arXiv:2303.01239v2 [cs.CV] 7 Jun 2023

MixPHM：面向低资源视觉问答的冗余感知参数高效微调

JingjingJiang Nanning Zheng*人工智能与机器人研究所，西安交通大学jingjingjiang2017@gmail.com nnzheng@mail.xjtu.edu.cn

摘要

近年来，微调预训练的视觉语言模型（*VLM*）已成为在视觉问答（*VQA*）中取得最先进性能的主流范式。然而，随着视觉语言模型规模不断扩大，在低资源设置下针对特定任务微调整个模型的参数，会带来高昂的计算成本、低效的存储开销，并且容易过拟合。当前的参数高效微调方法大幅减少了可调参数的数量，但与全量微调相比，性能仍存在显著差距。本文提出***MixPHM***，这是一种能够在低资源视觉问答中超越全量微调、具备冗余感知能力的参数高效微调方法。具体而言，*MixPHM* 是一个由多个 *PHM* 专家以混合专家方式实现的轻量级模块。为减少参数冗余，*MixPHM* 在低秩子空间中重新参数化专家权重，并在专家内部及专家之间共享部分权重。此外，基于对适配器的定量冗余分析，我们提出冗余正则化，以减少 *MixPHM* 表征中的任务无关冗余，同时促进任务相关性。在 *VQA v2*、*GQA* 和 *OK-VQA* 上开展的实验表明，*MixPHM* 超越了最先进的参数高效方法，并且是唯一一种始终优于全量微调的方法。

1. 引言

通过微调来适配预训练视觉语言模型（视觉语言模型） [5,6,25,30,31,53,60]以完成下游视觉问答任务 [2]，已成为实现最先进的性能的主流范式。然而，随着视觉语言模型的规模不断扩大，对包含数百万乃至数十亿参数的整个预训练模型进行微调，会大幅增加计算和存储成本，并带来低资源学习中过拟合（性能不佳）的风险。因此，参数高效的调

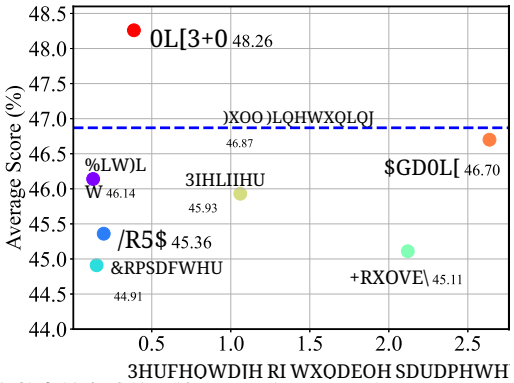


图1. **参数高效方法的比较**。在低资源设置（即，仅有64个训练样本）下，我们展示了VQA v2上五个随机种子的平均得分（纵轴），以及可调参数占预训练VL-T5参数的百分比（横轴）。

优方法应运而生。这类方法只更新新添加的轻量级模块（例如，Houlsby [16], Pfeiffer [41], Compacter [24], 和 AdaMix [54]），或仅更新预训练模型中的极少量参数（例如，BitFit [59]和 LoRA [17]），以应对这些挑战。

然而，如图1所示，前述参数高效微调方法大幅减少了可调参数的数量，但其性能仍落后于全量微调。在这些方法中，基于适配器的方法 [16,24,41,54]支持更灵活的参数共享 [46]，并且存储效率更高，因为它们只需存储一份适配器，而非整个预训练模型。具体而言，AdaMix [54]将适配器用作混合专家（MoE） [45] 架构中的专家，以提升适配器容量，并取得了与全量微调相当的性能。尽管如此，由于适配器之间存在参数冗余，这种混合专家模式会导致可调参数数量增加。

在本文中，我们以基于适配器的方法为基础，探索能够在低资源视觉问答中超越全量微调的参数效率更高的微调方法。具体而言，将预训练视觉语言模型适配到给定任务时，我们着重于两项关键改进：**(*i*)** 在维持适配器容量的同时减少参数冗余。要实现平衡

*通讯作者。

参数效率与模型容量之间的平衡至关重要，因为过度减少可调参数可能导致欠拟合，限制适配器捕获足够任务相关信息的能力 [24]。(*ii*) 减少任务无关冗余，同时促进表征中的任务相关关联。在实践中，适配器利用残差连接，整合从目标数据集中学到的任务特定信息，以及预训练视觉语言模型中已有的先验知识。然而，近期研究 [22, 35, 51] 指出，预训练模型不可避免地包含对目标任务而言冗余且无关的信息，导致表征与标签之间出现统计上的伪相关，从而损害性能和泛化能力 [49, 52]。因此，为提升适配器的有效性，我们的目标是最大限度地获取任务相关信息，同时摒弃预训练视觉语言模型中的任务无关信息。

为此，我们提出了**MixPHM**，这是一种冗余感知参数高效微调方法，能够有效减少可调参数和任务无关冗余，同时增强表征中的任务相关性。MixPHM 以混合专家的方式由多个 PHM 专家实现。为减少(*i*) 参数冗余，MixPHM 首先将专家权重分解并重新参数化到低秩子空间中。随后，它通过全局和局部权重共享进一步减少可调参数，并促进信息传递。为实现(*ii*)中的改进，我们首先量化适配器在表征空间中的冗余。结果表明，适配器表征与预训练视觉语言模型表征存在冗余，但与最终任务使用的表征相关性有限。受此发现启发，我们提出**冗余正则化**，将其纳入 MixPHM，通过去相关化 MixPHM 表征与预训练视觉语言模型表征之间的相似度矩阵，来减少任务无关冗余 。与此同时，通过最大化 MixPHM 表征与最终任务使用的表征之间的互信息，增强任务相关性 。

我们在 VQA v2 [12], 、GQA [20],和 OK-VQA [38], 上开展实验，结果表明，在将预训练视觉语言模型适配到低资源视觉问答时，所提出的 Mix-PHM 始终优于全量微调和最先进的参数高效微调方法。总而言之，我们的贡献如下：(1) 我们提出了 MixPHM，这是一种用于将预训练视觉语言模型适配到下游任务的冗余感知参数高效微调方法。(2) 我们提出了冗余正则化，这是一种新型正则化器，用于减少 MixPHM 表征中的任务无关冗余，同时促进任务相关性。(3) 大量实验表明，MixPHM 在性能与参数效率之间实现了更优的权衡，相较于现有方法带来了显著的性能提升。

2. 相关工作

视觉-语言预训练。 视觉-语言预训练 [6, 9, 19, 21, 25, 31, 48, 63, 65] 旨在学习与任务无关的多模态表征，以通过微调的方式提升下游任务的性能。近年来，一系列研究 [5, 18, 30, 31, 53] 致力于利用编码器-解码器框架和生成式建模目标，统一预训练与微调之间的架构和目标。采用编码器-解码器架构的视觉语言模型具有更好的泛化能力。在本文中，我们探究如何更好地将其适配到低资源视觉问答 [2]。

参数高效微调。 在给定的下游数据集上微调大规模预训练视觉语言模型，已成为视觉语言任务的主流范式。然而，微调由数百万个参数组成的整个模型既耗时又耗费资源。参数高效微调 [13, 36, 37, 44, 58, 59, 62] 通过调整轻量级可训练参数，同时冻结（大部分）预训练参数，在自然语言处理任务中取得了巨大成功。根据是否引入新的可训练参数，这些方法大致可分为两类：（1）调整预训练模型的部分参数，例如 BitFit [59]和 FISH Mask [47], ；（2）调整额外参数，例如提示（前缀）微调 [28, 32], 、适配器 [16, 41], 和低秩方法 [17, 24]。

受自然语言处理领域成功经验的启发，一些研究 [33, 46, 64]开始引入参数高效方法，以调整用于视觉语言任务的预训练视觉语言模型。具体而言，Lin 等人 [33] 研究了用于视觉语言导航的动作级提示。VL-Adapter [46]将适配器扩展到视觉语言模型，以适配各种视觉语言任务。HyperPELT [64]是一个面向视觉语言任务的统一参数高效框架，结合了适配器和前缀微调。此外，Frozen [50] 和 PICa [57]使用提示微调技术 [28], 将大规模预训练语言模型的少样本学习能力迁移到少样本视觉语言任务中。FewVLM [23] 设计了手工设计的提示，以微调预训练视觉语言模型，实现低资源适配。相比之下，低秩方法具有更高的参数效率，但鲜少得到探索。

混合专家。 混合专家 [45] 旨在扩大模型容量，并通过条件计算保持计算效率。近期大多数工作 [7, 8, 27, 29, 42, 43] 都在研究如何在预训练阶段，利用混合专家和精心设计的路由机制构建大规模视觉或语言 Transformer 模型。尽管混合专家在预训练中取得了成功，但其在参数高效微调中的应用尚未得到广泛探索。MPOE [11]和 AdaMix [54]是近期两项利用混合专家微调预训练语言模型的工作。具体而言，MPOE 将额外的 FFN 层视为专家，并通过 MPO 对专家的权重矩阵进行分解。AdaMix 则将新增的适配器视为专家，并通过随机路由策略提升适配器容量。

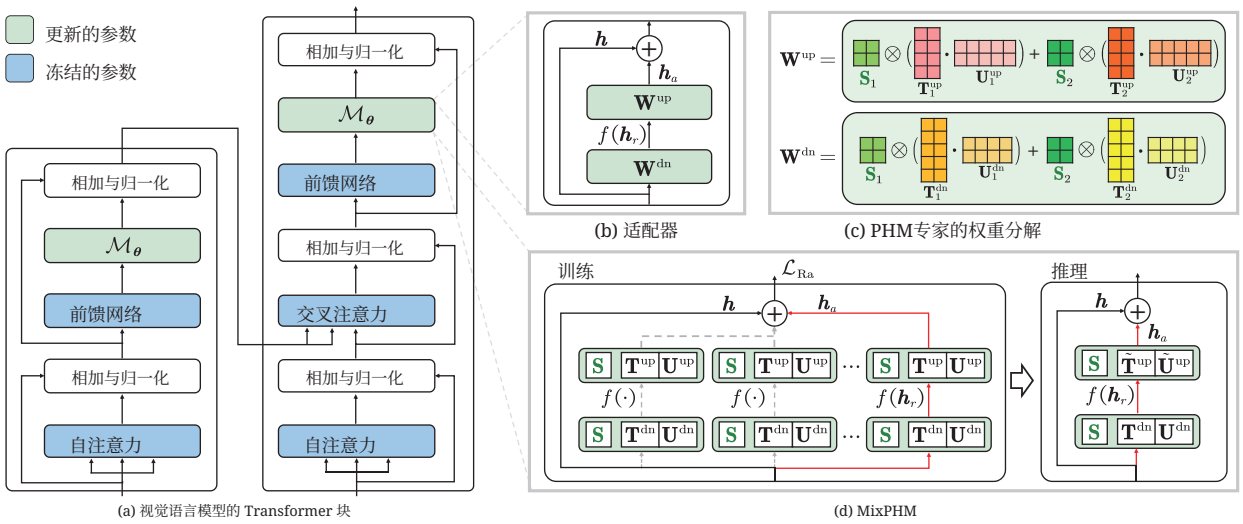


图2。**插入到视觉语言模型**（a）一个 Transformer 块中的 MixPHM 示意图。（b）标准适配器的结构。（c）式（13）中 PHM专家的权重矩阵分解示例（此处 $n=2$, $d=10$, $d_r=8$, $d_k=2$）。（d）具有 $N_e=3$ 个 PHM 专家的 MixPHM 架构。在训练期间，MixPHM 随机激活一个 PHM专家以学习鲁棒表征，并利用所提出的冗余正则化 $\mathcal{L}_{\text{Ra}}$ 来减少任务无关冗余，同时促进任务相关性。

3. 预备知识

问题定义。我们遵循近期工作 [5, 55] 将视觉问答建模为生成式建模任务，即，针对给定问题生成自由形式的文本答案，而非从预定义答案集中选择特定答案。形式上，我们用 $\mathcal{D}=\{(I,Q,y)\in\mathcal{I}\times\mathcal{Q}\times\mathcal{Y}\}$表示一个视觉问答数据集，其中 I 是一张图像，Q 是一个问题，而 y是一个答案。假设给定的预训练视觉语言模型 $\mathcal{M}_\Theta$由少量可调参数 $\boldsymbol{\theta}$参数化，那么将预训练视觉语言模型适配到视觉问答这一一般问题，就是以参数高效方式在 $\mathcal{M}_{\boldsymbol{\theta}}$ 上调优 $\mathcal{D}$。

混合专家。标准的 MoE [45]由一组 N_e 个专家 $\{E_i\}_{i=1}^{N_e}$ 和一个门控网络 G实现。每个专家都是具有独特权重的子神经网络，并且可以从输入的任务特定子集中学习。门控网络有条件地激活 N_a （$1\le N_a\le N_e$）个专家。形式化地，给定输入表示 $\boldsymbol{x}\in\mathbb{R}^d$，第 i 个专家将 $\boldsymbol{x}$ 映射到 d_e 维空间，即，$E_i(\cdot):\boldsymbol{x}\to\mathbb{R}^{d_e}$；门控网络生成一个稀疏的 N_e 维向量，即，$G(\cdot):\boldsymbol{x}\to\mathbb{R}^{N_e}$。然后，MoE 的输出 $\boldsymbol{y}\in\mathbb{R}^{d_e}$ 可表述为

$$\boldsymbol{y}=\sum_{i=1}^{N_e}G(\boldsymbol{x})_iE_i(\boldsymbol{x}),\tag{1}$$

其中，$G(\boldsymbol{x})_i$ 表示将 $\boldsymbol{x}$ 分配给第i个专家的概率，且满足 $\sum_{i=1}^{N_e}G(\boldsymbol{x})_i=1$。

参数化超复数乘法。PHM层 [61] 旨在通过从数据中学习乘法规则，将超复数乘法推广到全连接层。形式上，对于一个将输入 映射为输出的全连接层，将输入 $\boldsymbol{x}\in\mathbb{R}^d$ 映射为输出 $\boldsymbol{y}\in\mathbb{R}^{d_e}$，即。

$$\boldsymbol{y}=\mathbf{W}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{b},\tag{2}$$

其中，$\mathbf{W}\in\mathbb{R}^{d\times d_e}$。在 PHM 中，权重矩阵$\mathbf{W}$ 通过对 n 个$\mathbf{S}_j\in\mathbb{R}^{n\times n}$与$\mathbf{A}_j\in\mathbb{R}^{\frac{d}{n}\times\frac{d_e}{n}}$之间的克罗内克积求和来学习：

$$\mathbf{W}=\sum_{j=1}^{n}\mathbf{S}_j\otimes\mathbf{A}_j,\tag{3}$$

其中，超参数 $n\in\mathbb{Z}_{>0}$ 控制上述求和的次数，d 和 d_e 均可被 n整除，而$\otimes$ 表示克罗内克积，它将向量外积推广到实数空间的高维情形。例如，$\mathbf{S}\in\mathbb{R}^{m\times k}$ 与$\mathbf{A}\in\mathbb{R}^{p\times q}$ 之间的克罗内克积是一个块矩阵$\mathbf{S}\otimes\mathbf{A}\in\mathbb{R}^{mp\times kq}$，即，

$$\mathbf{S}\otimes\mathbf{A}=\begin{bmatrix}s_{11}\mathbf{A} & \cdots & s_{1k}\mathbf{A}\\ \vdots & \ddots & \vdots\\ s_{m1}\mathbf{A} & \cdots & s_{mk}\mathbf{A}\end{bmatrix},\tag{4}$$

其中，s_{ij} 表示矩阵$\mathbf{S}$ 中第 i行、第 j列的元素。因此，用 PHM 替换全连接层，最多可将其可训练参数减少 $1/n$。

4. 方法

我们提出 MixPHM，这是一种具有冗余感知能力的参数高效微调方法，用于适配预训练视觉语言模型。本节首先分析适用于低资源视觉问答的适配器表示空间中的冗余（第 4.1 节）。随后依次详细介绍 MixPHM 的架构（第 4.2 节）、冗余正则化（第 4.3 节）和推理（第4.4 节）。

4.1. 重新思考适配器中的冗余

如图 2 (b) 所示，适配器 [16] 本质上是一个轻量级模块，通常由一个具有瓶颈的两层前馈网络实现，并包含非线性函数和残差连接。在下游任务上微调预训练视觉语言模型时，适配器会插入视觉语言模型的 Transformer 层之间，且只有新添加的适配器参数会更新，而预训练视觉语言模型中的原始参数保持 Frozen。形式上，给定输入表示 $\boldsymbol{h} \in \mathbb{R}^d$，下投影层 $\mathbf{W}^{\text{dn}} \in \mathbb{R}^{d \times d_r}$ 将 $\boldsymbol{h}$ 映射到由瓶颈维度 d_r 指定的低维空间中，即，$\boldsymbol{h}_r \in \mathbb{R}^{d_r}$。上投影层 $\mathbf{W}^{\text{up}} \in \mathbb{R}^{d_r \times d}$ 将 $\boldsymbol{h}_r$ 映射回输入维度，即，$\boldsymbol{h}_a \in \mathbb{R}^d$。考虑到残差和非线性函数 f，适配器定义如下：

$$\boldsymbol{h}_a = f(\boldsymbol{h}\mathbf{W}^{\text{dn}})\mathbf{W}^{\text{up}}, \tag{5}$$

$$\boldsymbol{h} \leftarrow \boldsymbol{h}_a + \boldsymbol{h}. \tag{6}$$

理想情况下，通过结合从给定下游数据集（$\boldsymbol{h}_a$）中学习到的任务特定信息，以及预训练视觉语言模型（$\boldsymbol{h}$）中已编码的先验知识，适配器能够快速将预训练视觉语言模型迁移到新任务，同时避免过度参数化或参数不足。

适配器冗余分析。然而，近期研究表明，适配器捕获的部分信息与任务无关 [14]。为了解实际情况，我们采用表征相似性分析（RSA）[26]来评估表示空间中的冗余。具体而言，我们首先使用 Pfeiffer [41] 方法，在 VQA v2 训练集 [12]的 1千个样本上对预训练 VL-T5 [5] 进行微调。随后，从 VQA v2 验证集随机抽取 1千个样本，并提取每个 Transformer 层的词元级表征（即，$\boldsymbol{h}$ 和 $\boldsymbol{h}_a$），以及 Transformer 编码器/解码器的最终输出表征 $\boldsymbol{h}$。最后，对于每个样本，我们可以获得每一层的 N_t 词元级表征，即$\mathbf{H} = \{\boldsymbol{h}\}_{i=1}^{N_t} \in \mathbb{R}^{N_t \times d}$、$\mathbf{H}_a = \{\boldsymbol{h}_a\}_{i=1}^{N_t} \in \mathbb{R}^{N_t \times d}$ 和$\tilde{\mathbf{H}} = \{\tilde{\boldsymbol{h}}\}_{i=1}^{N_t} \in \mathbb{R}^{N_t \times d}$。在每一层中，我们计算 $\boldsymbol{h}_a$ 与 $\boldsymbol{h}$ 之间，以及 $\boldsymbol{h}_a$ 与 $\boldsymbol{h}$ 之间的RSA相似度，方法是

$$\text{RSA}(\boldsymbol{h}_a, \boldsymbol{h}) = f_\rho(f_\cup[\mathbf{H}_a\mathbf{H}_a^{\text{T}}], f_\cup[\mathbf{H}\mathbf{H}^{\text{T}}]), \tag{7}$$

$$\text{RSA}(\boldsymbol{h}_a, \tilde{\boldsymbol{h}}) = f_\rho(f_\cup[\mathbf{H}_a\mathbf{H}_a^{\text{T}}], f_\cup[\tilde{\mathbf{H}}\tilde{\mathbf{H}}^{\text{T}}]), \tag{8}$$

其中，$f_\cup[\cdot]$ 表示从矩阵中提取上三角元素的操作，f_ρ 是用于计算皮尔逊相关系数的函数。图3展示了1千个样本的平均RSA相似度，这表明在Transformer层中，适配器表征 $\boldsymbol{h}_a$ 与预训练 $\boldsymbol{h}$ 视觉语言模型的表征存在冗余，但与最终输出的相关性有限 $\boldsymbol{h}$。

直观而言，为了高效地将预训练视觉语言模型迁移到下游任务，适配器学习到的表示 $\boldsymbol{h}_a$ 需要尽可能多地包含

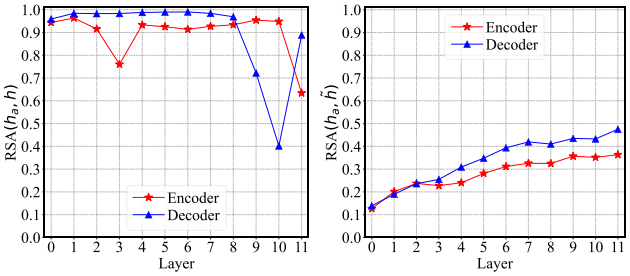


图3. **1千个样本的平均RSA相似度**，展示了各个Transformer层中 $\boldsymbol{h}_a$ 与 $\boldsymbol{h}$（左侧）以及 $\boldsymbol{h}_a$ 与 $\boldsymbol{h}$（右侧）之间的相似度。RSA值越高，表征空间就越相似（冗余）。

与任务相关的表示 $\boldsymbol{h}$，同时减少与预训练视觉语言模型的表示 $\boldsymbol{h}$ 之间与任务无关的冗余。然而，图3 展示了一个违反直觉的结果。因此，为了提高适配器的有效性，至关重要的是促进任务相关的相关性，存在于 $\boldsymbol{h}_a$ 和 $\boldsymbol{h}$ 之间，同时减少与任务无关的冗 $\boldsymbol{h}_a$ 余，存在于 $\boldsymbol{h}$之间。

4.2. MixPHM 架构

如图2所示，MixPHM也是一个轻量级模块，插入到视觉语言模型的每个Transformer块中。我们采用基于Transformer的编码器-解码器模型作为底层预训练视觉语言模型，这些模型由重复的 L编码器块和 L 解码器块组成。具体而言，对于第 l个（$1 \le l \le L$）块，我们在前馈层之后插入一个由一组 N_e PHM 专家 $\{E_i^l\}_{i=1}^{N_e}$ 组成的MixPHM，以捕获知识并学习任务特定信息。与适配器一样，每个PHM专家均由一个包含下投影层和上投影层的瓶颈网络构成。

为减少*MixPHM* 中的参数冗余，我们首先将 MixPHM 中专家的投影矩阵分解并重新参数化到低维子空间中。随后，我们进一步通过全局和局部专家权重共享策略减少参数数量并传递信息。此外，我们采用随机路由 [54,66] 进行专家选择，以避免门控网络引入额外的可训练参数。

参数高效微调。在每次训练（微调）迭代中，我们从插入的 N_e第 l个 Transformer 块中的 PHM 专家里随机选择一个。一旦选定专家E_i^l，给定批次中的所有输入都由同一个专家处理。形式化地，对于第 l个块[1], 中的词元输入表示 $\boldsymbol{h} \in \mathbb{R}^d$，随机选定的第 i个专家对其进行编码和更新 $\boldsymbol{h}$，方式如下：

$$\boldsymbol{h} \leftarrow f(\boldsymbol{h}\mathbf{W}_i^{\text{dn}})\mathbf{W}_i^{\text{up}} + \boldsymbol{h}. \tag{9}$$

在公式（9）中，下投影矩阵 $\mathbf{W}_i^{\text{dn}} \in \mathbb{R}^{d \times d_r}$ 和上投影矩阵 $\mathbf{W}_i^{\text{up}} \in \mathbb{R}^{d_r \times d}$ 首先被分解

[1]为简洁起见，后文省略上标 l。

通过 PHM 转换为低维矩阵，即，

$$\mathbf{W}_i^{\text{dn}} = \sum_{j=1}^{n} \mathbf{S}_{i,j} \otimes \mathbf{A}_{i,j}^{\text{dn}}, \quad \mathbf{W}_i^{\text{up}} = \sum_{j=1}^{n} \mathbf{S}_{i,j} \otimes \mathbf{A}_{i,j}^{\text{up}}, \quad (10)$$

其中，$\mathbf{S}_{i,j} \in \mathbb{R}^{n\times n}$，$\mathbf{A}_{i,j}^{\text{dn}} \in \mathbb{R}^{\frac{d}{n}\times d_r}$，$\mathbf{A}_{i,j}^{\text{up}} \in \mathbb{R}^{d_r \times \frac{d}{n}}$。为了进一步提高参数效率，矩阵$\mathbf{A}_{i,j}^{\text{dn}}$（$\mathbf{A}_{i,j}^{\text{up}}$）进一步分解为两个低秩矩阵，方法是

$$\mathbf{A}_{i,j}^{\text{dn}} = \mathbf{T}_{i,j}^{\text{dn}}(\mathbf{U}_{i,j}^{\text{dn}})^{\mathrm{T}}, \quad \mathbf{A}_{i,j}^{\text{up}} = \mathbf{T}_{i,j}^{\text{up}}(\mathbf{U}_{i,j}^{\text{up}})^{\mathrm{T}}, \quad (11)$$

其中，$\mathbf{T}_{i,j}^{\text{dn}} \in \mathbb{R}^{\frac{d}{n}\times d_k}$、$\mathbf{U}_{i,j}^{\text{dn}} \in \mathbb{R}^{\frac{d_r}{n}\times d_k}$、$\mathbf{T}_{i,j}^{\text{up}} \in \mathbb{R}^{\frac{d_r}{n}\times d_k}$、$\mathbf{U}_{i,j}^{\text{up}} \in \mathbb{R}^{\frac{d}{n}\times d_k}$，且 d_r为这些矩阵的秩。最后，我们通过以下方式学习第 i 个 PHM专家的权重矩阵：

$$\mathbf{W}_i^{\text{dn}} = \sum_{j=1}^{n} \mathbf{S}_{i,j} \otimes (\mathbf{T}_{i,j}^{\text{dn}}(\mathbf{U}_{i,j}^{\text{dn}})^{\mathrm{T}}), \quad (12)$$

$$\mathbf{W}_i^{\text{up}} = \sum_{j=1}^{n} \mathbf{S}_{i,j} \otimes (\mathbf{T}_{i,j}^{\text{up}}(\mathbf{U}_{i,j}^{\text{up}})^{\mathrm{T}}). \quad (13)$$

PHM专家间的信息共享。使用 MixPHM 在下游数据集上微调预训练视觉语言模型时，第 i 个 PHM专家的 n 矩阵 $\{\mathbf{S}_{i,j} \in \mathbb{R}^{n\times n}\}_{j=1}^{n}$ 在所有 Transformer 块中的 PHM 专家之间全局共享，以捕获目标任务的通用信息。换言之，一个专家的 $\mathbf{S} = \{\mathbf{S}_j\}_{j=1}^{n} \in \mathbb{R}^{n\times n\times n}$由所有新增的 MixPHM 共享。相反，$\{\mathbf{A}_{i,j}^{\text{dn}}\}_{j=1}^{n}$ 和 $\{\mathbf{A}_{i,j}^{\text{up}}\}_{j=1}^{n}$ 是专家专属权重矩阵，每个 PHM专家各不相同。为了更好地在 MixPHM 的 PHM 专家之间传递信息，并进一步减少参数冗余，我们在每个 MixPHM 中的 PHM 专家之间局部共享 $\{\mathbf{A}_{i,j}^{\text{dn}}\}_{j=1}^{n}$。

此时，使用MixPHM插入预训练视觉语言模型的可训练参数总数已从原来的 $4LN_e(dd_r)$减少至 $2Ld_k(d+d_r)(N_e+1)+n^3$。

4.3. 冗余正则化

受第 4.1 节所述洞见的启发，我们提出了冗余正则化。具体而言，对于 l 个 Transformer 块中的 MixPHM，我们将一个批次的 token 级输出表示 $\{\boldsymbol{h}_a\}_{i=1}^{N}$ 及其残差 $\{\boldsymbol{h}\}_{i=1}^{N}$分别集成为 $\mathbf{Z}_a \in \mathbb{R}^{N\times d}$ 和 $\mathbf{Z} \in \mathbb{R}^{N\times d}$，$N = N_b N_t$（$N_b$表示批次大小）。对于 Transformer编码器/解码器，我们沿 token 维度对一个批次的最终输出表示 $\{\tilde{\boldsymbol{h}}\}_{i=1}^{N}$进行平均，得到全局任务相关表示 $\{\bar{\boldsymbol{h}}\}_{i=1}^{N_b}$。随后，冗余正则化可表示为

$$\mathcal{L}_{\text{Ra}} \triangleq \sum_{i}^{N}\sum_{j\neq i}^{N} \frac{\mathbf{Z}_a\mathbf{Z}^{\mathrm{T}}}{\|\mathbf{Z}_a\|_2 \|\mathbf{Z}\|_2} - \sum_{i}^{N_b}\sum_{j}^{N_t} \hat{\mathcal{I}}(\boldsymbol{h}_{a\,i,j}; \bar{\boldsymbol{h}}_i), \quad (14)$$

其中，$\|\cdot\|_2$ 表示 L_2 范数，$\boldsymbol{h}_{a\,i,j}$ 是批次中第 i个样本的第 j个 token 的输出表示，并且 $\hat{\mathcal{I}}(\cdot;\cdot)$ 表示互信息（MI）的估计值，用于最大化两个表示之间的相关性，并通过 JSD互信息估计器计算 [15]。在冗余正则化 $\mathcal{L}_{\text{Ra}}$中，第一项是冗余减少项，它通过促使 $\boldsymbol{h}_a$ 与 $\boldsymbol{h}_a$ 之间的余弦相似度矩阵的非对角元素趋近于零，鼓励 $\boldsymbol{h}$ 从预训练视觉语言模型中丢弃任务无关信息。第二项旨在促使 $\boldsymbol{h}_a$ 包含更多来自下游数据集的任务相关信息，方法是最大化 $\bar{}$与 $\boldsymbol{h}_a$ 之间的互信息 $\boldsymbol{h}$。

将视觉问答表述为生成任务后，训练目标是最小化给定输入图像 I 和问题 Q时答案y 词元的负对数似然。因此，参数高效微调中的总训练损失为

$$\mathcal{L} = -\sum_{j=1}^{|y|} \log P_{\boldsymbol{\theta}}(y_j|y_{<j}; I, Q) + \alpha\mathcal{L}_{\text{Ra}}, \quad (15)$$

whe其中，α 是用于平衡冗余正则化的因子 项。

4.4. 推理

与训练期间使用的随机路由不同，我们采用权重聚合策略 [56]，在推理时为每个 MixPHM 得到一个最终的 PHM专家。具体而言，一个 MixPHM 有 N_e 个 PHM 专家。在低秩子空间中学习权重时，每个专家都有 $2n$专家专属矩阵 $\{\mathbf{T}_j^{\text{up}}、\mathbf{U}_j^{\text{up}}\}_{j=1}^{n}$，而这些 N_e 专家具有相同的 $2n$ 局部共享矩阵 $\{\mathbf{T}_j^{\text{dn}}、\mathbf{U}_j^{\text{dn}}\}_{j=1}^{n}$，以及 n 全局共享矩阵 $\{\mathbf{S}_j\}_{j=1}^{n}$。为获得最终 PHM专家的权重，我们首先对相应的上投影矩阵权重进行平均合并N_e。从数学上讲，第 j 个上投影矩阵可计算为

$$\widetilde{\mathbf{T}}_j^{\text{up}} = \frac{1}{N_e}\sum_{i=1}^{N_e} \mathbf{T}_{ji}^{\text{up}}, \quad \widetilde{\mathbf{U}}_j^{\text{up}} = \frac{1}{N_e}\sum_{i=1}^{N_e} \mathbf{U}_{ji}^{\text{up}}. \quad (16)$$

由于采用了全局和局部权重共享，我们无需对 $\mathbf{S}$ 和 $\{\mathbf{T}_j^{\text{dn}}, \mathbf{U}_j^{\text{dn}}\}_{j=1}^{n}$进行权重聚合。最后，我们使用合并后的专家，在每个 Transformer 块中计算 MixPHM 的输出表示。

5. 实验

5.1. 实验设置

数据集与指标。我们在三个数据集上进行实验：VQA v 2 [12], GQA [20], 和 OK-VQA [38]。为模拟 VQA 的低资源设置，我们遵循相关研究 [4]，并将 $N_{\mathcal{D}}$ 低资源视觉问答的训练数据量设定为少于 1,000。为了更贴近实际地开展低资源学习，我们遵循真正的少样本学习 [10, 40]，并使用开发集 $\mathcal{D}_{\text{dev}}$，其大小与训练集相同 $\mathcal{D}_{\text{train}}$（即，

数据集	方法	参数数量 (M)	参数数量 (%)	样本数量 $N_{\mathcal{D}}$=16	$N_{\mathcal{D}}$=32	$N_{\mathcal{D}}$=64	$N_{\mathcal{D}}$=100	$N_{\mathcal{D}}$=500	$N_{\mathcal{D}}$=1,000
VQA v2 [12]	微调	224.54	100%	41.82 ±1.58	43.09 ±3.10	46.87 ±0.57	48.12 ±0.87	53.46 ±0.41	55.56 ±0.13
	BitFit [59]	0.29	0.13%	40.61 ±4.15	43.86 ±2.19	46.14 ±1.00	47.53 ±0.67	51.91 ±0.40	53.18 ±0.58
	LoRA [17]	0.44	0.20%	41.60 ±2.27	42.62 ±2.41	45.36 ±1.66	47.57 ±0.91	51.93 ±0.38	54.15 ±0.45
	Compacter [24]	0.34	0.15%	39.28 ±1.87	42.47 ±2.76	44.91 ±1.27	46.28 ±1.37	51.21 ±0.90	53.39 ±0.54
	Houlsby [16]	4.76	2.12%	41.71 ±2.16	44.01 ±2.09	45.11 ±1.40	47.71 ±0.78	52.27 ±1.05	54.31 ±0.34
	Pfeiffer [41]	2.38	1.06%	41.48 ±1.86	44.18 ±2.13	45.93 ±1.11	47.42 ±1.15	52.35 ±0.52	53.98 ±0.38
	AdaMix [54]	5.92	2.64%	40.59 ±2.05	43.42 ±2.08	46.70 ±1.32	47.34 ±0.91	51.72 ±1.05	54.12 ±0.63
	MixPHM	0.87	0.39%	43.13 ±1.78	45.97 ±2.01	48.26 ±0.56	49.91 ±0.76	54.30 ±0.33	56.11 ±0.40
GQA [20]	微调	224.54	100%	28.24 ±2.08	30.80 ±2.49	34.22 ±0.59	36.15 ±0.99	41.49 ±0.54	43.04 ±0.57
	BitFit [59]	0.29	0.13%	26.13 ±2.83	29.00 ±4.81	34.25 ±1.16	35.91 ±1.22	40.08 ±0.42	41.84 ±0.15
	LoRA [17]	0.44	0.20%	26.89 ±2.74	30.40 ±2.27	34.40 ±0.99	36.14 ±1.10	40.20 ±1.02	42.06 ±1.12
	Compacter [24]	0.34	0.15%	23.70 ±2.10	27.18 ±2.61	32.70 ±1.30	35.28 ±1.45	38.68 ±0.50	41.17 ±0.95
	Houlsby [16]	4.76	2.12%	25.13 ±2.32	28.34 ±1.17	33.23 ±0.94	35.88 ±1.79	40.85 ±0.48	41.90 ±0.72
	Pfeiffer [41]	2.38	1.06%	25.08 ±1.81	29.18 ±1.32	32.97 ±0.84	35.08 ±1.01	40.30 ±0.40	41.39 ±0.27
	AdaMix [54]	5.92	2.64%	24.62 ±2.34	28.01 ±1.33	32.74 ±0.96	35.64 ±0.94	40.14 ±0.42	41.97 ±0.86
	MixPHM	0.87	0.39%	28.33 ±2.63	32.40 ±2.52	36.75 ±0.55	37.40 ±0.87	41.92 ±0.55	43.81 ±0.50
OK-VQA [38]	微调	224.54	100%	11.66 ±2.08	14.20 ±0.78	16.65 ±1.02	18.28 ±0.67	24.07 ±0.40	26.66 ±0.72
	BitFit [59]	0.29	0.13%	11.29 ±1.79	13.66 ±1.49	15.29 ±0.57	16.51 ±0.53	22.54 ±0.57	24.80 ±0.63
	LoRA [17]	0.44	0.20%	10.26 ±1.53	12.46 ±1.82	15.95 ±0.38	17.03 ±0.82	23.02 ±0.41	25.26 ±0.53
	Compacter [24]	0.34	0.15%	9.64 ±2.73	11.04 ±1.39	13.57 ±1.07	15.92 ±1.18	22.20 ±0.89	24.52 ±0.59
	Houlsby [16]	4.76	2.12%	9.79 ±1.71	12.25 ±2.13	15.04 ±1.25	16.58 ±0.65	22.67 ±0.77	25.04 ±0.44
	Pfeiffer [41]	2.38	1.06%	9.06 ±0.53	11.39 ±0.79	14.23 ±1.54	16.34 ±0.79	22.90 ±1.03	26.70 ±0.71
	AdaMix [54]	5.92	2.64%	8.39 ±1.20	11.55 ±1.37	13.66 ±2.29	16.27 ±0.92	23.20 ±0.78	26.34 ±0.88
	MixPHM	0.87	0.39%	13.87 ±2.39	16.03 ±1.23	18.58 ±1.42	20.16 ±0.97	26.08 ±0.88	28.53 ±0.85

表1。 **使用预训练VL-T5的实验结果**。 在 VQA v2 验证集、GQA test-dev集和 OK-VQA测试集上评估了 5 个随机种子的平均 VQA-Score 及其标准差。 最佳 和 次佳 参数高效微调方法以高亮显示。报告了调优参数的数量，以及相对于 VL-T5（即，224.54M）的调优参数百分比。

$|\mathcal{D}_{\text{dev}}| = |\mathcal{D}_{\text{train}}| = N_{\mathcal{D}}$），而不是使用大规模验证集来进行最佳模型选择和超参数调优。具体而言，我们在 $N_{\mathcal{D}} \in \{16, 32, 64, 100, 500, 1000\}$ 上进行实验。为构建 $\mathcal{D}_{\text{train}}$ 和 $\mathcal{D}_{\text{dev}}$，我们从 VQA v2 的训练集中随机抽取 $2N_{\mathcal{D}}$ 个样本，并将它们平均分为 $\mathcal{D}_{\text{train}}$ 和 $\mathcal{D}_{\text{dev}}$。同理，我们为 GQA 和 OK-VQA 构建 $\mathcal{D}_{\text{train}}$ 和 $\mathcal{D}_{\text{dev}}$。VQA-Score [2]是低资源VQA任务的准确率指标。

基线方法。 我们将我们的方法与几种最先进的参数高效微调方法和微调进行比较。为确保公平比较，我们对它们的关键超参数（KHP）进行超参数搜索，并报告其最佳性能。具体而言，

• **BitFit** [59] 仅调整预训练模型的偏置权重，同时冻结其余参数。

• **LoRA** [17] 调整额外的低秩矩阵，用于近似每个 Transformer 自注意力层和交叉注意力层中的查询权重和值权重。KHP 是矩阵秩（r）。

• **Compacter** [24]在 Transformer 块的每个前馈层后添加适配器，并使用低秩PHM层对适配器权重进行重新参数化 [61]。KHP 分别表示 Kronecker 积的求和项数量（n）、瓶颈维度（d_r）和 r。

• **Houlsby** [16] 在每个 Transformer 块的自注意力层和前馈层之后添加适配器。KHP 是 d_r。

方法	模型大小	参数数量（%）	VQAv2	GQA	OK-VQA
Frozen [50]	7B	-	38.2	12.6	-
PICa-Base [57]	175B	-	54.3	43.3	-
PICa-Full [57]	175B	-	**56.1**	**48.0**	-
FewVLM [23]	225M	100%	48.2	32.2	15.0
MixPHM†	226M	0.39%	49.3	33.4	**19.2**

表2。**与少样本学习器（$N_{\mathcal{D}}$=64）的比较**。FewVLM 是一种基于提示的全量微调方法。MixPHM† 表示以参数高效方式使用 MixPHM 对 FewVLM 进行微调。

• **Pfeiffer** [41] 根据先导实验确定适配器的位置。在这项工作中，我们将其放置在每个 Transformer 前馈层之后。KHP 是 d_r。

• **AdaMix** [54]以混合专家的方式，在每个 Transformer 前馈层之后添加多个适配器。KHP 是适配器的数量（N_e），并且 d_r。

实现细节。我们使用四种预训练视觉语言模型，即，VL-T5 [5], X-VLM [60], BLIP [30], 和 OFA_{Base} [53]作为底层编码器-解码器 Transformer，它们在微调时将视觉问答建模为生成任务，因此不会引入额外的视觉问答头参数。由于 VL-T5 使用的原始预训练数据集包含上述视觉问答数据集中的样本，我们改为加载权重[2]，这些权重由 Jin等人发布 [23],，并且是在剔除重叠样本后重新训练得到的。所有结果均基于五个随机种子报告 $\{13, 21, 42, 87, 100\}$。更多细节和超参数设置见附录D。

[2] https://github.com/woojeongjin/FewVLM

方法	VQA v2	GQA	OK-VQA
微调	46.87 ± 0.57	34.22 ± 0.59	16.65 ± 1.02
MixPHM*	47.30 ± 0.67	34.66 ± 0.78	18.05 ± 1.16
+ $\mathcal{L}_{\text{cs}}$	46.70 ± 0.66	34.83 ± 1.35	17.37 ± 1.38
+ $\mathcal{L}_{\text{Ra}}^{\text{I}}$	47.42 ± 0.71	34.69 ± 0.96	18.25 ± 1.54
+ $\mathcal{L}_{\text{Ra}}^{\text{II}}$	47.71 ± 0.85	36.10 ± 0.83	18.21 ± 1.08
+ $\mathcal{L}_{\text{Ra}}$	**48.26** ± 0.56	**36.75** ± 0.55	**18.58** ± 1.42

表3。 **使用不同正则化器的消融实验** $N_{\mathcal{D}} = 64$。 MixPHM* 表示不使用任何正则化器的基线。 $\mathcal{L}_{\text{cs}}$ 是一致性正则化器 [66]。 $\mathcal{L}_{\text{Ra}}^{\text{I}}$ 和 $\mathcal{L}_{\text{Ra}}^{\text{II}}$ 分别表示仅使用其中第一项和第二项 $\mathcal{L}_{\text{Ra}}$。

5.2. 低资源视觉问答

表1 展示了预训练 VL-T5 [5] 在三个数据集上的结果。总体而言，我们的 MixPHM 优于最先进的参数高效微调方法，也是唯一一种始终优于全量微调的方法。接下来，我们将详细讨论不同的比较结果。

与 AdaMix 的比较。AdaMix 和 MixPHM 都采用混合专家来提升适配器的容量，但 MixPHM 还考虑进一步减少参数冗余并学习更多任务相关表示。表 1显示，在不同 $N_{\mathcal{D}}$ 的所有数据集上，我们的 MixPHM 均显著优于 AdaMix 和全量微调，而 AdaMix 未能超越全量微调（VQA v2 在 $N_{\mathcal{D}} = 32$的情况下除外）。这一结果证明，MixPHM 在性能和参数效率方面都有效，也表明在减少参数冗余的同时，提示任务相关的修正信息十分重要。

与 Houlsby 和 Pfeiffer 的比较。MixPHM 中的 PHM专家与适配器具有相同的瓶颈结构。然而，由于减少了参数冗余，PHM专家具有更高的参数效率；并且得益于所提出的冗余正则化，它能够更好地捕捉任务相关信息。表 1中的结果显示，与微调相比，Houlsby 和 Pfeiffer 在大多数低资源设置下的表现远未达到上限。相反，所提出的 MixPHM 在所有数据集设置下均展现出性能和参数效率方面的优势。

与 Compacter 的比较。为减少参数冗余，Compacter 和 MixPHM 使用低秩PHM对适配器权重进行重新参数化。然而，在降低参数冗余的同时，MixPHM 通过冗余正则化鼓励与任务相关的校正，并借助混合专家提升模型容量，从而避免过拟合和参数不足的问题。表1显示，Compacter 的表现未如预期。一种可能的解释是，Compacter 在低资源VQA任务上参数不足。因为可训练参数过少，并不能保证模型在微调阶段捕获足够的任务相关信息。这表明，在针对低资源视觉问答对预训练视觉语言模型进行微调时，有必要

WD D1	WD D2	WS S	WS $\mathbf{A}^{\text{dn}}$	WS $\mathbf{A}^{\text{up}}$	参数数量 (M)	VQA v2	GQA	OK-VQA
微调					224.54			
✓					2.45	48.15 ± 0.89	36.42 ± 0.64	17.34 ± 1.59
✓		✓	✓		1.55	47.79 ± 1.11	36.59 ± 0.64	**18.** **77** ± 0.99
✓	✓	✓	✓		0.87	**48.26** ± 0.56		
✓	✓				1.37	47.67 ± 1.13	36.22 ± 0.89	17.65 ± 2.38
✓	✓	✓			1.36	48.05 ± 0.99	**36.76** ± 0.78	17.02 ± 1.70
✓	✓			✓	0.83	47.30 ± 0.92	36.05 ± 1.05	17.27 ± 0.63
✓	✓	✓		✓	0.82	47.83 ± 0.65	36.39 ± 0.84	17.75 ± 1.36
✓	✓		✓		0.88	47.78 ± 1.20	36.57 ± 0.81	18.07 ± 1.73
✓	✓	✓	✓		0.87	**48.26** ± 0.56		

表4. **权重分解（WD）和权重共享（WS）的消融实验**。D1：使用 PHM 对 Mix-PHM 中的专家权重进行分解。D2：对分解后的权重进一步进行低秩重参数化。

平衡效果与参数效率。

与 LoRA 和 BitFit 的比较。LoRA 和 BitFit 是两种典型的参数高效方法，它们只微调原始预训练视觉语言模型的部分参数，因此参数效率更高。结果见表1。我们观察到，与 MixPHM 相比，LoRA 和 BitFit 的可调参数量相对较少。然而，它们的表现远逊于 MixPHM。尤其是，随着 $N_{\mathcal{D}}$ 增加，性能差距也随之扩大。与对 Compacter 的讨论类似，微调过程中可训练参数过少，可能会导致模型在给定数据集上过拟合。

与最先进的小样本学习器的比较。小样本视觉问答是低资源视觉问答的一种特殊情况。与最先进的多模态小样本学习器的比较结果如表2所示。Frozen [50] 和 PICa [57]是两种上下文学习方法，它们采用提示微调，在不调整参数的情况下迁移大规模语言模型（即 GPT-2 和 GPT-3 [3]）。而 FewVLM [23] 是一种基于提示的全量微调方法，用于适配 VL-T5；它将手工设计的提示插入模型输入，并对模型的全部参数进行微调。我们采用 MixPHM 以参数高效方式微调 FewVLM。只需调整极少量参数，MixPHM 的小样本性能便优于 FewVLM，尤其是在 OK-VQA 上（**19.2** 比 15.0）。这表明 MixPHM 在性能和参数效率方面具有优势。

5.3. 消融研究

我们使用预训练的 VL- T5，在 VQA v2、GQA 和 OK-VQA 上进行所有消融实验，设置为 $N_{\mathcal{D}} = 64$。

冗余正则化的有效性。为了证明所提出的冗余正则化的有效性，我们首先引入一个一致性正则化器 $\mathcal{L}_{\text{cs}}$ [66]进行比较。此外，为了进一步分析 $\mathcal{L}_{\text{Ra}}$中不同项的作用，我们考虑 $\mathcal{L}_{\text{Ra}}$的两种变体：（*i*） $\mathcal{L}_{\text{Ra}}^{\text{I}}$：训练时仅使用公式（14）中的第一项作为正则化器。（*ii*） $\mathcal{L}_{\text{Ra}}^{\text{II}}$：训练时仅使用公式（14）中的第二项。表3 显示

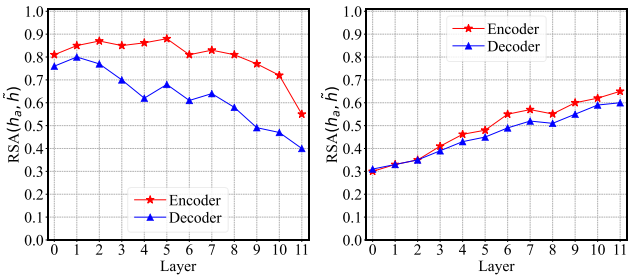


图4. **MixPHM 在每个 Transformer 层中，与（左）以及与（右）之间的平均 RSA 相似度。**

$\mathcal{L}_{cs}$ 仅提升了 MixPHM 在 GQA 上的性能，且提升幅度较小。相比之下，$\mathcal{L}_{Ra}$ 显著提升了 MixPHM 在所有数据集上的性能。这一观察结果证明了所提出的正则化器 $\mathcal{L}_{Ra}$的有效性和优越性。此外，在分析 $\mathcal{L}_{Ra}^{I}$ 和 $\mathcal{L}_{Ra}^{II}$ 对 MixPHM 性能的影响时，我们发现，仅减少 MixPHM 表征与预训练视觉语言模型表征之间的冗余（即., $\mathcal{L}_{Ra}^{I}$）对性能提升的贡献有限。然而，$\mathcal{L}_{Ra}^{I}$ 和 $\mathcal{L}_{Ra}^{II}$ 的联合效果优于单独使用$\mathcal{L}_{Ra}^{II}$。这表明，在减少任务无关冗余与促进任务相关性之间取得平衡，对 MixPHM 至关重要。

降低参数冗余的影响。MixPHM 首先使用 PHM 分解专家权重（D1），然后对分解后的权重重新参数化（D2），从而降低参数冗余。我们通过消融 D1 和 D2，分析它们对 MixPHM 性能的影响（即表4的第三列）。此外，权重共享还能进一步降低 MixPHM 中的参数冗余。因此，我们在表4的第四列中，对不同的共享权重组合进行了消融实验。除了全局共享矩阵（$\mathbf{S}$）之外，我们还在单个 MixPHM 中的专家之间局部共享下投影（$\mathbf{A}^{dn}$）或上投影（$\mathbf{A}^{up}$）矩阵。表4显示，参数效率与性能之间存在权衡，即过度减少参数可能会损害性能。因此，我们主张在降低参数冗余的同时维持模型容量。

超参数的影响。关于超参数（N_e、d_r、d_k、n、α）的结果见附录 A。

5.4. 讨论

MixPHM 的冗余分析。本文提出了一项见解：通过减少表征中的任务无关冗余并促进任务相关性，可以提升适配器的有效性。为评估基于这一见解提出的方法是否确实能够带来性能提升，我们在 MixPHM 的表示空间中，按照第 4.1 节所述的相同实验设置进行了冗余分析。图 4 展示了 1千个样本之间的 RSA相似度

VLMs	方法	参数数量 (M)	样本数量 $N_{\mathcal{D}}$=16	$N_{\mathcal{D}}$=64	$N_{\mathcal{D}}$=500	$N_{\mathcal{D}}$=1000
X-VLM [60]	微调	294	26.63	30.45	38.96	43.92
	MixPHM	0.66	**27.54**	**31.80**	**41.05**	**48.06**
BLIP [30]	微调	385	27.01	30.05	37.00	42.22
	MixPHM	0.87	**29.17**	**32.09**	**41.80**	**46.78**
OFA_{Base} [53]	微调	180	27.48	31.75	42.99	46.81
	MixPHM	0.70	**28.46**	**33.00**	**45.88**	**50.01**

表5. **MixPHM 在其他预训练视觉语言模型上的实验结果**。我们报告了不同低资源设置下，VQA v2 验证集上五个随机种子的平均 VQA-Score。

在 VQA v2 上。与图 1 所示的适配器冗余分析结果相比，我们观察到，MixPHM 显著降低了 $\boldsymbol{h}_a$ 与 $\boldsymbol{h}$之间的表征冗余，同时提高了 $\tilde{}$与 $\boldsymbol{h}_a$ 之间的任务相关性 $\boldsymbol{h}$。这一发现直观地证明了我们动机的合理性以及方法的有效性。

对其他预训练视觉语言模型的泛化能力。为展示我们的方法在其他预训练视觉语言模型上的泛化能力，我们将 MixPHM 应用于适配预训练 X-VLM [60], BLIP [30], 和 OFA_{Base} [53]，以用于低资源视觉问答任务。表 5 展示了我们的方法与 VQA v2 上全量微调的简要对比。我们观察到，MixPHM 在所有设置下都持续优于全量微调，且在 $N_{\mathcal{D}} \in \{500, 1000\}$时性能提升尤为显著。值得注意的是，X-VLM（**+4.14**）、BLIP（**+4.80**）和 OFA_{Base}（**+3.20**）在 $N_{\mathcal{D}}$=1000、$N_{\mathcal{D}}$=500和 $N_{\mathcal{D}}$=1000时，相较于微调取得了最大的性能差距。这些结果表明，MixPHM 能够泛化至多种预训练视觉语言模型。附录C中提供了参数高效方法的更多比较。

6. 结论与局限

本文提出了一种具有冗余感知能力的参数高效调优方法，用于将预训练视觉语言模型适配到低资源VQA任务。所提出的MixPHM通过引入冗余正则化，减少了任务无关冗余，同时促进了任务相关性。实验结果表明，该方法在性能和参数效率方面均具有有效性和优越性。

冗余是一把双刃剑。除了减少任务无关冗余，我们还可以利用预训练视觉语言模型已经学到的任务相关冗余来提升性能。尽管 MixPHM 强调减少任务无关冗余，但并不能明确保证所减少的冗余对给定任务毫无用处。因此，一个值得探索的方向是如何明确界定并尽量减少任务无关冗余。**致谢**。 本研究获得国家自然科学基金（项目编号62088102）的资助。

附录

在附录中，我们为实验提供了一些补充材料。具体而言，附录 A 探讨了不同路由机制和超参数对 MixPHM 的影响。附录B 展示了我们方法的一些可视化结果。附录C提供了使用预训练 X-VLM 在 VQA v2 上进行参数高效微调方法的更多比较。附录 D介绍了更多实现细节。

A. 消融研究与参数分析

在本节中，我们使用预训练视觉语言模型 VL-T5 [5]作为底层预训练视觉语言模型，在 VQA v2、GQA 和 OK-VQA 上开展了关于路由机制的额外消融实验以及超参数分析 $N_{\mathcal{D}} = 64$。

路由机制的影响。 在 MixPHM 中，除了性能之外，路由机制还会影响训练速度，即每次迭代时间（秒）。为了分析不同路由策略对性能和速度的影响，我们首先介绍两种随机路由方法，即令牌级路由和句子级路由 [8]。此外，我们通过对每个 MixPHM 中所有 PHM 专家的输出取平均，设计了一种简单的基于表示的舍入方法。表 6显示，随机路由机制速度最快，并且在 VQA v2 和 OK-VQA 上均具有最佳性能。

方法	每次迭代时间（秒）	VQA v2	GQA	OK-VQA
MixPHM-Token	0.693	47.67 ±0.92	36.23 ±0.89	17.77 ±0.89
MixPHM-Sent	0.683	47.69 ±0.99	36.13 ±0.86	17.83 ±1.32
MixPHM-Rep	0.675	48.00 ±0.95	**36.77** ±0.55	18.25 ±1.46
MixPHM	0.668	**48.26** ±0.56	36.75 ±0.55	**18.58** ±1.42

表 6 $N_{\mathcal{D}} = 64$。**不同路由机制的消融实验**。每次迭代时间（秒）为每次迭代的平均调优时间。

超参数的影响。 为了研究不同超参数对 MixPHM 的影响，我们通过改变 N_e、 d_r、 d_k、 n来开展实验。更具体地说，我们考虑以下设置： $N_e \in \{1, 2, 4, 8, 12\}$、$d_r \in \{48, 64, 96, 192\}$、 $d_k \in \{1, 8, 16, 24\}$和 $n \in \{2, 4, 8, 16\}$。表 7 中的结果表明，改变这些超参数只会对 MixPHM 的性能产生轻微影响。此外，采用不同超参数的 MixPHM 的性能始终优于全量微调。这表明，MixPHM 带来的性能提升并不显著依赖于超参数选择。

HP		参数数量	VQA v2	GQA	OK-VQA
微调		224.54	46.87 ±0.57	34.22 ±0.59	16.65 ±1.02
N_e	1	0.34	47.30 ±0.97	36.30 ±0.83	17.59 ±0.97
	2	0.52	47.90 ±0.65	**36.88** ±0.75	18.08 ±1.28
	4	0.87	**48.26** ±0.56	36.75 ±0.55	**18.58** ±1.42
	8	1.59	48.09 ±0.67	36.50 ±0.81	18.51 ±1.29
	12	2.30	47.80 ±0.72	36.30 ±0.80	18.43 ±1.50
d_r	48	0.86	**48.36** ±0.97	36.36 ±0.32	18.05 ±0.85
	64	0.87	48.26 ±0.56	**36.75** ±0.55	**18.58** ±1.42
	96	0.91	48.05 ±0.82	36.36 ±0.51	18.39 ±0.96
	192	1.00	47.97 ±1.17	36.37 ±0.82	18.26 ±1.20
d_k	1	0.18	47.87 ±0.73	35.74 ±0.70	17.04 ±0.81
	8	0.87	48.26 ±0.56	**36.75** ±0.55	18.58 ±1.70
	16	1.67	**48.35** ±1.14	36.62 ±0.35	18.22 ±1.36
	24	2.47	48.07 ±1.12	36.42 ±0.52	**18.79** ±1.18
n	2	0.87	48.17 ±0.93	36.53 ±0.32	18.43 ±0.75
	4	0.87	**48.26** ±0.56	**36.75** ±0.55	**18.58** ±1.42
	8	0.87	47.97 ±1.08	36.37 ±0.56	17.41 ±1.05
	16	0.88	46.65 ±1.10	35.46 ±0.55	17.52 ±0.63

表7。**超参数（HP）对 MixPHM 的影响**。 N_e：PHM 专家的数量，d_r：瓶颈维度， d_k：秩维度， n：克罗内克积的求和次数。

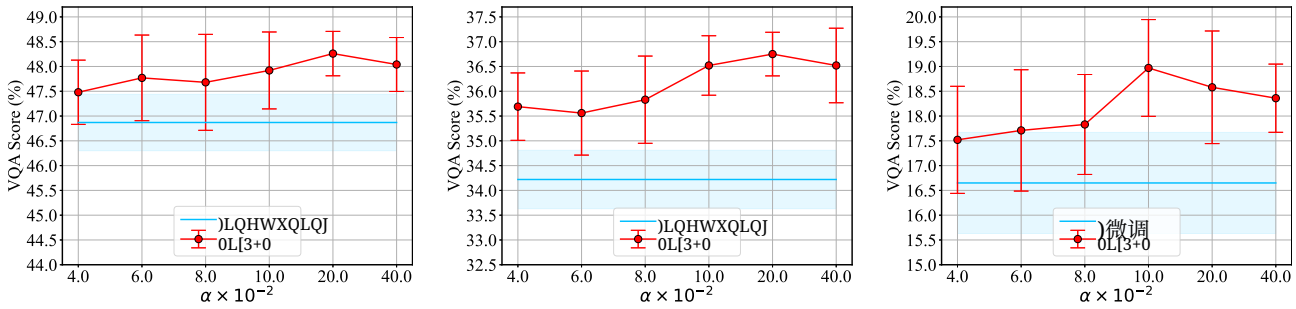
.....的影响 α。使用 Mix-PHM 调优预训练视觉语言模型时， α 控制着冗余正则化与生成建模损失之间的权衡。为研究 α 对 MixPHM 的影响，我们针对 α的不同取值（即） $\alpha \in \{0.04, 0.06, 0.08, 0.1, 0.2, 0.4\}$进行了实验。图 5 展示了 VQA-Score 随着 α 增大而变化的曲线。我们发现，在一定范围内改变 α， $[0.04, 0.4]$并不会削弱 MixPHM 相较于全量微调的优势。此外，根据三个数据集上的结果，我们将 α 经验性地设为 0.2。

B. 可视化结果

我们展示了所提出的 MixPHM 的一些示例。如图6 所示，这些预测由在 VQA v2 上使用 MixPHM 微调的 VL-T5 $N_{\mathcal{D}} = 64$生成。此外，我们展示了图像的自上而下注意力 [1]，并标出问题中排名前两位的任务相关词元。具体而言，我们遵循一项近期工作 [22]，计算任务相关表征与使用自下而上的 Faster R-CNN 提取的视觉输入特征之间的注意力分数 [1]，并展示得分最高和次高位置对应的自上而下注意力。同样，我们计算任务相关表征与问题的语言嵌入之间的分数，并标出得分最高和次高位置对应的词元。图6 定性地表明，在答案预测过程中，我们的 MixPHM 能够生成一致且与问题相关的视觉和文本注意力。

C. 使用预训练 X-VLM 的结果

作为正文表5结果的补充，我们使用预训练 X-VLM [60]作为代表模型，并在 VQA v2 验证集上将我们的方法与最先进的参数高效微调方法进行比较。这些参数高效方法的关键超参数设置与表9中的设置相同。我们在表8中观察到的结论与表



(a) VQA v2 (b) GQA (c) OK-VQA

图 5. **随着 α 变化，五个随机种子上的平均 VQA-Score 及其标准差。**

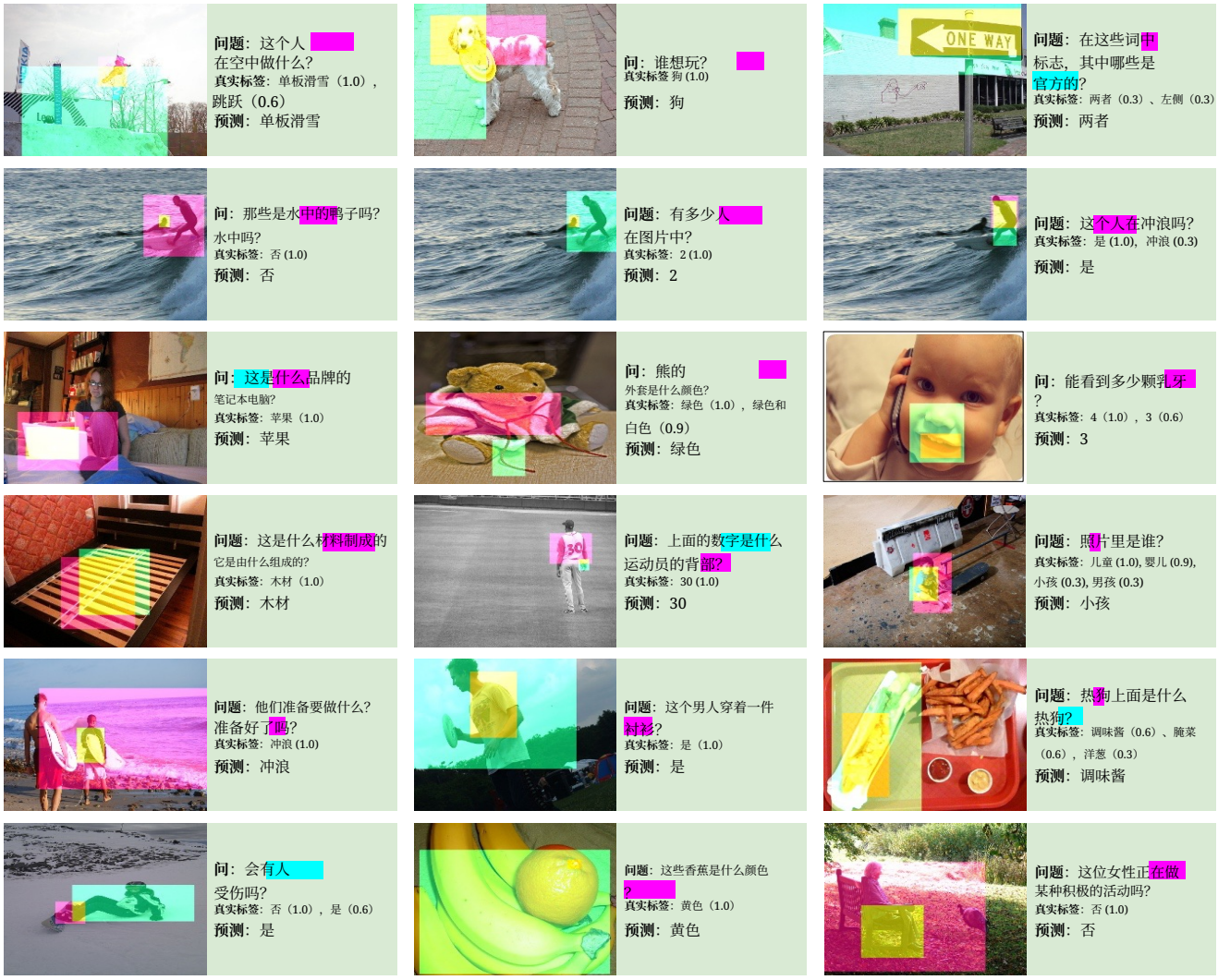


图6. **VQA v2 验证集上的定性结果。** 预测 **由采用所提出的 MixPHM 进行微调的 VL-T5 生成。** **真实标签** 表示标注的答案及其对应分数。我们将图像的自上而下注意力可视化 [1]，并标记出问题中与任务相关的词元中注意力分数最高的 第 一和第二 个词元。

5一致，即，当使用其他预训-

训练后的视觉语言模型，进一步证明了 MixPHM 的泛化能力。

方法	参数数量 (M)	参数数量 (%)	样本数量 $N_{\mathcal{D}}$=16	$N_{\mathcal{D}}$=32	$N_{\mathcal{D}}$=64	$N_{\mathcal{D}}$=100	$N_{\mathcal{D}}$=500	$N_{\mathcal{D}}$=1,000
微调	293.48	100%	26.63 $\pm$0.98	29.33 $\pm$1.68	30.45 $\pm$1.80	31.48 $\pm$1.57	38.96 $\pm$1.56	43.92 $\pm$1.22
BitFit [59]	0.29	0.13%	25.48 $\pm$3.81	28.90 $\pm$1.14	30.73 $\pm$1.18	31.92 $\pm$1.14	36.77 $\pm$1.32	40.77 $\pm$0.79
LoRA [17]	0.37	0.13%	25.31 $\pm$1.50	26.91 $\pm$3.09	30.52 $\pm$1.67	31.97 $\pm$1.11	36.13 $\pm$1.12	40.49 $\pm$0.87
Compacter [24]	0.25	0.09%	25.69 $\pm$2.34	28.04 $\pm$1.63	28.10 $\pm$2.06	31.35 $\pm$0.34	35.91 $\pm$0.65	40.44 $\pm$0.77
Houlsby [16]	3.57	1.20%	26.54 $\pm$2.57	29.34 $\pm$2.25	30.74 $\pm$1.20	31.71 $\pm$1.43	38.48 $\pm$0.91	41.96 $\pm$0.72
Pfeiffer [41]	1.78	0.60%	26.57 $\pm$2.00	28.46 $\pm$1.74	29.22 $\pm$2.56	31.95 $\pm$1.34	37.39 $\pm$0.73	40.96 $\pm$1.09
AdaMix [54]	4.44	1.49%	26.11 $\pm$1.58	28.91 $\pm$1.36	30.71 $\pm$2.05	31.15 $\pm$1.26	38.48 $\pm$1.53	43.26 $\pm$0.85
MixPHM	0.66	0.22%	27.54 $\pm$1.52	30.65 $\pm$1.09	31.80 $\pm$1.61	32.58 $\pm$1.09	41.05 $\pm$1.22	48.06 $\pm$0.64

表 8。 **使用预训练 X-VLM 的实验结果**。 在 VQA v2 验证集上评估了 5 个不同随机种子的平均 VQA-Score 和标准差。 最佳 和 次佳 参数高效微调方法以高亮显示。报告了调优参数的数量及其相对于 X-VLM（即，293.48M）的百分比。

方法	学习率	配置
微调	5×10^{-5}	—
BitFit	5×10^{-5}	—
LoRA	5×10^{-5}	$r = 4$
Compacter	5×10^{-3}	$d_r = 64, d_k = 8, n = 4$
Houlsby	5×10^{-5}	$d_r = 64$
Pfeiffer	5×10^{-5}	$d_r = 64$
AdaMix	5×10^{-4}	$N_e = 4, d_r = 64$
MixPHM	5×10^{-3}	$N_e = 4, d_r = 64, d_k = 8, n = 4$

表9. **所有参数高效微调方法的超参数设置**。 N_e：专家数量， d_r：瓶颈维度， d_k 和 r：秩维度， n：克罗内克积的求和次数。

D. 实现细节

对于参数高效微调方法，我们搜索瓶颈维度 d_r 至 $\{48, 64, 96, 192\}$，适用于所有基于适配器的方法（即 MixPHM、AdaMix、Pfeiffer、Houlsby 和 Compacter），搜索专家数量 N_e 至$\{1, 2, 4, 8, 12\}$，适用于 MixPHM 和 AdaMix；搜索秩维度 d_r （适用于 MixPHM 和 Compacter）、r （适用于 LoRA），范围为$\{1, 8, 16, 24\}$；此外，还搜索克罗内克积的求和次数 n，范围为 $\{2, 4, 8, 16\}$，适用于 MixPHM 和 Compacter。表9展示了我们实验中使用的超参数最终配置。对于 MixPHM，我们将权衡因子 α 设为 0.2。

所有方法均使用 Pytorch [39] 在 NVIDIA GeForce RTX 3090Ti GPU 上实现。此外，我们还通过网格搜索，从 $\{5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}\}$中选择最佳学习率。批次大小和训练轮次分别设为 16 和 1000。我们使用 AdamW优化器 [34]以及早停策略，耐心值为 200 个非递增轮次，停止指标为开发集上的视觉问答得分 $\mathcal{D}_{\text{dev}}$。

参考文献

[1]Peter Anderson、Xiaodong He、Chris Buehler、Damien Teney、Mark Johnson、Stephen Gould 和 Lei Zhang。《用于图像描述和视觉问答的自下而上与自上而下注意力》。发表于 CVPR，第 6077–6086 页，2018 年。9、10。[2] Stanislaw Antol、Aishwarya Agrawal、Jiasen Lu、Margaret Mitchell、Dhruv Batra、C Lawrence Zitnick 和 Devi Parikh。《VQA：视觉问答》。发表于 ICCV，第 2425–2433 页，2015 年。1、2、6。[3] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等。《语言模型是少样本学习器》。发表于 NeurIPS，第 1877–1901 页，2020。7[4] Guanzheng Chen、Fangyu Liu、Zaiqiao Meng 和 Shang-song Liang。《重新审视参数高效微调：我们真的已经做到位了吗？》。发表于 EMNLP，第 2612–2626 页，2022。5[5] Jaemin Cho、Jie Lei、Hao Tan 和 Mohit Bansal。《通过文本生成统一视觉与语言任务》。发表于 ICML，第 1931–1942 页，2021。1、2、3、4、6、7、9[6] Zi-Yi Dou、Yichong Xu、Zhe Gan、Jianfeng Wang、Shuohang Wang、Lijuan Wang、Chenguang Zhu、Pengchuan Zhang、Lu Yuan、Nanyun Peng 等。《端到端视觉与语言 Transformer 训练的实证研究》。发表于 CVPR，第 18166–18176 页，2022。1、2[7] Nan Du、Yanping Huang、Andrew M Dai、Simon Tong、Dmitry Lepikhin、Yuanzhong Xu、Maxim Krikun、Yanqi Zhou、Adams Wei Yu、Orhan Firat 等。《GLAM：利用混合专家高效扩展语言模型》。发表于 ICML，第 5547–5569 页，2022。2[8] William Fedus、Barret Zoph 和 Noam Shazeer。《Switch Transformer：以简单高效的稀疏性扩展至万亿参数模型》。arXiv 预印本 arXiv:2101 03961，2021。2、9。[9] Zhe Gan、Yen-Chun Chen、Linjie Li、Chen Zhu、Yu Cheng 和 Jingjing Liu。《面向视觉-语言表征学习的大规模对抗训练》。发表于 NeurIPS，2020。2[10] Tianyu Gao、Adam Fisch 和 Danqi Chen。《让预训练语言模型成为更好的少样本学习器》。发表于 ACL，第 3816–3830 页，2021。5

[11] Ze-Feng Gao、Peiyu Liu、Wayne Xin Zhao、Zhong-Yi Lu 和 Ji-Rong Wen。《面向预训练语言模型的参数高效混合专家架构》。发表于 COLING，第 3263–3273 页，2022。2[12] Yash Goyal、Tejas Khot、Douglas Summers-Stay、Dhruv Batra 和 Devi Parikh。《让 VQA 中的 V 发挥作用：提升图像理解在视觉问答中的作用》。发表于 CVPR，第 6904–6913 页，2017年。2、4、5、6[13] Junxian He、Chunting Zhou、Xuezhe Ma、Taylor Berg-Kirkpatrick 和 Graham Neubig。《迈向参数高效迁移学习的统一视角》。发表于 ICLR，2022。2[14] Ruidan He、Linlin Liu、Hai Ye、Qingyu Tan、Bosheng Ding、Liying Cheng、Jia-Wei Low、Lidong Bing 和 Luo Si。《基于适配器的微调对预训练语言模型适配的有效性》。发表于 ACL，第 2208–2222 页，2021年。4[15] R Devon Hjelm、Alex Fedorov、Samuel Lavoie-Marchildon、Karan Grewal、Phil Bachman、Adam Trischler 和 Yoshua Bengio。《通过互信息估计与最大化学习深层表征》。发表于 ICLR，2019年。5[16] Neil Houlsby、Andrei Giurgiu、Stanislaw Jastrzebski、Bruna Morrone、Quentin De Laroussilhe、Andrea Gesmundo、Mona Attariyan 和 Sylvain Gelly。《自然语言处理的参数高效迁移学习》。发表于 ICML，第 2790–2799 页，2019年。1、2、4、6、11[17] Edward J Hu、Yelong Shen、Phillip Wallis、Zeyuan Allen-Zhu、Yuanzhi Li、Shean Wang、Lu Wang 和 Weizhu Chen。《LoRA：大型语言模型的低秩适配》。发表于 ICLR，2022。1、2、6、11[18] Ronghang Hu 和 Amanpreet Singh。《Unit：采用统一变换器的多模态多任务学习》。发表于 ICCV，第 1439–1449 页，2021年。2[19] Zhicheng Huang、Zhaoyang Zeng、Yupan Huang、Bei Liu、Dongmei Fu 和 Jianlong Fu。《跳出框外看：面向视觉-语言表征学习的端到端预训练》。发表于 CVPR，第 12976–12985 页，2021年。2[20] Drew A Hudson 和 Christopher D Manning。《GQA：面向真实世界视觉推理和组合式问答的新数据集》。发表于 CVPR，第 6700–6709 页，2019年。2、5、6[21] Chao Jia、Yinfei Yang、Ye Xia、Yi-Ting Chen、Zarana Parekh、Hieu Pham、Quoc Le、Yun-Hsuan Sung、Zhen Li 和 Tom Duerig。《利用噪声文本监督扩展视觉及视觉-语言表征学习》。发表于 ICML，第 4904–4916 页，2021年。2[22] Jingjing Jiang、Ziyi Liu 和 Nanning Zheng。《相关性信息瓶颈：面向鲁棒视觉问答的预训练多模态模型适配》。arXiv 预印本 arXiv:2209.06954，2022。2、9[23] Woojeong Jin、Yu Cheng、Yelong Shen、Weizhu Chen 和 Xiang Ren。《好的提示价值数百万参数：面向视觉-语言模型的低资源提示学习》。发表于 ACL，第 2763–2775 页，2022。2、6、7[24] Rabeeh Karimi Mahabadi、James Henderson 和 Sebastian Ruder。《Compacter：高效的低秩超复数适配器层》。发表于 NeurIPS，第 1022–1035 页，2021。1、2、6、11

[25] Wonjae Kim、Bokyung Son 和 Ildoo Kim。ViLT：一种无需卷积或区域监督的视觉-语言 Transformer。载于 ICML，第 5583 5594, 2021 1, 2 – 页。

[26] Aarre Laakso 和 Garrison Cottrell。内容与聚类分析：评估神经系统中的表征相似性。《哲学心理学》，13(1):47–76，2000。4[27] Dmitry Lepikhin、HyoukJoong Lee、Yuanzhong Xu、Dehao Chen、Orhan Firat、Yanping Huang、Maxim Krikun、Noam Shazeer 和 Zhifeng Chen。GShard：通过条件计算和自动分片扩展巨型模型。载于 ICLR，2021。2[28] Brian Lester、Rami Al-Rfou 和 Noah Constant。规模对于参数高效提示调优的力量。载于 EMNLP，第 3045–3059 页，2021。2[29] Mike Lewis、Shruti Bhosale、Tim Dettmers、Naman Goyal 和 Luke Zettlemoyer。基础层：简化大型稀疏模型的训练。载于 ICML，第 6265–6274 页，2021。2[30] Junnan Li、Dongxu Li、Caiming Xiong 和 Steven Hoi。BLIP：通过自举进行语言-图像预训练，以实现统一的视觉语言理解与生成。载于 ICML，第 12888–12900 页，2022。1, 2, 6, 8[31] Junnan Li、Ramprasaath Selvaraju、Akhilesh Gotmare、Shafiq Joty、Caiming Xiong 和 Steven Chu Hong Hoi。先对齐再融合：通过动量蒸馏进行视觉与语言表征学习。载于 NeurIPS，第 9694–9705 页，2021。1, 2[32] Xiang Lisa Li 和 Percy Liang。Prefix-Tuning：优化用于生成的连续提示。载于 ACL，第 4582–4597 页，2021。2[33] Bingqian Lin、Yi Zhu、Zicong Chen、Xiwen Liang、Jianzhuang Liu 和 Xiaodan Liang。ADAPT：利用模态对齐动作提示进行视觉-语言导航。载于 CVPR，第 15396–15406 页，2022。2[34] Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。载于 ICLR，2019。11[35] Rabeeh Karimi Mahabadi、Yonatan Belinkov 和 James Henderson。用于有效低资源微调的变分信息瓶颈。载于 ICLR，2021。2[36] Rabeeh Karimi Mahabadi、Sebastian Ruder、Mostafa Dehghani 和 James Henderson。通过共享超网络对 Transformer 进行参数高效多任务微调。载于 ACL，第 565–576 页，2021。2[37] Yuning Mao、Lambert Mathias、Rui Hou、Amjad Almahairi、Hao Ma、Jiawei Han、Scott Yih 和 Madian Khabsa。UniPELT：用于语言模型参数高效调优的统一框架。载于 ACL，第 6253–6264 页，2022。2[38] Kenneth Marino、Mohammad Rastegari、Ali Farhadi 和 Roozbeh Mottaghi。OK-VQA：一种需要外部知识的视觉问答基准。载于 CVPR，第 3195–3204 页，2019。2, 5, 6[39] Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga 等。PyTorch：一种采用命令式风格的高性能深度学习库。载于 NeurIPS，第 8024 8035, 2019 11 – 页。

[40] Ethan Perez、Douwe Kiela 和 Kyunghyun Cho。借助语言模型实现真正的少样本学习。载于 NeurIPS，第 11054–11070 页，2021。5

[41] Jonas Pfeiffer、Aishwarya Kamath、Andreas R¨uckl´e、Kyunghyun Cho 和 Iryna Gurevych。AdapterFusion：用于迁移学习的非破坏性任务组合。arXiv 预印本 arXiv:2005.00247，2020。1, 2, 4, 6, 11[42] Carlos Riquelme、Joan Puigcerver、Basil Mustafa、Maxim Neumann、Rodolphe Jenatton、Andr´e Susano Pinto、Daniel Keysers 和 Neil Houlsby。利用稀疏专家混合扩展视觉模型。载于 NeurIPS，第 8583–8595 页，2021。2[43] Stephen Roller、Sainbayar Sukhbaatar、Jason Weston 等。面向大型稀疏模型的哈希层。载于 NeurIPS，第 17555–17566 页，2021。2[44] Andreas R¨uckl´e、Gregor Geigle、Max Glockner、Tilman Beck、Jonas Pfeiffer、Nils Reimers 和 Iryna Gurevych。AdapterDrop：论适配器在 Transformer 中的效率。载于 EMNLP，第 7930–7946 页，2021。2[45] Noam Shazeer、Azalia Mirhoseini、Krzysztof Maziarz、Andy Davis、Quoc Le、Geoffrey Hinton 和 Jeff Dean。极其庞大的神经网络：稀疏门控专家混合层。载于 ICLR，2017。1, 2, 3[46] Yi-Lin Sung、Jaemin Cho 和 Mohit Bansal。VL-Adapter：面向视觉与语言任务的参数高效迁移学习。载于 CVPR，第 5227–5237 页，2022。1, 2[47] Yi-Lin Sung、Varun Nair 和 Colin A Raffel。利用固定稀疏掩码训练神经网络。载于 NeurIPS，第 24193–24205 页，2021。2[48] Hao Tan 和 Mohit Bansal。LXMERT：从 Transformer 中学习跨模态编码器表征。载于 EMNLP，第 5099–5110 页，2019。2[49] Naftali Tishby 和 Noga Zaslavsky。深度学习与信息瓶颈原理。arXiv 预印本 arXiv:1503.02406，2015。2[50] Maria Tsimpoukelli、Jacob Menick、Serkan Cabi、SM Eslami、Oriol Vinyals 和 Felix Hill。利用冻结语言模型进行多模态少样本学习。载于 NeurIPS，第 200–212 页，2021。2, 6, 7[51] Boxin Wang、Shuohang Wang、Yu Cheng、Zhe Gan、Ruoxi Jia、Bo Li 和 Jingjing Liu。InfoBERT：从信息论视角提升语言模型的鲁棒性。载于 ICLR，2021。2[52] Haoqing Wang、Xun Guo、Zhi-Hong Deng 和 Yan Lu。重新思考对比学习中的最小充分表征。载于 CVPR，第 16041–16050 页，2022。2[53] Peng Wang、An Yang、Rui Men、Junyang Lin、Shuai Bai、Zhikang Li、Jianxin Ma、Chang Zhou、Jingren Zhou 和 Hongxia Yang。OFA：通过简单的序列到序列学习框架统一架构、任务与模态。载于 ICML，第 23318–23340 页，2022。1, 2, 6, 8[54] Yaqing Wang、Subhabrata Mukherjee、Xiaodong Liu、Jing Gao、Ahmed Hassan Awadallah 和 Jianfeng Gao。AdaMix：用于大型语言模型参数高效调优的适配器混合。载于 EMNLP，第 5744–5760 页，2022。1, 2, 4, 6, 11[55] Zirui Wang、Jiahui Yu、Adams Wei Yu、Zihang Dai、Yulia Tsvetkov 和 Yuan Cao。SimVLM：采用弱监督的简单视觉语言模型预训练。载于 ICLR，2022。3[56] Mitchell Wortsman、Gabriel Ilharco、Samir Ya Gadre、Rebecca Roelofs、Raphael Gontijo-Lopes、Ari S Morcos、Hongseok Namkoong、Ali Farhadi、Yair Carmon、Simon Ko- rnblith 等。模型汤：对多个微调模型的权重进行平均，无需增加推理时间即可提升准确率。发表于 ICML，第 23965–23998 页，2022 年。 5[57] Zhengyuan Yang、Zhe Gan、Jianfeng Wang、Xiaowei Hu、Yu- mao Lu、Zicheng Liu 和 Lijuan Wang。关于 GPT-3 少样本知识型视觉问答的实证研究。发表于 AAAI，第 3081–3089 页，2022。2、6、7[58] Zonghan Yang 和 Yang Liu。关于面向文本分类的稳健 Prefix-Tuning。发表于 ICLR，2022。2[59] Elad Ben Zaken、Shauli Ravfogel 和 Yoav Goldberg。BitFit：面向基于 Transformer 的掩码语言模型的简单参数高效微调。发表于 ACL，第 1–9 页，2022。1、2、6、11[60] Yan Zeng、Xinsong Zhang 和 Hang Li。多粒度视觉语言预训练：将文本与视觉概念对齐。发表于 ICML，第 25994–26009 页，2022。1、6、8、9[61] Aston Zhang、Yi Tay、Shuai Zhang、Alvin Chan、Anh Tuan Luu、Siu Cheung Hui 和 Jie Fu。超越四元数全连接层：用参数化超复数乘法来表示 $1/n$ 参数。发表于 ICLR，2021。3、6[62] Ningyu Zhang、Luoqiu Li、Xiang Chen、Shumin Deng、Zhen Bi、Chuanqi Tan、Fei Huang 和 Huajun Chen。可微提示使预训练语言模型成为更出色的少样本学习器。发表于 ICLR，2022。2[63] Pengchuan Zhang、Xiujun Li、Xiaowei Hu、Jianwei Yang、Lei Zhang、Lijuan Wang、Yejin Choi 和 Jianfeng Gao。VinVL：重新审视视觉语言模型中的视觉表征。发表于 CVPR，第 5579–5588 页，2021。2[64] Zhengkun Zhang、Wenya Guo、Xiaojun Meng、Yasheng Wang、Yadao Wang、Xin Jiang、Qun Liu 和 Zhenglu Yang。HyperPELT：统一的参数高效语言模型调优方法，适用于语言任务和视觉与语言任务。arXiv 预印本 arXiv:2203.03878，2022。2[65] Yiwu Zhong、Jianwei Yang、Pengchuan Zhang、Chunyuan Li、Noel Codella、Liunian Harold Li、Luowei Zhou、Xiyang Dai、Lu Yuan、Yin Li 等。RegionCLIP：基于区域的语言图像预训练。发表于 CVPR，第 16793 16803 页，2022 2 –。[66] Simiao Zuo、Xiaodong Liu、Jian Jiao、Young Jin Kim、Hany Hassan、Ruofei Zhang、Tuo Zhao 和 Jianfeng Gao。驯服具有随机专家的稀疏激活 Transformer。发表于 ICLR，2022。4、7