

# Tip-Adapter:CLIP的无训练适应少枪分类

张仁瑞\*<sup>1,2</sup>, 张伟\*<sup>1</sup>, 方荣耀<sup>2</sup>, 高鹏<sup>†</sup><sup>1</sup>, 李坤昌<sup>1</sup>, 戴继峰<sup>3</sup>, 于桥<sup>1</sup>, 李洪生<sup>2,4</sup>

<sup>1</sup>上海人工智能实验室

<sup>2</sup>香港中文大学

<sup>3</sup>商汤科技研究

<sup>4</sup>感知与交互智能中心(CPII)

{张仁瑞, 高鹏, 乔玉}@pjlab.org.cn, hsl@ee.cuhk.edu.hk

摘要对比视觉语言预训练, 被称为CLIP, 为使用大规模图像-文本对学习视觉表征提供了一种新的范例。它通过零样本知识迁移在下游任务上表现出令人印象深刻的性能。为了进一步提高CLIP的自适应能力, 现有方法提出对附加可学习模块进行微调, 这大大提高了少射性能, 但引入了额外的训练时间和计算资源。在本文中, 我们提出了一种CLIP进行少弹分类的无训练自适应方法, 称为Tip-Adapter, 它不仅继承了零弹CLIP无需训练的优点, 而且具有与需要训练的方法相当的性能。Tip-Adapter算法通过基于少拍训练集的键值缓存模型构建适配器, 并通过特征检索更新CLIP中编码的先验知识。最重要的是, Tip-Adapter的性能可以进一步提升到最先进的ImageNet上, 通过微调缓存模型, 比现有方法少10倍的epoch, 这是既有效又高效的。在11个数据集上进行了广泛的少样本分类实验, 以证明所提出方法的优越性。代码发布在<https://github.com/gaopengcuhk/Tip-Adapter>。

关键词:视觉-语言学习, 少样本分类, 缓存模型

## 1 介绍

视觉和语言是人类感知周围世界和与环境进行多样化交互的两种模态。由于更好的神经架构设计[22,59]和精心设计的框架[51,37,5,7,72], 视觉任务, 如分类[35,22,26,13,42,17,71]、检测[51,5,73,65,70,9]和3D理解[47,69,64,68]的精度得到了显著提升。

\* Indicates equal contributions, † Indicates corresponding author

表1. 在16张照片ImageNet上比较不同方法的分类精度(%)和时间效率[10], 其中我们提出的Tip-Adapter和Tip-Adapter-F实现了更好的精度-效率权衡。所有实验都在单个NVIDIA GeForce RTX 3090 GPU上以批处理大小32进行测试。蓝色列记录了相对于零弹CLIP的性能增益。

Models	Training	Epochs	Time	Accuracy	Gain	Infer. Speed	GPU Mem.
Zero-shot CLIP [48]	Free	0	0	60.33	0	10.22ms	2227MiB
Linear-probe CLIP [48]	Required	-	13min	56.13	-4.20	-	-
CoOp [74]	Required	200	14h 40min	62.95	+2.62	299.64ms	7193MiB
CLIP-Adapter [16]	Required	200	50min	63.59	+3.26	10.59ms	2227MiB
Tip-Adapter	<b>Free</b>	<b>0</b>	<b>0</b>	62.03	+1.70	10.42ms	2227MiB
Tip-Adapter-F	Required	20	5min	<b>65.51</b>	+5.18	10.53ms	2227MiB

由于大规模的自监督方法, 包括通过掩码预测[11]和收集的网络规模数据[49]进行预训练, 有关生成和理解的语言任务也得到了很大的改善。由于视觉和语言通常包含互补信息, 多模态表示的联合学习已被证明在各种任务上相当有效, 例如视觉问答[2,1,31], 图像描述[66,27], 以及参考表达[67]。与以往在独立数据集上独立学习视觉和语言表征的方法不同[1,40,56], CLIP[48]提出了从成对的自然语言监督中学习可转移的视觉特征, 并发挥了惊人的零镜头图像分类能力。由于语言和视觉之间的相互作用, 编码后的视觉表示可以用于开放词汇表识别, 而无需进一步重新训练。

许多后续研究都提出利用少射数据来提高CLIP对下游任务的适应能力。CoOp[74]遵循提示设计的方向[4,38], 通过可学习的文本标记对预训练的CLIP进行微调, 在少拍图像分类上取得了较强的性能。最近, CLIP-adapter[16]引入了参数化特征适配器, 为CLIP配置参数化特征适配器, 该参数化特征适配器生成自适应特征, 并通过残差连接将其与原始的CLIP编码特征结合。在不利用提示设计的情况下, 它在少样本分类方面表现出了有希望的性能。虽然CoOp[74]和CLIP-adapter[16]在少弹分类基准上显示出强大的能力, 但与零弹CLIP[48]和线性探针CLIP[48]相比, 它们需要更多的计算资源来微调新引入的可学习参数。因此, 我们提出了以下问题: 我们能否做到两全其美, 既能利用CLIP在零弹分类时无需训练的特性, 又能在少弹分类时拥有需要训练的方法的强大性能?

为了实现这一目标, 我们提出了一种CLIP的无训练自适应方法, 称为Tip-Adapter, 该方法在权重冻结的CLIP模型上附加了一个新的非参数适配器。与现有方法不同, 我们的方法不需要额外的训练, 而是将适配器设计为来自少样本数据集的查询键缓存模型[30,45,18]。具体来说, Tip-Adapter提取的视觉特征

通过CLIP的视觉编码器对少拍图像进行编码，并将其对应的标签转换为单热编码。然后，创建一个包含少次视觉特征和one-hot标签的缓存模型，它们被视为配对的键和值。

通过缓存模型，无需训练的Tip-Adapter构造比传统的随机梯度去气味(Stochastic Gradient Descent, SGD)微调具有更高的效率[32,39]。在推理过程中，测试图像首先计算其与缓存键的特征相似度，然后聚合缓存值形成适配器的预测，这可以视为从缓存模型检索少样本的知识。之后，通过残差连接将适配器的预测与原始CLIP的预测结合起来[22]。通过这种方式，Tip-Adapter同时利用预训练的CLIP和少量训练数据集的知识。令人惊讶的是，未经训练的Tip-Adapter的性能可以与经过微调的CoOp和CLIP-Adapter相媲美。此外，如果我们解冻缓存的键作为可学习的参数，并进一步微调它们，Tip-Adapter的性能可以在几个训练周期内得到显著提高。我们将这种微调版本称为Tip-Adapter- f，与CoOp和CLIP-Adapter采用的200个epoch相比，它只需要在ImageNet[10]上使用20个epoch就可以达到最先进的水平。在表1中，我们列出了所有现有方法在ImageNet上对16张照片分类的性能、训练时间和推理速度的比较，这表明我们的方法在精度和效率上有很大的权衡。

我们论文的贡献总结如下：

1. 我们提出了一种Tip-Adapter方法，这是一种无需训练的CLIP自适应方法，它通过直接使用缓存模型设置适配器来放弃传统的基于sgd的训练。
2. 将缓存模型的关键字解冻为可学习的参数，经过微调的Tip-Adapter，命名为Tip-Adapter- f，在ImageNet上以超快的收敛速度实现了最先进的性能。
3. 我们在11个广泛采用的数据集上对Tip-Adapter和Tip-Adapter- f进行了评估，并进行了广泛的消融研究，以证明它们的特性。

## 2 相关工作

数据高效的迁移学习。深度神经网络的能力在大规模高质量数据集[35]的辅助下得以展现。然而，由于长尾分布、噪声注释和不断增加的标记成本，收集这样的数据是具有挑战性和昂贵的。因此，迁移学习被提出来缓解这个问题，这已经成为一个热门的研究领域。图像分类[10]上的监督预训练已经被广泛采用作为下游任务微调的默认基础(例如检测[51]和分割[21])。自监督学习，如MoCo[20]和BYOL[19]，进一步抛弃了对监督信号的需求，并构建了一个用于鲁棒特征学习的对比性前置任务。最近，CLIP[48]、DeCLIP[36]和ALIGN [28]

证明从简单的对比视觉-语言对中学习,可以在不同的数据集上获得有希望的可迁移特征,用于零样本识别。除此之外,CoOp[74]、CLIP-Adapter[16]和WiSE-FT[61]通过冻结预训练权值和训练加性可学习模块,显著改善了训练数据有限的CLIP。相比之下,我们提出的Tip-Adapter旨在以无训练的方式直接将少镜头监督注入预训练的CLIP模型中。这样,Tip-Adapter的构造在时间和内存上都大大提高了效率,它只需要计算一次少量训练集的特征,然后缓存它们。

缓存模型。缓存模型将训练图像的特征及其标签存储为键值数据库。在推理过程中,从测试样本编码的特征被视为查询,通过基于相似性的检索[59]来聚合来自缓存模型的信息。整个过程是非参数[33],不涉及参数更新。缓存模型已经装备在各种模型上,以提高视觉或语言模型的性能,包括kNN-LMs[30]、无界缓存[18]、匹配网络[60]等[43,53]。虽然简单的缓存模型[45]已经显示出了有希望的结果,但训练数据的巨大存储预算在许多应用程序无法承受的。为了降低这样的成本,人们提出了基于高度优化的相似性搜索系统[29]的近似kNN,然而这种方法速度慢且容易出错。与以往的纯视觉或语言缓存不同,我们利用CLIP的对比多模态预训练构建了混合视觉语言缓存模型。重要的是,由于我们的训练样本有限的少样本设置,总缓存大小较小,并且可以通过两个级联矩阵乘法有效地计算检索。此外,Tip-Adapter中的缓存模型可以通过随机梯度下降(SGD)进行学习 and 动态更新,从而进一步提高了其性能。

### 3 方法

在3.1节中,我们首先介绍了我们提出的无需训练的Tip-Adapter及其微调变体Tip-Adapter-F。然后在3.2节中,我们将讨论我们的方法与之前的方法之间的关系,例如CLIP-Adapter和基于缓存的网络。

#### 3.1 免培训的CLIP改编

为了提高CLIP的少镜头分类性能,我们提出了一种无需训练的Tip-Adapter自适应方法。我们以非参数的方式从少样本训练集构建了一个键值缓存模型。令人惊讶的是,使用这种设计良好的缓存模型,无需微调的Tip-Adapter可以达到与那些需要训练的方法(包括CoOp[74]和CLIP-Adapter[16])相当的性能。此外,如果允许训练,Tip-Adapter-F通过微调高速收敛的缓存键来进一步超越最先进的性能。

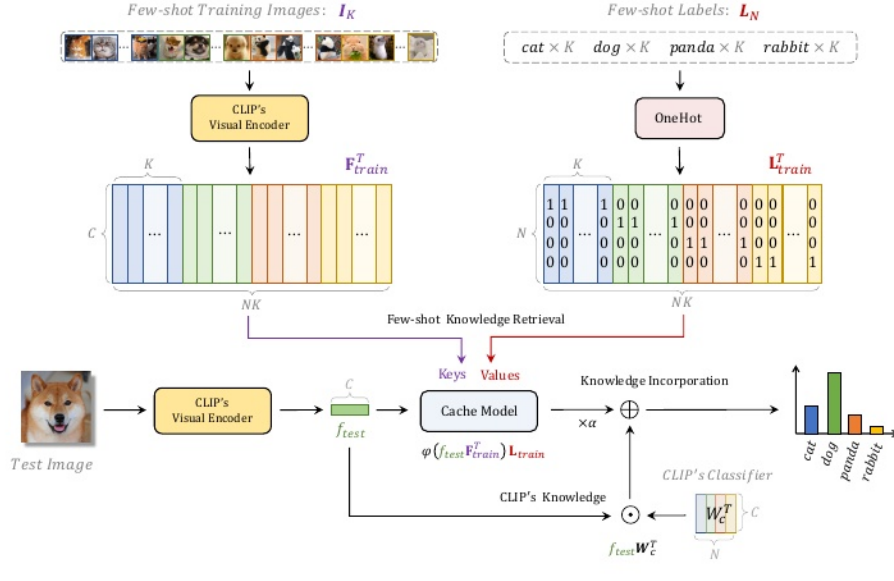


图1所示。Tip-Adapter的管道。给定一个K-shot n类训练集，我们构建了一个缓存模型来适应下游任务的CLIP。它包含由CLIP编码的少数镜头视觉特征 $F_{train}^T$ 及其在one-hot编码下的ground-truth标签 $L_{train}^T$ 。从缓存模型中检索后，将少拍知识与CLIP的预训练知识相结合，实现无训练自适应。

缓存模型构建。给定预训练的CLIP[48]模型和一个具有K-shot n类训练样本的新数据集，用于少镜头分类，N个类别中每个类别有K个带注释的图像，记为 $I_K$ ，其标签为 $L_N$ 。我们的目标是创建一个key-value缓存模型作为特征适配器，其中包含N类内的少样本知识。对于每个训练图像，我们利用CLIP预训练的视觉编码器提取其c维L2归一化特征，并将其真值标签转换为n维单热向量。对于所有NK训练样本，我们将其视觉特征和相应的标签向量表示为 $F_{train} \in \mathbb{R}^{NK \times C}$ 和 $L_{train} \in \mathbb{R}^{NK \times N}$ ,

$$F_{train} = \text{VisualEncoder}(I_K), \quad (1)$$

$$L_{train} = \text{OneHot}(L_N). \quad (2)$$

对于键-值缓存，将clip编码表示 $F_{train}$ 视为键，而将one-hot ground-truth向量 $L_{train}$ 用作其值。这样，键值缓存存储了从少拍训练集中提取的所有新知识，用于更新预训练CLIP中编码的先验知识。

Tip-Adapter。在构建缓存模型后，CLIP的自适应可以简单地通过两次矩阵-向量乘法来实现。在推理过程中，首先由CLIP的视觉编码器提取测试图像的L2归一化特征 $f_{\text{test}} \in \mathbb{R}^1 \times \mathbb{C}$ ，并作为从键值缓存中检索的查询。查询和键之间的亲和度可以估计为

$$A = \exp(-\beta(1 - f_{\text{test}} \mathbf{F}_{\text{train}}^T)), \quad (3)$$

其中 $A \in \mathbb{R}^{1 \times N_K}$ 和 $\beta$ 代表一个调制超参数。由于查询和关键特征都是L2归一化的，术语 $f_{\text{test}} \mathbf{F}_{\text{train}}^T$ 相当于测试特征 $f_{\text{test}}$ 和所有Few-shot训练特征 $\mathbf{F}_{\text{train}}^T$ 之间的余弦相似度。采用指数函数将相似度转换为非负值， $\beta$ 调制其锐度。然后，通过查询-键亲和度加权的缓存值的线性组合得到缓存模型的预测，记为 $\mathbf{A} \mathbf{L}_{\text{train}} \in \mathbb{R}^1 \times \mathbb{N}$ 。

除了从缓存模型中检索到的Few-shot知识外，通过 $f_{\text{test}} \mathbf{W}_c^T \in \mathbb{R}^1 \times \mathbb{N}$ 计算预训练CLIP的先验知识边缘，其中 $\mathbf{W}_c$ 为预训练的文本编码器生成的CLIP分类器的权重。通过残差连接混合两种预测，Tip-Adapter测试图像的输出logits被计算为

$$\begin{aligned} \text{logits} &= \alpha \mathbf{A} \mathbf{L}_{\text{train}} + f_{\text{test}} \mathbf{W}_c^T \\ &= \alpha \varphi(f_{\text{test}} \mathbf{F}_{\text{train}}^T) \mathbf{L}_{\text{train}} + f_{\text{test}} \mathbf{W}_c^T, \end{aligned} \quad (4)$$

其中 $\alpha$ 表示残差比率，我们定义 $\phi(x) = \exp(-\beta(1-x))$ 。因此，Tip-Adapter的预测包含两项，前一项自适应总结来自Few-shot训练数据集的信息，后一项保留来自CLIP分类器 $\mathbf{W}_c^T$ 的先验知识。这两个项通过权重 $\alpha$ 进行平衡。经验上，如果预训练和下游Few-shot任务之间的领域差距很大， $\alpha$ 被设置为大，因为需要更多来自Few-shot集的知识，否则 $\alpha$ 被设置为小。

Tip-Adapter与微调。Tip-Adapter可以大大提高CLIP通过纳入新的知识在Few-shot训练集。然而，经过更多的射击，未经训练的Tip-Adapter逐渐落后于需要训练的CoOp和CLIP-Adapter。为了在保持效率的同时减小这种差距，我们提出了Tip-Adapter-F，它将缓存模型中的键视为可学习参数的良好初始化，并通过SGD对它们进行微调。得益于缓存模型的有利起点，与CoOp和CLIP-Adapter的200 epoch训练相比，Tip-Adapter-F在ImageNet上只需要20 epoch微调就能达到最先进的性能[10]。

更具体地说，我们解冻了缓存的键 $\mathbf{F}_{\text{train}}$ ，但仍然冻结了预训练CLIP的值 $\mathbf{L}_{\text{train}}$ 和两个编码器。直观的感觉是，更新缓存模型中的密钥可以提升亲和性的估计，从而能够更准确地计算测试图像和训练图像之间的余弦相似度。相比之下，缓存模型中的值是表示ground-truth注释的one-hot编码，应保持冻结以很好地记忆类别信息。

### 3.2 与以前模型的关系

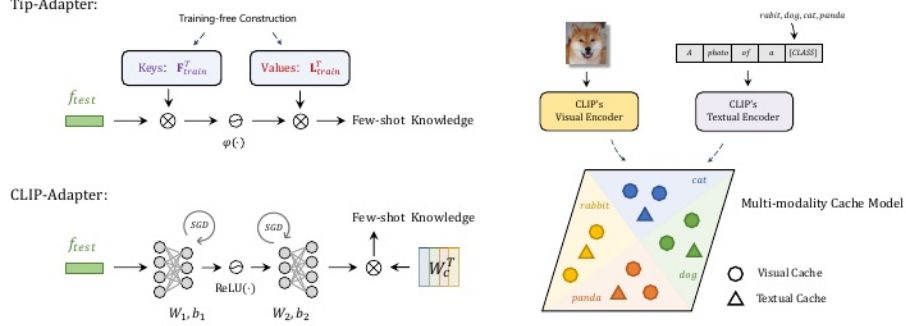


图2所示。Tip-Adapter与CLIP-Adapter的比较[16]获取 Few-shot 知识。Tip-Adapter 从构造的缓存模型中检索，但 CLIP-Adapter 通过可学习适配器对知识进行编码，并在 CLIP 的分类器  $W_c$  的帮助下获得知识。

图3所示。Tip-Adapter 的多模态缓存模型。与以往仅具有视觉缓存的网络不同，Tip-Adapter 通过 CLIP 的编码器同时缓存视觉和文本知识。

与 CLIP-Adapter 的关系。继神经语言处理中的适配器[25]之后，CLIP-Adapter[16]在预训练的权值固定的 CLIP 模型中附加了一个轻量级的两层多层 Perceptron (MLP)，并通过 SGD 优化其参数。具体而言，对于输入测试图像，首先通过 CLIP 预训练的视觉编码器获得其视觉特征  $f_{\text{test}}$ 。然后，附加随机初始化参数为  $W_1, b_1, W_2, b_2$  的基于 mlp 的适配器，输出适应后的特征，

$$f_{\text{test}}^a = \varphi(f_{\text{test}} W_1^T + b_1) W_2^T + b_2, \quad (5)$$

式中  $\varphi$  为 MLP 中的激活函数。然后，将自适应特征  $f_{\text{test}}^a$  与预训练的 CLIP 特征  $f_{\text{test}}$  线性组合，输出具有超参数  $\alpha \in [0, 1]$  的最终分类逻辑，

$$\text{logits} = \alpha f_{\text{test}}^a W_c^T + f_{\text{test}} W_c^T, \quad (6)$$

其中  $W_c^T$  为 CLIP 分类器的权重。两个等式的第一项。(4) 和 (6) 分别表示 Tip-Adapter 和 CLIP-Adapter 获取 Few-shot 知识的方法。如图 2 所示，Tip-Adapter 通过从缓存模型中检索获取知识，但 CLIP-Adapter 首先利用可学习的适配器来预测适应的特征，然后将其与 CLIP 的  $W_c^T$  相乘，形成最终的知识输出。

通过对等式的进一步分析。(4) 和 (6)，CLIP-Adapter 可以看作是我们提出的 Tip-Adapter 的一种特殊形式；

$$W_1 = F_{\text{train}}, W_2 = L_{\text{train}}^T W_c^{-1}, b_1 = 0, b_2 = 0, \quad (7)$$

$$\varphi(x) = \exp(-\beta(1-x)), \text{ where } x \in [0, 1]. \quad (8)$$

它们有两个关键的区别。首先，CLIP-Adapter随机初始化缓存模型中的键和值分别为 $W_1$ 和 $W_2$ ，并通过SGD进行学习，而Tip-Adapter直接使用缓存的训练特征 $F_{\text{train}}$ 和非参数、无需训练的ground-truth标签的单热编码 $L_{\text{train}}$ 构造它们。其次，Tip-Adapter的瓶颈维数等于NK，而CLIP-Adapter为了防止训练导致的过拟合，选择了一个更低维的瓶颈。这表明我们的缓存模型可以更好地缓解Few-shot数据集的过拟合问题，进一步释放大规模预训练模型的拟合能力。第三，Tip-Adapter引入Eq.(7)所示的激活函数。由于它的输入是归一化特征空间中的距离，因此它自然在0和1之间有界。但是，对于CLIP-Adapter，选择通用激活函数 $\text{ReLU}(\cdot)$ 来处理无界输入。简而言之，Tip-Adapter在没有经过训练的情况下得到了一个性能良好的适配器，在Few-shot分类上效率更高。

与基于缓存的网络的关系。从Few-shot训练数据中获取缓存模型已经被许多先前的方法所探索，包括匹配网络[60]、原型网络[53]、MAML[15]、关系网络[55]等[12,7,57,6]。我们的模型在特定的方法和实验设置上都有两点不同于它们。

首先，以往的工作只构建了视觉特征的缓存，而Tip-Adapter采用了一种多模态异构缓存模型，通过CLIP提取视觉和文本缓存特征，如图3所示。具体来说，前面提到的键为 $F_{\text{train}}$ 、值为 $L_{\text{train}}$ 的缓存模型作为视觉缓存，这里记为 $F_{\text{vis}}$ 和 $L_{\text{vis}}$ 。由于CLIP的分类器 $W_c$ 是由文本编码器从类别文本中计算出来的，因此 $W_c \in \mathbb{R}^{N \times C}$ 可以看作是作为文本缓存键 $F_{\text{tex}}$ 的语言特性。然后，文本缓存的值用单位矩阵 $L_{\text{tex}} \in \mathbb{R}^{N \times N}$ 表示，因为 $W_c$ 分别对N个类别知识进行编码，其每个行向量对应于某个类别。从这个角度出发，将Eq.(4)重新表述为

$$\text{logits} = \alpha \varphi(f_{\text{test}} \mathbf{F}_{\text{vis}}^T) \mathbf{L}_{\text{vis}} + (f_{\text{test}} \mathbf{F}_{\text{tex}}^T) \mathbf{L}_{\text{tex}}, \quad (9)$$

其中，两个术语表示从视觉和文本缓存的知识中进行知识检索。

其次，之前的工作将相同的数据集划分为不同类别的三个子集，分别作为训练集、支持集和查询集。尽管他们用一组新的类别在查询集上进行测试，但它仍然在同一个语义域内。相比之下，Tip-Adapter将预训练的CLIP调整为一个全新的数据集进行评估，这将泛化到一个新的领域，因此更具挑战性。重要的是，我们在完整的测试集上测试我们的模型，与由完整训练集训练的传统方法[22,13]相同。与在小查询集上的现有工作[60,53]相比，我们的有效性通过更多新类别的测试图像得到了验证。



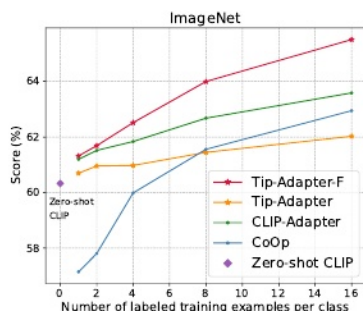


图4所示。Ima-geNet上不同模型的Few-shot分类准确率[10]。

Few-shot Setup	1	2	4	8	16
Zero-shot CLIP [48]	60.33				
Linear-probe CLIP [48]	22.17	31.90	41.20	49.52	56.13
CoOp [74]	57.15	57.81	59.99	61.56	62.95
CLIP-Adapter [16]	61.20	61.52	61.84	62.68	63.59
Tip-Adapter	60.70	60.96	60.98	61.45	62.03
Tip-Adapter-F	<b>61.32</b>	<b>61.69</b>	<b>62.52</b>	<b>64.00</b>	<b>65.51</b>
	+0.62	+0.73	+1.54	+2.55	+3.48

表2。不同模型在Ima-geNet[10]上的分类准确率(%), 并给出定量值。蓝色的最后一行记录了通过对Tip-Adapter进行进一步微调而获得的Tip-Adapter-F的性能增益。

## 4 实验

### 4.1 培训设置

我们在11个广泛使用的图像分类数据集上进行了Tip-Adapter和Tip-Adapter-F的实验: ImageNet[10]、StanfordCars[34]、UCF101[54]、Caltech101[14]、Flowers102[44]、SUN397[63]、DTD[8]、EuroSAT[23]、FGV-CAircraft[41]、OxfordPets[46]和Food101[3]。对于Few-shot学习, 我们比较了1、2、4、8、16个Few-shot训练集的性能, 并在完整的测试集上进行测试。对于CLIP主干, 我们使用ResNet-50[22]作为视觉编码器, 使用变压器[13]作为文本编码器。我们从[48]中获得两个编码器的预训练权重, 并在训练过程中冻结它们。我们遵循CLIP[48]中的数据预处理协议, 该协议由随机裁剪, 重新调整大小和随机水平翻转组成。除了CoOp中可学习的提示外, 我们遵循CLIP采用提示集成, 特别是在ImageNet上, 并在其他10个数据集上使用单个手工制作的提示。Tip-Adapter-F使用批大小256、学习率0.001和带有余弦调度器的AdamW[32]优化器进行微调。我们对EuroSAT数据集设置了100 epoch的训练, 对其他10个数据集只设置了20 epoch的训练。

对零射击CLIP[48]、线性探针CLIP[48]、CoOp[74]和CLIP-Adapter[16]进行性能比较。其中, 零射击CLIP不使用额外的训练样本, 纯粹通过预训练的知识进行分类。线性探针CLIP在Few-shot训练集上对权重冻结的CLIP进行训练后, 再训练一个额外的线性分类器。CoOp采用可学习的提示进行训练, 我们选择其表现最好的变体进行比较, 也就是说, 将类令牌放在16个令牌提示的末尾, 没有特定于类的上下文。CLIP-Adapter附加了一个特征适配器[25], 以缩小预训练的特征和下游任务之间的域差距。我们还报告了仅使用可学习的可视适配器的CLIP-Adapter的最佳性能变体。我们在论文中报告了他们的官方分数, 以便进行公平的比较。

表3. 不同视觉编码器在16张ImageNet上的分类准确率(%) [10]。viti - b /32和viti - b /16分别表示补丁大小为 $32 \times 32$ 和 $16 \times 16$ 的viti - base [13], RN50 $\times$ 16表示计算量为16倍的ResNet-50[22][48]。

Models	ResNet-50	ResNet-101	ViT-B/32	ViT-B/16	RN50 $\times$ 16
Zero-shot CLIP [48]	60.33	62.53	63.80	68.73	70.94
CoOp [74]	62.95	66.60	66.85	71.92	-
CLIP-Adapter [16]	63.59	65.39	66.19	71.13	-
Tip-Adapter	62.03	64.78	65.61	70.75	72.95
Tip-Adapter-F	<b>65.51</b>	<b>68.56</b>	<b>68.65</b>	<b>73.69</b>	<b>75.81</b>

## 4.2 ImageNet的比较

性能分析。如图4和表2所示, Tip-Adapter和Tip-Adapter-F都比其他方法表现出出色的性能。与零射击CLIP相比, Tip-Adapter在没有任何训练的情况下始终超越它。当训练样本数量有限时, Tip-Adapter在一次射击和两次射击设置中大大超过线性探针CLIP +38.53%, +29.06%。通过进一步的微调, Tip-Adapter-F更新了缓存模型中的键, 并在所有Few-shot设置中实现了所有方法的最佳性能。随着训练样本数量的增加, Tip-Adapter的性能增益变得更大, 从1次射击+0.62%到16次射击+3.44%。这表明, 使用更多训练样本进行微调, 使网络能够构建更强大的缓存模型。在表3中, 我们还在ResNet[22]和ViT[13]骨干网上使用各种视觉编码器实现了不同的模型, 其中我们的Tip-Adapter-F仍然表现最好。

效率的比较。在表1中, 我们展示了不同模型的训练时间和推理速度的比较。CLIP-Adapter、Tip-Adapter和Tip-Adapter-F能够在开始时缓存CLIP中的文本特征, 并在训练或推理时加载它们, 但CoOp采用可学习提示, 每次迭代都需要在线计算整个文本编码器。线性探针CLIP使用逻辑回归[62], 因此不能通过epoch和GPU上的推理速度来测量训练时间。通过比较, 我们观察到CoOp在学习提示上花费了最多的训练时间, 并且比零射击CLIP的性能提高了+2.26%。CLIP-Adapter显著减少了训练时间, 性能提高了+3.26%, 但仍然需要200 epoch的训练。在缓存模型的帮助下, Tip-Adapter获得了+1.70%的改进, 但不需要额外的训练时间, 这使得它在性能和效率之间取得了很好的平衡。Tip-Adapter-F进一步达到了最先进的精度, 只有CLIP-Adapter和CoOp的训练时间的1/10, 实现了两全其美。至于推理速度和GPU内存消耗[52], 我们的Tip-Adapter和Tip-Adapter - f在零射击CLIP上只产生边际的额外延迟, 与CoOp相比节省了大量GPU内存, 这对于应用程序来说是相当高效的。

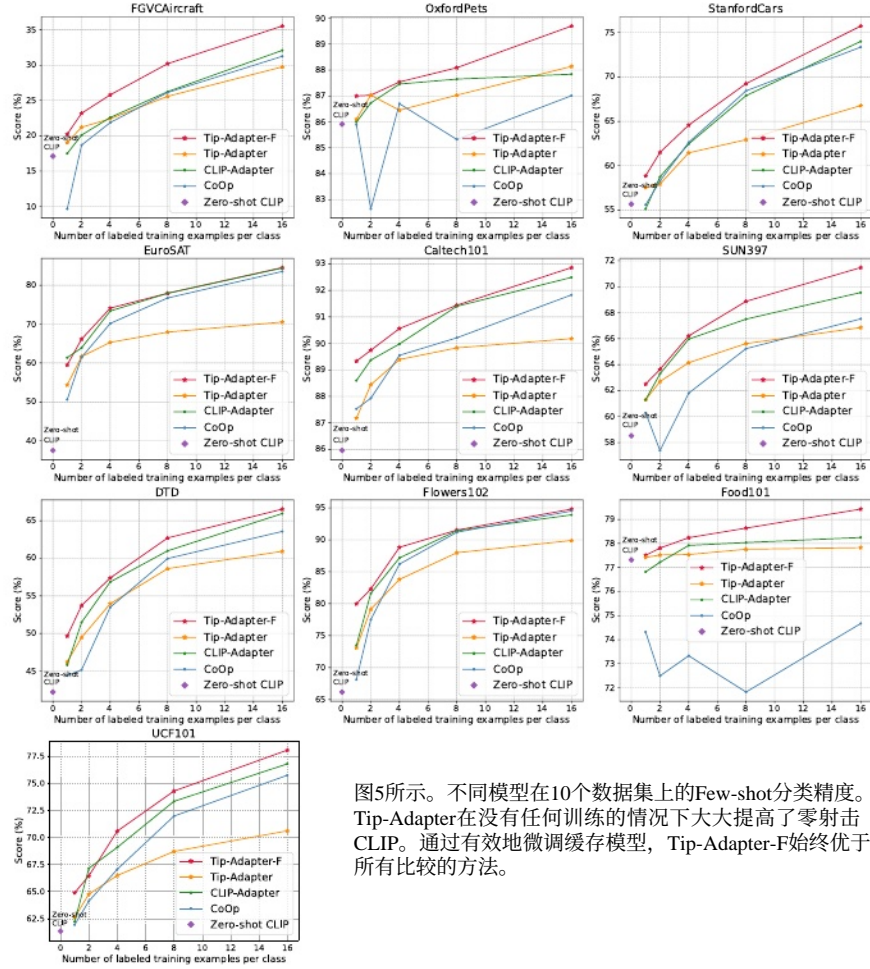


图5所示。不同模型在10个数据集上的Few-shot分类精度。Tip-Adapter在没有任何训练的情况下大大提高了零射击CLIP。通过有效地微调缓存模型，Tip-Adapter-F始终优于所有比较的方法。

### 4.3 在其他数据集上的性能

图5显示了在4.1节列出的其他10个数据集上的性能对比。我们的无需训练的Tip-Adapter显著提高了零射击CLIP的分类精度，并且在大多数数据集上超过了1或2次射击训练的CoOp。虽然Tip-Adapter不如CoOp和CLIP- adapter，但Tip-Adapter-F通过更少的微调可以消除差距，进一步超越所有其他型号，实现全面领先的性能。Tip-Adapter-F在10个数据集上的一致性优势充分证明了我们所提出的缓存模型的有效性和通用性。

### 4.4 烧蚀研究

在本节中，我们对ImageNet上的Tip-Adapter进行了几项消融研究[10]。所有实验均采用16镜头设置，无需训练。

表4。ImageNet上Tip-Adapter的四项消融研究(%) [10]，从上到下:残差比 $\alpha$ ，锐度比 $\beta$ ，缓存模型的大小，以及固定缓存大小16时给出更多镜头的性能。

Ablation Studies on Tip-Adapter						
Residual Ratio $\alpha$	0.0	0.5	<b>1.0</b>	2.0	3.0	4.0
	60.33	61.44	<b>62.03</b>	61.41	60.36	59.14
Sharpness Ratio $\beta$	1.5	3.5	<b>5.5</b>	7.5	9.5	11.5
	61.82	61.91	<b>62.03</b>	61.76	61.62	61.40
Cache Size	0	1	2	4	8	<b>16</b>
	60.33	61.45	61.71	61.79	61.83	<b>62.03</b>
More Shots than 16	Shot Setup		16	32	64	128
	Tip-Adapter		62.03	62.51	62.88	63.15
	Tip-Adapter-F		65.47	66.58	67.96	69.74

残差比 $\alpha$ 。超参数 $\alpha$ 控制了将缓存模型中新调整的预测与预训练的CLIP模型相结合的程度，这也可以解释为像Eq. 9中那样权衡视觉和文本缓存。如上所述，较大的 $\alpha$ 表示从Few-shot训练集中使用更多的知识，否则使用较少的知识。我们将 $\alpha$ 从0.0变化到5.0，并将超参数 $\beta$ 设置为5.5。当 $\alpha = 0.0$ 时，模型在不使用Few-shot知识的情况下等效于零射击CLIP。从表4的上半部分，我们观察到随着 $\alpha$ 从0.0增加到1.0，分类精度正在提高，在1.0时达到了最好的62.03%。这表明来自CLIP模型的先验知识和来自缓存模型的Few-shot知识同等重要。

锐度比 $\beta$ 。在Eq.(3)中，激活函数 $\phi$ 中的 $\beta$ 控制亲和性的锐度。当 $\beta$ 很大时，只有在嵌入空间中与测试图像最相似的训练样本对预测有较大的影响，反之亦然。在表4的第二部分中， $\alpha$ 为1.0时，我们观察到 $\beta$ 的变化具有有限的影响， $\beta$ 的适度5.5导致性能最佳的Tip-Adapter。

缓存模型的大小。我们探讨了Tip-Adapter中大小对缓存模型的影响。给定16次训练集，而不是每个类别缓存所有16个样本，我们构建了大小大于0但小于16的缓存。以8为例，我们将16个样本随机分成8个均匀的组，通过平均每组中2个样本的特征得到8个原型。考虑到这样随机划分样本可能会影响性能，我们进行了5次实验，并报告了平均分数。表4第三部分的结果表明，我们缓存的样本越多，以保留更多的Few-shot知识，Tip-Adapter可以实现的精度就越高。

放大到更多的镜头。给定超过16个镜头，我们探索了一种方法，仍然将缓存大小限制为16，并避免对两种内存的潜在负担

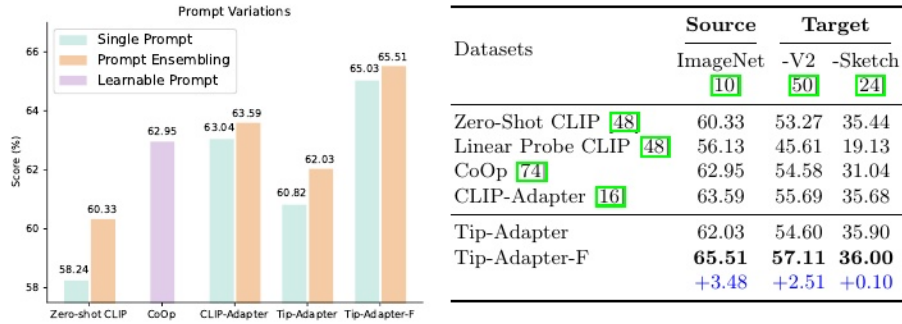


图6所示。不同提示设计下的分类性能:单一提示(青色)、提示集成(橙色)和可学习提示(紫色)。

表5所示。不同方法对分布偏移的鲁棒性(%)。蓝色的最后一行记录了通过对Tip-Adapter进行进一步微调而带来的Tip-Adapter-F的性能增益。

和计算。以64次射击为例，遵循上一段的划分策略，我们从4组中获得16个原型来构建缓存模型。表4的最后一部分表明，即使缓存大小限制为16, Tip-Adapter也可以很好地从每个类别的32、64和128个训练样本中捕获知识。此外，当提供更多样本时，性能提升会逐渐放缓，这意味着在没有训练的情况下，缓存大小可能限制为16。然而，Tip-Adapter-F可以通过微调琴键来打破这种限制，通过更多的击球训练来达到更好的表现。

提示的设计。我们使用来自[48]的7个模板的提示集成作为默认值，用于零射击CLIP, CLIP- adapter和Tip-Adapter。在图6中，我们只使用一个提示符来测试它们，“一个[类]的照片。”，并观察到稍差的性能。Tip-Adapter-F和CLIP-adapter的精度下降较小，而Tip-Adapter和零射击CLIP的精度下降较大，这表明性能较好的模型受提示变化的影响较小。

#### 4.5 分布变化

我们通过从一个数据集学习而在另一个数据集上测试来评估我们提出的Tip-Adapter和Tip-Adapter- f的分布能力。我们将ImageNet[10]作为提供16个shot训练集的源数据集，并采用两个目标数据集进行测试:ImageNetV2[50]和ImageNet- sketch[24]，它们包含与ImageNet兼容但存在语义缺口的类别。如表5所示，未经训练的Tip-Adapter对分布移位的鲁棒性更强，超过了ImageNet-V2上的CoOp[74]和ImageNet- sketch上的CLIP-Adapter[16]。这表明缓存模型更有利于分布外评估，其无需训练的构造缓解了源数据集上过度拟合的风险。此外，Tip-Adapter- f实现了两全其美:缓存模型带来的强大的out- distribution性能和微调带来的领先的in-distribution能力。

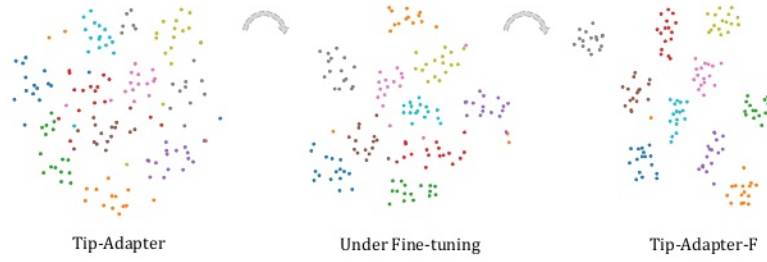


图7所示。Tip-Adapter中 $F_{\text{train}}$ 的t-SNE可视化。不同颜色的点代表不同类别的嵌入。从左到右，三种分布表示微调期间缓存模型中键的变化。

## 5 可视化

为了更好地显示微调期间缓存模型的变化，我们在图7中使用t-SNE[48]来可视化键 $F_{\text{train}}$ 。不同颜色的点表示16张ImageNet中的10个类别[10]，它们的相对距离反映了类别嵌入的高维分布。从左到右，三个子图分别表示未训练的Tip-Adapter、微调期间的Tip-Adapter和最终的Tip-Adapter-f。可以观察到，在训练之前，由于合理设计了缓存模型构造，分布已经显示出良好的区分度。在微调过程中，相同类别的嵌入逐渐收敛在一起，不同的簇变得更具对比性和独立性，有助于更强的分类能力。

## 6 结论

提出了一种CLIP的非参数自适应方法——Tip-Adapter，该方法通过Few-shot训练集构建缓存模型获取适配器。通过这种方式，从缓存模型中检索Few-shot知识，并以无训练的方式与CLIP预训练的知识相结合。最重要的是，Tip-Adapter可以通过微调缓存的键来进一步增强，只针对几个epoch，命名为Tip-Adapter-f，这在现有方法中实现了最先进的性能。考虑到局限性，尽管是边际的，Tip-Adapter-f仍然需要在ImageNet上进行20 epoch的微调，以学习性能最好的缓存模型。我们未来的工作将集中在探索新的无需培训的CLIP方法，以充分释放其视觉表现的力量。

确认。本研究由感知及互动智能研究中心有限公司、香港研究资助局资助基金(编号:14204021、14207319)、香港中文大学策略基金及上海市科学技术委员会(资助号:21DZ1100100)资助。



## 附录

### A 微调设置

与未经训练的Tip-Adapter相比，Tip-Adapter- f在缓存模型中微调键 $F_{train}$ ，但冻结值 $L_{train}$ 、CLIP的[48]视觉编码器和文本编码器。在这里，我们将探讨是否可以对Tip-Adapter中的其他模块进行微调以提高性能。在表6中，我们对Tip-Adapter不同模块的解冻进行了7次微调实验。注意，为了训练稳定性，我们将两个CLIP编码器的学习率设置为 $F_{train}$ 和 $L_{train}$ 的1/1000，并在ImageNet [10]上使用16次训练集训练每个设置20个epoch。如图所示，前两行表示Tip-Adapter的62.03%和Tip-Adapter- f的65.51%的性能。第三行通过微调缓存值 $L_{train}$ 将性能降低到60.90%，微调所有缓存模型甚至会导致训练过程中的崩溃，这符合我们的假设，即不更新one-hot ground-truth标签以保留Few-shot知识。此外，我们尝试固定缓存模型中的所有参数，并微调预训练的CLIP的权重。如果视觉编码器或文本编码器独立微调，性能可以分别提高到62.84%和63.15%，但当两个编码器联合微调时，分类精度将显著下降到51.22%。这是因为这样一个从Few-shot训练集学习的大参数模型存在严重的过拟合。与解冻CLIP编码器相比，仅微调 $F_{train}$ 就能带来更大的性能提升，但耗时更少，这充分展示了我们的Tip-Adapter- f的优势。

表6所示。微调Tip-Adapter的不同模块。”†表示微调，`-`表示冻结。的活力。还有“特克斯。表示CLIP的视觉编码器和文本编码器。准确率(%)和训练时间在16发ImageNet[10]和单个NVIDIA GeForce RTX 3090 GPU上进行了测试。

Vis.	Tex.	$F_{train}$	$L_{train}$	Accuracy	Time
-	-	-	-	62.03	<b>0</b>
-	-	✓	-	<b>65.51</b>	5min
-	-	-	✓	60.90	5min
-	-	✓	✓	Collapsed	-
✓	-	-	-	62.84	8min
-	✓	-	-	63.15	1h 20min
✓	✓	-	-	51.22	1h 27min

## B 未经训练的性能增益

在图8中，我们展示了在16次射击设置下的11个分类数据集上，Tip-Adapter比零射击CLIP[48]带来的绝对精度提高。无需任何训练，Tip-Adapter可以将EuroSAT上的零射击CLIP提高33.02%，将Fowers102提高23.87%。既然CLIP是在日常场景的大规模网络收集的图像-文本对上进行预训练的，当下游数据集和预训练数据之间的域差距较大时，Tip-Adapter的性能增益通常会更高。以EuroSAT和DTD为例，它们分别包含具有不同语义的土地覆盖和细节纹理图片，因此需要缓存模型中存储更多的Few-shot知识来更新预训练的CLIP知识，以获得更好的性能。

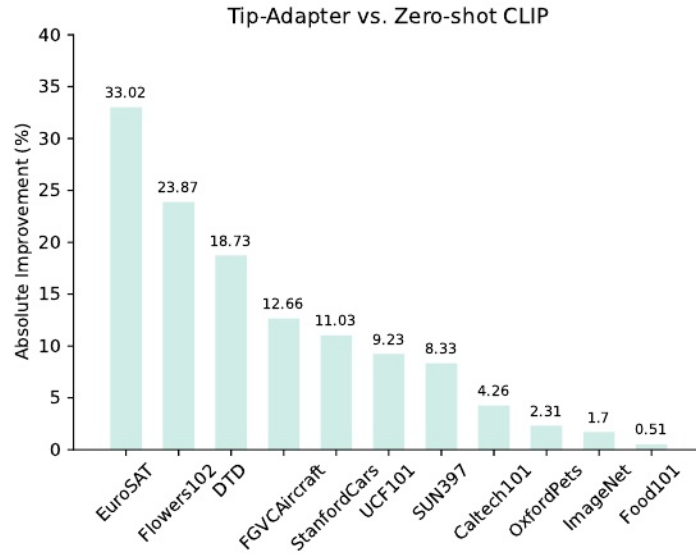


图8所示。性能增益来自提出的免训练缓存模型，该模型由11个分类数据集上的16次训练集构建。

## C 与完全训练的方法进行比较

虽然我们的Tip-Adapter和Tip-Adapter-f是基于Few-shot训练集，但它们是由完整测试集来评估的，与传统的由完整训练集训练的方法[22,13]相同。在表7中，我们比较了我们的和ResNet[22]和DeiT[58]系列之间的可学习参数和训练设置。我们采用ViT-Large[13]作为Tip-Adapter的视觉骨干



Tip-Adapter-F。如图所示，仅通过16次训练集，未经pa参数或训练的Tip-Adapter分别优于ResNet-50和DeiT-T +1.9% 和+3.9%。Tip-Adapter-F通过6分钟的高效微调进一步实现更高的性能。这证明了所提出方法在低数据和资源有限的体制下的优越性。

表7所示。Tip-Adapter、Tip-Adapter-F与ImageNet上全训练集训练方法的比较[10]。训练时间在单个NVIDIA GeForce RTX 3090 GPU上进行测试。

Method	Acc. (%)	Param. (M)	Train. Set	Train. Time
ResNet-50 [22]	74.2	25.6	full set	>1 day
ResNet-101 [22]	77.4	44.5	full set	>1 day
DeiT-T [58]	72.2	6.0	full set	>1 day
DeiT-S [58]	<b>79.9</b>	22.1	full set	>1 day
Tip-Adapter	76.1	<b>0</b>	<b>16-shot</b>	<b>0</b>
Tip-Adapter-F	79.4	6.2	<b>16-shot</b>	6 min