语义区分度较高的图片特征。此后, 一系列工作^[11-15] 尝试将图文预训练知识迁移到视频领域, 在侧重于场景理解的视频数据集 Kinetics^[4] 上取得良好的效果。引入图文多模态预训练进行视频理解的最大优点在于, 它预先提供了一个空间建模充分强大的基础, 这可以让其它结构专注于图像预训练所不能建模的视频时序、细粒度动作等视频的专属特征。

2021 年, Wang 等人提出了 ActionCLIP^[12], 它是首个将 CLIP 预训练模型用于视频分类的工作, 其使用方法较为直接, 对视频抽帧并通过 CLIP 初始化的 ViT 后对各帧结果进行简单融合, 即得到最终结果。ActionCLIP 直接开放 ViT 的全部参数, 在 Kinetics-400 数据集上进行微调, 利用如此简单的方法就达到了良好的精度。2022 年, Luo 等人提出了 CLIP4Clip^[11], 以端到端的方式将 CLIP 模型的知识转移到视频-语言检索中, 探索图像特征是否足以进行视频-文本检索, 并探索时序建模的机制, 最终用实验表明从 CLIP 迁移的 CLIP4Clip 模型可以在视频-文本检索数据集 (如 MSR-VTT、MSVC 等) 上实现最优结果。同年, Ni 等人提出了 X-CLIP^[13], 它将 CLIP 特征迁移到视频领域并引入了多粒度对比。为解决先前工作仅从单一粒度进行对比, 无法全面感知特征的问题, X-CLIP 提出了注意力超相似性矩阵 (AOSM) 模块, 使模型专注于关键帧和文本之间的对比, 降低非必要帧和冗余单词对检索结果的影响, 在多个广泛使用的视频-文本检索数据集上取得了出色的性能。

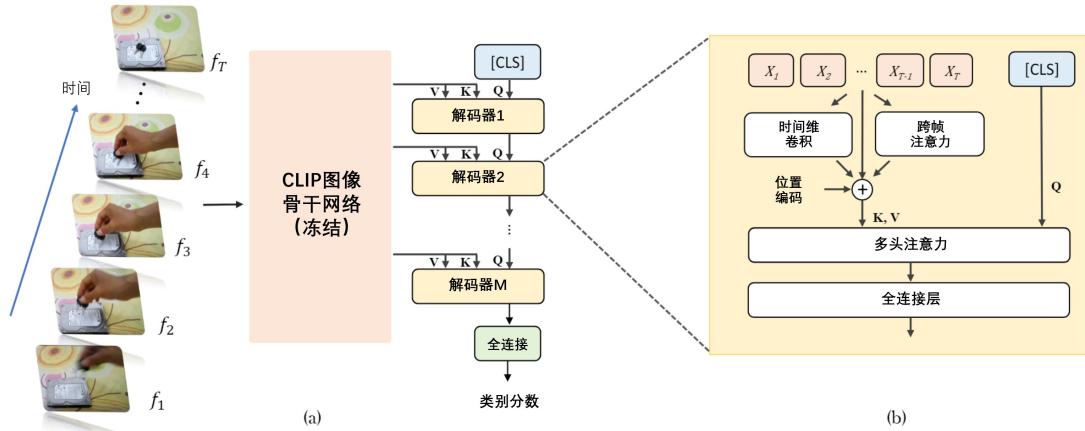


图 2.2 EVL^[14], 一种典型的高效视频理解方法

将 CLIP 知识迁移到视频领域时, 除了上述直接采用微调权重或部分微调权重的迁移方式以外, 一些工作还探索了其它迁移方案。2022 年, Lin 等人提出了 EVL^[14], 它在引入 CLIP 特征时不采用端到端的方式, 而是直接使用冻结的 CLIP 特征。为了让 CLIP 特征有效且高效地迁移到视频, EVL 采用轻量级 Transformer 解码器, 学习一个查询向量, 从 CLIP 图像编码器动态收集帧级空间特征, 并加入本地时间编码, 如图 2.2。该方法在保证训练高效性的同时, 也在视频数据上学习到了高质量的特征表示。EVL 所的思路对本文方法产生了重要的启发。

同年, Ju 等人提出了 Efficient-Prompt^[15], 它引入 CLIP 特征时不采用传统的微

调方法，而是在视觉侧引入了可学习的连续提示向量（continuous prompt vectors），由此让视频相关的特征可以转换到与预训练同分布的格式中。这种方式改变了传统的“让模型适应数据”的范式，转而变为“让数据适应模型”。Efficient-Pormpt 在全量监督学习、少样本学习、零样本推理的情况下分别汇报了性能，证明方法的普适性。

2.2.2 小样本视频分类

小样本视频分类是小样本任务的一种，它基于图像的小样本学习任务发展而来。和图像任务类似，小样本视频任务也主要分为 5 类 1 样本任务和 5 类 5 样本任务。主流的小样本视频分类方法为保证公平比较，普遍采用统一的 ImageNet-1K^[43] 预训练的 ResNet50^[23] 作为骨干网络，主要差异在于设计特征融合结构来帮助对于视频数据的建模。

2018 年，Zhu 等人提出了针对小样本视频理解的 CMN^[16]，其中复合记忆结构的每个“记忆”包含多个组成键值，这些组成键可以协同工作，关注视频中的不同部分。在组成键值之上可以形成“抽象记忆”，它位于更高层的位置，使 CMN 形成了可扩展的多层结构。2020 年，Cao 等人提出了 OTAM^[6]，它通过对两个视频进行有序的时间对齐，来显式地利用视频数据中的时序信息，大大提高了小样本学习的数据效率。2021 年，Zhang 等人提出了 ITANet^[17]，针对 OTAM 进行进一步优化，将 OTAM 中的显式时间对齐改为了隐式时间对齐，可以更加普适地对齐不同视频之间的特征。此外，ITANet 还采用了上下文编码模块以融合空间和特征通道上下文信息，更好地建模类内变化。

2021 年以来，注意力机制的运用越发普遍。2021 年，Perrett 等人提出了 TRX^[18]，使用交叉注意机制来构建整个类别的原型，更好地利用了类内样本的所有信息，而不是使用类平均值或单个最佳匹配。TRX 将视频表示为不同数量帧的有序元组，从而允许比较不同速度和时间偏移的动作子序列，在主流小样本视频数据集上达到领先精度。2022 年，Wu 等人提出了 MTFAN^[44]，其中的运动调制器可以对每个任务进行特定的动作特征嵌入，而不是进行普适嵌入。此外，MTFAN 还提出了多层次的片段注意力机制，对帧间、片段间、帧与片段之间都进行对齐。同年，Luo 等人提出了 LSTC^[45]，改变了传统的对视频均匀采样的方式，采用长短时交叉注意力。对于“短”的部分，运动的变化（运动矢量）中包含的线索可以反映帧的重要性；对于“长”的部分，一个视频可以被切分为一系列帧组。最终 LSTC 的“长短时选择模块”可以选择信息量丰富的数据，长/短期模块则用于挖掘时空信息。

近期，相关工作在精度方面进一步提升。2022 年，Thatipelli 等人提出了 STRM^[20]，它在图像块（Patch）层面对具体动作特征进行帧内增强，进而在跨帧层面对时间上下文进行特征增强，方法简单而有效。同年，Wang 等人提出了 HyRSM^[19]，它利用支持集、查询集内所有样本进行跨视频融合来帮助每个视频的特征编码，又提出了一种基于双向平均 Hausdorff 距离的度量方法用于度量两个视频的相似度，用集合匹配的方式替代了简单的顺序对齐。HyRSM 在多个主流小样本视频理解数据集上达到最优性能，是目前的 State-of-the-Art 方法。

2.3 图文多模态预训练

2021 年，OpenAI 研究者 Radford 等人提出 CLIP^[1]模型，它在 4 亿个私有的图文对数据上进行大规模对比学习，在推理时根据图-文的特征相似度给出预测标签。实验表明 CLIP 的预训练权重在各类下游任务上有强大的泛化能力，说明在预训练阶段模型学到了对各类 2D 场景视觉任务的通用、普适知识。

在此之后，一系列图文多模态预训练模型陆续被研究者提出。来自 Salesforce Research 的研究者 Li 等人在 2021 年提出了 ALBEF^[7]，该工作的思想正如其标题“Align Before Fuse”的含义，它先对图文特征使用对比学习进行对齐，而后共同输入特征融合模块进行信息交互。ALBEF 使用了图文特征对比损失（Image Text Contrastive Loss, ITC Loss）、二分类的图文匹配损失（Image Text Matching Loss, ITM Loss），以及语言掩码损失（Masked Language Modeling Loss, MLM Loss）进行联合训练。在此工作之后，ITC、ITM、MLM 这三个损失函数成为多模态预训练的主流。此外，ALBEF 为提高数据质量，沿用 MoCo^[46]提出的动量模型来生成伪标签，通过类似于隐式 E-M 算法^[47]的方式提高了数据质量，实现模型自我增强。Li 等在 2022 年又进一步提出 BLIP^[8]，它维护一个图像标签器（Captioner）和一个标签过滤器（Filter）首先对训练数据进行增强和清洗，一定程度上解决了大规模网络图文对数据质量低下的问题。相比于 ALBEF，BLIP 还加入了解码器结构，更加便于进行各类文本生成任务。

与此同时，其它研究机构的学者也陆续提出新的多模态与预训练方法。2022 年，DeepMind 研究者 Alayrac 等人提出了 Flamingo^[10]，Flamingo 是具有 800 亿参数规模的大模型，它桥接了强大的预训练视觉模型和语言模型，能够处理任意交错的视觉和文本数据序列，且可以无缝将图像或视频作为输入。Flamingo 具有惊人的泛化性、灵活性，在广泛的评估中发现它在视觉问答、视觉标注等多模态任

务中表现出强大的性能，且可以根据特定任务快速迁移，在小样本学习中达到领先水平。

同年，阿里达摩院的研究者 Wang 等人提出了 OFA^[9]，实现了架构统一、模态统一和任务统一，其最大特点是将任务统一表达为 Seq2Seq 形式，从而可以让一个模型获得多种能力，包括文本生成、图像生成、跨模态理解等。目前，约 10 亿参数的 OFA-huge 模型在训练数据少一到两个数量级的情况下，在图文描述、物体指代理解等任务中超越了 Flamingo，同时具备高质量的图像生成能力。

2.4 迁移学习

迁移学习的目标是将预训练特征迁移到下游数据分布上，不需要微调全部特征，仅有选取地优化部分特征即可完成迁移。这类方法最初由自然语言处理研究者提出，主要基于 Transformer 结构设计。近年来，研究者将这类方法普遍称为“参数高效的微调”（Parameter-Efficient Fine-Tuning, PEFT）。

2019 年，Houlsby 等人提出了 Adaptor^[48]方法，它创新地在预训练模型中添加一些小的可训练模块，这些模块只针对特定的任务进行微调，并且在不影响其他任务的情况下进行扩展。这种方式规避了对整个模型权重进行微调，从而让模型在多个任务之间共享预训练模型的参数，仅需要选择不同的 Adaptor 即可，显著减少了训练参数的数量。Adaptor 具有简单、高效、易于扩展等优点，已被广泛应用于自然语言处理领域，取得了良好的效果，是 PEFT 方面的开山之作。

在此之后，一系列类似的 PEFT 工作被陆续提出。2021 年，Li 等人提出了 Prefix-tuning^[49]，其思想是保持语言模型参数固定，但是对每个 Transformer 块新增一个可优化的连续任务特定向量（称为前缀）。它与提示微调（prompt-tuning）不同的是，提示微调仅对输入数据产生作用，而 prefix-tuning 对模型内部每层均产生作用。Prefix-tuning 也在少量数据情况下获得比整体微调更好的性能，且只需学习 0.1% 的参数。同年，Lester 等人提出了 Prompt-tuning^[50]。有别于直接对输入数据进行模板化调整的“硬提示微调”（Hard Prompt Tuning），本文提出了以可学习参数为载体的“软提示”，它通过反向传播学习。作者在实验中发现 Prompt-tuning 在规模扩大后变得更具竞争力：随着模型超过数十亿个参数，该方法达到了与全面微调同样的性能。该方法只对输入数据进行优化，可以视为 Prefix-tuning 的简化版。同年，Hu 等人又提出了 LoRA^[51]即低秩适应方法。它和前述工作类似，也冻结预训练模型权重，但它的微调位置和 Prefix、Adaptor 均不相同。具体来说，LoRA 是通

过将可训练的秩分解矩阵注入到 Transformer 架构的每一层中，减少了下游任务的可训练参数数量。

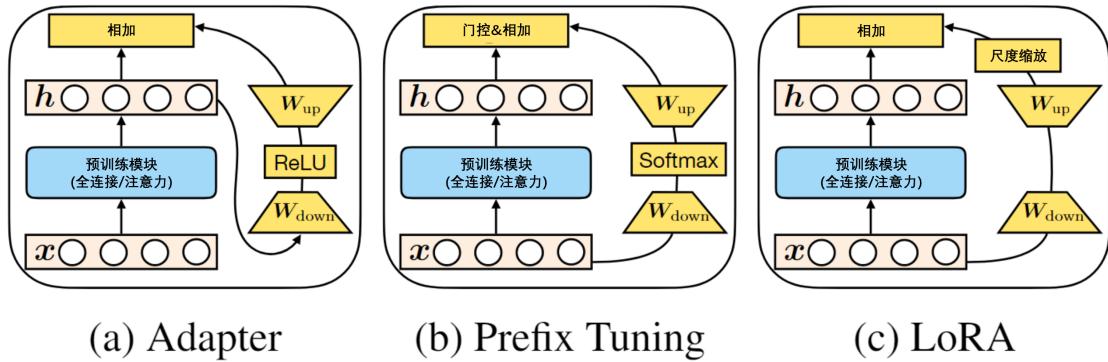


图 2.3 用统一的框架理解各类 PEFT 方法^[52]

2022 年，Lian 等人^[52]总结了前述各类 PEFT 方法^[48-49,51]，将它们纳入到了统一的数学框架中，如图 2.3 所示。作者发现虽然他们结构各异，但本质十分相近。根据这一统一的数学框架，Lian 等人提出了一种新的 MAM Adaptor^[52]，发现其效果更优。

迁移学习的方法在计算机视觉领域也在被不断探索。2022 年，Lian 等人提出了 SSF^[53]，它仅对每层增加尺度缩放（Scale）、特征偏移（Shift）两个可学的线性模块，而冻住所有预训练参数。实验表明 SSF 在视觉任务上的表现竟优于 Adaptor 或全面微调。SSF 的另一大优势在于：神经网络的各个已有组件已经具备缩放和偏移的线性操作能力，在训练完成之后可以通过权重的重参数化，将训练完毕的线性模块融合到已有网络模型之中。这样推理阶段的开销和预训练模型相比没有任何增加。

前述工作探索了各类参数高效的微调方法，然而它们都在网络中间层插入了可学参数。这种方式与全面微调相比虽然能够大幅减少可学的参数量，但梯度在反向传播阶段还是需要流过整个预训练的骨干网络，才能达到待优化的模块处，导致这类方法的显存占用仍然较高。2022 年，Sung 等人提出了 LST^[54]，该方法在骨干网络外部搭建“过墙梯”，仅优化旁路的可学参数，避免了梯度流过骨干网络，如图 2.4 所示。这种方式也被基于 CLIP^[1]的视频理解方法 EVL^[14]所采纳，它同时兼顾了参数高效和显存高效，在真正意义上让预训练权重的迁移过程变得更加便利，对本文的方法有重要启发。

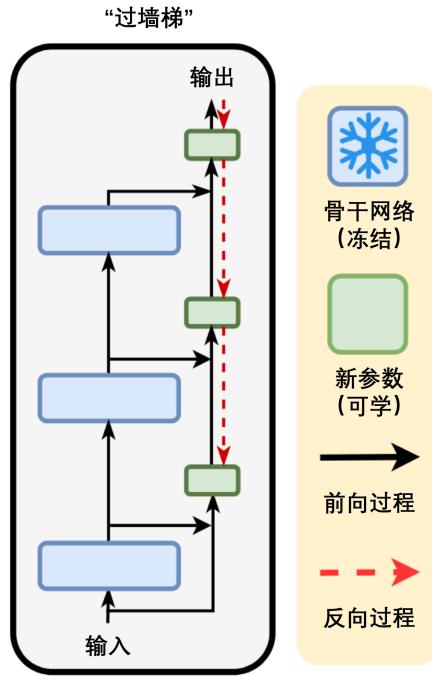


图 2.4 “过墙梯”^[54], 一种典型的迁移学习方法

2.5 本章小结

本章分几方面介绍了本文的主要相关工作，包括小样本学习方面、视频理解方面、图文预训练方面以及迁移学习方面。其中视频理解方面又可以细分为基于 3D 卷积的视频理解、基于 Transformer 的视频理解、基于多模态预训练的视频理解，以及小样本视频理解。

第3章 方法论述

3.1 学习范式

3.1.1 小样本学习

N类K样本(N-way, K-shot)分类任务是小样本学习的典型任务。该任务要求模型根据支持集的样本及标签，判断出查询集样本的标签。其数学定义如下：

现有 N 个类别标签构成集合 $\mathcal{N} = \{l_1, l_2, \dots, l_N\}$ ，每类包含 K 个标签已知的支持样本，共 $N \times K$ 个，构成了支持集

$$\mathcal{S} = \{(x_1^s, l_1), \dots, (x_K^s, l_1), (x_{K+1}^s, l_2), \dots, (x_{N \times K}^s, l_N)\} \quad (3.1)$$

另有 M 个待分类的查询样本构成了查询集

$$\mathcal{Q} = \{(x_1^q, l_1^q), \dots, (x_M^q, l_M^q)\}, l_i^q \in \mathcal{N}, i \in [1, M] \quad (3.2)$$

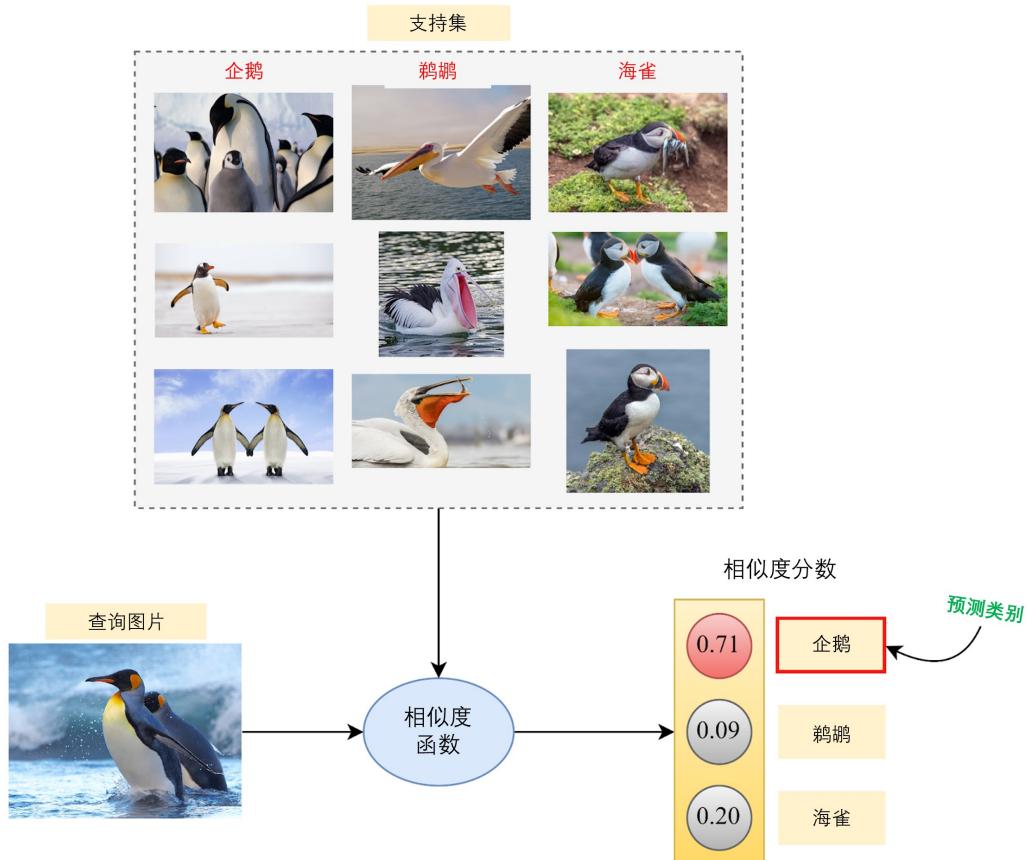


图 3.1 3类3样本分类任务示意图^[55]

任务的目标是根据标签集合 \mathcal{N} 与支持集 \mathcal{S} 来对查询集 \mathcal{Q} 中的样本进行分类，也即确定 $l_1^q, l_2^q, \dots, l_M^q$ 的值。以图 3.1 的 3 类 3 样本任务为例，支持集内有企鹅、鹈鹕、海雀三种动物图片各 3 张，模型利用支持集图片信息，判断出了查询图片的类别属于企鹅。

本文研究视频小样本学习任务，以单个视频作为一个样本。值得说明的是，小样本学习任务本质上是一个多分类任务，任务形式适用于任何可分类的样本，并不局限于图片、视频等。

3.1.2 改进的孪生网络

为解决小样本学习任务，本文沿用孪生网络^[25](Siamese Neural Networks)并加以改进。孪生网络基于度量学习，是解决小样本任务的常见方法。对于前述 N 类 K 样本小样本学习问题，孪生网络使用一个参数完全共享的网络分别将支持集、查询集映射到特征空间。

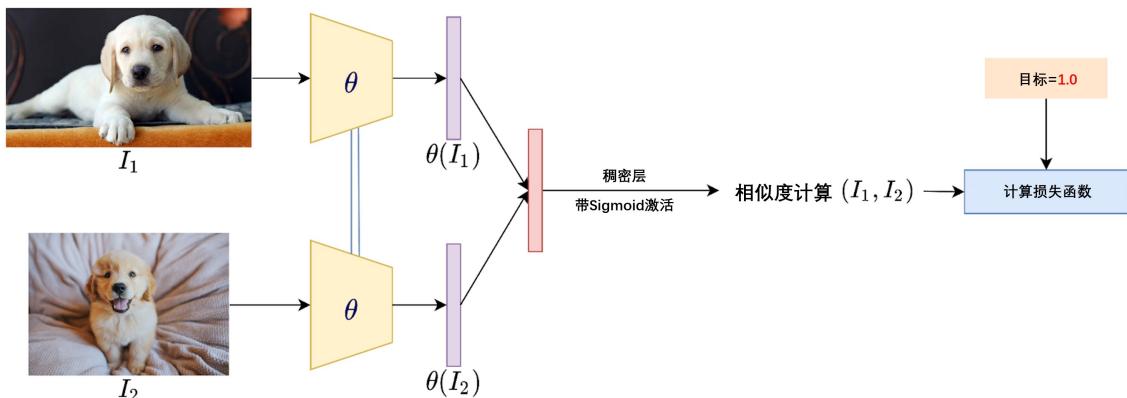


图 3.2 孪生网络示意图^[55]

如图 3.2 所示，在标准的孪生网络中，令 θ 作为共享的网络映射，则有：

$$h_i = \theta(x_i^s), \quad i = 1, 2, \dots, N \times K \quad (3.3)$$

其中 h_i 表示第 i 个支持集样本的特征。

类似地，对于支持集：

$$p_i = \theta(x_i^q), \quad i = 1, 2, \dots, M \quad (3.4)$$

其中 p_i 表示第 i 个查询集样本的特征。

标准孪生网络的核心思想在于共享同一个映射 θ 来对支持集和查询集样本进行编码。然而，标准孪生网络对每个样本独立编码，无法进行跨样本的信息融合。在保持孪生网络“共享映射”的思想的条件下，为充分融合所有支持集、查询集样

本的信息，本文将特征映射 θ 的定义扩展为：

$$h_1, h_2, \dots, h_{N \times K}, p_1, p_2, \dots, p_M = \theta(x_1^s, x_2^s, \dots, x_{N \times K}^s, x_1^q, x_2^q, \dots, x_M^q) \quad (3.5)$$

为计算支持集和查询集样本的相似度，可以使用各类相似度度量，本文采用余弦相似度 (cosine similarity)。令 $\text{sim}(h_i, p_j)$ 表示第 i 个支持集样本和第 j 个查询集样本的特征相似度，则对于余弦相似度有：

$$\text{sim}(h_i, p_j) = \frac{h_i \cdot p_j}{\|h_i\| \cdot \|p_j\|} \quad (3.6)$$

其中 \cdot 表示点积， $\|\cdot\|$ 表示欧几里得范数 (L-2 norm)。

计算支持集所有样本与该查询样本的余弦相似度，并与温度超参数 Temp 相乘再通过 Softmax，得到总和为 1 的相似度分数：

$$\begin{aligned} & \text{score}(h_1, p_j), \text{score}(h_2, p_j), \dots, \text{score}(h_{N \times K}, p_j) \\ &= \text{Softmax} [\text{Temp} \cdot (\text{sim}(h_1, p_j), \text{sim}(h_2, p_j), \dots, \text{sim}(h_{N \times K}, p_j))] \end{aligned} \quad (3.7)$$

为训练孪生网络，根据支持集样本与查询集样本的相似度分数 $\text{score}(h_i, p_j)$ ，使用负对数相似度损失 (Negative Log Likelihood Loss, NLLLoss)。这种损失让相似度分数的预测值向目标真值靠拢，从而达到训练目标。

令 $\text{sim}(h_i, p_j)$ 表示第 i 个支持集样本和第 j 个查询集样本的特征相似度，则可以定义损失函数如下：

$$L(i, j) = \text{NLLLoss} \left[\text{predict} = \text{score}(h_i, p_j), \text{target} = \mathbb{1}_{(l_i = l_j^q)} \right] \quad (3.8)$$

其中 $L(i, j)$ 表示第 i 个支持集样本和第 j 个查询集样本之间的损失， $\mathbb{1}_{(\cdot)}$ 是示性函数，当 (\cdot) 内的条件为真时为 1 否则为 0， l_i^s 是第 i 个支持集样本的真实类别标签， l_j^q 是第 j 个查询集样本的真实类别标签。该损失的直观理解为：当支持集样本与查询集样本同类别时，引导 score 向 1 靠拢，否则向 0 靠拢。

总体损失由每对样本的损失累加而得：

$$L = \frac{1}{N \times K \cdot M} \sum_{i=1}^{N \times K} \sum_{j=1}^M L(i, j) \quad (3.9)$$

训练过程中，网络映射 θ 被不断优化。在推理阶段，即可通过特征相似度分数来对查询集样本进行归类。

对于第 j 个查询集样本 x_j^q ，它与标签为 l_r 的第 r 类样本的总体分数，等于它

与支持集中所有第 r 类样本的分数均值：

$$\text{avg_score}(l_r, x_j^q) = \frac{1}{K} \sum_{i=rK+1}^{rK+K} \text{score}(h_i, p_j) \quad (3.10)$$

进而可以预测与第 j 个查询集样本 x_j^q 平均相似度最高的类别就是该样本的类别 l_j^q ：

$$l_j^q = \underset{l_r}{\text{Argmax}} \left[\text{avg_score}(l_r, x_j^q) \right] \quad (3.11)$$

总的来看，改进的孪生网络的训练本质是优化一个映射 θ ，它将支持集和查询集样本映射到特征空间，从而可以进一步计算特征相似度，在训练时计算损失或在推理时输出预测标签。可见，映射 θ 的具体网络结构设计是整体方法的重点。

在 3.2 节中将简要阐述映射 θ 使用的骨干网络：使用图文多模态预训练 CLIP 权重初始化的 Vision Transformer。在 3.3 节将进一步整体阐述 θ 的所采用的结构。

3.2 骨干网络

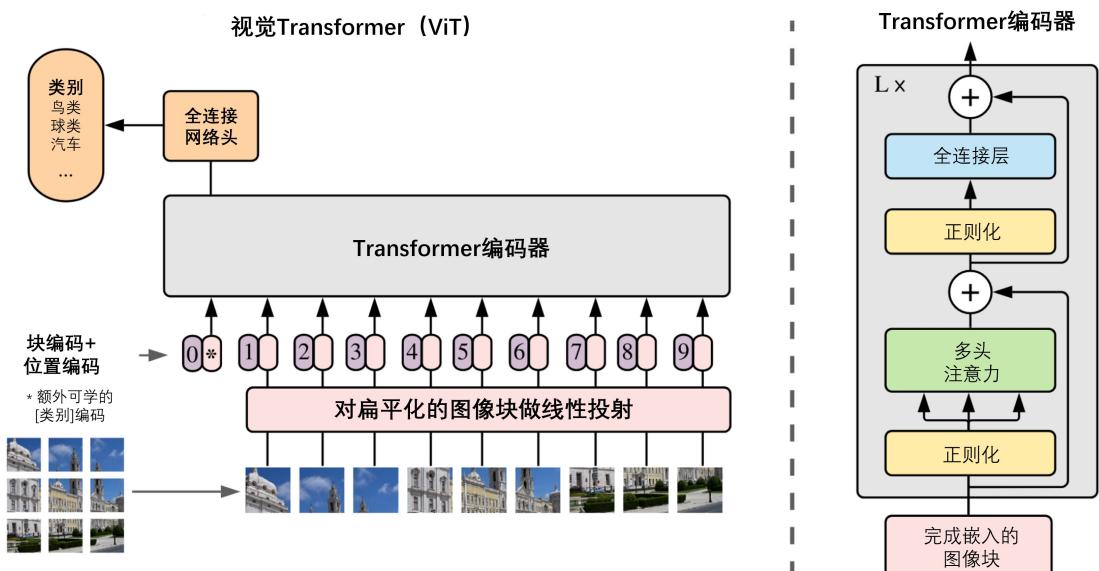


图 3.3 视觉 Transformer^[2]模型

Transformer^[36]是一种广泛应用于自然语言处理任务的模型，其核心思想是使用自注意力机制建立输入序列各元素间的关联。Vision Transformer(ViT)^[2]将这种思想应用于图像数据，首先将图像分解为一系列的图像块，然后使用图像块嵌入操作 (Patch Embedding) 将它们转为序列数据后输入到 Transformer 编码器来捕捉图

像内部的关联信息，最终通过全连接 (MLP) 分类头给出分类结果，如图 3.3 左侧所示。Transformer 编码器的内部由多个编码器块 (Block) 堆叠而成，块的内部又由层归一化 (Layer Norm)、多头自注意力 (Multi-Head Self Attention)、全连接层 (MLP) 等交替串连并引入残差连接，如图 3.3 右侧所示。

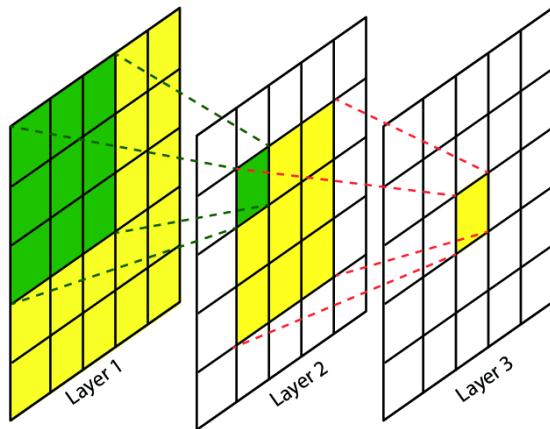


图 3.4 卷积的感受野

传统的卷积神经网络具有局部连接的特点，如图 3.4 所示，输入的特征图中只有被卷积核覆盖的部分可以产生信息流动，特征图中两个位置无法超越卷积核的感受野 (Receptive Field) 范围进行交互。局部连接的设计源于视觉数据的一个先验特点：相邻区域之间的特征具有较高的相关性。局部连接减少了网络的参数量，让网络的层数得以加深，给卷积网络带来出色的效果，使其长期主导视觉骨干网络^[21-24]。

ViT 的出现引入了全新的思路，自注意力机制使得序列中的每个元素能够与其他元素相互交互，这意味着相互距离较远的图像块也可以通过一步自注意力进行信息融合。这一特性将它强烈区别于依赖近邻数据交互单层感受野受限的卷积神经网络，让每个 ViT 编码块都具有很大的感受野，在模型的表达容量方面得到增强，在许多主流视觉任务上超越了卷积网络^[41-42,56]。然而，Zhu 等^[57]发现 ViT 在小规模数据上表现不及传统 CNN 模型，主要由于 ViT 摒弃了“相邻区域的特征相关性高”这一先验知识，在获得更高的模型容量上限时也需要更大规模的数据来训练。这提示我们数据规模对 ViT 的必要性。

基于 ViT 的大规模图文预训练良好地契合了 ViT 对数据量的需求。CLIP^[1]基于图、文两个 Transformer 结构进行对比学习，在 4 亿个图文对上进行预训练，如图 3.5 所示。这样在推理时即可根据图-文的特征相似度给出预测标签，如图 3.6。实验表明 CLIP 的预训练权重在各类下游任务上有强大的泛化能力，说明在预训练阶

(1) 对比学习预训练

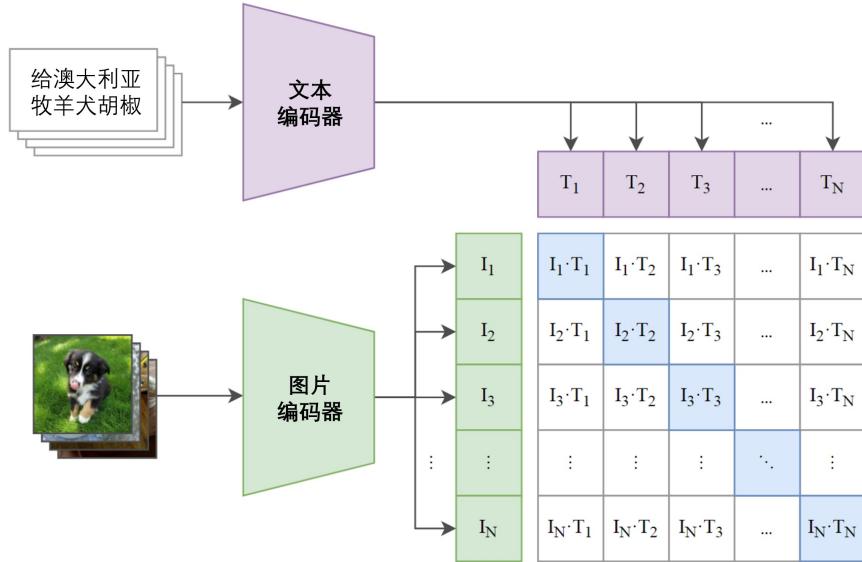


图 3.5 CLIP 的训练方法

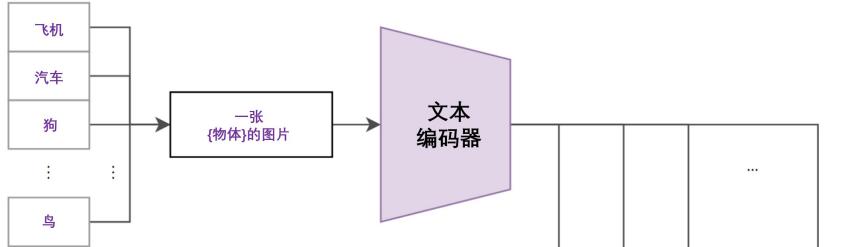
段模型学到了对各类 2D 场景视觉任务的通用、普适知识。这恰恰解决了小样本视频理解中场景建模困难的问题。

因此，为充分利用图文预训练知识，同时避免骨干网络在小规模数据上的过拟合，本文采用了冻结的 CLIP 预训练 ViT 权重作为骨干网络（也即帧编码器），来提取视频帧特征。

对于整个批次的 B 个视频，每个抽取 T 帧输入，帧编码器首先将每帧变形为 $H \times W$ 。而后将每帧划分为 $P \times P$ 的图像块共 L 个，经过 Transformer 嵌入编码后整个视频的特征张量尺寸变为 $T \times L \times D$ ，其中 D 为特征维数。将张量依次通过 M 个堆叠的冻结权重的 Transformer 编码器。选取尾部共 S 层，序号分别为 $[m_0, m_1, \dots, m_S]$ 的编码器块，在其前方加入可训练的轻量化时序模块（下文详述），并收集上述编码器的输出备用。输出张量的总尺寸可表示为 $B \times S \times T \times L \times D$ 。

如图 3.7 所示，以 Batch 大小为 10 个视频，采用 8 帧视频输入和 ViT-B/32 模

(2) 给带标签的文本创建分类数据集



(3) 零样本预测

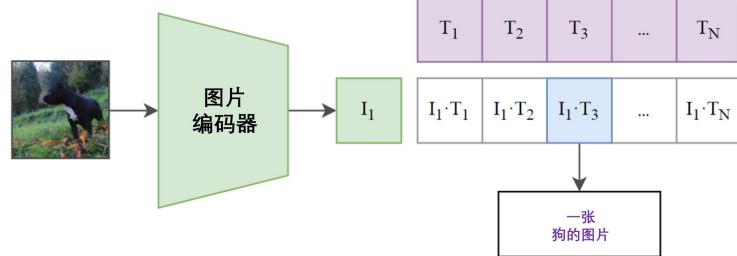


图 3.6 CLIP 的推理方法

型, 取末 4 层输出为例, 则:

$$B = 10,$$

$$T = 8,$$

$$H = W = 224,$$

$$P = 32,$$

$$L = \frac{224}{32} \times \frac{224}{32} = 49, \quad (3.12)$$

$$D = 768,$$

$$M = 12,$$

$$S = 4,$$

$$[m_0, m_1, \dots, m_4] = [8, 9, 10, 11].$$

注意到 ViT 是纯图像骨干网络, 对每个视频帧独立编码, 无法进行多帧信息之间的交互。这对于纯场景理解类型的视频数据 (如 Kinetics^[4]) 并无大碍, 但不利于模型在涉及时序的视频数据 (如 Something-Something V2^[5]) 上的表现。为此, 本文的帧编码器中还针对时序数据集加入了可训练的时序融合模块。

如前文所述, 时序融合模块仅会插入在序号为 $[m_0, m_1, \dots, m_S]$ 的编码器块里。模块在骨干网络中的位置如图 3.7 中红色标记所示, 插入在指定序号的所有编码器

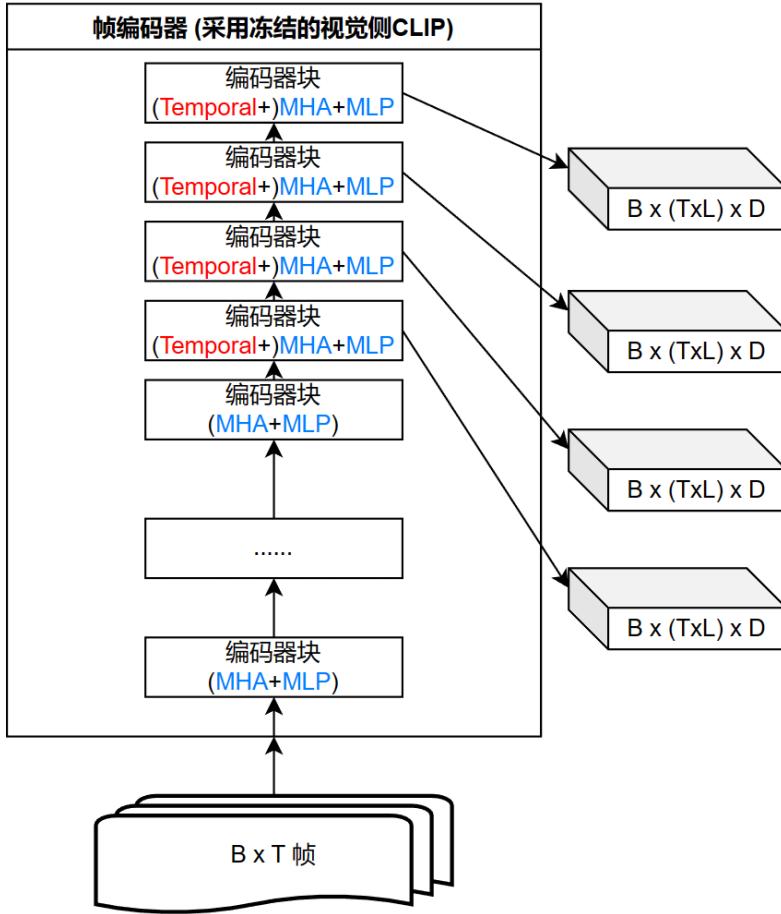


图 3.7 帧编码器：CLIP 初始化的 ViT-B/32

块的最初始位置，用其输出作为多头自注意力 (MHA) 的输入。时序融合模块是整个骨干网络中唯一随机初始化的模块，也是唯一可被更新权重的模块。

本文帧编码器中的时序模块，采用 1 维卷积和深度可分离卷积 (Depthwise Convolution)^[58] 交叠的结构。如 3.8 所示，对于 T 帧、序列长度 L 、特征维数 D 的视频特征，其张量尺寸为 $T \times L \times D$ 。首先经过一个特征压缩器，它是一个卷积核尺寸为 $1 \times 1 \times 1$ 的 3D 卷积，来压缩特征维数为原来的 $\frac{2}{3}$ 。而后经过一个时序融合器，它是一个轻量的深度可分离卷积，也即组数等于特征维数的组卷积 (Grouped Convolution)^[21]，其卷积核采用 $3 \times 1 \times 1$ ，即仅在时间维度进行卷积。最后经过一个特征扩展器，它也是一个卷积核尺寸为 $1 \times 1 \times 1$ 的 3D 卷积，来扩展特征维数为原来的 1.5 倍。注意到时序融合模块的输入、输出尺寸一致，因此无需对骨干网络的其它结构进行调整即可直接嵌入。

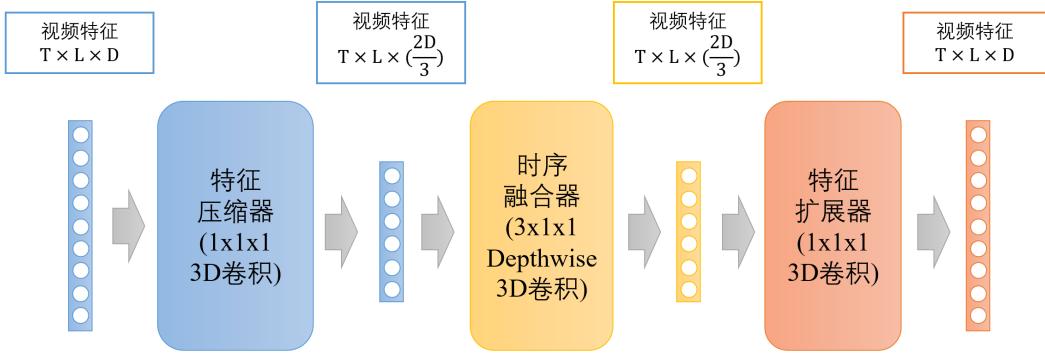


图 3.8 帧编码器中的时序融合模块（以单视频输入为例）

3.3 视频旁路聚合器 VBA

本节阐释映射 θ 的结构设计：视频旁路聚合器 (Vision Bypass Aggregator, VBA)。分别介绍 VBA 是如何基于骨干网络的输出特征，对输入进行跨帧信息聚合、跨视频特征增强、以及跨层信息聚合。

3.3.1 跨帧的信息聚合

由于骨干网络的主体部分——CLIP 预训练的 ViT 并不具备时序理解能力，虽然引入了轻量化的时序融合模块，仍不足以充分建模视频时序信息。为此，本文 VBA 结构的首个聚合目标就是进行跨帧的信息聚合，进一步增强时序理解力。

骨干网络共有 S 层的输出被保留，其层编号依次为 $[m_0, m_1, \dots, m_S]$ 。记第 i 层（下文统一采用此表示），也即骨干网络的第 m_i 个编码器块的输出视频特征张量为 A_i ，则 A_i 的尺寸为

$$A_i \in \mathbb{R}^{B \times (T \cdot L) \times D} \quad (3.13)$$

其中 B 为批次数量， T 为每个视频抽取的帧数， L 为序列长度， D 为特征维数。

如图 3.9 所示， A_i 将首先经过时序嵌入模块添加时序信息，而后计算交叉注意力。

具体而言，如图 3.10 所示， A_i 首先通过一个和骨干网络中结构完全相同的时序融合器，所得到的时序信息 α_i 。将 α_i 与 A_i 自身叠加，再进一步加入可学习的时间编码 β_i ，就得到了输出 A_i^* 。此处的 β_i 可以被视为一种可学习的提示语 (Learnable Prompt)，它们被初始化为零，在训练阶段依靠梯度不断更新，在推理阶段固定不变，与输入无关。

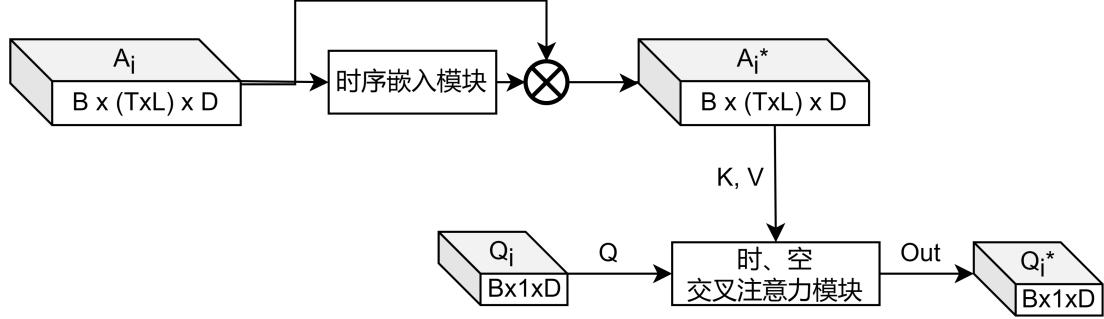


图 3.9 跨帧的信息聚合

给出时间嵌入模块 (Temporal Embedding, TE) 的整体数学表达为:

$$A_i^* = \text{TE}(A_i) = A_i + \alpha_i + \beta_i \quad (3.14)$$

而后进入解码环节。注意图3.9中省略了正则化 (LayerNorm) 与 MLP 等操作，仅画出了交叉注意力部分。注意力机制的表达式为:

$$\begin{aligned} & \text{Attention}(Q, K, V) \\ &= \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V \end{aligned} \quad (3.15)$$

在完整解码流程中，首先获得查询张量 Q_i ，其来源为:

$$Q_i = \begin{cases} \text{随机初始化的可学参数, } i = 0 \\ \text{CrossVideo}(Q_{i-1}^*), \quad i \geq 1 \end{cases} \quad (3.16)$$

其中 CrossVideo 表示跨视频的特征增强，将在下一小节中详述。

其尺寸为:

$$Q_i \in \mathbb{R}^{B \times 1 \times D} \quad (3.17)$$

然后对 A_i^* 与外部张量 Q_i 进行正则化，并以正则化的 A_i^* 为 key 和 value，以正则化的 Q_i 为 query，计算时序、空间的交叉注意力:

$$\begin{aligned} & \text{Attention}(\text{LN}(Q_i), \text{LN}(A_i^*), \text{LN}(A_i^*)) \\ &= \text{softmax}\left(\frac{\text{LN}(Q_i)\text{LN}(A_i^*)^T}{\sqrt{D}}\right)\text{LN}(A_i^*) \end{aligned} \quad (3.18)$$

其中 LN 表示 LayerNorm 正则化，D 为特征维数。

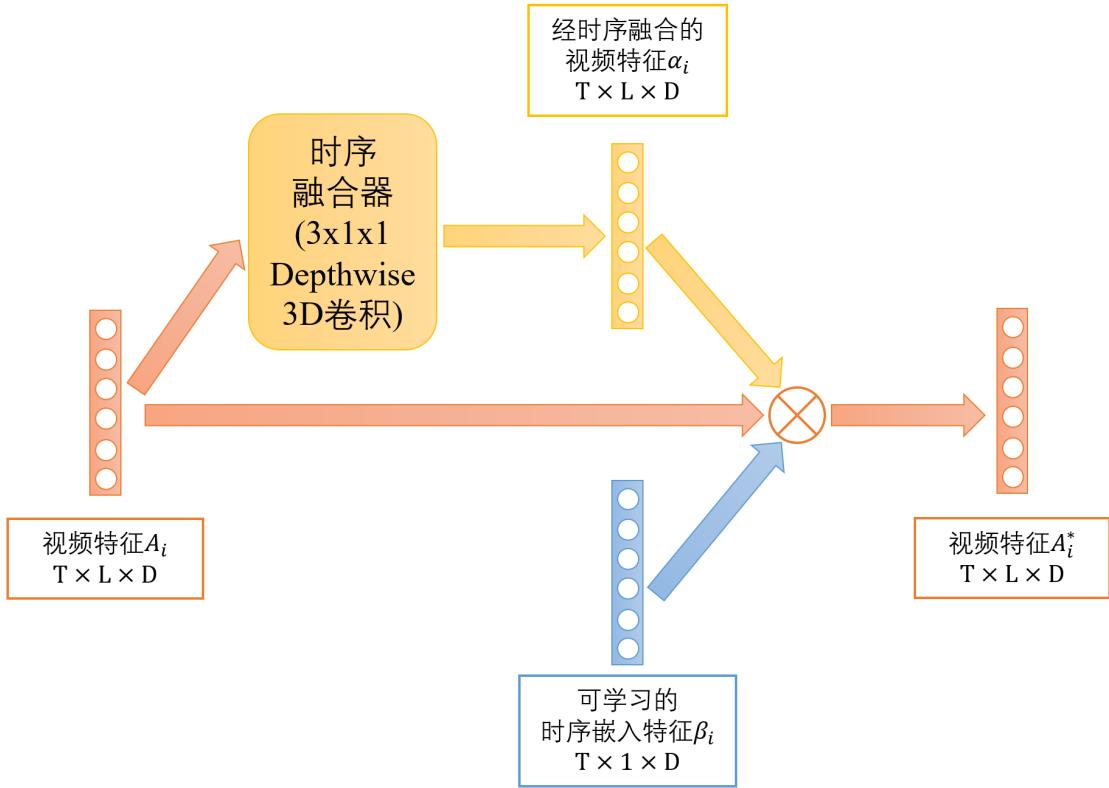


图 3.10 时序嵌入模块（以单视频输入为例）

最终再经 MLP 与残差连接，完成解码，给出跨帧信息聚合 CrossFrame 的整体计算式：

$$\begin{aligned} Q_i^* &= \text{CrossFrame}(Q_i, A_i^*) \\ &= Q_i + \text{Attention}(\text{LN}(Q_i), \text{LN}(A_i^*), \text{LN}(A_i^*)) + \text{MLP}(\text{LN}(Q_i)) \end{aligned} \quad (3.19)$$

3.3.2 跨视频的特征增强

注意到本小节之前的所有方法模块，即骨干网络中的帧编码器以及 VBA 中的跨帧信息聚合，均对每个视频样本独立理解，不涉及批次维度的信息交互。由于小样本任务的数据稀缺性，需要尽力挖掘每个批次数据中的全部信息，因此本文的 VBA 结构中设计了跨视频的特征增强模块。

具体来说，对于第 i 层的跨帧解码输出

$$Q_i^* \in \mathbb{R}^{B \times D} \quad (3.20)$$

其批次中包含了该轮前向过程中的所有支持集与查询集样本。为了让不同样本之间的特征相互增强，本文希望特征相近的样本可以更多地互相融合，而避免特征

不相近的样本互相干扰。



图 3.11 跨视频的信息聚合

为此，采用跨视频（跨批次）的自注意力，如图 3.11 所示。注意图中省略了正则化（LayerNorm）与 MLP 等操作，仅画出了自注意力部分。参考注意力机制表达式 3.15，得到跨视频自注意力表达式：

$$\begin{aligned} & \text{Attention}(\text{LN}(Q_i^*), \text{LN}(Q_i^*), Q_i^*) \\ &= \text{softmax}\left(\frac{\text{LN}(Q_i^*)\text{LN}(Q_i^*)^T}{\sqrt{D}}\right)Q_i^* \end{aligned} \quad (3.21)$$

最终再经 MLP 与残差连接，完成增强，其输出将作为下一层的查询输入，如表达式 3.16。给出跨视频信息聚合 CrossVideo 整体计算式：

$$\begin{aligned} Q_{i+1} &= \text{CrossVideo}(Q_i^*) \\ &= Q_i^* + \text{Attention}(\text{LN}(Q_i^*), \text{LN}(Q_i^*), Q_i^*) + \text{MLP}(\text{LN}(Q_i^*)) \end{aligned} \quad (3.22)$$

3.3.3 跨层的信息聚合

在计算机视觉任务中，神经网络的不同层次往往负责捕捉不同粒度的特征，其中浅层特征更加低级，对应图片的边缘、颜色等局部特征；深层特征语义级别更高，对应更高级的对象和场景特征。为了聚合 ViT 不同层次的输出特征，且保证特征具备基本的语义含义，本文选取了共 12 层的骨干网络 ViT-B/32 的末 4 层特征进行融合。

如图 3.12，骨干网络的输出特征 A_i 与查询张量 Q_i 在 CrossFrame、CrossVideo 两个模块的处理之后，输出张量 Q_{i+1} 恰好作为下一层的查询张量使用。

综合表达式 3.14，3.19 与 3.22，得到跨层信息聚合得数学表示：

$$\begin{aligned} Q_{i+1} &= \text{CrossVideo}(\text{CrossFrame}(Q_i, \text{TE}(A_i))) \\ i &= 0, 1, 2, 3 \\ \text{Output} &= Q_4 \end{aligned} \quad (3.23)$$

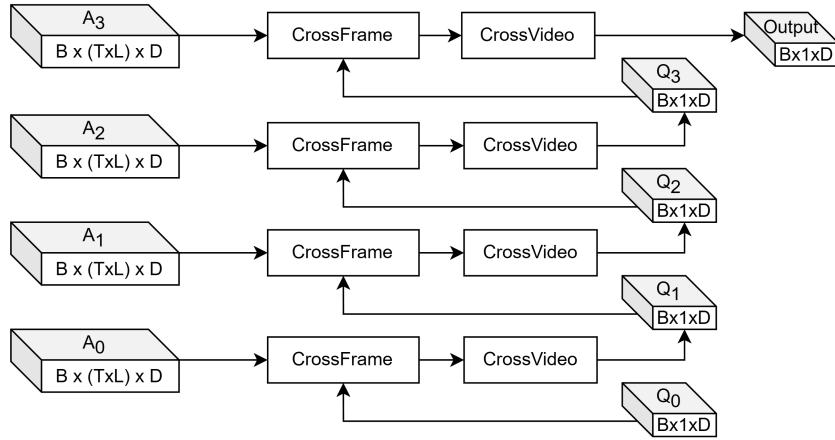


图 3.12 跨层的信息聚合

3.4 模型整体结构与本章小结

如图 3.13 所示，本文模型结构可总体概述为以下三点：

- 在学习范式上采用改进的孪生网络，通过支持集、查询集特征得余弦相似度进行分类。
- 骨干网络采用 CLIP 初始化并冻结权重的 ViT-B/32，其中部分层插入时序融合模块。
- 解码部分本文提出视频旁路聚合器 (VBA)，实现跨帧、跨视频、跨层的信息聚合。

具体地，本章第一节首先介绍了小样本学习的问题定义，引出了本文使用的改进孪生网络的形式，它与常规孪生网络的区别在于扩展了孪生网络映射 θ 的定义，引入了跨样本的信息融合。

而后，本章第二节介绍了骨干网络所采用的视觉 Transformer (ViT)^[2] 模型结构，分析了 ViT 结构为何对数据规模要求较高，进而引出了图文预训练方法 CLIP^[1]，此外还引入了时序融合模块来强化骨干网络的时序理解能力。

最后，本章第三节阐述了视频旁路聚合器 (VBA)，该结构从帧特征融合、同批次内的视频特征融合、以及跨层次的特征聚合三个角度进行设计，实现了信息的全面融合与建模。

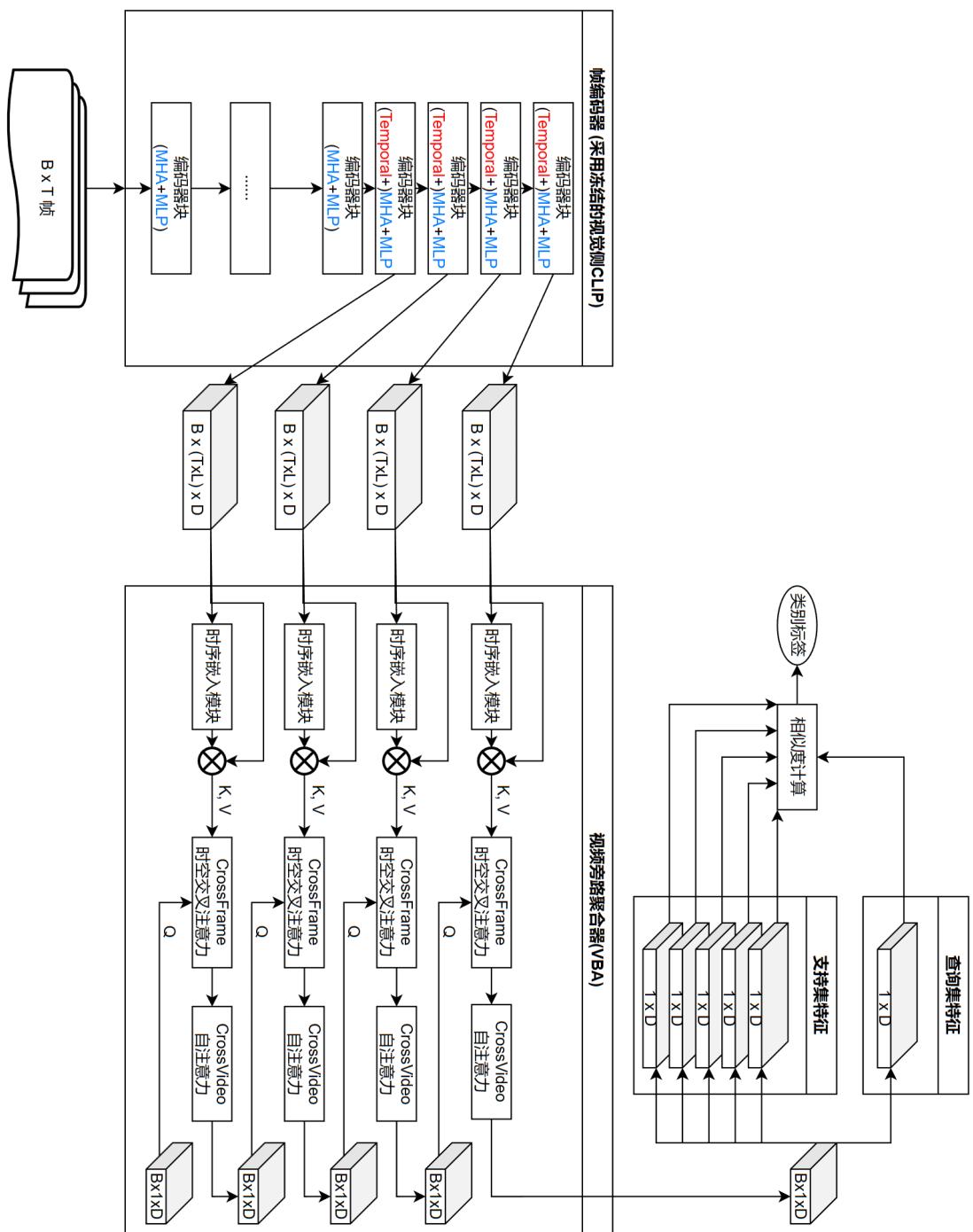


图 3.13 模型整体结构

第 4 章 实验结果

4.1 数据集

本文在 7 个科研用途下可公开的小样本视频数据集上进行了实验，它们的示例视频如图 4.1 所示。它们的小样本训练、验证、测试集划分如下：

对于 Kinetics^[4], SSV2-Small^[5], SSV2-Full^[5], UCF101^[30], 以及 HMDB51^[59] 数据集，我们采用已有工作中提出的划分方式^[18]。本文还额外使用了 Diving48^[60], 以及 Gym99^[61] 两个数据集，并进行了随机划分。数据集具体划分情况见附录 B。表 4.1 列出了不同数据集的划分类数与样本总数。

数据集	训练集类数	验证集类数	测试集类数	视频总数	训练视频总数
Kinetics ^[4]	64	12	24	10000	6400
SSV2-Small ^[5]	64	12	24	10000	6400
SSV2-Full ^[5]	64	12	24	82283	77500
UCF101 ^[30]	70	10	21	13320	9154
HMDB51 ^[59]	31	10	10	6766	4280
Diving48 ^[60]	27	10	10	16997	10789
Gym99 ^[61]	63	12	24	29005	16867

表 4.1 不同数据集的划分类数与样本总数

下面分别介绍每个数据集的特性。

如图 4.1(a) 的“健身”视频所示，Kinetics 数据集侧重 2D 场景理解，视频的每一帧都可以独立理解出视频的类别。

如图 4.1(a) 的“从杯中往出倒水”视频所示，SSV2 数据集侧重时序理解，其视频类别关注动作本身。典型类别如“把 xx 放在 xx 之上”、“用 xx 盖住 xx”。该数据集仅凭单帧难以进行分类，需要模型对多帧特征进行联合理解，从而判断出动作本身的类别。

如图 4.1(c) 的“刷牙”视频所示，UCF101 是一个广泛使用的行为识别数据集，包含 101 个常见的动作和活动，包括人类运动、日常生活、社交互动等。它在 2012 年即被提出，是最经典的视频分类数据集之一。UCF101 的建模难度相对较低，它的监督学习分类任务多年前就被基本攻克。在作为小样本数据集时它仍具有一定



(a) Kinetics 示例视频



(b) SSV2 示例视频



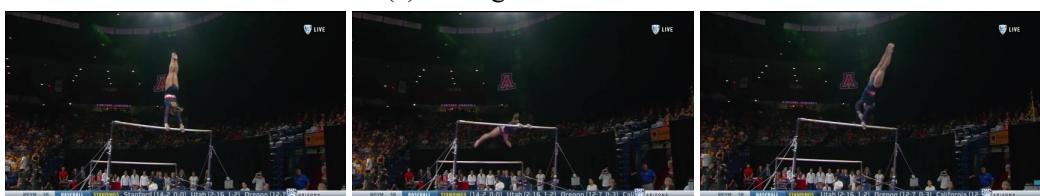
(c) UCF101 示例视频



(d) HMDB51 示例视频



(e) Diving48 示例视频



(f) Gym99 示例视频

图 4.1 各数据集示例视频

的挑战性，能够有效评估模型能力。

如图 4.1(d)的“喝水”视频所示，HMDB51 关注人类动作，由 51 个人类动作类别组成，如“亲吻”、“喝水”、“骑马”和“踢球”等。HMDB51 的拍摄环境多样性高，且部分动作类别涉及到精细的某个人体部位的运动，因此具有一定挑战性。HMDB51 也依赖于模型对多帧特征进行联合理解。

如图 4.1(e)的“跳水”视频所示，Diving48 的所有视频均为室内场馆的运动员跳水动作，该数据集不同类别视频的区别在于起跳、凌空、入水时的姿势各不相同。因此，为了理解 Diving48 数据集，模型不仅需要对多帧特征进行联合理解，还需要对动作进行细粒度建模和分辨，且几乎不能依赖任何场景信息辅助判断，具有较高的挑战性。

如图 4.1(f)的“体操”视频所示，Gym99 的所有视频均为室内场馆的体操运动员动作，视频持续时间长，标注精细。该数据集不同类别视频的区别在于不同体操动作的差异。因此，与 Diving48 类似，模型为了理解 Gym99 数据集不仅需要对多帧特征进行联合理解，还需要对动作进行细粒度建模和分辨，且几乎不能依赖任何场景信息辅助判断，具有较高的挑战性。

4.2 实现细节

软硬件环境方面，本文基于深度学习框架 PyTorch^[62]开展实验，并基于两个开源算法库：MMACTION2^[63]视频理解算法库、MMFewShot^[64]小样本学习算法库，搭建了自己的代码环境。所有实验均在单张 NVIDIA A100 80G GPU 上完成。得益于本文方法的显存高效性，主实验结果的峰值显存占用未超过 16G，可在单张 NVIDIA RTX3090 24G GPU 或更高规格的其它型号 NVIDIA GPU 上复现。为保证高效的训练与推理，本文使用了特征缓存机制，对于单个任务预留的服务器内存应不少于 150G。

参数设置方面，由于小样本学习不存在周期（epoch）的概念，因此本文所有实验采用基于迭代轮次的运行器（Iter-based Optimizer）。优化器采用 AdamW^[65]，对于 Kinetics 与 UCF101 数据集学习率 $lr = 1e-5$ ，对于其它数据集学习率 $lr = 3e-5$ ，其它优化器参数 $\beta = [0.9, 0.999]$, $weight_decay = 0.02$ 。对于 SSV2-Full 数据集，余弦相似度的温度 $Temp = 50$ ，其余数据集 $Temp = 100$ 。所有随机种子均设置为 42，所有训练与推理均开启 PyTorch 的 deterministic 选项，保证 GPU 运算没有随机性，以确保结果可复现。

模型结构方面，对于不涉及时序建模的数据集 Kinetics 与 UCF101，本文未启用骨干网络的时序融合模块。对于 SSV2-Full 与 UCF101，由于启用 VBA 中的跨视频特征增强会带来负面影响，因此不启用。除此以外，模型结构均包含了第三章所阐述的全部模块。对于 Diving48 和 Gym99，使用 5 类 1 样本任务对模型进行训练，并在 5 类 1 样本、5 类 5 样本任务上进行测试。对于其它所有数据集，均使用 5 类 5 样本任务对模型进行训练，并在 5 类 1 样本、5 类 5 样本任务上进行测试。

抽帧策略方面，采用 TSN^[32]提出的均匀抽帧策略，对每个视频抽取 8 帧。具体来说，首先将视频帧均分为 8 份，而后在每一份中随机抽取一帧，兼顾均匀性与随机性。

特征增强策略方面，所有数据集均在训练阶段使用了随机裁剪（Random Resized Crop）或中心裁剪（Center Crop），也都按照 ImageNet 的平均 RGB 参数（ $\text{mean} = [123.675, 116.28, 103.53]$, $\text{std} = [58.395, 57.12, 57.375]$ ）对输入帧进行了正则化。其中 HMDB51 和 UCF101 对应的实验还使用了随机增强 ImgAug^[66]。

混合精度方面，所有训练与推理均开启 MMAction2^[63]提供的动态 fp16 半精度方法。

4.3 定量实验结果与分析

4.3.1 主实验结果及对比

表 4.2 与表 4.3 分别列出了 5 类 1 样本、5 类 5 样本任务在 5 个数据集上的精度。其中 K 表示 Kinetics、S 表示 SSV2-Small、F 表示 SSV2-Full、H 表示 HMDB51、U 表示 UCF101。注意到表 4.2 与表 4.3 均分为上下两部分，上半部分的骨干网络均为 ‘IN1K-pre R50’，表示 ImageNet-1K^[43]预训练的 ResNet50^[23]；下半部分的骨干网络均为 CLIP^[1]预训练的 ViT^[2]。其中本文方法 *VBA* 的精度来自本人实验，HyRSM^[19]在 CLIP 预训练 ViT 上的精度来自本人的复现，其余方法的精度均来该方法对应原文，或引用该文章的后续学术文章。需要补充的是，ResNet50 系列方法保证公平比较，一律采用的都是原始的 ImageNet-1K 预训练权重，并非 torchvision 使用更新的训练策略得到的新权重^[67]。由于 STRM^[20]与 Eff-Prompt^[15]未公布 5 类 1 样本任务的精度，因此它们的精度在表 4.2 中未列出。

表 4.4 列出了本文方法 *VBA*，以及本文复现的更换骨干网络的 HyRSM 在 Diving48, Gym99 两个数据集上的精度。其中 D-1shot 表示使用 Diving48 数据集进行 5 类 1 样本任务，D-5shot 表示使用 Diving48 数据集进行 5 类 5 样本任务，G-1shot

方法			数据集				
名称	收录	骨干网络	K	S	F	H	U
CMN ^[16]	ECCV18	IN1K-pre R50	60.5	36.2	/	45.6	71.8
OTAM ^[6]	CVPR20		73.0	36.4	42.8	54.5	79.9
ITANet ^[17]	IJCAI21		73.6	39.8	49.2	/	/
TRX ^[18]	CVPR21		63.6	/	42.0	53.1	78.2
MTFAN ^[44]	CVPR22		74.6	/	45.7	59.0	84.8
LSTC ^[45]	IJCAI22		73.4	/	46.7	60.9	85.7
HyRSM ^[19]	CVPR22		73.7	40.6	54.3	60.3	83.9
HyRSM ^[19]	CVPR22	CLIP ViT-B/32	82.0	48.3	59.0	71.1	92.6
VBA	/		86.2	52.1	61.5	73.4	91.9
VBA	/	CLIP ViT-B/16	88.9	53.6	62.8	78.4	93.7

表 4.2 5 类 1 样本任务的 Top-1 精度

表示使用 Gym99 数据集进行 5 类 1 样本任务，G-5shot 表示使用 Gym99 数据集进行 5 类 5 样本任务。由于已公开的文献中，未找到使用 Diving48, Gym99 两个数据集进行小样本视频理解的方法，因此本表仅列出本文实验得出的结果。

由表 4.2 与表 4.3 的实验结果可见，对于 5 类 1 样本、5 类 5 样本两个主流的小样本学习任务，本文提出的 VBA 模型在 5 个主流视频数据集上都大幅超越了一系列基于 ResNet50 且未引入预训练模型的方法。

为了进一步证明本文 VBA 模型的有效性，表 4.2、表 4.3 与表 4.4 还为目前的 SOTA 方法：收录于 CVPR2022 的 HyRSM^[19] 模型更换了骨干网络，从 ImageNet-1K 预训练的 ResNet50 替换为了 CLIP 预训练的 ViT-B/32。对比更换骨干网络前后的 HyRSM，可以发现它在 5 个数据集上的精度均大幅提高，这证明了引入预训练知识的有效性。再用 VBA 对比 HyRSM，在同样骨干网络和同样预训练的条件下，不论是 5 类 1 样本任务还是 5 类 5 样本任务，在 7 个数据集上的相对精度都有明显提高或达到可比程度。这说明本文 VBA 方法的结构设计有效地迁移了预训练知识，比直接在已有方法中引入预训练的效果更优。

表 4.3 还对比了直接使用 CLIP 预训练权重的方法 Efficient-Prompt^[15]。该方法采用 CLIP 预训练的 ViT-B/16 作为骨干网络，为此本文将 VBA 也换用 ViT-B/16

方法			数据集				
名称	收录	骨干网络	K	S	F	H	U
CMN ^[16]	ECCV18	IN1K-pre R50	78.9	48.8	/	62.6	88.7
OTAM ^[6]	CVPR20		85.8	48.0	52.3	68.0	88.9
ITANet ^[17]	IJCAI21		84.3	53.7	62.3	/	/
TRX ^[18]	CVPR21		85.9	59.1	64.6	75.6	96.1
MTFAN ^[44]	CVPR22		87.4	/	60.4	74.6	95.1
LSTC ^[45]	IJCAI22		86.5	/	66.7	76.8	96.5
STRM ^[20]	CVPR22		86.7	/	68.1	77.3	96.9
HyRSM ^[19]	CVPR22		86.1	56.1	69.0	76.0	94.7
HyRSM ^[19]	CVPR22	CLIP ViT-B/32	91.4	64.1	74.9	83.9	96.8
VBA	/		95.1	71.2	74.7	87.1	97.5
			(+3.7)	(+7.1)	(-0.2)	(+3.2)	(+0.7)
Eff-Prompt ^[15]	ECCV22	CLIP ViT-B/16	96.4	/	/	85.3	98.3
VBA	/		95.8	72.9	76.8	90.4	98.4
			(-0.6)			(+5.1)	(+0.1)

表 4.3 5 类 5 样本任务的 Top-1 精度

方法			数据集			
名称	收录	骨干网络	D-1shot	D-5shot	G-1shot	G-5shot
HyRSM ^[19]	CVPR22	CLIP ViT-B/32	74.4	85.6	90.0	93.6
VBA	/		75.7	90.1	91.3	95.4
			(+1.3)	(+4.5)	(+1.3)	(+1.8)

表 4.4 Diving48, Gym99 数据集的 Top-1 精度

进行公平比较。由表 4.3 可见同样设定下，本文的 VBA 在三个数据集上超越了 Efficient-Prompt 或达到可比程度，在涉及到时序建模的 HMDB51 上提升尤其大 (+5.1)，从一个侧面反映了 Efficient-Prompt 时序建模能力很可能偏弱。从另一个侧面，虽然 Efficient-Prompt 未公开时序小样本数据 SSV2-Small 和 SSV2-Full 的精度导致无法对比，但该方法所公开的其在监督学习范式下的 SSV2 数据集分类精度明显偏低（详见 Efficient-Prompt 原文 Table3），这也暗示该方法很可能没有良好的时序建模能力。本文的 VBA 则具有强大的时序建模能力，真正地将预训练的图像知识迁移到了视频知识域中。

4.3.2 各组件的有效性验证

方法		数据集				
名称	时序融合	S	F	H	D	G
VBA	无	45.2	51.3	72.4	62.5	84.6
VBA	有	52.1 (+6.9)	61.5 (+10.2)	73.4 (+1.0)	75.7 (+13.2)	91.3 (+6.7)

表 4.5 使用 5 类 1 样本任务，验证骨干网络中时序融合的有效性

如表 4.5，对于本文所使用的涉及时序理解的数据集，启用骨干网络中的时序融合模块均带来了精度的提升。这说明了帧编码器中时序融合模块的有效性。

方法		数据集						
名称	VBA 层数	K	S	F	H	U	D	G
VBA	1	78.5	47.0	60.7	70.5	91.9	70.5	90.3
VBA	4	86.2 (+7.7)	52.1 (+5.1)	61.5 (+0.8)	73.4 (+2.9)	91.9 (+0)	75.7 (+5.2)	91.3 (+1.0)

表 4.6 使用 5 类 1 样本任务，验证跨层次融合的有效性

如表 4.6，相比于只利用骨干网络单层输出的 VBA，启用多层次结构均带来了精度的提升。这说明本文所采用的多层结构能够融合跨层次、跨粒度的特征，证明了 VBA 跨层融合 (CrossFrame) 的有效性。

如表 4.7，相比于使用每个视频独立编码，启动跨视频的特征融合之后给大多数数据集精度带来了提升，证明了 VBA 跨视频融合 (CrossVideo) 的有效性。然而

方法		数据集						
名称	跨视频融合	K	S	F	H	U	D	G
VBA	无	79.1	49.1	61.5	67.2	91.9	75.1	90.1
VBA	有	86.2 (+7.1)	52.1 (+3.0)	58.1 (-3.4)	73.4 (+6.2)	90.2 (-1.7)	75.7 (+0.6)	91.3 (+1.2)

表 4.7 使用 5 类 1 样本任务，验证跨视频融合的有效性

对于不同数据集效果差异较大，对 Kinetics 和 HMDB51 数据集提升幅度最大，而对 SSV2-Full 和 UCF101 数据集反而降低了精度。猜测这是因为跨视频的特征增强是使用自注意力实现的，本质上依赖特征相似度进行加权，而特征相似度对更偏语义的场景信息较为敏感，对时序信息则不太敏感。对于 Kinetics 和 HMDB51 都是比较依赖于场景理解的视频，跨视频的特征融合可以让每个样本在编码过程中“见到”其它的同类或异类样本，从而增加场景的丰富性。而对于 SSV2-Full 这类依赖动作时序理解的数据集，同一场景下的视频也可能属于不同类别；且该划分的训练数据量很大（77500 个训练视频），使得跨视频特征增强这一本意解决数据稀缺性的方法变得没有价值了。

4.3.3 跨域理解

方法名称	训练数据集	测试数据集	5 类 1 样本	5 类 5 样本
Random Pick	/	/	20	20
VBA	Gym99	Gym99	91.3	95.4
VBA	Diving48	Gym99	47.6	67.9
VBA	Diving48	Diving48	75.7	90.1
VBA	Gym99	Diving48	28.1	37.9

表 4.8 跨域理解

传统监督学习往往采用分类头进行多分类问题，只能对训练集出现过的类别进行归类。本文所采用的改进型孪生网络，发挥了小样本学习可以对新类进行归类的能力，通过相似度度量进行归类的方法突破了传统监督学习的限制。

然而，本文与其它已有的小样本学习方法都是将某个数据集进行划分，这样能测试出模型在新类的泛化能力，却不能考察模型在新的数据域内的泛化能力。若

测试数据和训练数据的分布完全不同，那么模型是否仍能依靠相似度度量，给出有价值的分类结果？

为了探索这一问题，本文实验探索了 VBA 模型在 Diving48、Gym99 两个完全不同分布的数据集的相互跨域理解能力。具体来说，测试了 Diving48 训练集上训练的 VBA 模型在 Gym99 测试集上的表现，以及 Gym99 训练集上训练的 VBA 模型在 Diving48 测试集上的表现。如表 4.8 所示，在 Diving48 训练的模型竟在 Gym99 数据集的 5 类 1 样本任务中，给出了高达 47.6% 的 Top-1 准确度，远高于随机选取的 20%。而对于难度更高的 Diving48 数据集的 5 类 1 样本任务，在 Gym99 训练的模型仅能给出 28.1% 的 Top-1 准确度，但也仍高于随机。

上述实验结果表明，以特征相似度为基础的小样本学习方法可以学到通用的视觉知识。即使在数据域差距巨大的条件下，在其它数据分布上学到的特征映射也对新的数据分布具有一定的价值。也正是特征映射的通用性让视觉预训练、图文多模态预训练等工作纷纷采用了对比学习的方式，其底层原理和本文所采用的学习范式是相通的。

4.3.4 训练开销

以 5 类 5 样本任务为例，支持集内样本总数为 $5 \times 5 = 25$ ，查询集内样本总数为 $5 \times 5 = 25$ ，共计 50。设置 $meta_batch = 3$ 时，则每个批次内的样本总数为 150。每个视频样本采样 8 帧。所有实验均在单张 NVIDIA A100 80G GPU 上完成。

数据集	迭代总数	骨干网络 时序模块	50 轮 迭代耗时	总耗时	显存占用
Kinetics	1k	无	30s	10min	8G
SSV2-Small	3k	有	60s	1h	16G
SSV2-Full	15k	有	60s	5h	16G
HMDB51	1k	有	60s	20min	16G
UCF101	2k	无	30s	20min	8G
Diving48	15k	有	60s	5h	16G
Gym99	15k	有	60s	5h	16G

表 4.9 实测的训练开销

训练开销如表 4.9 所示，可见在单个批次的视频数量高达 150 个时，场景数据集（未启用骨干网络的时序模块）的显存占用低至 8G，且每次迭代耗时仅 0.6s。使

用 1 张 GPU，仅需 10 分钟即可将 CLIP 特征迁移到 Kinetics 小样本数据上。对于骨干网络启用了时序模块的情况，显存占用、迭代耗时均翻倍，但仍很高效。

本文方法算力、显存双维度高效性的核心原因在于：

- 对场景数据，骨干网络的全部权重无需更新；对时序数据，骨干网络的绝大部分权重无需更新。
- 对场景数据，梯度完全不流经骨干网络，仅更新视频旁路聚合器；对时序数据，梯度仅需传回骨干网络的末 4 层，并更新视频旁路聚合器。
- 混合精度训练，进一步降低了算力开销，并节约显存。
- 编写了特征缓存机制，整个批次的特征全部提取完毕后再统一计算相似度，从而减少了前向过程的总次数。

4.4 可视化展示

图 4.2 对 VBA 中时空交叉注意力模块里的注意力图 (Attention Map) 进行了可视化。图中是 Diving48 数据集某视频抽取的 8 帧，及其对应的注意力图。可以发现，注意力被良好地集中在了人体及周边附近，集中在了和分类决策最相关的部分上，说明 VBA 学到了如何关注该数据集中与类别最相关的部分。

4.5 本章小结

本章介绍实验相关的内容，首先给出本文所使用的 7 个数据集，而后从软硬件环境、参数设置、模型结构、训练时策略等多方面介绍实现细节。之后给出了主要实验结果，证明本文提出的 VBA 结构在同等设定下可以超越已有的最领先方法或与之达到可比水平，通过消融实验证明各组件有效性，并介绍了训练、测试数据不同分布的跨域理解结果，给出训练开销证明本文 VBA 结构的高效性。最后给出可视化结果，定性证明了本文方法的有效性。

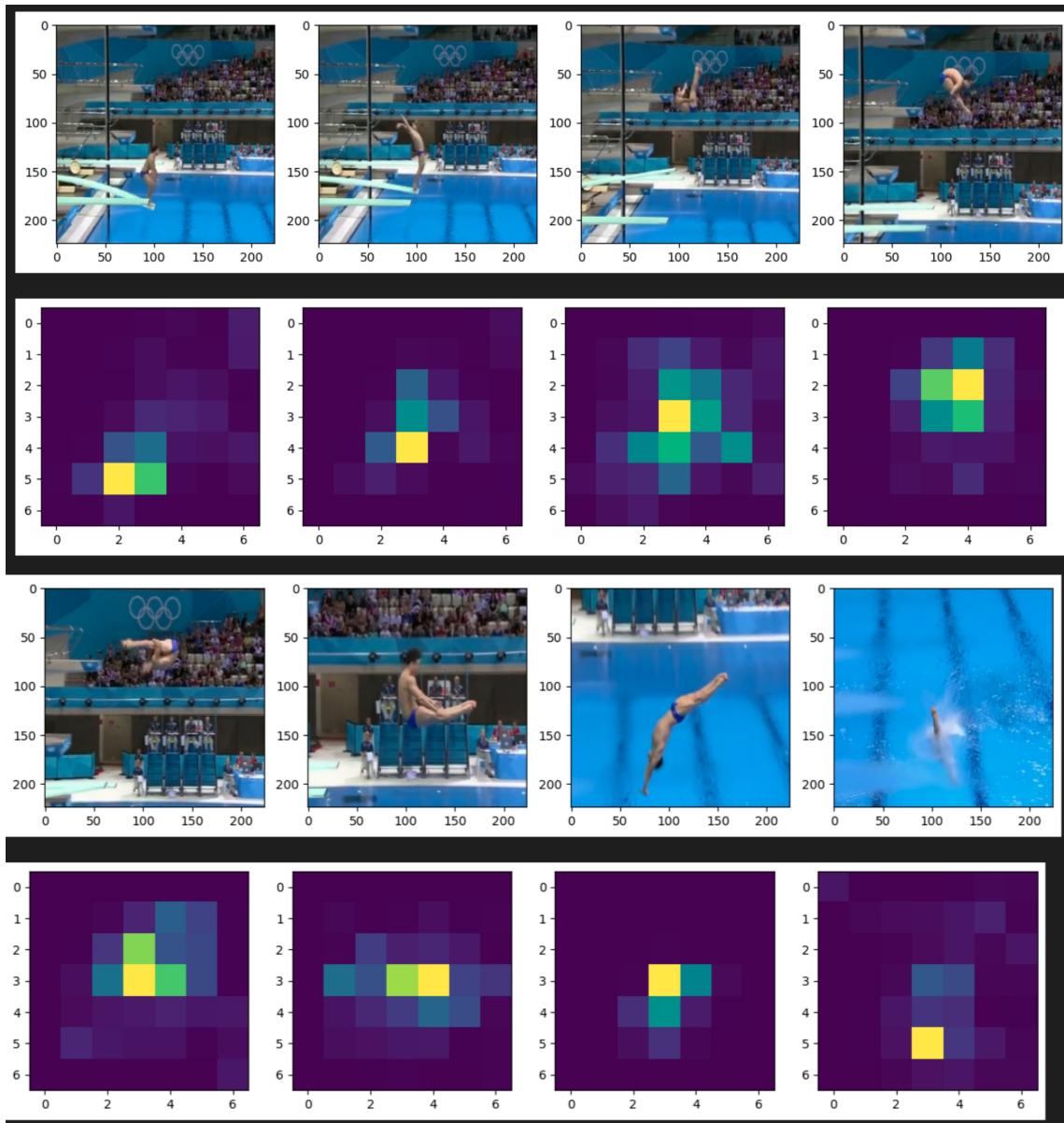


图 4.2 注意力图可视化

第 5 章 结论

小样本视频分类任务是计算机视觉的任务之一，该任务要求模型在极少量样本条件下达成针对视频的分类能力，考察模型在数据极度匮乏的条件下，在避免过拟合的条件下对视频的场景、时间序列等特征进行综合建模的能力。

本文针对小样本视频分类任务，提出了**视频旁路聚合器（Video Bypass Aggregator, VBA）**模型。本方法的骨干网络利用了在互联网级规模的图文对数据上学习得到的预训练模型 CLIP^[1]，它具备强大的特征表示能力，可以将 2D 图片编码为语义信息丰富的特征向量，但图像模型并不具备视频时间序列的建模能力。为此，本文在骨干网络的一旁搭建了旁路结构，称为“视频旁路聚合器”。它既可以将预训练的场景建模能力迁移到视频数据域，又可以在视频数据中新学到时序建模能力，还可以联合建模小样本一幕（episode）中的全部支持集、查询集样本，获得鲁棒的特征表示。

具体而言，本文方法整体采用扩展的孪生网络，基于特征相似度对查询集视频进行归类。在孪生网络具体的映射部分，本文骨干网络采用 CLIP 预训练初始化的 Vision Transformer^[2]并冻结权重，而本文的旁路结构采用本文提出的 VBA 结构。首先，VBA 通过可学习的查询向量作为提示词（prompt），层层传递来解码出骨干网络的每层输出特征，有效地迁移并聚合了预训练特征的多粒度场景建模能力。其次，VBA 通过轻量化的时间融合模块，以及交叉注意力机制进行跨帧信息聚合，充分学习到针对视频的时序建模能力。此外，VBA 还通过批次维度的自注意力，实现了跨视频样本的信息聚合，联合利用支持集、查询集的所有样本信息，对每个样本获得更加鲁棒的特征表示。

在 6 个主流小样本视频数据集共 7 种小样本划分下，本文的方法相比同等设定下的已有最佳方法，达到更高或可比精度，证明本文方法的有效性。其中，VBA 在相同的骨干网络下，在精度上超越了本任务的已有最优方案^[19]或与之可比；VBA 还超越了原生基于预训练特征 CLIP 开发的方法^[15]，尤其在时序建模能力方面大幅提高，是真正为视频而设计的方法。

广泛的消融实验证明了 VBA 内各个模块的有效性，包括：多层次聚合结构、多帧时间融合结构，以及跨样本特征增强结构。特别创新地，本文进行了跨域理解实验，证明本文的方法具备一定的跨数据分布的普适性。本文还论证了方法在训练速度、显存占用两方面的高效性。最后，本文给出了定性可视化结果。

随着各类图文多模态预训练方法的蓬勃发展，对于如何在小样本视频分类任务中有效地利用这类预训练知识，本文所提出的方法提供了有益的参考。

未来的改进方向可能包括以下方面：

- VBA 结构对不同骨干网络或预训练的普适性。本文论证了 VBA 结构在 CLIP 预训练的 ViT-B/32 的有效性。对于其它类型的预训练，如大规模监督学习数据集 ImageNet-21K^[68]预训练的 ViT，又如视觉掩码学习预训练 BEiT^[69]初始化的 ViT 等，与 VBA 结构联合使用时的有效性尚不明确。对于其它类型的骨干网络，如该任务常用的 ResNet50^[23]，VBA 的有效性也尚不明确。
- VBA 跨数据域零样本能力的来源。此前，小样本视频理解任务普遍关注同数据域的不同类别迁移能力，而很少考虑跨域数据的普适性。在第四章的 4.3.3 小节中，本文创新地给出了 VBA 方法在跨数据域任务中的表现，说明 VBA 方法具备一定的零样本跨域迁移能力。然而，该能力究竟是源于预训练知识，还是源于 VBA 结构中的某个模块，这一问题还有待进行更深入的定量、定性探索。

插图索引

图 1.1 5 类 1 样本任务示例 ^[3]	1
图 1.2 SSV2 数据集示例：“拉近 xxx 与 xxx 的距离” ^[5]	2
图 1.3 整体结构示意图	4
图 2.1 MatchingNet ^[26] ，一种典型的度量学习方法	8
图 2.2 EVL ^[14] ，一种典型的高效视频理解方法	10
图 2.3 用统一的框架理解各类 PEFT 方法 ^[52]	14
图 2.4 “过墙梯” ^[54] ，一种典型的迁移学习方法	15
图 3.1 3 类 3 样本分类任务示意图 ^[55]	16
图 3.2 孪生网络示意图 ^[55]	17
图 3.3 视觉 Transformer ^[2] 模型	19
图 3.4 卷积的感受野	20
图 3.5 CLIP 的训练方法	21
图 3.6 CLIP 的推理方法	22
图 3.7 帧编码器：CLIP 初始化的 ViT-B/32	23
图 3.8 帧编码器中的时序融合模块（以单视频输入为例）	24
图 3.9 跨帧的信息聚合	25
图 3.10 时序嵌入模块（以单视频输入为例）	26
图 3.11 跨视频的信息聚合	27
图 3.12 跨层的信息聚合	28
图 3.13 模型整体结构	29
图 4.1 各数据集示例视频	31
图 4.2 注意力图可视化	40

表格索引

表 4.1 不同数据集的划分类数与样本总数	30
表 4.2 5 类 1 样本任务的 Top-1 精度	34
表 4.3 5 类 5 样本任务的 Top-1 精度	35
表 4.4 Diving48, Gym99 数据集的 Top-1 精度	35
表 4.5 使用 5 类 1 样本任务，验证骨干网络中时序融合的有效性	36
表 4.6 使用 5 类 1 样本任务，验证跨层次融合的有效性	36
表 4.7 使用 5 类 1 样本任务，验证跨视频融合的有效性	37
表 4.8 跨域理解	37
表 4.9 实测的训练开销	38

参考文献

- [1] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PMLR, 2021: 8748-8763.
- [2] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[A]. 2020.
- [3] Li W, Wu Z, Zhang J, et al. Lgsim: local task-invariant and global task-specific similarity for few-shot classification[J]. Neural computing and applications, 2020, 32: 13065-13076.
- [4] Carreira J, Zisserman A. Quo vadis, action recognition? a new model and the kinetics dataset [C]//proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 6299-6308.
- [5] Goyal R, Ebrahimi Kahou S, Michalski V, et al. The” something something” video database for learning and evaluating visual common sense[C]//Proceedings of the IEEE international conference on computer vision. 2017: 5842-5850.
- [6] Cao K, Ji J, Cao Z, et al. Few-shot video classification via temporal alignment[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 10618-10627.
- [7] Li J, Selvaraju R, Gotmare A, et al. Align before fuse: Vision and language representation learning with momentum distillation[J]. Advances in neural information processing systems, 2021, 34: 9694-9705.
- [8] Li J, Li D, Xiong C, et al. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation[C]//International Conference on Machine Learning. PMLR, 2022: 12888-12900.
- [9] Wang P, Yang A, Men R, et al. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework[C]//International Conference on Machine Learning. PMLR, 2022: 23318-23340.
- [10] Alayrac J B, Donahue J, Luc P, et al. Flamingo: a visual language model for few-shot learning [J]. Advances in Neural Information Processing Systems, 2022, 35: 23716-23736.
- [11] Luo H, Ji L, Zhong M, et al. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning[J]. Neurocomputing, 2022, 508: 293-304.
- [12] Wang M, Xing J, Liu Y. Actionclip: A new paradigm for video action recognition[A]. 2021.

- [13] Ni B, Peng H, Chen M, et al. Expanding language-image pretrained models for general video recognition[C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV. Springer, 2022: 1-18.
- [14] Lin Z, Geng S, Zhang R, et al. Frozen clip models are efficient video learners[C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV. Springer, 2022: 388-404.
- [15] Ju C, Han T, Zheng K, et al. Prompting visual-language models for efficient video understanding [C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV. Springer, 2022: 105-124.
- [16] Zhu L, Yang Y. Compound memory networks for few-shot video classification[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 751-766.
- [17] Zhang S, Zhou J, He X. Learning implicit temporal alignment for few-shot video classification [A]. 2021.
- [18] Perrett T, Masullo A, Burghardt T, et al. Temporal-relational crosstransformers for few-shot action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 475-484.
- [19] Wang X, Zhang S, Qing Z, et al. Hybrid relation guided set matching for few-shot action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 19948-19957.
- [20] Thatipelli A, Narayan S, Khan S, et al. Spatio-temporal relation modeling for few-shot action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 19958-19967.
- [21] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
- [22] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [A]. 2014.
- [23] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [24] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.
- [25] Koch G, Zemel R, Salakhutdinov R, et al. Siamese neural networks for one-shot image recognition[C]//ICML deep learning workshop: volume 2. Lille, 2015.

- [26] Vinyals O, Blundell C, Lillicrap T, et al. Matching networks for one shot learning[J]. Advances in neural information processing systems, 2016, 29.
- [27] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning[J]. Advances in neural information processing systems, 2017, 30.
- [28] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks [C]//International conference on machine learning. PMLR, 2017: 1126-1135.
- [29] Satorras V G, Estrach J B. Few-shot learning with graph neural networks[C]//International conference on learning representations. 2018.
- [30] Soomro K, Zamir A R, Shah M. Ucf101: A dataset of 101 human actions classes from videos in the wild[A]. 2012.
- [31] Tran D, Bourdev L, Fergus R, et al. Learning spatiotemporal features with 3d convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2015: 4489-4497.
- [32] Wang L, Xiong Y, Wang Z, et al. Temporal segment networks: Towards good practices for deep action recognition[C]//European conference on computer vision. Springer, 2016: 20-36.
- [33] Tran D, Wang H, Torresani L, et al. A closer look at spatiotemporal convolutions for action recognition[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018: 6450-6459.
- [34] Wang X, Girshick R, Gupta A, et al. Non-local neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7794-7803.
- [35] Feichtenhofer C, Fan H, Malik J, et al. Slowfast networks for video recognition[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6202-6211.
- [36] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [37] Bertasius G, Wang H, Torresani L. Is space-time attention all you need for video understanding? [C]//ICML: volume 2. 2021: 4.
- [38] Zhang Y, Li X, Liu C, et al. Vidtr: Video transformer without convolutions[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 13577-13587.
- [39] Arnab A, Dehghani M, Heigold G, et al. Vivit: A video vision transformer[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 6836-6846.
- [40] Fan H, Xiong B, Mangalam K, et al. Multiscale vision transformers[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 6824-6835.
- [41] Liu Z, Ning J, Cao Y, et al. Video swin transformer[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 3202-3211.

- [42] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.
- [43] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database[C]//2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009: 248-255.
- [44] Wu J, Zhang T, Zhang Z, et al. Motion-modulated temporal fragment alignment network for few-shot action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 9151-9160.
- [45] Luo W, Liu Y, Li B, et al. Long-short term cross-transformer in compressed domain for few-shot video classification[C/OL]//Raedt L D. Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22. International Joint Conferences on Artificial Intelligence Organization, 2022: 1247-1253. <https://doi.org/10.24963/ijcai.2022/174>.
- [46] He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9729-9738.
- [47] Moon T K. The expectation-maximization algorithm[J]. IEEE Signal processing magazine, 1996, 13(6): 47-60.
- [48] Houlsby N, Giurgiu A, Jastrzebski S, et al. Parameter-efficient transfer learning for nlp[C]// International Conference on Machine Learning. PMLR, 2019: 2790-2799.
- [49] Li X L, Liang P. Prefix-tuning: Optimizing continuous prompts for generation[A]. 2021.
- [50] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning [C/OL]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021: 3045-3059. <https://aclanthology.org/2021.emnlp-main.243>. DOI: 10.18653/v1/2021.emnlp-main.243.
- [51] Hu E J, Shen Y, Wallis P, et al. Lora: Low-rank adaptation of large language models[A]. 2021.
- [52] He J, Zhou C, Ma X, et al. Towards a unified view of parameter-efficient transfer learning [C/OL]//International Conference on Learning Representations. 2022. <https://openreview.net/forum?id=0RDcd5Axok>.
- [53] Lian D, Zhou D, Feng J, et al. Scaling & shifting your features: A new baseline for efficient model tuning[A]. 2022.
- [54] Sung Y L, Cho J, Bansal M. Lst: Ladder side-tuning for parameter and memory efficient transfer learning[A]. 2022.

- [55] Kundu R. Everything you need to know about few-shot learning[EB/OL]. [2023-05-20]. <https://blog.paperspace.com/few-shot-learning/>.
- [56] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]// Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. Springer, 2020: 213-229.
- [57] Zhu H, Chen B, Yang C. Understanding why vit trains badly on small datasets: An intuitive perspective[A]. 2023.
- [58] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[A]. 2017.
- [59] Kuehne H, Jhuang H, Garrote E, et al. Hmdb: a large video database for human motion recognition[C]//2011 International conference on computer vision. IEEE, 2011: 2556-2563.
- [60] Li Y, Li Y, Vasconcelos N. Resound: Towards action recognition without representation bias [C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 513-528.
- [61] Shao D, Zhao Y, Dai B, et al. Finegym: A hierarchical video dataset for fine-grained action understanding[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 2616-2625.
- [62] Paszke A, Gross S, Massa F, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library[C/OL]//Wallach H, Larochelle H, Beygelzimer A, et al. Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 2019: 8024-8035. <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [63] Contributors M. Openmmlab’s next generation video understanding toolbox and benchmark [EB/OL]. 2020. <https://github.com/open-mmlab/mmaction2>.
- [64] Contributors M. Openmmlab few shot learning toolbox and benchmark[EB/OL]. 2021. <https://github.com/open-mmlab/mmfewshot>.
- [65] Loshchilov I, Hutter F. Decoupled weight decay regularization[A]. 2017.
- [66] Jung A B, Wada K, Crall J, et al. imgaug[EB/OL]. 2020. <https://github.com/aleju/imgaug>.
- [67] Vryniotis V. How to train state-of-the-art models using torchvision’ s latest primitives[EB/OL]. (2021-11-18)[2023-05-27]. <https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives/>.
- [68] Ridnik T, Ben-Baruch E, Noy A, et al. Imagenet-21k pretraining for the masses[A]. 2021.
- [69] Bao H, Dong L, Piao S, et al. BEit: BERT pre-training of image transformers[C/OL]// International Conference on Learning Representations. 2022. <https://openreview.net/forum?id=p-BhZSz59o4>.

致 谢

衷心感谢周杰老师，唐彦嵩老师对我的悉心指导。周老师、唐老师是我的科研引路人，他们敏锐的研究洞察力，严谨的研究作风，以及卓越的研究品味带给我启迪和思考。

感谢一直陪伴我的父母、亲人和朋友，你们始终如一的支持让我顺利完成了本科的学习，并即将踏上新的旅程。

四年时光飞逝，但收获的知识和锻炼将受益终生。期待今后在更广阔的平台上，争做更有影响力的研究。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：王逸敏 日 期：2023-5-31

附录 A 外文资料的书面翻译

利用冻结的 CLIP 模型进行高效的视频理解

A.1 摘要

视频识别一直被端到端学习范式所主导 - 首先使用预训练图像模型的权重初始化视频识别模型，然后在视频上进行端到端训练。这使得视频网络能够受益于预训练的图像模型。然而，这需要大量的计算和显存资源来微调模型，若直接使用预训练的图像特征而不微调图像骨干网络则导致非最优的结果。所幸，最近在对比视觉语言预训练（CLIP）方面的进展为视觉识别任务开辟了一条新路。这些在互联网规模的图像-文本对数据上预训练的模型，具有丰富语义的强大视觉表示。在本文中，我们提出了高效视频学习（Efficient Video Learning, EVL）- 一种直接使用冻结的 CLIP 特征训练高质量视频识别模型的有效框架。具体来说，我们采用轻量级 Transformer 解码器，并学习一个查询令牌，以动态地从 CLIP 图像编码器中收集帧级空间特征。此外，我们在每个解码器层中采用局部时间模块，以从相邻帧及其注意力映射中发现时间线索。我们的结果展示了：虽然使用了冻结的骨干网络来保证训练高效，但我们的模型仍在各种视频识别数据集上学习到了高质量的视频表示。

A.2 简介

学习时空表示是视频理解的基本组件，在近年来一直是一个活跃的研究领域。自深度学习时代开始以来，已经提出了许多体系结构来学习时空语义，例如传统的双流网络、3D 卷积神经网络和时空 Transformer。

由于视频具有高维度和大量的时空冗余性，从头开始训练视频识别模型非常低效，可能导致性能较差。直观地，视频片段的语义意义与其各个单独帧高度相关。以往的研究表明，图像识别的数据集和方法也能够惠及视频识别。由于图像和视频识别之间的密切关系，大多数现有的视频识别模型都利用预训练的图像模型，通过使用它们进行初始化，然后以端到端的方式重新训练所有参数，以实现视频理解。

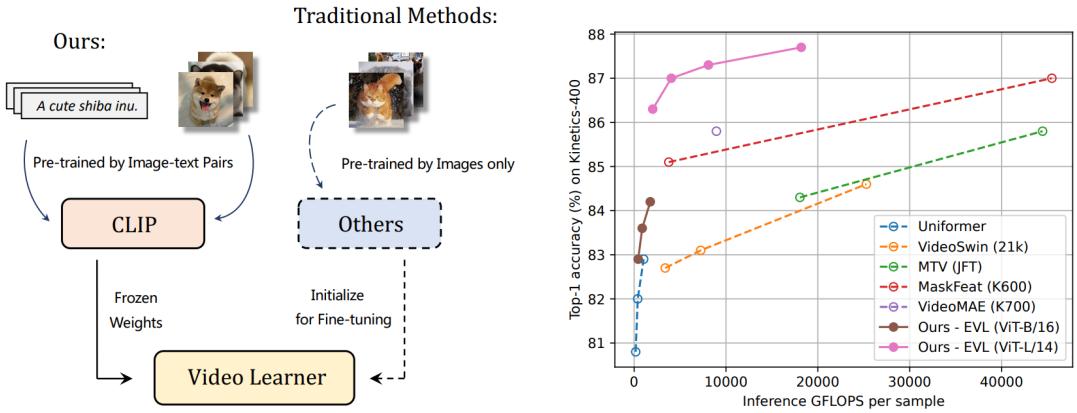


图 A.1 左：我们的 **EVL** 训练流程与其他视频识别方法之间的差异的示意图。右：尽管 **EVL** 旨在实现高效训练，但我们的模型在准确性与推理 *FLOPS* 的帕累托边界上树立了新的标准。在 Kinetics-400 数据集上，8 帧 ViT-B/16 模型只需进行 60 个 V100 GPU 小时的训练就能达到 82.9% 的 top-1 准确率。

然而，端到端微调机制存在两个主要缺点。第一个是效率。视频识别模型需要同时处理多个帧，且在模型大小方面比其图像对应物大几倍。微调整整个图像主干不可避免地会带来巨大的计算和显存消耗成本。因此，这个问题限制了在受限计算资源下采用和扩展一些最大的图像架构用于视频识别的范围。第二个问题在转移学习的背景下被称为灾难性遗忘。在下游视频任务上进行端到端微调时，我们冒着破坏从图像预训练中学习到的强大视觉特征并获得次优结果的风险，如果下游视频信息不足。这两个问题表明，从预训练的图像模型进行端到端微调并不总是一个理想的选择，这需要一种更有效的学习策略来将知识从图像传递到视频。

通过对对比学习、视觉掩码学习建模和传统的监督学习等方法，已经做出了大量努力来学习高质量和通用的视觉表示。视觉掩码学习方法，如 MAE，训练编码器-解码器架构，从潜在表示和掩码标记中重构原始图像。基于监督学习的方法使用固定的预定义类别标签训练图像主干。由于它们通常是单模态训练的，它们都缺乏表示丰富语义的能力。相比之下，对比视觉语言模型，如 CLIP，是使用大规模开放词汇的图像-文本对进行预训练的。它们可以学习与更丰富语义对齐的更强大的视觉表示。CLIP 的另一个优点是其有前途的特征可转移性，为各种下游任务上的一系列迁移学习方法提供了强有力的基础。

以上原因启发我们重新思考图像和视频特征之间的关系，并设计有效的迁移学习方法，利用冻结的 CLIP 图像特征进行视频识别。为此，我们提出了一种基于轻量级 Transformer 解码器的高效视频学习（**EVL**）框架。**EVL** 与其他视频识别模型的差异如图 A.1 左所示。具体而言，**EVL** 学习一个查询标记，动态地从 CLIP 图

像编码器的每个层中收集帧级空间特征。在此基础上，我们引入一个局部时间模块，借助于时间卷积、时间位置嵌入和跨帧注意力来收集时间线索。最后，使用全连接层预测视频类别的分数。我们进行了大量实验来展示我们的方法的有效性，并发现 EVL 是一个简单有效的流程，具有更高的准确性，但训练和推理成本更低，如图 A.1 右所示。我们的贡献如下：

- 我们指出了当前视频理解中端到端学习范式的缺陷，并提出了利用冻结的 CLIP 图像特征来促进视频识别任务的方法。
- 我们开发了一种高效的从图像到视频识别的传递学习流程 (EVL)，其中我们训练一个轻量级的 Transformer 解码器模块在固定的可传递的图像特征之上进行时空融合。
- 大量实验证明了 EVL 的有效性和高效性。它的训练时间比端到端微调短得多，但实现了竞争性能。这使得视频识别对于拥有平均计算资源的更广泛社区来说更加可行。

A.3 相关工作

视频识别。 最近视频识别的进展可以分为两个主要方向——改进模型体系结构和提出新的训练策略。随着 Transformer 在图像识别中的成功，视频识别也从 3D-CNN 向基于 Transformer 的体系结构转变。Uniformer 是一种定制的融合 CNN-Transformer 体系结构，实现了良好的速度-准确性平衡。Yan 等提出了一种多流 Transformer，在不同分辨率上操作并具有横向连接。先前的工作已经显示出图像预训练对于视频识别任务的好处。然而，端到端微调仍然很昂贵，特别是由于大显存占用。在新的训练策略方面，自监督学习的预设任务设计和多任务协同训练是两个主流方向。然而，两者的成本比常规监督训练更高。与以前的工作不同，我们利用固定的 CLIP 图像特征，并直接学习一个高效的视频识别模型，加上一个额外的 Transformer 编码器。

大规模图像表示学习。 随着大规模弱标记数据的可用性，我们见证了通用视觉表示学习的新模型的激增。使用常规监督学习构建的图像模型规模已经大幅增长。例如，Zhai 等人在大型 JFT-3B 数据集上训练了一个 ViT-G 模型。Riquelme 等人创建了一个规模超过 100 亿个参数的混合专家视觉模型。为了进一步提升视觉表示能力，人们开始专注于大规模对比学习和自监督学习。BERT 的成功引发了使用视觉掩码学习的大规模视觉模型构建的新方向。与此同时，CLIP 和 ALIGN 在由开放

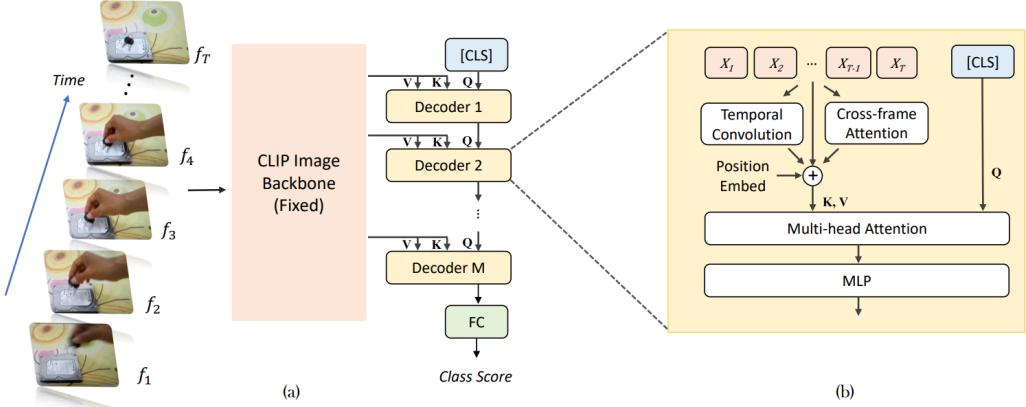


图 A.2 模型架构概述。**(a)** 顶层架构：从大规模预训练的图像骨干网络中提取多个中间特征图，并将它们送到 Transformer 解码器中以从中获取信息。**(b)** 增强运动的 Transformer 解码器块：在原始帧特征 X_i 之上添加时间建模，以保留时空特征的结构信息。

词汇图像-文本对组成的大规模数据集上使用对比损失预训练视觉-语言模型。多模态预训练环境使它们适用于需要丰富语义的下游任务。

高效迁移学习。 这些工作与我们的方法最相关。大多数关于高效传递学习的以前的工作是针对自然语言处理和图像识别的。一些方法在微调过程中学习参数高效的权重差异向量，利用稀疏性或低秩分解。一系列方法训练适配器，这是额外的带有残差连接的全连接层，保持预训练模型中原始权重不变。另一类方法学习提示，这是附加的可学习令牌，附加到输入或中间特征序列上，用于特定任务的自适应。虽然这些方法与我们的目标相同，但我们采用 Transformer 解码器，这更加灵活，也更加高效，如我们在方法部分中所分析的那样。

在视频识别方面，关于高效传递学习的探索仍然有限。Ju 等人通过学习提示和时间建模将 CLIP 模型转移到视频识别中。Wang 等人利用传统的端到端微调方法将 CLIP 模型用于视频识别。我们将在实验部分与它们进行比较。还有一些利用可传递的图像特征进行视频文本任务的工作，但这些工作更侧重于跨模态建模。相反，我们的工作旨在提高单模态视频表示，这应该是大多数视频文本学习方法的补充。

A.4 方法

我们的图像到视频传递学习流程有三个主要目标：(1) 能够总结多帧特征并推断视频级别的预测；(2) 能够捕获跨多个帧的运动信息；(3) 高效性。因此，我们提出了高效视频学习（EVL）框架，我们将在接下来详细介绍。

A.4.1 总体结构

EVL 的总体结构如图 A.2 所示，它是一个固定的 CLIP 骨干网络之上的多层时空 Transformer 解码器。CLIP 骨干网络独立地从每个帧中提取特征。然后将帧特征堆叠起来形成一个时空特征体积，加上时间信息进行调制，然后输入到 Transformer 解码器中。Transformer 解码器对多层特征进行全局聚合：学习一个视频级别的分类令牌 [CLS] 作为查询，将来自不同骨干块的多个特征体积作为键和值输入到解码器块中。线性层将最后一个解码器块的输出投影到类别预测上。形式上，Transformer 解码器的操作可以表示为：

$$\mathbf{Y}_i = \text{Temp}_i ([\mathbf{X}_{N-M+i,1}, \mathbf{X}_{N-M+i,2}, \dots, \mathbf{X}_{N-M+i,T}]), \quad (\text{A.1})$$

$$\tilde{\mathbf{q}}_i = \mathbf{q}_{i-1} + \text{MHA}_i (\mathbf{q}_{i-1}, \mathbf{Y}_i, \mathbf{Y}_i), \quad (\text{A.2})$$

$$\mathbf{q}_i = \tilde{\mathbf{q}}_i + \text{MLP}_i (\tilde{\mathbf{q}}_i), \quad (\text{A.3})$$

$$\mathbf{p} = \text{FC} (\mathbf{q}_M), \quad (\text{A.4})$$

其中， $\mathbf{X}_{n,t}$ 表示从 CLIP 骨干网络的第 n 层提取的第 t 帧的特征， \mathbf{Y}_i 表示输入到 Transformer 解码器的第 i 层的时空调制特征体积， \mathbf{q}_i 是逐步精炼的查询令牌，其中 \mathbf{q}_0 是可学习的参数， \mathbf{p} 是最终的预测结果。 N 、 M 分别表示骨干图像编码器和时空解码器中块的数量。MHA 表示多头注意力，其中三个参数分别为查询、键和值。Temp 是时间建模，它产生由更细粒度的时间信息调制的特征令牌，如下一节所述。

该网络通过使用交叉熵损失函数和真实标签进行优化，作为标准分类模型进行训练，但是在图像特征 \mathbf{X} 处停止反向传播，并且不会更新图像编码器中的任何权重。

A.4.2 从空间特征中发掘时序信息

虽然 CLIP 模型能够生成强大的空间特征，但它们完全缺乏时间信息。尽管 Transformer 解码器能够进行加权特征聚合，这是一种全局时间信息，但精细的和局部的时间信号也可能对视频识别有价值。因此，在将特征输入到 Transformer 解码器之前，我们引入了以下时间模块来编码这些信息。

时间卷积. 时间深度卷积能够捕捉沿时间维度的局部特征变化，已知其高效和有效。形式上，该卷积编码的特征表示为 \mathbf{Y}_{conv} ，并且：

$$\mathbf{Y}_{\text{conv}}(t, h, w, c) = \sum_{\Delta t \in \{-1, 0, 1\}} \mathbf{W}_{\text{conv}}(\Delta t, c) \mathbf{X}(t + \Delta t, h, w, c) + \mathbf{b}_{\text{conv}}(c). \quad (\text{A.5})$$

时间位置编码. 我们学习了一组 T 个维度为 C 的向量, 表示为 $\mathbf{P} \in \mathbb{R}^{T \times C}$, 作为时间位置嵌入。根据它们的时间位置 t , 将图像特征与其中一个向量相加, 形式上表示为:

$$\mathbf{Y}_{\text{pos}}(t, h, w, c) = \mathbf{P}(t, c). \quad (\text{A.6})$$

虽然时间卷积也可以隐式地捕获时间位置信息, 但通过使用位置嵌入, 可以更明确地使不同时刻的相似特征区分开来。对于长程时间建模, 位置嵌入也更加强大, 需要堆叠多个卷积块才能实现大的感受野。

时间交叉注意力. 另一个有趣但经常被忽视的时间信息来源是注意力图。由于注意力图反映了特征对应关系, 因此在两个帧之间计算注意力图自然地揭示了物体的运动信息。具体而言, 我们首先使用 CLIP 中原始的查询和键投影构建相邻帧之间的注意力图:

$$\begin{aligned} \mathbf{A}_{\text{prev}}(t) &= \text{Softmax}\left((\mathbf{QX}(t))^T (\mathbf{KX}(t-1))\right), \\ \mathbf{A}_{\text{next}}(t) &= \text{Softmax}\left((\mathbf{QX}(t))^T (\mathbf{KX}(t+1))\right). \end{aligned} \quad (\text{A.7})$$

为了简单起见, 我们省略了注意力头, 并在我们的实现中对所有头进行了平均。然后, 我们将其线性投影到特征维度:

$$\begin{aligned} \mathbf{Y}_{\text{attn}}(t, h, w, c) &= \sum_{h'=1}^H \sum_{w'=1}^W \mathbf{W}_{\text{prev}}(h-h', w-w', c) \mathbf{A}_{\text{prev}}(t, h', w') + \\ &\quad \mathbf{W}_{\text{next}}(h-h', w-w', c) \mathbf{A}_{\text{next}}(t, h', w'). \end{aligned} \quad (\text{A.8})$$

实验证明, 尽管查询、键和输入特征都是从纯 2D 图像数据中学习的, 这种注意力图仍然提供了有用的信号。

最终的调制特征是通过将时序特征与原始空间特征以残差的方式混合得到的, 具体而言, 它们是: $\mathbf{Y} = \mathbf{X} + \mathbf{Y}_{\text{conv}} + \mathbf{Y}_{\text{pos}} + \mathbf{Y}_{\text{attn}}$.

A.4.3 复杂度分析

推理 由于只使用一个查询令牌, 因此额外的 Transformer 解码器只引入了可以忽略不计的计算开销。为了说明这一点, 我们将 ViT-B/16 作为我们的图像骨干网络, 并将 Transformer 块的 FLOPS 写成如下形式:

$$\text{FLOPS} = 2qC^2 + 2kC^2 + 2qkC + 2\alpha qC^2 \quad (\text{A.9})$$

这里， q 、 k 、 C 、 α 分别表示查询令牌的数量、键（值）令牌的数量、嵌入维度的数量以及多层感知机（MLP）扩展因子。通过这个公式，我们可以大致比较编码器块和解码器块的 FLOPS (h 、 w 、 t 是沿着高度、宽度、时间维度的特征大小，我们采用常见的选择 $\alpha = 4$, $h = w = 14$, $C = 768$ 进行估计)：

$$\frac{\text{FLOPS}_{\text{dec}}}{\text{FLOPS}_{\text{enc}}} \approx \frac{2hwtC^2}{t(12hwC^2 + 2h^2w^2C)} \approx \frac{1}{6} \quad (\text{A.10})$$

通过这个公式，我们可以看到解码器块相比编码器块更加轻量级。即使采用完整配置（在每个编码器输出上放置一个解码器块，不进行通道缩减，并启用所有时间模块），FLOPS 的增加也仅在骨干网络的 20% 以内。

训练 由于我们使用固定的骨干网络和非侵入式 Transformer 解码器头（即我们插入的模块不改变任何骨干网络层的输入），因此我们完全可以避免通过骨干网络进行反向传播。这大大减少了显存消耗和训练迭代时间。

A.5 实验

本文基于 Kinetics-400 和 Something-Something-v2 两个数据集对我们的方法进行了基准测试。附录提供了额外的实现细节。

A.5.1 主实验结果

本节中，我们将与最近工作中的重要基线进行比较。

与最先进的方法的比较。 表A.1中提供了与最近的视频识别模型的比较。虽然我们旨在构建一个快速的迁移学习管道，但我们发现我们的模型在常规视频识别方法中实现了具有竞争力的准确性。表A.1中列出的模型实现了与我们类似的准确性，但需要比我们的方法更多的计算。

与基于 CLIP 的方法的比较。 据我们所知，有两项先前的研究利用 CLIP 模型进行视频识别。如表A.2所示，我们在使用较少帧和较少新参数的情况下获得了更高的准确性，显示出更有效的 CLIP 使用。

训练时间和减少的内存。 我们高效的迁移学习管道的主要优势之一是大大减少了

表 A.1 在 Kinetics-400 数据集上与最先进方法的比较。我们引用了一系列在准确度范围类似于我们的模型，并比较了它们的 FLOPS。帧数报告为每视图帧数 \times 视图数。

方法	预训练	Acc. (%)	帧数	GFLOPS
Uniformer-B	ImageNet-1k	82.9	32×4	1,036
Swin-B	ImageNet-21k	82.7	32×12	3,384
irCSN-152	IG-65M	82.6	32×30	2,901
MViT-S	ImageNet-21k	82.6	16×10	710
Omnivore-B	IN1k + SUN	83.3	32×12	3,384
ViViT-L FE	JFT	83.5	32×3	11,940
TokenLearner 8at18 (L/16)	JFT	83.2	32×6	6,630
MViT-L	MaskFeat, K600	85.1	16×10	3,770
MTV-L	JFT	84.3	32×12	18,050
<hr/>				
		82.9	8×3	444
EVL ViT-B/16 (本文方法)		CLIP	16×3	888
			32×3	1,777
<hr/>				
		86.3	8×3	2,022
EVL ViT-L/14 (本文方法)		CLIP	16×3	4,044
			32×3	8,088
EVL ViT-L/14 (336px, 本文方法)			87.7	18,196
<hr/>				

训练时间。我们引用了表A.4中几项先前研究报告的训练时间进行比较。^① 在这种情况下，强大的预训练导致大约 $10\times$ 的训练时间缩短，而我们高效的迁移学习方案进一步缩短了约 $8\times$ 的时间。我们还在表A.5中比较了理想化设置下的训练时间：我们使用虚假数据在单个 GPU 上报告单个步骤时间（前向传播 + 反向传播 + 更新）。这样可以绕过数据加载和分布式通信开销，这些可能是未优化和难以控制的混淆因素。

^① Uniformer-B 的训练时间是通过将其在 Kinetics-600 数据集中提供的值减半得出的。TimeSformer 的训练时间是我们自己复现的，我们发现其比其论文中报告的数字小几倍（报告值约为 400 小时）。ActionCLIP 的训练时间是通过将其论文中 8 帧变体报告的值加倍得出的。

表 A.2 在 Kinetics-400 数据集上与基于 CLIP 的方法的比较。所有模型均使用 ViT-B/16 作为骨干网络。由于 Efficient-Prompt 论文中对细节描述模糊不清，我们估计它们的新参数为 3 个 Transformer 块，特征大小为 512，MLP 扩展因子为 4。对于 ActionCLIP，我们不计算文本分支中的参数。

方法	新的可训参数 (M)	帧数 × 视图数	Acc. (%)
Efficient-Prompting (A5)	9.43*	16×5	76.9
		8×1	81.1
		16×1	81.7
ActionCLIP	105.15	32×1	82.3
		16×3	82.6
		32×3	83.8
EVL ViT-B/16 (本文方法, 1 层)	7.41	8×3	81.1
EVL ViT-B/16 (本文方法, 4 层)	28.70	8×3	82.9
EVL ViT-B/16 (本文方法, 4 层)	28.78	32×3	84.2

表 A.3 在实际硬件上测量的推理延迟和吞吐量。两个模型在 Kinetics-400 上均达到 82.9% 的准确率。使用 PyTorch 内置混合精度在 V100-32G 上获得了结果。使用批量大小为 1 来测量延迟，并使用尽可能大的批量大小来测量吞吐量，直至耗尽显存。

模型 (帧数)	Acc (%)	GFLOPS	延时 (ms)	吞吐量 (V/s)
Uniformer-B (32)	82.9	1036 (1.00×)	314.58 (1.00×)	3.42 (1.00×)
EVL ViT-B/16 (本文方法, 8)	82.9	454 (0.44×)	102.88 (0.33×)	25.53 (7.47×)

推理延迟和吞吐量。 尽管我们的方法没有专门针对推理速度进行优化，但我们展示了利用大规模预训练模型的重要优势。在小数据集上进行训练需要注入手工编写的归纳偏差，这些偏差不一定对现代加速器友好。相反，ViT 模型几乎完全由标准线性代数运算组成。ViT 的简单性通常能够更高效地利用硬件资源。如表 A.3 所示，延迟和吞吐量甚至比理论 FLOPS 改进更好。

A.5.2 消融实验

我们提供了详细的消融研究，以阐明我们设计的每个部分的影响。除非另有说明，否则我们在 Kinetics-400 数据集上使用 ViT-B/16 骨干网络、8 个输入帧和 3 个测试视图获得结果。

表 A.4 训练时间比较

方法 (每视图的帧数)	Acc (视图数)	预训练	训练的 GPU 时
Uniformer-B (32)	82.9 (4)	ImageNet-1k	$5000 \times V100$
TimeSformer (8)	82.0 (3)	CLIP	$100 \times V100$
ActionCLIP (16)	82.6 (3)	CLIP	$480 \times RTX3090$
EVL ViT-B/16 (8)	82.9 (3)	CLIP	60 × V100

表 A.5 理想化的训练步骤时间。使用 4 个解码器层。所有数据都在单个 V100-16G GPU 上测量。步骤时间是使用 64 个训练样本测量的。

骨干网络	网络头	最大批量	单步迭代耗时 (s)
CLIP (冻结)	全局平均池化	无穷大	0.57
CLIP (可训)	全局平均池化	8	3.39
CLIP (冻结)	EVL	64	1.03
CLIP (可训)	EVL	8	4.41

中间特征。我们改变特征数和 Transformer 解码器层数，并在表A.6(a)和表A.6(b)中呈现结果。利用多个解码器块将准确性提高了 1.0%。将每个解码器块与多层中间特征一起使用可以进一步提高 0.8% 的准确性。另一个观察结果是，深层特征提供了更有效的用于视频识别的特征。

时空特征。我们发现实现高迁移性能的关键设计是使用高分辨率、未池化的特征图。表A.6(c)展示了实验结果，从中可以看出，在时间或空间维度上进行汇总会导致显著的精度下降。我们推测，这表明了任务特定的重新注意力的重要性，例如对于类似 Kinetics-400 的人类动作识别数据集，与人体相关的特征非常重要，这可能与预训练阶段不同。

预训练质量 一个推动从微调到冻结骨干的范式转变的主要因素是预训练模型质量的提高。我们在表A.7中展示了我们的方法在高质量 CLIP 骨干上优于以前的完全微调骨干权重的方法。表中的所有模型使用相同的骨干架构。虽然在 ImageNet-21k 预训练骨干上，我们的方法落后于完全微调，但在 CLIP 骨干上，我们的方法优于

表 A.6 多层高分辨率特征图的影响。 (a) 不同数量的 Transformer 解码器块。 (b) 不同数量的特征图。 (c) 不同的特征分辨率。

(a)		(b)		(c)		
深度	Acc (%)	特征层	Acc (%)	特征尺寸	缩减方式	Acc (%)
1	81.1	$[-4, -3, -2, -1]$	82.9	仅时间	令牌	79.8
2	82.1	$[-2, -2, -1, -1]$	82.7	仅时间	平均	75.8
3	82.6	$[-1, -1, -1, -1]$	82.1	仅空间	平均	80.1
4	82.9	$[-2, -1, -2, -1]$	82.4	时 + 空	-	82.9
5	83.0	$[-7, -5, -3, -1]$	82.0			

竞争的完全微调基线。

我们还发现，尽管我们的模型架构是为冻结骨干设计的，但通过微调骨干，我们的模型架构也可以成为强大的完全微调基线。然而，对于高质量预训练模型，冻结骨干具有更高的训练效率的趋势仍然存在，如图A.3所示。我们的模型架构在 ViT-B/16 上的完全微调产生了类似的效率曲线，但对于更大的 ViT-L/14，达到相同准确率所需的训练时间差距变得更加明显。我们指出，即使 ViT-L/14 按现代标准来看也是一个相对较小的预训练模型，大约有 3 亿个参数（作为比较，用于自然语言处理的 GPT-3 有 1750 亿个参数，用于计算机视觉的 ViT-G 有 18 亿个参数）。我们认为，如果将来发布更大的预训练模型，冻结骨干可能会带来进一步的好处。

A.5.3 时序信息分析

我们方法的一个有趣特性是为视频识别提供了一种分解的方法：空间信息几乎完全编码在固定的高质量 CLIP 骨干中，而时间信息仅编码在 Transformer 解码器头中。如表A.8所示，两个数据集上的时间建模表现有着非常不同的行为：在 Kinetics-400 上，时间模块带来的精度提升不到 0.5%，而在 Something-Something-

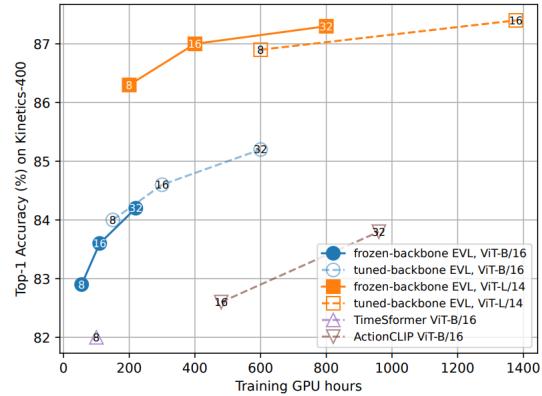


图 A.3 冻结或微调骨干时的训练时间与准确率比较。标记中的数字表示每个视图的帧数。当预训练质量更高时，冻结骨干更加高效。

表 A.7 不同预训练图像特征的结果比较，除非另有说明，否则使用 ViT-B/16 骨干和 8 帧。我们与 TimeSformer 和 ActionCLIP 进行比较。它们都进行了广泛的实验，以确定在视频数据集上进行端到端训练的竞争性设置。

模型	预训练	是否冻结骨干网络	K-400 Acc (%)
TimeSformer - SOnly	ImageNet-21k	✗	76.9
TimeSformer - JointST	ImageNet-21k	✗	77.4
TimeSformer - DividedST	ImageNet-21k	✗	78.0
EVL (本文方法, 8 帧)	ImageNet-21k	✓	75.4
TimeSformer - DividedST	CLIP	✗	82.0
EVL (本文方法, 8 帧)	CLIP	✓	82.9
ActionCLIP (16 帧)	CLIP	✗	82.6
EVL (本文方法, 16 帧)	CLIP	✓	83.3
ActionCLIP (32 帧)	CLIP	✗	83.8
EVL (本文方法, 32 帧)	CLIP	✓	84.2

v2 上，添加时间模块会带来惊人的 +13.8% 的精度提升。这表明了两个基准所需的时间信息之间的明显差异。对于 Kinetics-400，时间信息主要以全局加权特征聚合的形式捕获，如表A.6所示。对于 Something-Something-v2，局部时间特征（例如对象运动、特征变化）也是实现强结果的重要信号来源。

与 Kinetics-400 相比，Something-Something-v2 也更容易从深度解码器中受益。如表A.8(b)所示，Something-Something-v2 受益于使用全部 12 个解码器块，而对于 Kinetics-400，只需要大约 4 个块（见表A.6(a)）。

最后，我们在表A.9中提供了我们在 Something-Something-v2 数据集上的主要结果。虽然 Something-Something-v2 是一个重视运动的数据集，但我们轻量级的时间学习模块仍然学习到了有意义的运动信息，并达到了主流性能（作为比较，CLIP ViT-B/16 的线性探测仅达到约 20% 的准确率）。我们也是第一个基于 CLIP 的方法在 Something-Something-v2 上报告结果，希望这对未来的参考有所帮助。

A.5.4 基于 CLIP 的模型的知识互补性

另一个发现是，我们基于 CLIP 的模型学习到的知识与常规监督学习的知识高度互补。为了展示这一点，我们考虑将我们的模型与监督模型组合起来，观察性

表 A.8 时间信息对视频识别的影响。 (a) 两个数据集的局部时间信息。 *T-Conv*: 时间卷积。*T-PE*: 时间位置嵌入。*T-CA*: 时间交叉注意力。(b) Something-Something-v2 需要更深的解码器块。

(a)				(b)		
T-Conv	T-PE	T-CA	K-400 Acc (%)	SSv2 Acc (%)	深度	SSv2 Acc (%)
✗	✗	✗	82.5	47.2	4	58.6
✓	✗	✗	82.9	57.1	6	60.1
✗	✓	✗	82.5	58.5	8	60.2
✗	✗	✓	82.6	59.5	10	60.5
✓	✓	✗	82.9	59.4	12	61.0
✓	✗	✓	82.7	60.0		
✗	✓	✓	82.7	60.7		
✓	✓	✓	82.9	61.0		

表 A.9 Something-Something-v2 数据集上的主要结果. *Ens* 实验将 EVL 与在 Kinetics-600 上预训练的 *Uniformer-B (32)* 相结合。

Method	SSv2 Acc (%)	帧数	GFLOPS
EVL ViT-B/16	61.0	8×3	512
EVL ViT-B/16	61.7	16×3	1,023
EVL ViT-B/16	62.4	32×3	2,047
EVL ViT-L/14	65.1	8×3	2,411
EVL ViT-L/14	66.7	32×3	9,641
EVL ViT-L/14 (336px)	68.0	32×3	24,259
EVL ViT-B/16 Ens	72.1	$32 \times 3 + 32 \times 3$	2,824

能提升。集成是通过加权平均视频级别的预测分数完成的，平均权重 $\alpha \in [0, 1]$ 在验证集上进行粗略的 0.1 粒度搜索。如表A.10和表A.11所示，对于 Kinetics-400 和 Something-Something-v2，我们一致观察到，如果集成中含有基于 CLIP 的模型，则性能提升更多。

这些集成实验的含义有两个方面。首先，它们表明，在实践中，我们基于 CLIP 的模型可以以双流形式使用。与基于光流的第二流相比，基于 CLIP 的第二流避免

表 A.10 不同组合的集成结果. 我们将具有相似准确度的不同模型结合在一起，并测量准确率的提升。

模型 1	Acc 1	模型 2	Acc 2	模型 1+2 Acc (Δ)
Uniformer-B (16)	82.0	Uniformer-B (32)	82.9	83.6 (+1.6)
		Swin-B	82.7	83.7 (+1.7)
		EVL ViT-B/16 (8)	82.9	84.5 (+2.5)
Swin-B	82.7	Uniformer-B (32)	82.9	84.7 (+2.0)
		EVL ViT-B/16 (8)	82.9	85.0 (+2.3)
Uniformer-B (32)	82.9	Swin-B	82.7	84.7 (+1.8)
		EVL ViT-B/16 (8)	82.9	85.2 (+2.3)

表 A.11 **Something-Something-v2** 数据集上的集成结果. 尽管 *EVL* (32) 的准确率要低得多，但它仍然提升了 *Uniformer-B* 模型的性能。相比之下，准确率略高的 TimeSformer 模型带来的收益微不足道。

模型 1	Acc 1	模型 2	Acc 2	模型 1+2 Acc (Δ)
Uniformer-B (32)	71.2	TimeSformer-L	62.4	71.4 (+0.2)
		EVL ViT-B (32)	62.4	72.1 (+0.9)

了昂贵的光流计算，并且训练速度更快。其次，结果表明，数据集中仍存在我们基于 CLIP 学习范式未捕获的知识。这显示了基于 CLIP 的模型进一步提高的潜力，一旦可以利用数据集中的更多知识。

A.6 结论

我们提出了一种新的视频动作识别流程：在固定可迁移的图像特征之上学习一个高效的迁移学习头。通过冻结图像骨干，大大缩短了训练时间。此外，通过利用来自骨干的多层高分辨率中间特征图，可以在很大程度上弥补因冻结骨干而导致的精度损失。因此，我们的方法有效地利用了强大的图像特征进行视频识别，同时避免了对非常大的图像模型进行沉重或禁止的完全微调。我们进一步表明，在开放世界环境中学习的可迁移图像特征具有高度互补的知识，这可能激发更有效的方法来构建最先进的视频模型。我们相信我们的观察结果有潜力使视频识别更加普及，并以更有效的方式推动视频模型的最新技术发展。

书面翻译对应的原文索引

- [1] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 388–404. Springer, 2022.

附录 B 数据集标签划分

B.1 Kinetics

训练集标签:

'air drumming', 'arm wrestling', 'beatboxing', 'biking through snow', 'blowing glass', 'blowing out candles', 'bowling', 'breakdancing', 'bungee jumping', 'catching or throwing baseball', 'cheerleading', 'cleaning floor', 'contact juggling', 'cooking chicken', 'country line dancing', 'curling hair', 'deadlifting', 'doing nails', 'dribbling basketball', 'driving tractor', 'drop kicking', 'dying hair', 'eating burger', 'feeding birds', 'giving or receiving award', 'hopscotch', 'jetskiing', 'jumping into pool', 'laughing', 'making snowman', 'massaging back', 'mowing lawn', 'opening bottle', 'playing accordion', 'playing badminton', 'playing basketball', 'playing didgeridoo', 'playing ice hockey', 'playing keyboard', 'playing ukulele', 'playing xylophone', 'presenting weather forecast', 'punching bag', 'pushing cart', 'reading book', 'riding unicycle', 'shaking head', 'sharpening pencil', 'shaving head', 'shot put', 'shuffling cards', 'slacklining', 'sled dog racing', 'snowboarding', 'somersaulting', 'squat', 'surfing crowd', 'trapezing', 'using computer', 'washing dishes', 'washing hands', 'water skiing', 'waxing legs', 'weaving basket'

验证集标签:

'baking cookies', 'crossing river', 'dunking basketball', 'feeding fish', 'flying kite', 'high kick', 'javelin throw', 'playing trombone', 'scuba diving', 'skateboarding', 'ski jumping', 'trimming or shaving beard'

测试集标签:

'blasting sand', 'busking', 'cutting watermelon', 'dancing ballet', 'dancing charleston', 'dancing macarena', 'diving cliff', 'filling eyebrows', 'folding paper', 'hula hooping', 'hurling (sport)', 'ice skating', 'paragliding', 'playing drums', 'playing monopoly', 'playing trumpet', 'pushing car', 'riding elephant', 'shearing sheep', 'side kick', 'stretching arm', 'tap dancing', 'throwing axe', 'unboxing'

B.2 SSV2-Small

训练集标签:

'Pouring something into something', 'Poking a stack of something without the stack collapsing', 'Pretending to poke something', 'Lifting up one end of something without letting it drop down', 'Moving part of something', 'Moving something and something away from each other', 'Removing something, revealing something behind', 'Plugging something into something', 'Tipping something with something in it over, so something in it falls out', 'Stacking number of something', "Putting something onto a slanted surface but it doesn't glide down", 'Moving something across a surface until it falls down', 'Throwing something in the air and catching it', 'Putting something that cannot actually stand upright upright on the table, so it falls on its side', 'Holding something next to something', 'Pretending to put something underneath something', "Poking something so lightly that it doesn't or almost doesn't move", 'Approaching something with your camera', 'Poking something so that it spins around', 'Pushing something so that it falls off the table', 'Spilling something next to something', 'Pretending or trying and failing to twist something', 'Pulling two ends of something so that it separates into two pieces', 'Lifting up one end of something, then letting it drop down', "Tilting something with something on it slightly so it doesn't fall down", 'Spreading something onto something', 'Touching (without moving) part of something', 'Turning the camera left while filming something', 'Pushing something so that it slightly moves', 'Uncovering something', 'Moving something across a surface without it falling down', 'Putting something behind something', 'Attaching something to something', 'Pulling something onto something', 'Burying something in something', 'Putting number of something onto something', 'Letting something roll along a flat surface', 'Bending something until it breaks', 'Showing something behind something', 'Pretending to open something without actually opening it', 'Pretending to put something onto something', 'Moving away from something with your camera', 'Wiping something off of something', 'Pretending to spread air onto something', 'Holding something over something', 'Pretending or failing to wipe something off of something', 'Pretending to put something on a surface', 'Moving something and something so they collide with each other', 'Pretending to turn something upside down', 'Showing something to the camera', 'Dropping something onto something', "Pushing something so that it almost falls off but doesn't", 'Piling something up', 'Taking one of many similar things on the table', 'Putting something in front of something', 'Laying something on the table on its side, not upright', 'Lifting a surface with something on it until it starts sliding down',

'Poking something so it slightly moves', 'Putting something into something', 'Pulling something from right to left', 'Showing that something is empty', 'Spilling something behind something', 'Letting something roll down a slanted surface', 'Holding something behind something'

验证集标签:

'Lifting something up completely without letting it drop down', 'Pouring something into something until it overflows', 'Putting something, something and something on the table', 'Trying to bend something unbendable so nothing happens', 'Pouring something out of something', 'Throwing something onto a surface', 'Putting something onto something else that cannot support it so it falls down', 'Pretending to pour something out of something, but something is empty', 'Pulling something out of something', 'Holding something in front of something', 'Tilting something with something on it until it falls off', 'Moving something away from the camera'

测试集标签:

'Twisting (wrapping) something wet until water comes out', 'Poking a hole into something soft', 'Pretending to take something from somewhere', 'Putting something upright on the table', 'Poking a hole into some substance', 'Rolling something on a flat surface', 'Poking a stack of something so the stack collapses', 'Twisting something', 'Something falling like a feather or paper', 'Putting something on the edge of something so it is not supported and falls down', 'Pushing something off of something', 'Dropping something into something', 'Letting something roll up a slanted surface, so it rolls back down', 'Pushing something with something', 'Opening something', 'Putting something on a surface', 'Taking something out of something', 'Spinning something that quickly stops spinning', 'Unfolding something', 'Moving something towards the camera', 'Putting something next to something', 'Scooping something up with something', 'Squeezing something', 'Failing to put something into something because something does not fit'

B.3 SSV2-Full

训练集标签:

'Showing a photo of something to the camera', 'Holding something', "Putting something onto a slanted surface but it doesn't glide down", 'Lifting a surface with something

on it but not enough for it to slide down', 'Covering something with something', 'Tearing something into two pieces', 'Pulling two ends of something so that it separates into two pieces', 'Dropping something in front of something', 'Turning the camera upwards while filming something', 'Lifting something with something on it', 'Moving something and something closer to each other', 'Pretending to close something without actually closing it', "Tilting something with something on it slightly so it doesn't fall down", 'Moving away from something with your camera', 'Putting something and something on the table', 'Pulling something onto something', 'Putting number of something onto something', 'Holding something next to something', 'Tilting something with something on it until it falls off', 'Dropping something behind something', 'Moving something and something so they collide with each other', 'Pushing something so that it slightly moves', 'Pouring something onto something', 'Spilling something next to something', 'Tipping something with something in it over, so something in it falls out', 'Dropping something into something', 'Moving something up', 'Plugging something into something', 'Moving something across a surface until it falls down', 'Pushing something off of something', 'Something colliding with something and both are being deflected', 'Stuffing something into something', 'Spinning something that quickly stops spinning', 'Closing something', 'Poking something so that it falls over', 'Bending something until it breaks', 'Moving something down', 'Taking something from somewhere', 'Spreading something onto something', 'Turning the camera left while filming something', 'Pretending to take something out of something', "Poking something so lightly that it doesn't or almost doesn't move", 'Pouring something into something', 'Showing something behind something', 'Poking a hole into something soft', 'Pulling two ends of something so that it gets stretched', 'Folding something', 'Twisting (wringing) something wet until water comes out', 'Pretending to be tearing something that is not tearable', 'Pouring something into something until it overflows', 'Pushing something so that it falls off the table', 'Putting something similar to other things that are already on the table', 'Pretending to put something next to something', 'Uncovering something', 'Spilling something onto something', 'Squeezing something', 'Letting something roll along a flat surface', 'Pushing something from left to right', 'Pretending to pick something up', 'Putting something onto something', 'Twisting something', 'Turning the camera downwards while filming something', 'Letting something roll down a slanted surface', 'Pretending to spread air onto something'

验证集标签:

'Plugging something into something but pulling it right out as you remove your hand', 'Scooping something up with something', 'Lifting something up completely, then letting it drop down', 'Pushing something so it spins', 'Holding something over something', 'Taking one of many similar things on the table', 'Showing something to the camera', 'Lifting up one end of something, then letting it drop down', 'Throwing something in the air and catching it', 'Putting something in front of something', 'Pretending or failing to wipe something off of something', 'Pretending to pour something out of something, but something is empty'

测试集标签:

'Pretending to open something without actually opening it', 'Pretending to put something behind something', 'Poking a stack of something without the stack collapsing', 'Digging something out of something', 'Pouring something out of something', 'Pulling something from left to right', 'Pretending to put something underneath something', 'Picking something up', 'Removing something, revealing something behind', 'Taking something out of something', 'Pushing something from right to left', 'Failing to put something into something because something does not fit', 'Pulling something out of something', 'Spilling something behind something', 'Showing something next to something', 'Showing that something is empty', 'Pretending to put something into something', 'Dropping something onto something', 'Pretending to sprinkle air onto something', 'Lifting up one end of something without letting it drop down', 'Dropping something next to something', 'Throwing something in the air and letting it fall', 'Tipping something over', 'Approaching something with your camera'

B.4 HMDB51

训练集标签:

'brush_hair', 'catch', 'chew', 'clap', 'climb', 'climb_stairs', 'dive', 'draw_sword', 'dribble', 'drink', 'fall_floor', 'flic_flac', 'handstand', 'hug', 'jump', 'kiss', 'pullup', 'punch', 'push', 'ride_bike', 'ride_horse', 'shake_hands', 'shoot_bow', 'situp', 'stand', 'sword', 'sword_exercise', 'throw', 'turn', 'walk', 'wave'

验证集标签:

'cartwheel', 'eat', 'golf', 'hit', 'laugh', 'shoot_ball', 'shoot_gun', 'smile', 'somersault', 'swing_baseball'

测试集标签:

'fencing', 'kick', 'kick_ball', 'pick', 'pour', 'pushup', 'run', 'sit', 'smoke', 'talk'

B.5 UCF101

训练集标签:

'ApplyEyeMakeup', 'Archery', 'BabyCrawling', 'BalanceBeam', 'BandMarching', 'BaseballPitch', 'Basketball', 'BasketballDunk', 'BenchPress', 'Biking', 'Billiards', 'BlowDryHair', 'BodyWeightSquats', 'Bowling', 'BoxingPunchingBag', 'BoxingSpeedBag', 'BreastStroke', 'BrushingTeeth', 'CricketBowling', 'Drumming', 'Fencing', 'FieldHockeyPenalty', 'FrisbeeCatch', 'FrontCrawl', 'Haircut', 'Hammering', 'HeadMassage', 'HulaHoop', 'JavelinThrow', 'JugglingBalls', 'JumpingJack', 'Kayaking', 'Knitting', 'LongJump', 'Lunges', 'MilitaryParade', 'Mixing', 'MoppingFloor', 'Nunchucks', 'ParallelBars', 'PizzaTossing', 'PlayingCello', 'PlayingDhol', 'PlayingFlute', 'PlayingPiano', 'PlayingSitar', 'PlayingTabla', 'PlayingViolin', 'PoleVault', 'PullUps', 'PushUps', 'Rafting', 'RopeClimbing', 'Rowing', 'ShavingBeard', 'Skijet', 'SoccerJuggling', 'SoccerPenalty', 'SumoWrestling', 'Swing', 'TableTennisShot', 'TaiChi', 'ThrowDiscus', 'TrampolineJumping', 'Typing', 'UnevenBars', 'WalkingWithDog', 'WallPushups', 'WritingOnBoard', 'YoYo'

验证集标签:

'ApplyLipstick', 'CricketShot', 'HammerThrow', 'HandstandPushups', 'HighJump', 'HorseRiding', 'PlayingDaf', 'PlayingGuitar', 'Shotput', 'SkateBoarding'

测试集标签:

'BlowingCandles', 'CleanAndJerk', 'CliffDiving', 'CuttingInKitchen', 'Diving', 'FloorGymnastics', 'GolfSwing', 'HandstandWalking', 'HorseRace', 'IceDancing', 'JumpRope', 'PommelHorse', 'Punch', 'RockClimbingIndoor', 'SalsaSpin', 'Skiing', 'SkyDiving', 'StillRings', 'Surfing', 'TennisSwing', 'VolleyballSpiking'

B.6 Diving48

训练集标签:

0, 1, 2, 3, 7, 8, 10, 13, 15, 17, 20, 21, 22, 23, 24, 25, 26, 29, 32, 34, 35, 37, 38, 41, 42, 45, 46

验证集标签:

4, 11, 14, 16, 28, 31, 33, 36, 39, 47

测试集标签:

5, 6, 9, 12, 18, 19, 27, 40, 43, 44

B.7 Gym99

训练集标签:

0, 2, 3, 4, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 37, 38, 40, 41, 43, 44, 46, 47, 48, 49, 53, 55, 58, 59, 68, 70, 71, 72, 73, 74, 75, 78, 80, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94

验证集标签:

10, 19, 32, 56, 57, 62, 66, 69, 79, 82, 95, 96

测试集标签:

1, 5, 8, 18, 20, 31, 39, 42, 45, 50, 51, 52, 54, 60, 61, 63, 64, 65, 67, 76, 77, 81, 97, 98

综合论文训练记录表

学生姓名	王逸钦	学号	2019010850	班级	自 93	
论文题目	基于图文对预训练权重的小样本视频理解					
主要内容以及进度安排	<p>论文研究的主要内容为设计一种利用图文对预训练知识的小样本视频理解算法，算法能够挖掘视频时序特征。该算法相比已有方法应在有效性、高效性、普适性等方面有实质性提升，并在多个主流数据集上进行定量实验证。论文的主要进度安排如下：</p> <p>2022 年 10-11 月：进行文献调研、前期准备工作；</p> <p>2022 年 12 月：探索最优的迁移学习范式；</p> <p>2023 年 1 月：确定基线方法和骨干网络结构，并开始探索解码器结构设计；</p> <p>2023 年 2-3 月：用定量实验确定全部结构设计与参数设定；</p> <p>2023 年 4 月：完成消融实验以论证各模块有效性，进行可视化分析；</p> <p>2023 年 5 月：论文撰写。</p>					
	指导教师签字: <u>周生</u> 考核组组长签字: <u>冯建江</u> 2022 年 11 月 24 日					
	中期考核意见	论文工作进展良好，达到了中期目标。				
		考核组组长签字: <u>冯建江</u> 2023 年 3 月 16 日				

指导教师评语	<p>论文研究基于多模态预训练权重，设计了一种小样本的视频理解方法，选题具有重要的理论意义与应用价值。论文提出了“视频旁路聚合器”结构，通过融合跨视频、跨帧与跨粒度特征，验证了方法的有效性。论文工作量大，写作规范，叙述清楚，达到了综合论文训练的要求。</p> <p>指导教师签字:  2023年5月28日</p>
评阅教师评语	<p>论文研究基于小样本的视频分类，选题具有理论意义和应用价值。论文提出融合多帧多粒度信息，提高了分类性能。论文写作规范，工作量大，达到了综合论文训练的要求。</p> <p>评阅教师签字:  2023年5月28日</p>
答辩小组评语	<p>论文选题具有理论意义和应用价值。回答问题正确，工作达到了综合论文训练的要求。</p> <p>答辩小组组长签字:  2023年6月1日</p>

总成绩: B+
教学负责人签字: 13484

2023年6月12日