**PEMT: Multi-Task Correlation Guided Mixture-of-Experts Enables Parameter-Efficient Transfer Learning**

PEMT:多任务关联引导的专家混合实现参数高效迁移学习

**太长不看版**

为了克服现有方法限制，本文提出了⼀种基于多任务迁移学习的参数⾼效微调(Parameter-Efficient)框架PEMT，扩展了专家混合(MoE)框架，将可转移的知识作为经过源任务训练的适应器的加权组合来捕获。这些权重由⼀个⻔控单元决定，使⽤任务描述提⽰向量衡量⽬标和每个源任务之间的相关性，并提出了任务稀疏性损失(Task Sparsity Loss)来提⾼⻔控单元的稀疏性。实验结果表明，PEMT⽐完全微调和最先进的PEFT和知识转移⽅法在各种任务上取得了稳定的改进。

**摘要**

⽬前，将知识从⼀个或多个任务迁移到下游⽬标任务的⽅法，要么在单个任务上训练适应器，要么从源任务中提取共享知识，未能充分利⽤任务特定知识以及源任务和⽬标任务之间的相关性。为了克服这些限制，本文提出了⼀种基于多任务迁移学习的参数⾼效(Parameter-Efficient)微调框架PEMT，扩展了专家混合(MoE)框架，将可转移的知识作为经过源任务训练的适应器的加权组合来捕获。这些权重由⼀个⻔控单元决定，使⽤任务描述提⽰向量衡量⽬标和每个源任务之间的相关性。为了充分利⽤特定任务的知识，还提出了任务稀疏性损失(TaskSparsityLoss)来提⾼⻔控单元的稀疏性。实验结果表明，PEMT⽐完全微调和最先进的PEFT和知识转移⽅法在各种任务上取得了稳定的改进。

**1引言**

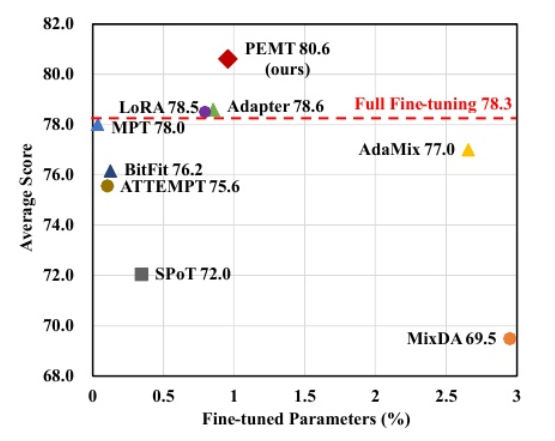


图1:不同参数高效微调方法的性能。所有结果都基于T5-base。

目前，从一个或多个源任务中提取知识，并将其适应于各种下游目标任务的方法有以下局限性:(1)专注于利用所有源任务的共享知识，忽略了任务特定知识。(2)独立训练源任务和目标任务的特定任务(3)源任务和目标任务的制定不一致，阻碍了跨任务适应。(4)用作初始化的来自源任务的知识与下游任务交织在一起，并逐渐被遗忘。

为了解决这些挑战，本文提出了PEMT，一种促进多任务迁移学习的两阶段参数高效微调方法。通过对源任务进行训练的适应器组合捕获可转换的知识，有效地利用task-specific的知识。作者提出了一个基于任务相关性的门控单元，通过测量源和目标下游任务之间的相关性来确定每个源适应器的权重。为了捕获任务之间的相互依赖关系，引入了一系列提示向量来描述每个任务。

实验结果表明，在广泛的任务范围内，PEMT始终优于全微调和最先进的PEFT方法，对于使用4-32标签的少量学习也是有效的，证明了方法的鲁棒性和适应性。此外，作者分析了在各种设置下性能如何变化，为所提出的方法的有效性提供了清晰的解释。

**2 相关工作**

关于Parameter-Efficient微调和多任务迁移学习。

**3 方法**

给定一组K个源任务S = {, ，…，}和一组M个目标任务T = {, ，…，}，目标是捕获S的知识并使其适应于任何∈T的目标任务。

**3.1 源训练**

第1阶段的目标是捕获每个源任务的特定任务知识。为此使用基于适应器的PEFT方法对多个源任务上的PLM进行微调。

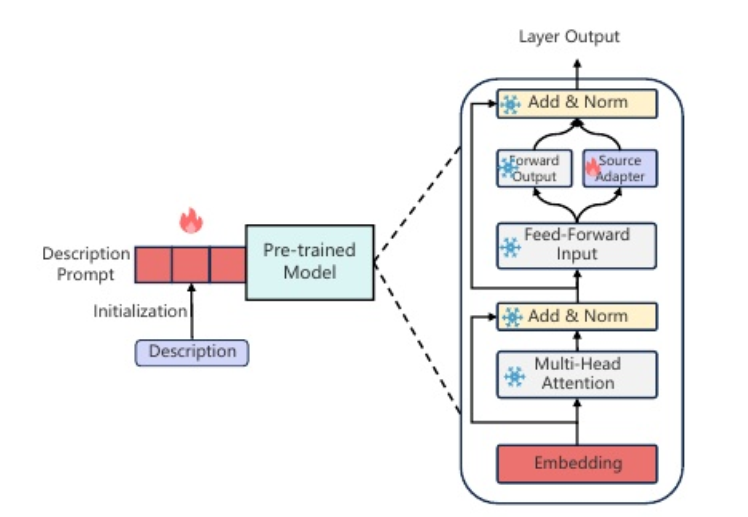


图2:阶段1的训练过程。

如图2所示，任务适配器被注入到每个transformer层中。一个 transformer FFN由两个堆叠层组成，一个向上投影层和一个向下投影层。作者使用并行适配器或与向上投影层并行工作的LoRA将适配器模块集成到FFN。为了减少开销，任务适应器被实现为两个堆叠的低秩矩阵。

作者为每个源任务引入了一个任务描述提示，用来衡量任务之间的相关性，并使用手工的任务描述作为提示向量的初始化。具体来说，给定源任务∈S，在PLM的输入token前加上一个可训练的提示矩阵，其中*d*为嵌入维数，为任务描述的长度(即提示长度)。任务描述是一句话，由基于各种任务特征的任务定义和输入输出格式组成。

任务适应器和任务描述提示都按照典型的PEFT过程进行训练，但对源任务没有特别的要求，统一制定为文本到文本的生成任务。

**3.2 目标适应**

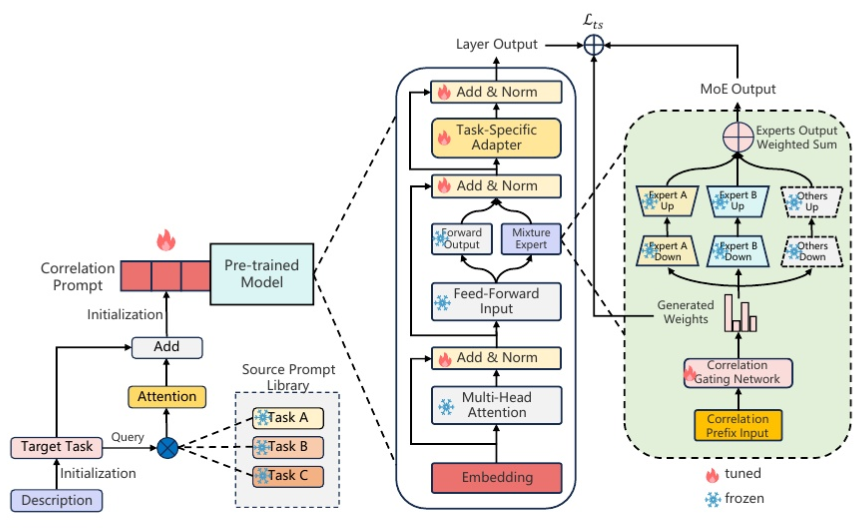
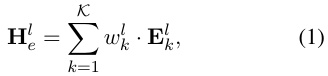


图3:第二阶段的训练过程。

在第二阶段，PEMT以任务之间的相关性为指导，利用所有源任务的提炼知识来适应下游目标任务。

作者使用专家混合(MoE)模块将源适应器组合为可转移的知识，在适应下游任务期间维护每个源任务的特定任务信息。如图3所示，在阶段1中训练的任务适应器被用作MoE模块中的专家。作者冻结专家的参数，而不是对源任务适应器进行微调，以避免灾难性的遗忘问题。通常，第l层MoE模块的输出计算为:



其中 是在第K个源任务上训练的第l个Transformer层中的任务适应器，K是源任务的总数。为的权重，由MoE门计算得到，计算公式为:

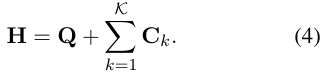


其中是一个可训练矩阵，avg是一个平均池化层。*H*是当前目标任务的提示矩阵，它捕获了任务之间的相关性。

该文提出，将在源任务上训练的提示信息融入基于注意力机制的目标适应过程中。注意力函数*attn(Q, K, V)*接受query，key和value三个输入，目标提示作为query，源提示作为key和value。形式上，让作为目标任务的可训练提示矩阵，初始化为T个token的任务描述，表示第T个提示向量。给定第k个源任务的提示，得到源任务与目标任务的相关特征为:



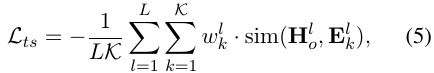
目标任务的最终提示矩阵通过以下计算得到:



这个设计的灵感来自于词嵌入的加法构成。目标任务的提示捕获源任务和目标任务之间的表示信息和相互依赖性，并用于度量每个源任务适应器的权重。PEMT的所有Transformer层共享相同的H以提高效率。

为了适应下游任务，作者将另一个task-specific的适应器合并到每个Transformer层中。目标任务适应器(插入在MoE模块之后)被用来挖掘在源任务上受过培训的专家未涵盖的知识。源任务适应器和目标任务适应器的组合使模型既能利用从每个源任务中学习到的丰富知识，又能利用目标任务的特定任务知识。

作者还提出了任务稀疏损失(Task Sparsity Loss, TSL)来提高MoE模块的稀疏性。这种方法通过测量特定专家输出与最后一层输出之间的相似性，来确保MoE门为top-1源任务专家分配更高的优先级。形式上，TSL定义为:



其中表示第L个Transformer层的最终隐藏状态，L是总层数，sim是相似度得分函数，本文选择余弦相似度。

与源任务上的训练过程类似，将目标任务表述为文本到文本的生成问题。训练目标是最小化输出y在输入文本x和任务提示h条件下的负对数似然。最后，目标任务上的微调损失定义为:



其中α是一个平衡损失的超参数。

**4 实验**

作者在广泛的NLP数据集上进行了实验，以证明PEMT的有效性。并在全数据集和少样本设置下比较了不同方法的性能。

**4.1 数据集和任务**

源任务:MNLI、QNLI、QQP、SST-2、SQuAD和ReCoRD。

目标任务:MultiRC、BoolQ、WiC、WSC和SuperGLUE 的CB;RTE、CoLA、STSB、MRPC来自 GLUE ; 来自MRQA 的Natural Questions (NQ)、HotpotQA、NewsQA 和SearchQA;Wino-Grande，Yelp- 2的F1，sciail和PAWS-Wiki。

方法比较: (1)全微调(2)提示微调，包括vanilla提示微调，SPoT，ATTEMPT和MPT。(3)基于适应器的调优，包括 vanilla 适应器、AdaMix 和MixDA。(4)其他参数高效调优方法，包括LoRA和BitFit。

**4.2 实现**

使用公开的预训练T5-Base 模型，以HuggingFace 1的220M参数作为骨干。

如果没有公开可用的带有注释的测试分割，就使用完整的开发集或测试集进行测试。PEMT在4 × NVIDIA A800 gpu上进行训练。

作者用不同的随机种子运行了所有实验三次，并报告了平均值和标准差。在少样本设置下，对于每个样本数k∈{4,16,32}，随机从下游任务数据中收集k个样本。随机种子由所有比较方法共享。

**4.3 结果**

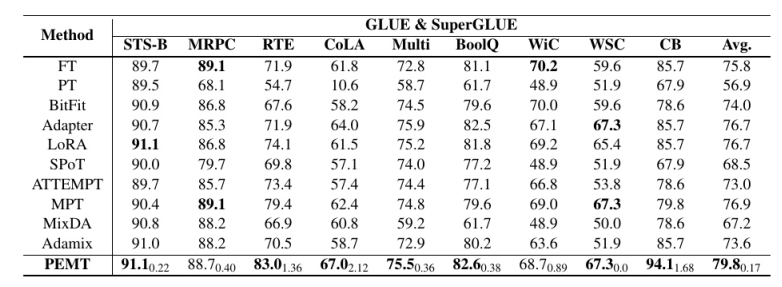


表1:GLUE和SuperGLUE的结果。指标是STS-B的Pearson相关性，MultiRC (Multi)的F1，以及其他任务的准确性作为评估指标。结果在三次运行中取平均，下标表示标准偏差。

表1和表2中的实验结果表明，PEMT明显优于全微调和所有其他参数高效微调方法。

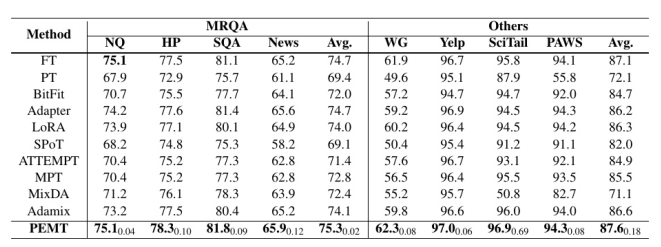


表2:MRQA和其他基准的结果。结果是三次运行的平均值，下标表示标准偏差。

表2显示了不同方法在MRQA和其他基准测试上的性能。与GLUE和SuperGLUE相比，这两个数据集的数据量更大，样本的上下文更长，导致之前的PEFT方法性能明显不如全微调。从结果来看，PEMT在这些数据集上成功地优于完全微调，这表明PEMT在不同数据大小和上下文长度上的稳定性和鲁棒性。

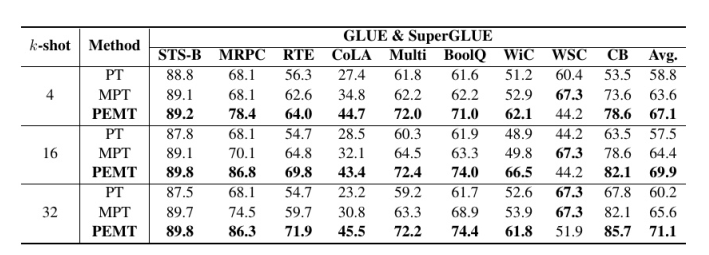
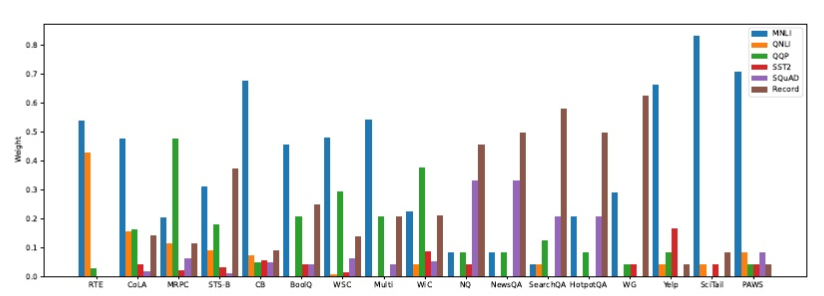


表3:4、 16、 32个训练样例在GLUE上的少样本学习结果。

在GLUE和SuperGLUE基准上进行了几次实验，以测量PEMT对新任务的泛化程度，只有少量可用的训练样例(k∈{4,16,32})。表3显示，作者的方法仍然产生了显著的改进。并且随着训练样本数量的增加，PEMT在基线上的改进变得更加明显。这强调了当提供更多的训练数据时，在下游任务的训练过程中，MPT的任务共享知识逐渐消失。相比之下，PEMT冻结了源任务适应器，不仅最大程度地保留共享知识，还可以充分利用各种任务之间的关联和区别。

**5 分析**

作者进行了进一步的分析，以调查不同成分的PEMT的有效性。

为了探索源专家的权重在各种目标任务上的变化情况，收集了MoE门的输出。如图4所示，权重分布上存在明显的倾向和优先级。对于GLUE和SuperGLUE基准测试，MNLI的知识起主导作用，权重超过所有任务的50%。部分个体任务的贡献在任务稀疏性损失的约束下接近0。相比之下，MRQA的权重分布则完全不同，SQuAD和ReCorD两个任务约占权重的80%。原因是MRQA、 SQuAD和ReCorD三个数据集都属于问答类，这也说明相关性引导MoE模块和任务稀疏度损失有效地工作。图4:在GLUE、 SuperGLUE、 MRQA和其他基准测试中的源专家权重分布。

为了衡量用一个任务描述语句初始化任务提示这种方法在保持任务表示的一致性方面的有效性，将其替换为随机初始化的提示，并保持提示长度相同。如表5所示，在没有使用任务描述初始化的情况下，两个基准的平均得分降低了0.8%。

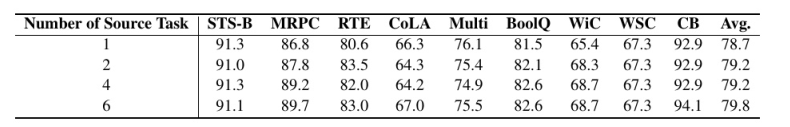


表4:不同源任务数的GLUE和SuperGLUE基准的平均得分。

为评估任务相关性特征在促进模型选择最优来源专家方面的有效性，在保留MoE模块的同时，去掉了源任务训练和目标适应中的整个提示模块。对于MoE门，对前一个FFN层的隐藏状态使用平均池化作为输入。表5中的消融研究显示，任务相关性特征产生1.5%的平均性能提升。

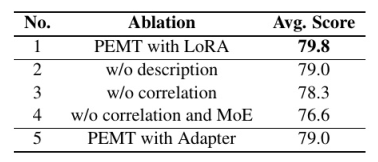


表5:GLUE和SuperGLUE基准的消融研究结果。

作者删除了目标提示和源适应器，同时只在每个FFN层之后维护task-specific的适应器。这种变化将模型退化为适应器的简单变体，它将适应器模块插入每个多头关注层和FFN层。表5显示，在没有对源任务进行训练的情况下，性能平均大幅下降3.2%。

**5.1 源任务的数量**

为了证实PEMT的可伸缩性，作者还研究了在阶段1中使用不同数量的源任务时性能的变化。如表4所示，与MixDA相比，随着源任务数量的增加，PEMT表现出逐渐改善的趋势，表明PEMT能够充分捕捉各种任务之间的共性和差异，体现了一定程度的持续学习熟练程度。

**6 结论**

本文中，提出了一种新的参数高效微调框架PEMT，能够利用来自多个任务的知识适应下游目标任务。任务之间的相关性特性促进了PEMT，并充分利用源任务的task-specific的知识，及时调优和混合专家架构。还引入了新的方法来改善提示初始化和模型稀疏性。在涉及多个任务和领域的广泛数据集上进行了实验，结果表明PEMT显著优于现有的SOTA方法。

**局限**

模型的推理延迟随着专家数量的增加而成比例地上升，这表明有必要确定一个稳定的重新参数化来合并多个专家的权重或探索一个可靠的简化方法。此外，该框架两阶段的架构既带来了训练开销和数据成本，也引入了数据泄露或模型攻击的潜在风险。