摘要

多任务学习（MTL）旨在同时训练多个相关任务，从而提高单个任务的性能。典型的多任务网络结构由共享的主干和特定于任务的解码器组成。然而，随着任务数量的增加，解码器的复杂性也随之增加。为了应对这一挑战，我们集成了无解码器的视觉-语言模型CLIP，该模型展现出强大的零样本预测能力。

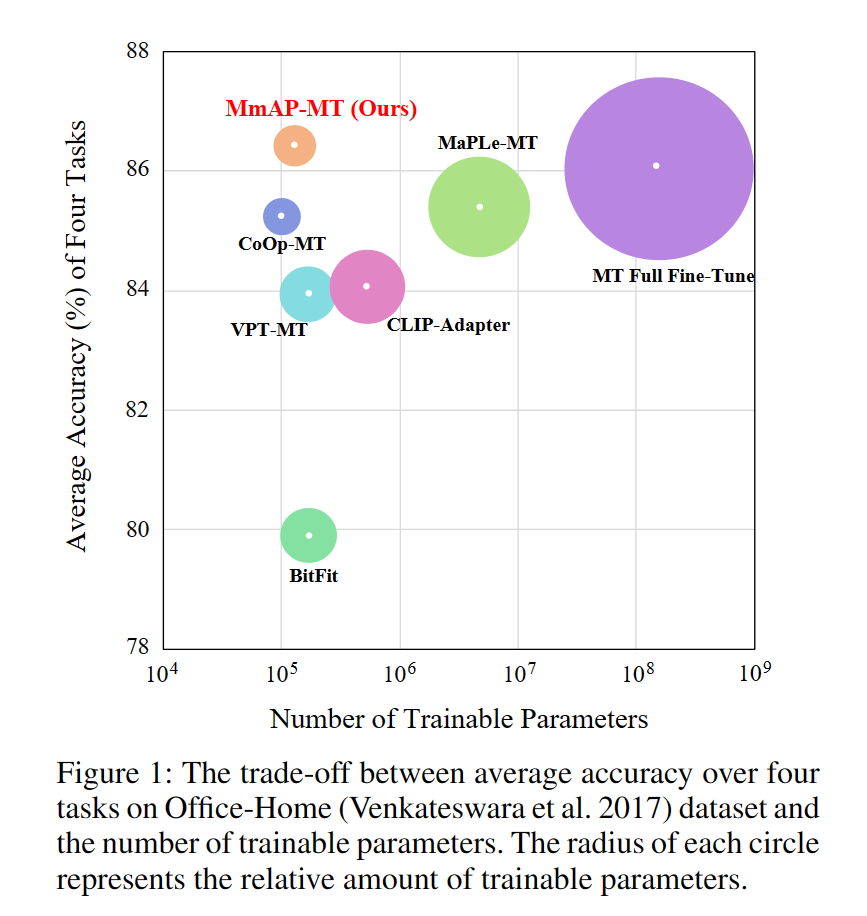
最近，针对CLIP的参数高效的迁移学习方法已被广泛探索，用于适应下游任务，其中提示调整展现出强大的潜力。然而，这些方法仅对单一模态（文本或视觉）进行微调，破坏了CLIP的模态结构。在本文中，我们首先为CLIP提出多模态对齐提示（MmAP），它在微调过程中对文本和视觉模态进行对齐。在MmAP的基础上，我们开发了一个创新的多任务提示学习框架。一方面，为了最大化高相似性任务的互补性，我们使用基于梯度的任务分组方法，将任务划分为几个不相交的组，并为每个组分配一个共享的MmAP。另一方面，为了保留每个任务的独特特征，我们为每个任务分配一个特定于任务的MmAP。在两个大型多任务学习数据集上的全面实验表明，与完全微调（Full fintuing）相比，我们的方法实现了显著的性能提升，同时仅使用约0.09%的可训练参数

**引言**

研究现状：多任务学习（MTL）是深度学习中一种强大的方法，它通过在统一的网络架构中同时训练多个相关任务，提高了模型性能。MTL通过学习任务共享和任务特定的表示来增强泛化能力并减轻过拟合。在视觉多任务学习领域，先前的研究主要集中在设计多任务模型训练框架，包括基于编码器和基于解码器的方法。随着视觉预训练模型的发展，直接微调这些模型用于下游多任务成为主流方法。

存在问题：尽管视觉预训练模型的微调带来了性能提升，但整个CLIP参数的更新对计算和存储资源提出了挑战。此外，现有的参数高效迁移学习技术主要针对预训练的视觉模型或语言模型，它们对更复杂的视觉-语言模型的适用性不确定。这些方法倾向于强调单任务适应，而多任务适应仍然是一个挑战。

本文作者贡献：作者引入了预训练的视觉-语言模型CLIP，该模型能够对齐语言和视觉模态，消除了为每个任务建立额外解码器结构的需求。作者探索了参数高效的迁移学习技术，以实现可训练参数和性能之间的最佳平衡。基于CLIP，作者提出了一种新颖的多模态对齐提示（MmAP）和多任务提示学习框架，旨在解决单任务适应的问题，并提高多任务学习的效率。实验结果表明，作者的方法在两个大型多任务学习数据集上取得了显著的性能提升，同时仅使用约0.09%的可训练参数。



**相关工作**

在“相关工作”部分，本文概述了三个主要领域的研究进展：多任务学习（MTL）、视觉-语言模型（V-L模型）和参数高效的迁移学习（PETL）。

多任务学习（MTL）：MTL致力于通过共享知识和计算资源来同时学习多个相关任务，从而提高模型性能。在计算机视觉领域，MTL主要分为两类：一是密集场景理解多任务，涉及语义分割、表面法线估计等任务；二是跨域分类多任务，处理具有域偏移的多个数据集。当前研究主要关注解码器结构设计以及在不同域之间学习共享和私有信息。

视觉-语言模型（V-L模型: **Vision-Language Model.**）：基础视觉-语言模型如CLIP和ALIGN在视觉任务中展现出显著能力，它们编码了丰富的多模态表示。尽管这些预训练模型能有效学习丰富的表示，将它们适应到下游视觉任务仍是一个挑战。

参数高效的迁移学习（PETL: **Parameter-Efficient Transfer Learning**）：PETL旨在通过训练少量参数使预训练模型适应新的下游任务。现有方法分为参数调整(parameter tuning)、适配器调整(adapter tuning)和提示调整(prompt tuning)。参数调整直接修改模型参数，适配器调整在模型中插入可训练的架构，而提示调整则通过设计特定模板将下游任务统一到预训练任务中，充分利用基础模型的能力。

**方法**

CLIP(**Contrastive Language-Image Pre-training**)：对比语言-图像预训练

1、图像编码器：

**输入图像**：给定一个输入图像*I*∈R*H*×*W*×3，其中 �*H* 和 �*W* 分别是图像的高度和宽度，3 表示 RGB 三个颜色通道。

**图像分割**：图像编码器将图像分割成 *M* 个固定大小的小块，并将它们投影到嵌入空间中，得到初始的小块嵌入*E*0​∈R*M*×*dv*​，其中 *dv*​ 是嵌入的维度。

**Transformer 层**：图像编码器由 *K* 个 Transformer 层组成。每个层接收一个可学习的类别标记 *ck*​ 和小块嵌入*Ek*​，并通过 Transformer 层 *Vk*+1​ 进行处理，得到下一层的输出 [*ck*+1​,*Ek*+1​]。这个过程从 *k*=0 重复到 *K*−1。

**图像表示**：最终的图像表示 *x* 通过将最后一个 Transformer 层的类别标记 *cK*​ 投影到 V-L 潜在嵌入空间来获得。

**2、文本编码器**

**文本处理**：文本编码器使用包含 *K* 层的 Transformer 来对输入词进行分词，并将它们投影到词嵌入空间中，得到初始的词嵌入 *W*0​∈R*N*×*dl*​，其中 *N* 是词的数量，*dl*​ 是嵌入的维度。

**Transformer 层**：类似于图像编码器，文本编码器的每个 Transformer 层 *Lk*+1​ 处理词嵌入 *Wk*​，得到下一层的输出*Wk*+1​。这个过程从*k*=0 重复到 *K*−1。

**文本表示**：最终的文本表示 *z* 通过将最后一个 Transformer 层的最后一个标记的文本嵌入 *WK*​ 投影到 V-L 潜在嵌入空间来获得。

**Zero Shot Prediction（零样本预测）：**

假设有一个零样本图像分类任务，目标是将图像分类为“狗”、“猫”或“马”，但模型在训练过程中没有看过这些类别的图像，使用CLIP模型进行预测。

**步骤 1: 设计提示**

对于每个类别设计一个提示，例如：

* “一张狗的照片”
* “一张猫的照片”
* “一张马的照片”

**步骤 2: 提取文本特征**

使用CLIP的语言分支，将上述每个提示转换为文本特征向量：

* *z*狗​：表示“一张狗的照片”的特征向量
* *z*猫​：表示“一张猫的照片”的特征向量
* *z*马​：表示“一张马的照片”的特征向量

**步骤 3: 提取图像特征**

对于给定的待分类图像（比如一张未知动物的照片），使用CLIP的图像分支提取图像特征向量 *x*。

**步骤 4: 计算余弦相似度**

对于每个类别，计算图像特征向量 *x* 与对应类别的文本特征向量 类别*z*​ 之间的余弦相似度：

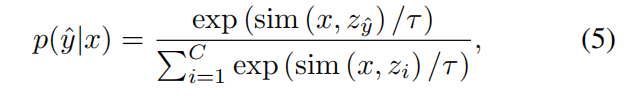
* *sim*(*x*,*z*狗​)
* *sim*(*x*,*z*猫​)
* *sim*(*x*,*z*马​)

**步骤 5: 预测标签**

根据余弦相似度的得分，选择得分最高的类别作为预测标签 *y*^​。例如，如果 *sim*(*x*,*z*猫​) 是三者中最高的，那么预测标签 *y*^​ 就是“猫”。

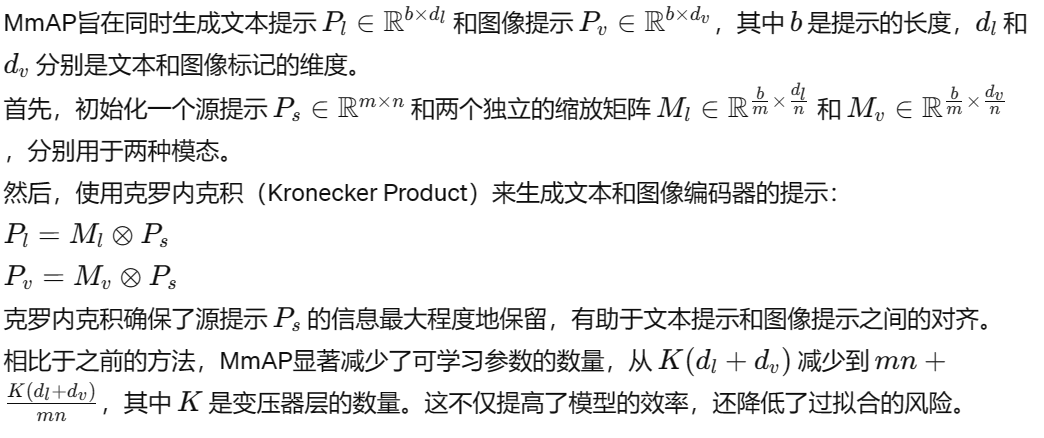
**步骤 6: 计算概率**

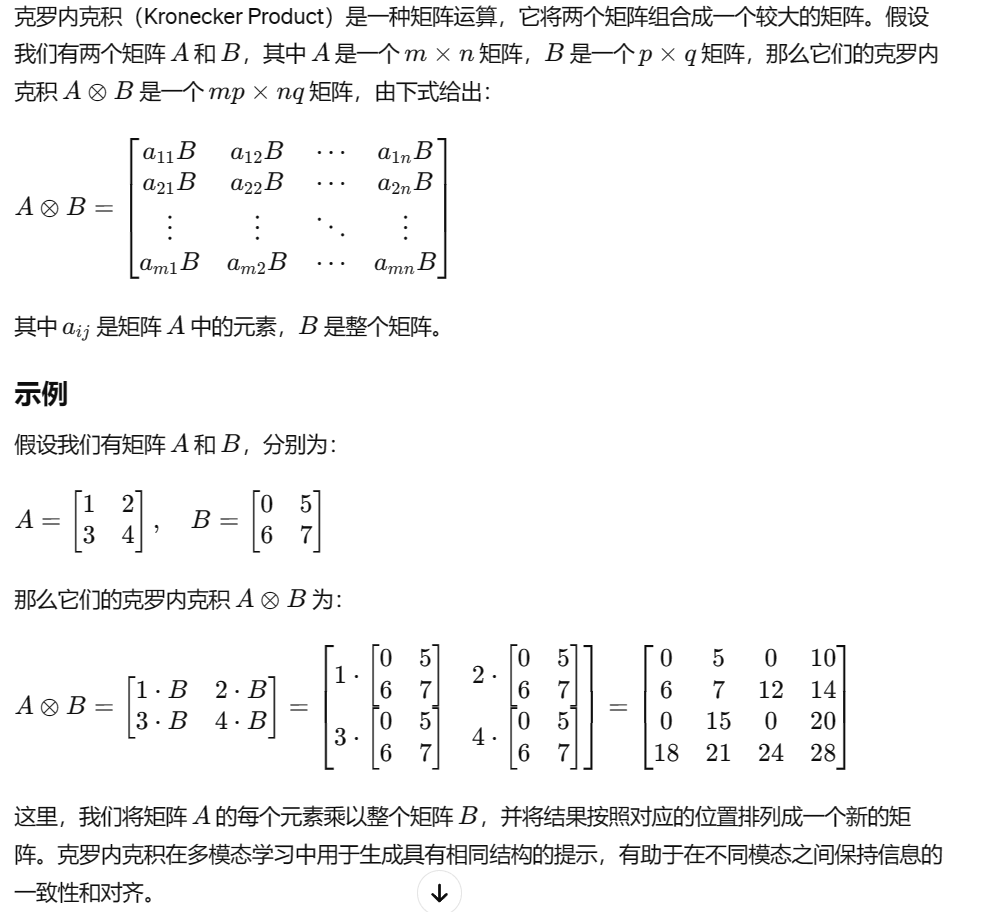
使用softmax函数计算预测标签的概率



通过这个过程，即使模型在训练过程中没有看过“狗”、“猫”或“马”的图像，它也能够利用CLIP的语言理解能力和设计的提示来进行零样本图像分类。

多模态对齐提示（MmAP）：





多任务提示学习框架：

**多任务学习**

* 在多任务学习中，同时训练多个相似任务可以带来相互有益的结果。通常，任务之间的相似度可以通过评估任务间的梯度冲突来量化。

**任务分组**

* 根据任务之间的相似度，将相似的任务分组在一起。这有助于在同一组内的任务之间进行相互学习和增强。

**共享和独立提示**

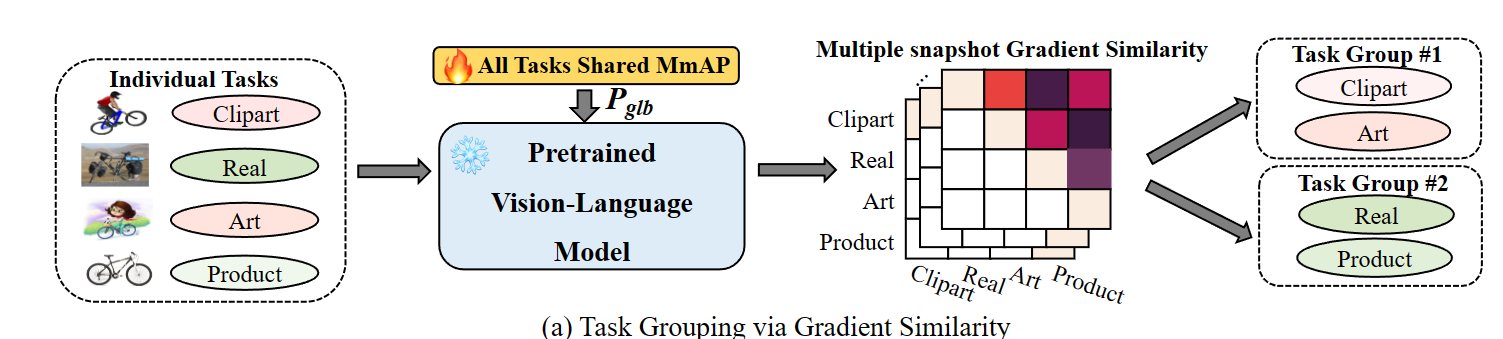
* 对于每个任务组，分配一个共享的多模态对齐提示（MmAP）。这个共享的MmAP促进了组内任务的相互学习和增强。
* 然而，为了保持每个任务的独特特征，也为每个任务分配一个独立的MmAP。这个独立的MmAP确保了每个任务的独特特征和需求得到了充分的满足。

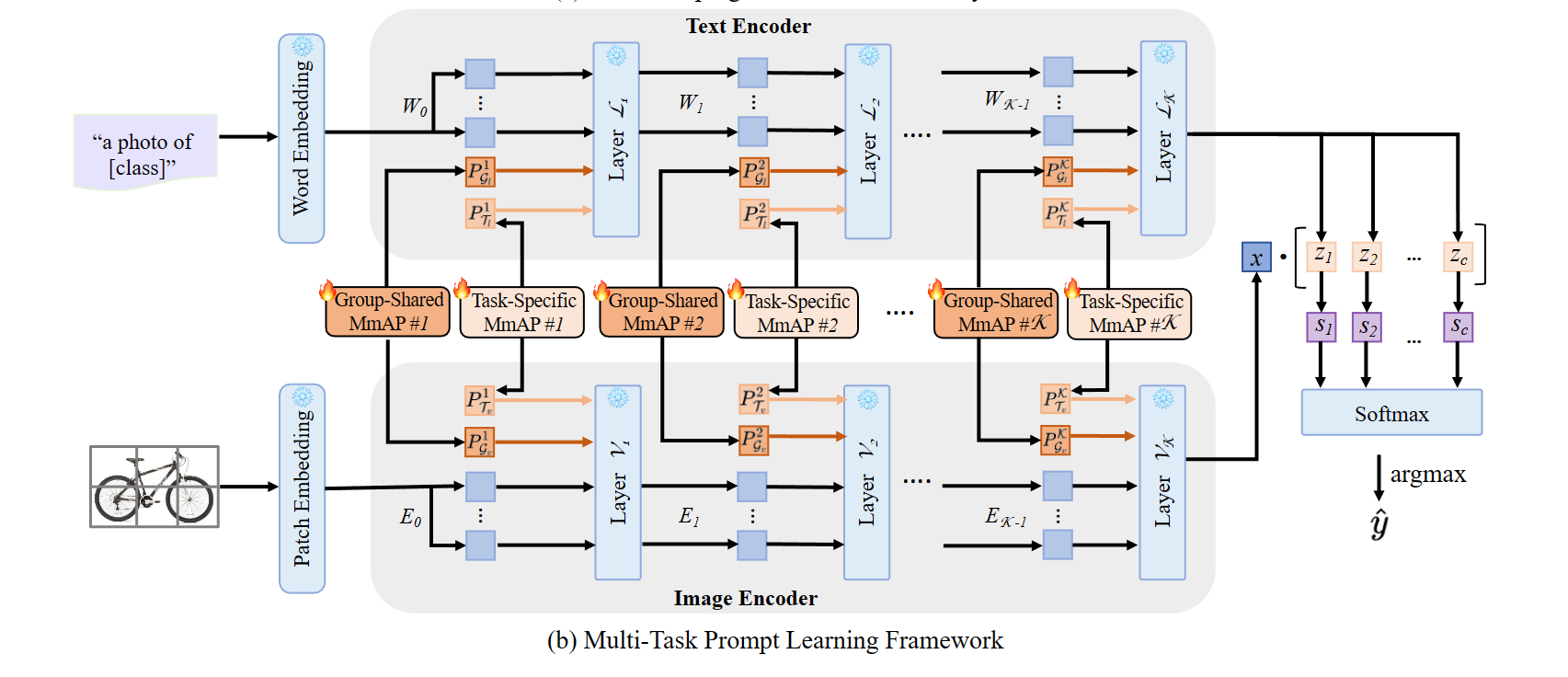
**总体框架**

* 总体的多任务提示调整框架通过结合共享和独立的MmAP，实现了在保持任务独特性的同时促进任务间的相互学习。这种框架可以提高模型在多任务学习中的表现，使得模型能够更好地适应和优化多个相关任务。

**总结**

多任务提示学习框架通过将相似任务分组，并为每个组和每个任务分配适当的提示，有效地结合了任务间的相互学习和任务独特性的保持。这种方法可以提高模型在处理多个相关任务时的性能和效率。





**实验**

数据集：

Office-Home包含来自四个任务（Art、Clipart、Product和Real World）的图像，每个任务涵盖65个对象类别，共约15500张图像。

MiniDomainNet是DomainNet的一个子集，是一个极具挑战性的多任务学习数据集。MiniDomainNet包含140000张图像，分布在126个类别中，包含四个不同的任务：Clipart、Painting、Sketch和Real。

Baseline

Zero-shot使用人工撰写的文本提示（“a photo of [class]”）模板进行零样本预测。

Single-Task Full Fine-Tuning为每个任务更新一个预训练模型，Multi-task Full Fine-Tuning为所有任务更新一个共享的预训练模型。

单模态提示调整方法：包括标准的CoOp（针对文本提示调整）和标准的VPT（针对视觉提示调整）。CoOp-MT和VPT-MT是多任务版本，使用来自所有任务的样本训练一个共享的提示。

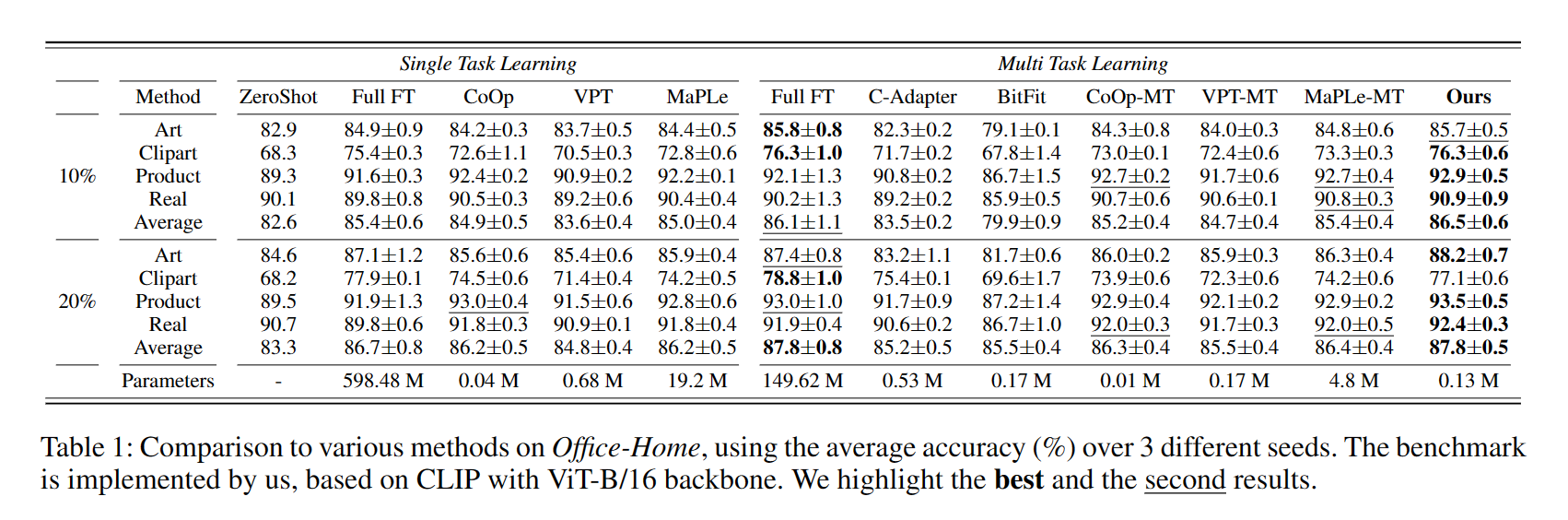
MaPLe：使用文本提示生成视觉提示。构建了一个多任务版本，称为MaPLe-MT。

其他参数高效的调整方法，包括CLIP-Adapter（在视觉或语言分支上学习新特征）和BitFit（调整预训练模型的偏置参数）

结果

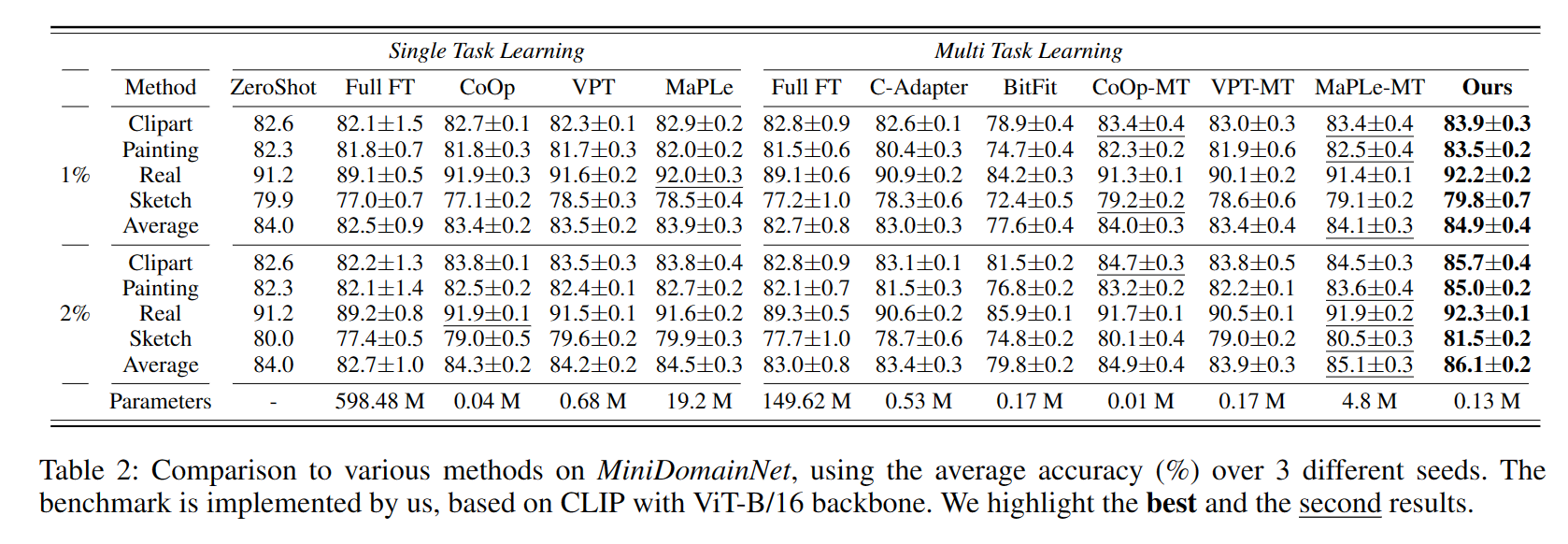
Office-Home：结果如表1所示。首先，观察到我们的方法在不同数据划分（10%或20%）上与Multi-Task Full Fine-Tuning相当，同时只需要0.09%（0.13M vs. 149.62M）的可训练参数。这代表了在CLIP的多任务图像识别中参数高效调整的重大突破。

其次，我们的方法一致地优于其他参数高效的调整方法与提示方法（即MaPLe-MT、CoOp-MT和VPT-MT）相比，我们的方法表现出显著的改进，强调了在调整CLIP时整合视觉和文本模态的必要性，以及结合共享和特定知识的重要性。



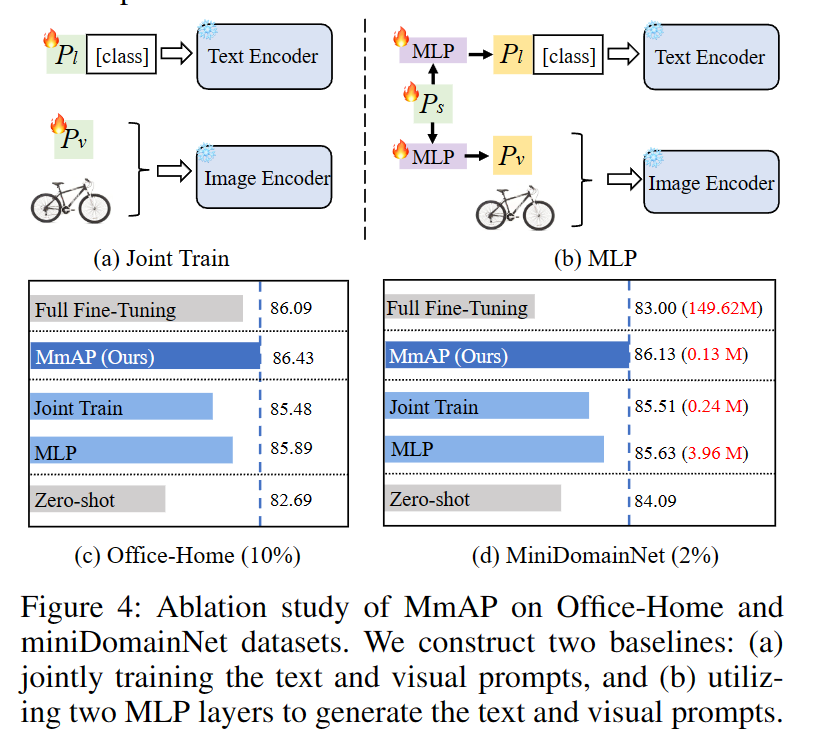
MiniDomainNet：结果如表2所示。我们可以得出与Office-Home一致的结论。我们的方法表现最佳，在1%划分上达到84.9%，在2%划分上达到86.1%。然而，我们观察到Full Fine-Tuning的性能并不十分令人满意，在大多数参数高效的调整方法中表现较差，这是由于过拟合造成的。

具体来说，与Office-Home相比，MiniDomainNet的任务难度显著增加，同时训练数据的数量有限。此外，BitFit方法表现最差。它使用少量数据更新CLIP的少数参数，严重损害了CLIP的原始零样本预测能力。CoOp-MT、VPT-MT和MaPLe-MT的效果在1%划分上只能接近零样本，但当训练数据量达到2%时，CoOp-MT和MaPLe-MT分别超过零样本0.9%和1.1%。因此，为了探索在不同训练数据大小下的性能，我们设置了详细的消融研究实验。

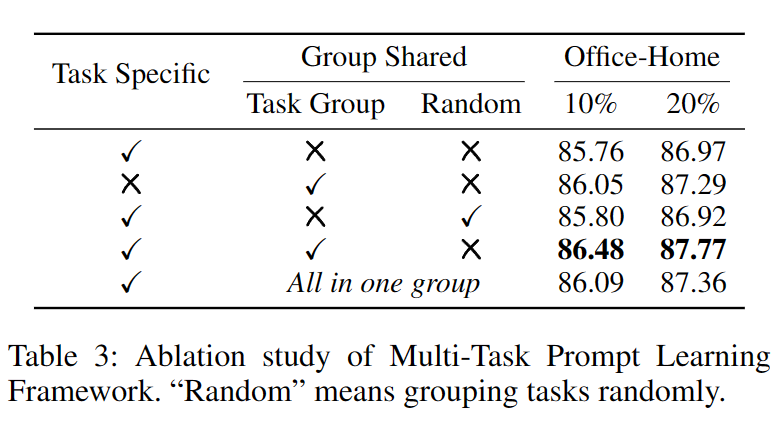


**消融实验**

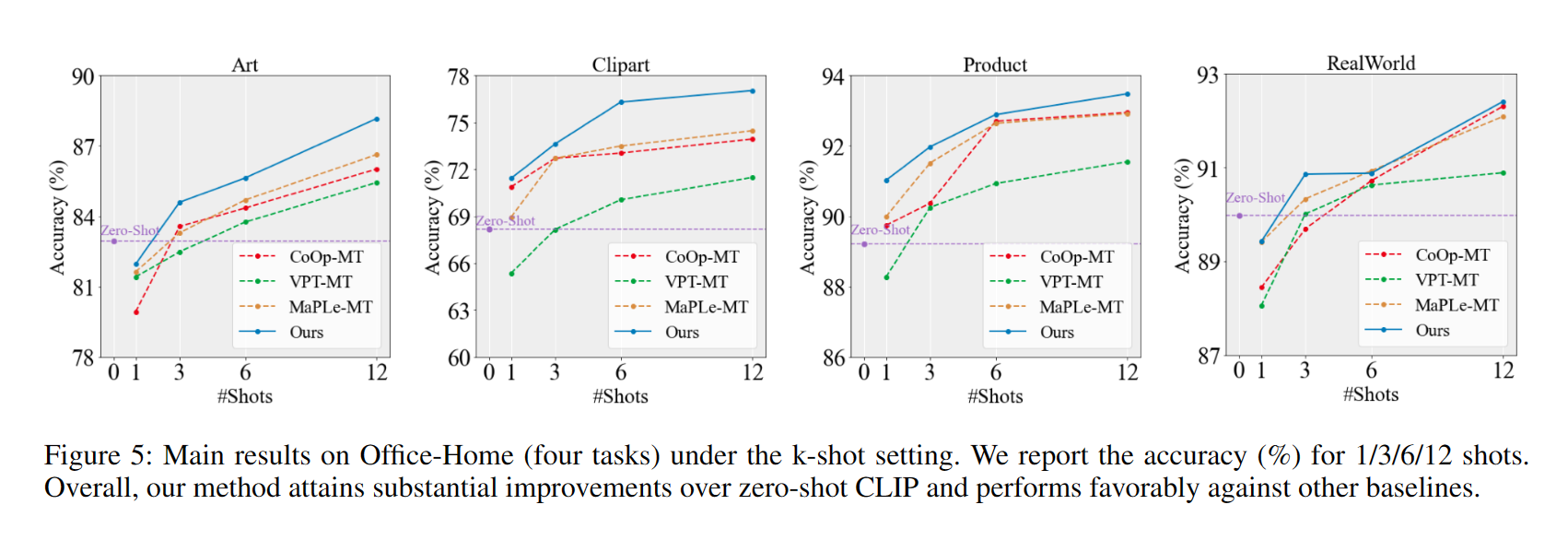
MmAP的有效性：如图4a所示，一种直接的多模态提示方法是同时调整文本和视觉提示。另一种直接的解决方案涉及共享文本和视觉提示。然而，由于CLIP的文本编码器（dl = 512）和图像编码器（dv = 768）的尺寸不相等，它们不能直接共享。因此，我们设计了MLP提示基线作为另一种比较方案，它使用两个MLP层来生成文本和视觉提示，如图4b所示。结果显示在图4c中。在Office-Home的四个任务中，MLP基线与联合训练基线相比显示出0.5%的改善，证明了建立文本提示和视觉提示之间连接的有效性。此外，我们观察到MmAP与MLP基线相比取得了0.54%的改善，表明MmAP方法通过Kronecker Product在文本和视觉提示之间更有效地实现了信息共享。同时，与MLP基线相比，MmAP的可训练参数大大减少了（0.13M vs. 3.96M）。



多任务提示学习框架的有效性：在多任务提示学习框架中，任务特定的MmAP和任务共享的MmAP是主要组成部分。为了验证每个模块的重要性，我们在Office-Home上进行了相关的消融实验，结果如表3所示。为了证明任务分组策略的有效性，我们将随机分组作为比较的基准。实验结果表明，我们框架中的每个模块都发挥了关键作用，累积贡献了我们的多任务提示学习框架所取得的优越性能。与随机分组相比，我们的任务分组在10%和20%的设置下分别表现出0.68%和0.85%的提升。与所有任务在一个组中相比，我们的任务分组在10%和20%的设置下分别表现出0.39%和0.41%的提升。从另一个角度来看，任务特定的MmAP超过了CoOp和VPT（表1中的结果），进一步证明了我们的MmAP的有效性。



不同下游数据大小的影响：我们研究了训练数据大小对Office-Home（四个任务）的影响。我们为每个类别选择1/3/6/12张图片，并将我们的MmAP与CoOp-MT、VPT-MT和MaPLe-MT进行比较。在不同训练数据规模下，每个任务和方法的结果如图5所示。结果表明，我们的方法在所有数据规模上都超过了所有其他方法，在四个任务上确认了我们方法的强大泛化能力。然而，我们观察到，在1-shot设置下，所有方法在Art和Real World任务上的表现都不如Zero-Shot。这可能是因为1-shot对于整个任务来说太具体，无法作为一般性表示。当提供3个或更多的训练样本时，我们方法引入的平均性能差距是显著的。



**结论**

在这项工作中，我们提出了多模态对齐提示（MmAP）来适应CLIP到下游任务，它在可训练参数和性能之间实现了最佳的权衡，优于大多数现有方法。同时，MmAP解决了之前的单模态提示方法（例如CoOp和VPT）破坏CLIP的模态对齐的问题。基于MmAP，我们设计了一个多任务提示学习框架，不仅使相似任务能够一起训练以增强任务互补性，而且还保留了每个任务的独立特性。我们的方法在两个大型多任务学习数据集上与全微调相比取得了显著的性能改善，在有限的下游数据下仅使用约0.09%的可训练参数。