AdaTT：自适应任务间融合网络：推荐中的多任务学习

Danwei Li*
lidli@meta.com
Meta AI
门洛帕克 , 加利福尼亚州 , USA

Zhengyu Zhang
zhengyuzhang@meta.com
Meta Platforms , Inc.
门洛帕克 , 加利福尼亚州 , USA

Siyang Yuan
syyuan@meta.com
Meta AI
门洛帕克 , 加利福尼亚州 , USA

Mingze Gao
gaomingze@meta.com
Meta Platforms , Inc.
门洛帕克 , 加利福尼亚州 , USA

Weilin Zhang
weilinzzz@meta.com
Meta AI
门洛帕克 , 加利福尼亚州 , USA

Chaofei Yang
yangcf10@meta.com
Meta AI
门洛帕克 , 加利福尼亚州 , USA

Xi Liu
xliu1@meta.com
Meta AI
门洛帕克 , 加利福尼亚州 , USA

Jiyan Yang
chocjy@meta.com
Meta AI
门洛帕克 , 加利福尼亚州 , USA

arXiv:2304.04959v2 [cs.IR] 5 Jun 2023

摘要

多任务学习（MTL）旨在通过同时在多个任务上进行训练，提升机器学习模型的性能和效率。然而，多任务学习研究面临两项挑战：1）有效建模任务之间的关系，以实现知识共享；2）联合学习任务特定知识和共享知识。本文提出了一种名为自适应任务间融合网络（AdaTT）[1] 的新模型，以应对这两项挑战。AdaTT 是一种深度融合网络，由多个层级上的任务特定融合单元和可选的共享融合单元构成。通过在任务间融合中利用残差机制和门控机制，这些单元能够自适应地学习共享知识和任务特定知识。为了评估 AdaTT 的性能，我们在公开基准和一个工业推荐数据集上，使用不同的任务组开展实验。结果表明，AdaTT 的表现显著优于现有最先进的基线方法。此外，我们的端到端实验显示，与其他方案相比，该模型性能更佳。

CCS 概念

• **计算方法** → **多任务学习**；•**信息系统** →**推荐系统**。

关键词

多任务学习；神经网络；推荐系统

*联系邮箱：lidli@meta.com。[1]代码可在 https://github.com/facebookresearch/AdaTT 获取。

特此授权，可免费为个人或课堂使用目的制作本作品的全部或部分数字副本或纸质副本，但不得为营利或商业利益而制作或分发副本，且副本首页必须载有本声明和完整引文。对于本作品中由作者以外的其他人拥有的内容，其版权必须得到尊重。注明出处后可以摘录。除此之外，复制、再版、发布到服务器或分发至列表，均须事先获得明确许可和／或支付费用。请向 permissions@acm.org 申请许可。
*KDD '23，2023*年*8*月*6*日至*10*日，美国加利福尼亚州长滩© 2023年版权归所有者／作者所有。出版权授权予 ACM。ACM ISBN 979-8-4007-0103-0/23/08...$15.00
https://doi.org/10.1145/3580305.3599769

ACM 引用格式：
Danwei Li、Zhengyu Zhang、Siyang Yuan、Mingze Gao、Weilin Zhang、Chaofei Yang、XiLiu 和 Jiyan Yang。2023年。AdaTT：自适应任务间融合网络，用于推荐中的多任务学习。载于 第*29*届*ACM SIGKDD*知识发现与数据挖掘会议（*KDD '23*）论文集，*2023*年*8*月*6–10*日，长滩，加利福尼亚州，美国。ACM，纽约，纽约州，美国，10页。
https://doi.org/10.1145/3580305.3599769

1 引言

在线推荐系统旨在为用户生成个性化、高质量的推荐。这些系统的有效性通常取决于其准确学习用户偏好的能力，而这通常需要同时优化多个目标。例如，短视频推荐系统应同时考虑用户观看视频的可能性以及用户点赞视频的可能性。多任务学习（MTL）是此类应用场景的典型解决方案。通过在单一框架内联合训练多个任务，多任务学习可带来多项益处。首先，它能够提高计算效率，这对于大规模在线推荐系统十分重要。此外，它还通过跨任务正则化和知识共享来提升模型性能。

然而，多任务学习也面临着独特的挑战。其中一个主要挑战是对任务之间的关系进行建模。由于各任务与其他任务的相关程度可能不同，仅对所有任务共有的一般共性进行建模是不够的。随着任务数量增加，这个问题会变得更加复杂。有效的任务关系建模是实现高效任务自适应知识共享的关键。例如，对于“分享视频”这一任务，可以重点共享与“给视频点赞”等相似任务相关的知识，同时也借鉴其他样本丰富的任务（如“观看视频”）中不同方面的知识。另一方面，对于高度不相关的任务，则会尽量减少共享学习。以往的工作 [2, 19]通常采用静态共享表示。其他工作，例如图2 (c)所示的交叉缝合网络 [24], , 通过学习矩阵来对关系进行建模

多个子网络之间的关系。然而，所有样本的权重都保持不变，而且这些子网络仅在较宽泛的意义上针对特定任务。近期方法，如图2 (b)所示的 MMoE [22] 和图2 (e)所示的 PLE [29]，使用专门的门控网络动态组合共享子模块，以实现灵活的共享，但这些方法所建模的任务之间关系仍然晦涩且间接。

除了共享学习之外，任务特定学习也是多任务学习不可或缺的一部分。在两者之间取得恰当的平衡，对于解决任务冲突和实现跨任务正则化至关重要。一方面，多任务学习可能会遭遇负迁移，即针对一个任务的优化会对另一个任务的性能产生负面影响，尤其是在各任务的目标相互冲突时。在这种情况下，多任务学习模型应自适应地侧重于任务特定学习。另一方面，过度进行任务特定学习而共享不足，则可能导致过拟合，削弱跨任务正则化的益处。每个任务的训练数据量及其分布也会影响学习侧重点：数据较多的任务可以更多地依赖自身的特定学习，而数据较少或数据高度偏斜的任务则可以更多地侧重于共享学习。考虑样本差异可以使两者之间的权衡更加动态。因此，自动学习如何平衡这两种学习方式至关重要。许多软参数共享模型无需繁琐的手动调优 [2]，也无需基于简化假设为所有样本学习静态结构 [23, 28, 30]，便能实现这一点。然而，仍需进一步研究如何对共享学习与任务特定学习之间的交互进行建模，以提升性能。

为共同应对这些挑战，我们提出了一种新颖的多任务学习模型——自适应任务间融合网络（AdaTT）。为提升共享学习效果和可解释性，我们提出引入任务特定专家、共享专家和门控模块，在任务对级别和全任务级别显式建模任务间交互。为协同实现任务特定学习与共享学习，我们在独立的融合模块中分别对二者进行区分和建模，并在各模块中采用不同的专家和融合策略。随后，通过残差机制组合融合结果 [12]。此外，我们采用多个融合层级，每个层级专门负责不同功能，以提升学习性能。

为评估 AdaTT 的性能，我们在一个真实的短视频推荐系统上进行了实验。我们调整实验分组，以考察模型对不同任务关系的适应能力。此外，我们还使用一个公开基准进一步验证其泛化能力。在所有这些实验中，AdaTT 在不同数据集和任务组上均持续优于基线模型。

为评估 AdaTT 在大规模场景下的性能，我们研究了其超参数，重点关注融合层级和专家的数量。此外，我们设计了消融实验和可视化分析，以深入了解 AdaTT 的内部机制。消融实验验证了采用独立建模的融合模块的残差设计在实现任务特定学习与共享学习互补方面的有效性。对深层和浅层融合层级中专家权重的可视化，有助于进一步理解不同融合层级、任务和任务组之间所学习到的独特且有意义的在不同融合层级、任务和任务组之间共享所学习的模式。

总而言之，本文的贡献如下：

- 我们提出了一种新颖的多任务学习模型——自适应任务到任务融合网络（AdaTT），能够同时实现自适应的任务到任务知识共享和稳健的任务特定学习。
- 通过在真实世界基准数据和大规模视频推荐系统上进行全面实验，我们评估了 AdaTT 相较于各种基线方法的有效性。
- 我们通过对模型的各个融合模块进行消融研究，并考察其针对浅层和深层知识的融合单元运行机制，展示了模型的可解释性。

2 相关工作

多任务学习在多个领域有广泛应用，包括计算机视觉 [16, 19, 24, 34], 自然语言处理[5, 11], 语音识别 [6], 机器人技术 [32], 和推荐系统 [10, 22, 29, 35]。许多研究致力于开发创新的多任务学习架构。这些模型可分为两类：硬参数共享和软参数共享。**硬参数共享** 是指采用预定义的模型架构，其中某些层由所有任务共享，而其他层则专用于各个任务。共享底层模型 [2] 是硬参数方法中应用最广泛的模型之一。该模型利用共享的底层进行表征学习，并在其之上设置特定于任务的层。多线性关系网络 [20] 通过对特定于任务的层的参数施加张量正态先验，改进了这一结构。另一个例子是 UberNet [16], ，它采用图像金字塔方法联合解决多种低级、中级和高级视觉任务。它通过特定于任务的层和共享层处理金字塔中的每个分辨率。硬参数共享模型通常结构紧凑，但需要投入大量人工来确定共享哪些内容，且缺乏适应性。此外，在无关或相互冲突的任务之间过度共享可能导致负迁移，从而对模型性能产生负面影响。

为更好地应对这些挑战，研究者提出了许多**软参数共享**多任务学习模型。交叉缝合网络 [24]和 sluice 网络 [26] 使用可训练参数对每一层的输出进行线性组合。然而，它们采用的线性组合是固定的，因此无法充分体现不同样本中的任务关系差异。其他研究则提出使用以输入为条件的注意力或门控模块，为每个任务动态组合或提取知识。例如，MTAN [19] 采用注意力模块生成逐元素掩码，从共享表示中提取任务特定知识。MMoE [22] 引入专家混合，并采用门控网络为每个任务动态融合专家。最近，PLE [29] 被提出，以进一步增强知识共享的灵活性。PLE 明确引入了任务特定专家，并结合共享专家。此外，PLE 提出使用门控模块进行渐进式分离路由，以有选择地、动态地融合知识。在这一系列工作中。

相关工作中，PLE 与我们的工作最为相关。不同的是，我们的工作引入了两种互补的融合模块，分别用于建模任务特定学习和共享学习。此外，除了显式引入共享模块来学习所有任务之间的共性之外，我们还基于输入采用直接任务对融合，以最大限度地提高知识共享的灵活性。

神经架构搜索（NAS） [8, 17, 18, 25, 36]方法已应用于多任务学习（MTL），以自动学习模型结构。分支式多任务网络 [30] 通过根据亲和度分数对任务进行聚类，并将不相似的任务分配到不同分支，生成树形结构。 [9]在分支操作中采用 Gumbel-Softmax采样，而非预先计算的亲和度分数，从而实现端到端训练。软层排序技术 [23] 指出了多任务学习模型中传统固定顺序共享方法的局限性，并提出学习特定于任务的缩放参数，为每个任务灵活安排共享层的顺序。AdaShare [28] 学习特定于任务的策略，以选择每个特定任务要执行的层。子网络路由（SNR） [21] 将共享层拆分为子网络，并利用潜变量学习它们之间的连接。NAS 方法减少了大量人工工作，并提高了多任务学习模型中共享模式的灵活性。然而，由于穷举搜索所有可能的模型配置在组合上非常复杂，这些方法通常依赖分支 [9, 30], 路由 [21], 层排序 [23], 层选择 [28], 等简化假设来限制搜索空间。此外，生成的结构无法针对单个样本进行调整。

在关注多任务学习架构设计的工作之外，另一类工作旨在改进多任务学习优化。不确定性加权 [15] 根据任务不确定性学习每个任务的权重。GradNorm [3] 控制不同任务的梯度幅值，以平衡它们的训练速率。GradDrop [4] 以概率方式选择符号，并移除符号相反的梯度。梯度手术（PCGrad） [33] 将相互冲突的任务梯度投影到彼此的法平面上。RotoGrad [14] 同时操控任务梯度的幅值和方向，以缓解冲突。 [27] 将多任务学习视为一个多目标优化问题，目标是寻找帕累托最优解。 [31] 引入带有欠参数化小型塔的自辅助损失，以平衡帕累托效率与跨任务泛化。尽管这些方法能够带来改进，但如果没有强大的模型架构作为支撑而仅依赖这些方法，可能会限制模型性能的上限。

3 模型架构

为了共同学习自适应共享表示并增强任务特定学习，我们提出了一种新模型——自适应任务间融合网络（AdaTT）。AdaTT 利用门控和残差机制，在多个融合层级中自适应地融合专家。考虑一个包含两个预测任务的多任务学习场景。我们在图1中用两个融合层级展示了 AdaTT 的架构。AdaTT 由多层融合网络和任务塔组成。融合网络由任务专属融合单元和可选的共享融合单元构成，任务塔则构建在融合网络之上，并与最终融合层级中的任务专属单元相连。我们的框架具有通用性，支持灵活选择专家模块、任务塔网络、门控模块，以及可配置数量的专家和融合层级。在以下章节中，我们将首先介绍 AdaTT 的一个特殊版本，称为AdaTT-sp，该版本仅使用任务专属融合单元（如图1（a）所示）。然后，我们将描述通用的 AdaTT 设计，如图1（b）所示。

3.1 AdaTT-sp

AdaTT-sp 的详细设计如下。给定针对 T 个任务的输入 x，任务 t（$t = 1, 2, \ldots\ldots, T$）的预测可表示为：

$$y_t = h_t(f_t^L(x)), \tag{1}$$

其中，L 是融合层级的数量，h_t 表示任务 t 的任务塔，f_t^L 表示用于生成任务 t 在第 L 个融合层级的融合单元输出的函数。这里，f_t^L（x）通过应用融合层（从底部到顶部）并使用公式 2 和 3 计算得出：

$$f_1^0(x) = f_2^0(x) = \cdots = f_T^0(x) = x \tag{2}$$

$$f_t^l(x) = FU_t^l(f_1^{l-1}(x), f_2^{l-1}(x), \ldots, f_T^{l-1}(x)), \quad l = 1 \ldots L \tag{3}$$

这里，FU 表示融合单元 .

3.1.1 融合单元。 下面详细介绍公式 3 中引入的 FU_t^l。对于任务 t，在接收来自上一融合层级的所有输出后，我们首先使用函数 $e_{t,\,i}^l$和输入 f_t^{l-1}（x）构建该任务的 m_t 个原生专家，记为 $E_{t,\,i}^l$。也就是说，

$$E_{t,i}^l = e_{t,i}^l(f_t^{l-1}(x)), \tag{4}$$

其中 $i = 1, 2、\ldots\ldots、 m_t$以及$E_{t,\,i}^l \in \mathbb{R}^{1\times d^l}$。级别 l中的每个专家网络都会生成一个长度为 d^l的向量。为便于表示，在级别l，我们分别使用 E_t^l和 E^l表示属于任务 t的专家以及跨所有任务的专家的纵向拼接。具体而言， E_t^l和 E^l表示为：

$$E_t^l = [E_{t,1}^l, E_{t,2}^l, \ldots, E_{t,m_t}^l], \tag{5}$$

$$E^l = [E_1^l, E_2^l, \ldots, E_T^l], \tag{6}$$

其中 $E_t^l \in \mathbb{R}^{m_t\times d^l}$和 $E^l \in \mathbb{R}^{(m_1+m_2+\ +m_T)\times d^l}$。在上述公式中，[,] 表示将向量或子矩阵纵向堆叠成一个更大的矩阵。

由于任务与其他任务的相关程度可能不同，FU_t^l直接通过门控模块 $AllExpertGF_t^l$对任务间知识融合进行建模，该模块结合了所有任务的专家 E^l。此外，我们利用轻量级线性组合 $NativeExpertLF_t^l$来融合任务 t的原生专家， E_t^l。从概念上讲，门控模块对共享学习进行建模，而原生专家的线性组合则对任务特定学习进行建模。具体而言，任务t在第 l层特定单元的输出表示为：

$$f_t^l(x) = AllExpertGF_t^l(E^l, G_t^l) + NativeExpertLF_t^l(E_t^l), \tag{7}$$

其中，门控权重 G_t^l用于组合 E^l。相较于公式 3中的表示， E^l取决于所有 $f_1^{l-1}(x)$ $f_2^{l-1}(x) \ldots, f_T^{l-1}(x)$，而 G_t^l和 E_t^l仅取决于 $f_t^{l-1}(x)$。

具体而言，在公式 7 中，专家融合如下：

$$NativeExpertLF_t^l(E_t^l) = v_t^{l\,\intercal} E_t^l, \tag{8}$$

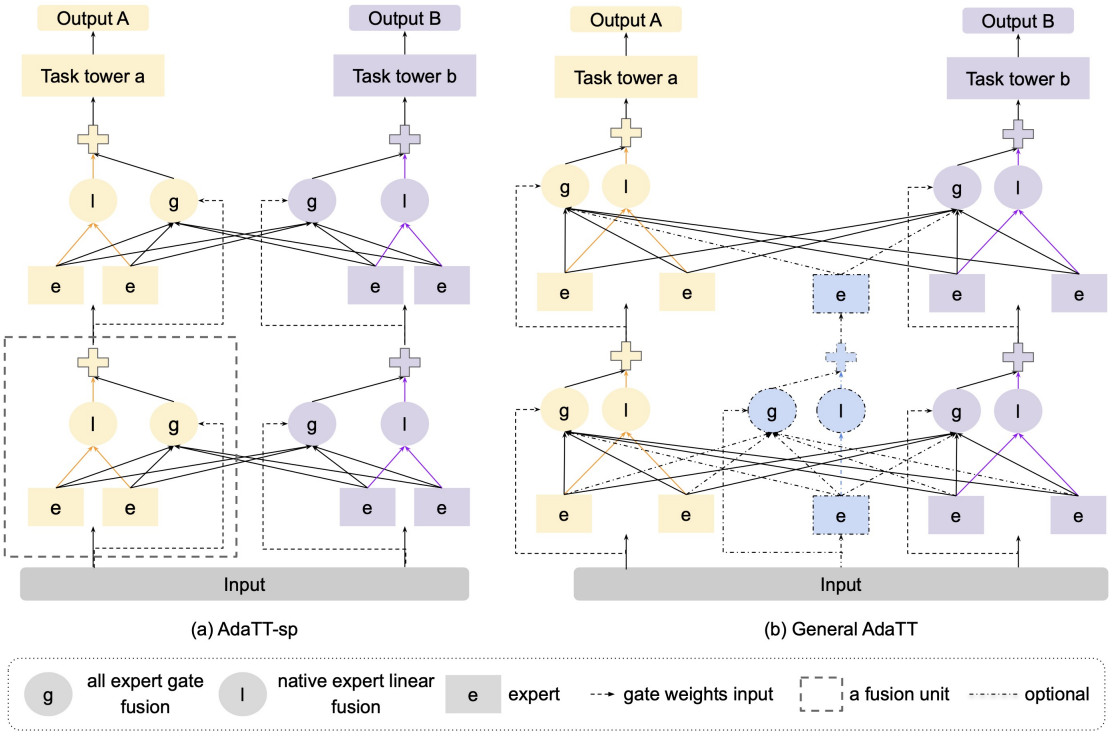


图1：具有2个融合层级的 AdaTT-sp 和通用 AdaTT。任务 A 和 B 的专属模块与共享模块通过颜色加以区分：黄色表示 A，紫色表示 B，蓝色表示共享模块。为便于说明，我们为每个任务专属单元使用2个专家。在通用 AdaTT 中，我们添加了一个共享融合单元，并以其中包含单个专家为例。请注意，通用 AdaTT 中的共享模块并非必需，因此使用虚线表示。当不存在共享模块时，通用 AdaTT 将退化为 AdaTT-sp。

$$AllExpertGF_t^l(E^l, G_t^l) = G_t^{l\top} E^l, \tag{9}$$

其中，E^l 与函数 g_t^l 在 $AllExpertGF$ 中生成的门控 $G_t^l \in \mathbb{R}^{(m_1+m_2+\ldots+m_T)\times 1}$ 相乘，而 E_t^l 则仅通过 $NativeExpertLF$中的一个可学习向量 $v_t^l \in \mathbb{R}^{m_t\times 1}$进行组合。当 $m_1 = m_2 = \ldots = m_T = 1$时，也就是说，所有融合单元都只有一个专家，$NativeExpertLF_t^l(E_t^l)$会退化为 E_t^l，为简单起见，为原生专家分配单位权重。g_t^l有许多设计选项。一种常见做法是使用由 softmax 激活的单层多层感知机：

$$g_t^l(f_t^{l-1}(x)) = softmax(W_t^l f_t^{l-1}(x)^\top). \tag{10}$$

这里，$W_t^l \in \mathbb{R}^{(m_1+m_2+\ldots+m_T)\times d^{l-1}}$是一个学习得到的矩阵。

3.1.2 简化。为了提高实现效率，根据公式 8 和公式 9，我们实际上可以在 $v_t^{l\top}$ 中填充零，使其与 $G_t^{l\top}$ 的大小匹配，然后将权重相加，并执行一次乘法来组合所有专家。因此，公式 7 可简化为：

$$f_t^l(x) = (pad(v_t^{l\top}) + G_t^{l\top})E^l. \tag{11}$$

如我们所见，加入线性融合模块仅会使计算量略有增加。

3.2 通用 AdaTT

如图1(b)所示，在通用形式下，AdaTT采用可选的共享融合单元。从概念上讲，任务专属模块对之间的融合建模了细粒度共享，而任务专属模块与共享模块之间的融合则传递适用于所有任务的广泛知识。这实现了高效且灵活的任务间知识共享。通用 AdaTT 的计算方式与 AdaTT-sp 类似，但最后一个融合层级除外：在该层级中，共享融合单元不执行任何融合操作，仅生成专家输出，供任务专属融合单元处理。

总之，AdaTT 显式地学习任务特定知识，并将其与共享知识自适应融合。该融合具有任务自适应性，具体体现在：1. 门控模块学习相对于各任务原生专家的残差。2. 每个任务专属单元都使用一个以输入为条件的专用门控模块来融合专家（从第二个融合层级开始，这一特性尤为独特）。通过允许每个任务直接、灵活地从其他任务中学习共享知识，与仅依赖共享专家作为媒介的 PLE 相比，AdaTT 提供了更大的灵活性。此外，AdaTT 还可以选择仅使用任务特定专家。与在单个门控模块中处理所有选定专家的 PLE 不同，AdaTT 在每个融合单元内使用不同的线性融合模块，分别融合原生专家。这一设计增强了任务特定

学习在每个融合层级之后的效果。尽管方法简单，我们的实验表明，它优于 PLE；PLE 会在不同融合单元中选择专家，并使用不同的路由路径来区分这些专家。

4 实验篇章

在本节中，我们展示了全面的实验结果，以突出我们所提出的 AdaTT 模型的有效性，并帮助读者更好地理解该模型。

本节分为四个部分。首先，我们在第 4.1 节简要介绍基线模型。其次，我们通过在真实工业数据集和公开数据集上进行实验，评估 AdaTT 相较于最先进的多任务学习模型的有效性。对于工业数据集，我们使用三组不同的预测任务，考察这些多任务学习模型在各种场景下的表现。结果见第 4.2 节和第 4.3 节。接下来，我们在第 4.4 节和第 4.5 节展示各个组件的研究。我们对 $NativeExpertLF$ 模块进行消融，以验证 AdaTT 的残差设计的重要性；该设计采用独立模块来融合不同的专家。我们还将每个任务专属单元中学习到的专家权重可视化，以展示 AdaTT 如何学习任务之间恰当的交互，这对实现有效的知识共享至关重要。最后，在第 4.6 节中，我们研究 AdaTT 的超参数，以了解融合层级和专家数量与 AdaTT 性能之间的关系。

4.1 基线模型

我们采用共享底层、MMoE、多层MMoE（原始单层MMoE的扩展）、PLE 和 Cross-stitch网络作为基线模型。在这些模型中，MMoE、PLE 和 Cross-stitch网络均采用软参数共享技术。

- **MMoE** [22]：该模型为每个任务学习一个专用的门控模块，以融合多个共享专家。给定 n 专家模块 e_1，e_2，..... e_n、任务 t 的任务塔模块 h_t 和门控模块 g_t，任务 t 的预测计算如下：

$$y_t = h_t(f_t(x)), \tag{12}$$

其中

$$f_t(x) = g_t(x)[e_1(x), e_2(x), ..., e_n(x)]. \tag{13}$$

这里，[,] 表示将向量按垂直方向堆叠成矩阵。

- **多层 MMoE (ML-MMoE)**：该模型通过引入多个融合层扩展了原始单层MMoE。在 ML-MMoE 中，高层专家将由不同门控模块融合的低层专家作为输入。与原始 MMoE类似，所有门控模块均以相同的原始输入为条件。

- **交叉缝合** [24]：该模型引入交叉缝合单元，通过学习得到的权重对不同任务的隐藏层进行线性组合。

- **PLE** [29]：该模型明确引入了任务专属专家和共享专家，并采用渐进式分离路由策略。门控模块用于融合任务专属单元和共享单元中的选定专家。在 PLE 中，共享单元可以融合同一层级的所有专家，而任务专属单元仅融合其原生专家和共享专家。该模型与 AdaTT 最为接近。

上述讨论的所有模型均在图2中展示，以供比较。

4.2 大规模短视频推荐评估

本节介绍短视频推荐系统上的实验结果。该系统会展示一系列推荐视频，并根据各项任务的得分进行排序。这些任务大致可分为两类：参与度任务会考虑用户的显式反馈，例如对视频发表评论；消费任务则反映用户的隐式反馈，例如观看视频。

4.2.1 任务组。 我们创建了三组任务，以全面评估这些模型在不同任务关系下的性能。

- 第一组包括一个参与度任务和一个消费任务，预计二者的相关性相对较低。

- 第二组由两个相关性较高的消费任务组成。第一个任务与第一组中的消费任务相同。第二个任务的正事件率与第一组中的参与度任务相近。第一组和第二组都只包含二元分类任务。

- 在第三组中，我们将任务数量增加到五个，并选择多样性较高的任务。其中，三个是消费任务，两个是参与度任务。一个消费任务是回归任务，其余四个任务是二元分类任务。从用户情感来看，我们纳入了一个反映用户不喜欢的任务，以及四个针对正向事件的任务。我们将一个正事件极其稀疏的参与度任务用作辅助任务。

报告所有任务组的结果时，如果存在回归任务，我们会先报告该任务，随后按照正例率从高到低的顺序报告分类任务。

4.2.2 实验设置。我们收集了约 700 亿条样本用于训练模型，并在约 100 亿条样本的测试集上评估其性能。在特征处理中，我们将所有稀疏特征转换为稠密嵌入，并与稠密特征拼接。所有任务使用相同的输入。所有模型均使用相同的框架进行训练和测试，优化设置也相同，例如优化器、学习率和批次大小。训练时，二元分类任务使用交叉熵损失，回归任务使用均方误差损失。所有任务的损失以相同权重求和并进行优化。测试时，二元分类任务使用归一化熵（NE）[13]，回归任务使用均方误差。

4.2.3 模型超参数。 在我们的实验中，所有模型都具有由 ReLU 激活的 3 个隐藏层。对于每组实验，我们进行了两项比较。

首先，我们将 MMoE、PLE和 AdaTT 与共享底层模型进行比较。为确保公平比较，PLE、ML-MMoE 和 AdaTT 都有

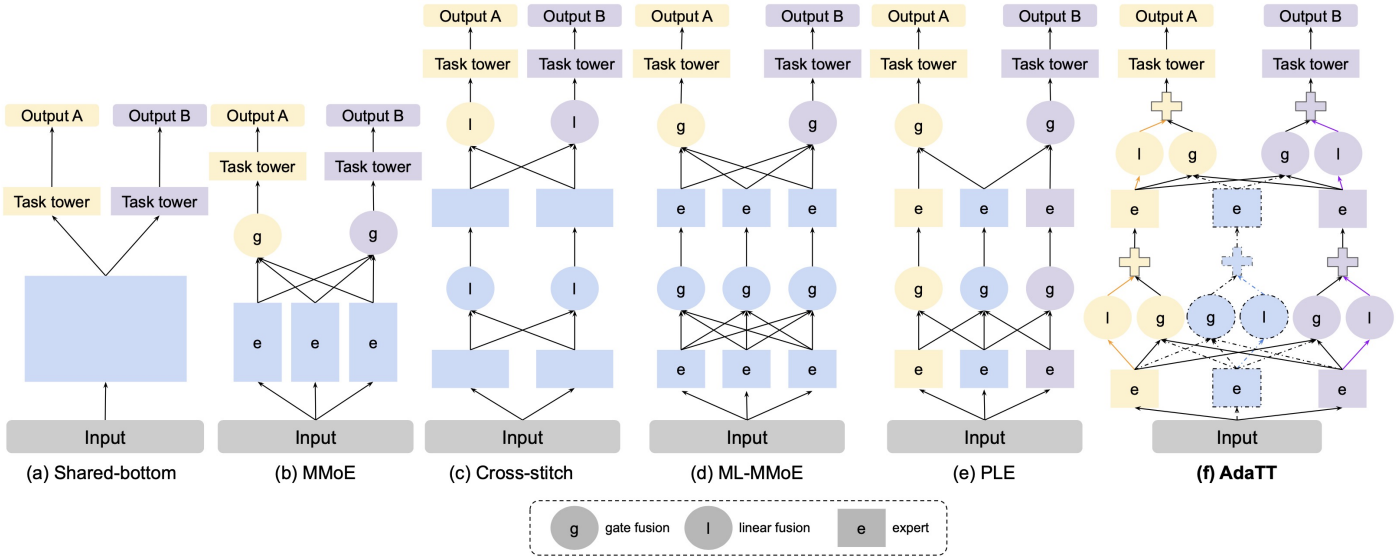


图2：我们实验中使用的多任务学习模型。在多层级多任务学习模型中，使用两个融合层级来展示其设计。不同模块以不同颜色表示：共享模块为蓝色，任务 A 的专属模块为黄色，任务 B 的专属模块为紫色。

2 个融合层级。对于这 2 个融合层级，我们分别采用隐藏维度为 256 和 128 的单层多层感知机专家。MMoE 由具有 [256, 128] 隐藏维度的两层多层感知机专家构成。我们还对每个融合层级的专家总数设置了上限。这些模型中的所有门控模块都使用带有 softmax 激活函数的单层多层感知机。值得注意的是，与专家模块相比，门控模块所需的计算量要小得多。尽管这两类模块的输入维度相同，但门控模块的总输出维度几乎小了两个数量级。所有模型的任务塔均包含一个由 64 个单元组成的隐藏层。采用这一设置后，所有模型的计算量相当，因为计算主要由任务塔和专家模块构成。在实验中，我们调整 PLE 和 AdaTT 的任务专属专家数量与共享专家数量；对于 MMoE，则调整专家总数。

在另一项独立实验中，我们比较了 AdaTT、交叉缝合模型与共享底层模型的性能。为便于与交叉缝合模型比较，AdaTT 使用与前一项实验类似的超参数，但每个任务仅使用 1 个专家，且不使用共享专家。交叉缝合模型包含 2 个交叉缝合单元，并使用与 AdaTT 相同的隐藏层。

4.2.4 参与度任务和消费任务任务组上的实验。对于这组任务，我们展示了各模型在训练 100 亿、300亿和700 亿个样本后，相对于共享底层模型的 NE 差异结果。我们还提供了测试结果。表 1 和表 2 分别展示了消费任务和参与度任务的结果。结果表明，AdaTT 在两个任务上都优于其他所有模型，不仅收敛速度更快，而且质量更高。在训练 100亿个样本后，两个 AdaTT 模型在两个任务上都已经表现出显著的 NE 提升。至于基线

在模型中，PLE 在消费任务上的收敛时间要长得多。另一方面，交叉缝合模型的表现远逊于 AdaTT，这表明自适应融合对于任务关系建模至关重要。值得注意的是，与消费任务相比，PLE 和 AdaTT 在参与度任务上的提升更大，而该任务的正向事件较少。然而，这一趋势在 MMoE 和 ML-MMoE 中并不明显，凸显了任务特定学习的重要性。有趣的是，尽管通过额外的融合操作获得了更大的灵活性，ML-MMoE 在这两项任务上的表现都不如 MMoE，表明其专家融合效果较差。这可能是由于 ML-MMoE 的设计缺乏区分度，且引入了先验知识。共享专家之间高度对称，每个门控模块都会使用所有共享专家，并且没有显式建模任务特定专家。此外，所有门控模块接收的都是相同的原始输入。融合层级的增加带来了更多路径，使 ML-MMoE 更难学会针对每个特定任务预测不同的权重组合。

4.2.5 两个消费任务任务组上的实验。由于多任务学习模型的性能可能对任务相关性较为敏感，我们设计了一个实验组，用于评估这些模型在两个相关消费任务上的性能，并与任务组1进行对比；任务组1中的任务相关性较低。如表3所示，结果表明，与基线相比，该组中的所有模型在两个任务上都取得了更为相近的提升。这并不令人意外，因为当任务之间的相关性更高时，负迁移的影响较小，且两个任务都能从更高程度的共享知识中受益。即使采用较简单共享机制的多任务学习模型也能取得良好性能，因此模型在NE方面的差异并不明显。然而，在所有多任务学习模型中，AdaTT仍然表现最佳。

表 1：消费任务组中消费 +参与度任务的性能

模型	10B 示例处的NE差异	NE差异 @30B 示例	70B 个样本下的 NE差异	测试集 NE差异
共享底层	-	-	-	-
MMoE	-0.334%	-0.421%	-0.498%	-0.481%
ML-MMoE	-0.307%	-0.400%	-0.480%	-0.463%
PLE	-0.162%	-0.385%	-0.482%	-0.448%
AdaTT	**-0.391%**	**-0.464%**	**-0.526%**	**-0.508%**
交叉缝合	-0.024%	-0.133%	-0.166%	-0.140%
AdaTT-sp（单任务专家）	**-0.259%**	**-0.261%**	**-0.277%**	**-0.231%**

表2：消费 + 参与度任务组中的参与度任务表现

模型	10B个样本时的NE差异	30B个样本时的NE差异	70B个样本时的NE差异	测试 NE差异
共享底层	-	-	-	-
MMoE	-0.370%	-0.436%	-0.542%	-0.532%
ML-MMoE	-0.260%	-0.386%	-0.496%	-0.494%
PLE	-0.360%	-0.627%	-0.698%	-0.691%
AdaTT	**-0.677%**	**-0.795%**	**-0.845%**	**-0.863%**
交叉缝合	-0.046%	-0.197%	-0.232%	-0.225%
AdaTT-sp（单任务专家）	**-0.393%**	**-0.367%**	**-0.397%**	**-0.362%**

表 3：消费任务组上的性能

模型	任务 1 NE差异	任务 2 NE差异
共享底层	-	-
MMoE	-0.343%	-0.372%
ML-MMoE	-0.415%	-0.372%
PLE	-0.446%	-0.368%
AdaTT	**-0.487**%	**-0.443**%
交叉缝合	-0.170%	-0.136%
AdaTT-sp（单任务专家）	**-0.233%**	**-0.194%**

4.2.6 五个多样化任务实验。在这一任务组中，我们利用5个高度多样化的任务，评估模型处理复杂跨任务关系的能力。我们针对4个主要任务对模型进行调优，并将结果列于表4中。由于辅助任务中的正向事件较为稀疏，噪声水平较高且性能不稳定，因此我们未将其纳入。结果表明，在所有主要任务上，AdaTT都以显著优势超越了所有对比模型，说明其在处理复杂任务关系方面具有优越性。

4.3 在公共数据集上的评估

4.3.1 数据集描述。我们使用从1994年和1995年当前人口调查中提取的 人口普查收入数据集 [7] 。该数据集包含40个特征和299,285个实例，包括199,523个训练样本和99,762个测试样本。我们将测试样本按相同比例随机划分为验证集和测试集。任务包括：1）预测收入是否超过50K；2）预测婚姻状况是否为从未结婚；3）预测教育程度是否至少为大学学历。

*4.3.2*模型超参数。本实验采用一个改编自 [1], 的框架来训练和测试ML-MMoE、PLE 和AdaTT。模型结构与第4.2.3 节中的结构类似，但隐藏维度和专家数量有所调整。实验分为两组进行，每个融合层级分别设置 6 个和9 个专家。对 PLE 和 AdaTT的共享专家数量 m_s进行调优。任务特定专家的数量分别计算为 $6 - m_s$ 和 $9 - m_s$。为确保公平性，所有其他超参数在各模型之间保持一致。在调优 m_s 后，每个模型使用不同的初始化训练 100 次，并报告测试集上的平均 AUC。

4.3.3 结果。结果如表 5 所示。AdaTT 在所有任务上均优于基线模型。

4.4 *NativeExpertLF*模块的消融实验

本节考察融合单元中 *NativeExpertLF* 模块的残差机制所产生的影响。我们对*NativeExpertLF*模块进行消融，仅使用 *AllExpertGF*模块来组合每个融合单元中所有专家的输出。我们采用与第 4.2.3 节类似的模型结构，并将每个任务的专家数量固定为三个，不设置共享专家。两个模型均使用 700 亿个样本进行训练，并使用 100 亿个样本进行测试。结果见表 6。

尽管 *AllExpertGF* 模块理论上能够学习灵活的专家组合，但我们的实验仍表明，分别组合原生专家并将 *AllExpertGF* 的输出作为残差添加至其中非常重要。具体而言，移除 *NativeExpertLF*

表4：5项任务组上的模型性能

模型	消费任务1 MSE差异	消费任务2 NE差异	消费任务3 NE差异	参与度任务1 NE差异
共享底层	-	-	-	-
MMoE	-0.770%	-0.632%	-0.708%	-1.182%
ML-MMoE	-0.697%	-0.608%	-0.685%	-1.013%
PLE	-0.697%	-0.599%	-0.698%	-1.221%
AdaTT	**-0.873%**	**-0.738%**	**-0.815%**	**-1.346%**
交叉缝合	-0.520%	-0.454%	-0.486%	-0.818%
AdaTT-sp（单任务专家）	**-0.613%**	**-0.543%**	**-0.589%**	**-0.930%**

表 5：UCI 人口普查收入数据集上的 3 个任务的性能。我们比较了采用两级融合的 PLE、ML-MMoE 和 AdaTT。专家网络和任务塔网络均为单层多层感知机，其隐藏维度列于表中。仅使用任务特定专家的 AdaTT-sp 配置，使 AdaTT 达到了最佳效果。

模型	专家总数	专家隐藏维度	任务塔隐藏维度	任务 1 AUC	任务 2 AUC	任务 3 AUC
ML-MMoE	6	128, 64	32	0.8729	0.9178	0.9731
PLE	6	128, 64	32	0.8683	0.9164	0.9697
AdaTT	6	128, 64	32	**0.8766**	**0.9202**	**0.9783**
ML-MMoE	9	96, 48	32	0.8688	0.9139	0.9730
PLE	9	96, 48	32	0.8645	0.9134	0.9680
AdaTT	9	96, 48	32	**0.8744**	**0.9174**	**0.9786**

这一项会导致所有任务的损失上升：分类任务的 NE 增加 0.107%–0.222%，回归任务的均方误差增加 0.158%。

4.5 门控模块专家权重分布可视化

在图3中，我们将 *NativeExpertLF* 和 *AllExpertGF* 模块的权重相加后，对专家权重的分布进行可视化，以探究 AdaTT 的内部融合机制。为评估专家利用率，我们选取了三个任务：两个消费任务和一个参与度任务。具体而言，我们从消费任务中选取一个回归任务，并从参与度任务和消费任务中选取正事件率最高的两个分类任务。我们采用两个融合层级，每个任务各配置一个专家，且不设共享专家。专家为单层多层感知机，在两个融合层级中的隐藏维度分别为 256 和 128。模型训练完成后，我们将其应用于测试数据集，计算所有测试样本的平均权重，并为每个融合层级绘制一个 3×3 权重矩阵。值得注意的观察如下：

首先，在较低的融合层级（层级 0），模型能够识别任务之间的关系。消费任务组与参与度任务组之间存在明显区别。此外，两个消费任务之间呈现非对称共享模式：分类消费任务主要使用专家 2，而分类回归任务则大致平均使用专家 1 和专家 2。

在更高层级的融合（level1）中，监督信号更接近且捕获了丰富的语义信息，我们的模型表现出通过跨任务的共享模式，实现软参数共享的优势。虽然原生专家在任务特定学习中发挥着重要作用，但所有专家都得到了灵活利用，为共享学习作出贡献。在这一层级，消费分类任务通过利用专用于参与度分类任务的专家 3，以及专用于消费回归任务的专家 1，旨在实现学习多样化。同时，正向信号较少的参与度任务受益于来自两个消费任务的知识迁移。相比之下，消费回归任务主要依赖其原生专家 1 和专用于另一个消费任务的专家。在所有专家中，专家 1 从第 0 层的专家混合 1 和 2 中获得了最为多样化的学习，因此在所有任务中都被赋予了较高权重。

总的来说，我们可以看到明显的专门化现象：在每个任务、任务分组和融合层级上，都会学习到不同的权重分布模式。

4.6 超参数研究

我们开展超参数研究，以探究专家数量和融合层级数量的影响。两项研究均使用与第 4.2.6 节类似的 5 个预测任务，训练数据包含 700 亿个样本，测试数据包含 100 亿个样本。两项研究均采用 AdaTT-sp 作为模型。

4.6.1 任务特定专家数量的影响。为了考察任务特定专家数量的影响，为简化起见，我们使所有任务的任务特定专家数量保持一致，并将其设为 1 到 4。每个融合层级使用由单层多层感知机构成的专家，隐藏维度分别为 256 和 128。此项分析的结果见表7。我们

表 6：对 *NativeExpertLF* 模块进行的消融实验。每个任务上的性能下降都证明了带独立融合的残差机制的重要性。

模型	消费任务1 MSE差异	消费任务2 NE差异	消费任务3 NE差异	参与度任务1 NE差异
AdaTT	-	-	-	-
AdaTT（消融）	+0.158%	+0.107%	+0.107%	+0.222%

表7：AdaTT 在每个任务采用不同数量专家时的性能。

专家数量 每个任务	消费任务1 MSE差异	消费任务2 NE差异	消费任务3 NE差异	参与度任务1 NE差异
1	-	-	-	-
2	-0.171%	-0.150%	-0.159%	-0.075%
3	-0.248%	-0.203%	-0.239%	-0.296%
4	-0.260%	-0.230%	-0.273%	-0.398%

表8：随着融合层级增加，AdaTT 的性能结果。我们在第一列标明了各融合层级对应的专家隐藏维度。

专家 隐藏维度	消费任务1 MSE差异	消费任务2 NE差异	消费任务3 NE差异	参与度任务1 NE差异
256, 128	-	-	-	-
512, 256, 128	-0.225%	-0.242%	-0.284%	-0.409%
1024, 512, 256, 128	-0.503%	-0.448%	-0.522%	-0.619%
2048, 1024, 512, 256, 128	-0.664%	-0.587%	-0.655%	-0.766%

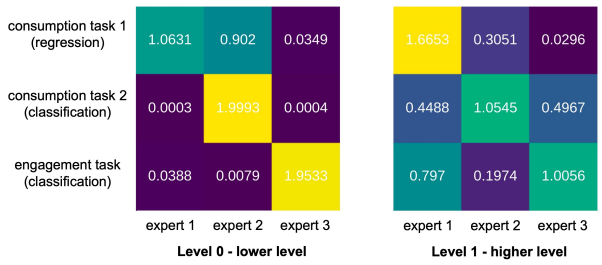


图3：双层 AdaTT-sp 中各融合层级学习到的专家权重分布可视化。任务和专家按消费回归任务、消费分类任务和参与度任务的顺序排列。请注意，此图展示的是*NativeExpertLF* 和 *AllExpertGF* 模块的权重总和。由于每个任务只有一个原生专家，*NativeExpertLF* 模块会为其分配单位权重（在图中对应对角网格）。

可以观察到，随着专家数量的增加，所有任务的性能都有所提升。然而，这些提升并不一致：在本研究中，当专家数量增加到 2 个时，与消费任务相比，参与度任务在 NE方面仅表现出轻微提升。不过，随着专家数量进一步增加到3 个和4个，这一趋势发生逆转，参与度任务在各项指标上表现出更明显的差异。

4.6.2 融合层级的效果。我们还通过为每个任务使用单个专家，研究了各种融合层级配置。我们逐步增加融合层级的数量，并为每个层级使用单层多层感知机。我们训练了四个模型，每个模型在不同融合层级的每个多层感知机专家中使用

[256, 128], [512, 256, 128], [1024, 512, 256, 128], [2048, 1024, 维隐藏512, 256, 128]

维度。对于任务塔，每个模型都使用隐藏维度为 64 的单层多层感知机。结果见表8。正如预期，增加融合层级的数量带来了更大的性能提升。即使融合层级增加到五层，所有任务仍然都取得了显著改善。

5 结论

在这项工作中，我们提出了一种新的多任务学习模型，称为自适应任务间融合网络（AdaTT）。借助其自适应融合机制，AdaTT 能够有效地建模复杂的任务关系，并促进任务专属知识与共享知识的联合学习。通过在包含多样化任务组的真实工业数据集以及公开数据集上进行全面评估，我们验证了 AdaTT 的有效性和泛化能力。结果表明，AdaTT 的表现显著优于最先进的多任务学习模型。我们希望这项工作能惠及多任务学习以外的更广泛应用场景，使不同的相关专用模块能够协同学习。

参考文献

[1] Raquel Aoki、Frederick Tung 和 Gabriel L. Oliveira。2021。具有专家多样性的异构多任务学习。载于*BIOKDD*。[2] Rich Caruana。1997。多任务学习。机器学习 28, 1（1997），41–75。[3] Zhao Chen、Vijay Badrinarayanan、Chen-Yu Lee 和 Andrew Rabinovich。2018。Gradnorm：用于深度多任务网络自适应损失平衡的梯度归一化。载于国际机器学习会议。PMLR，794–803。[4] Zhao Chen、Jiquan Ngiam、Yanping Huang、Thang Luong、Henrik Kretzschmar、Yuning Chai 和 Dragomir Anguelov。2020。只需选择一个符号：使用梯度符号丢弃优化深度多任务模型。神经信息处理系统进展 33（2020），2039–2050。[5] Ronan Collobert 和 Jason Weston。2008。一种统一的自然语言处理架构：采用多任务学习的深度神经网络。载于第*25*届国际机器学习会议论文集。160–167。[6] Li Deng、Geoffrey Hinton 和 Brian Kingsbury。2013。用于语音识别及相关应用的新型深度神经网络学习方法：综述。载于*2013*年*IEEE*国际声学、语音与信号处理会议。IEEE，8599–8603。[7] Dheeru Dua 和 Casey Graff。2017。UCI 机器学习仓库。http: //archive.ics.uci.edu/ml[8] Thomas Elsken、Jan-Hendrik Metzen 和 Frank Hutter。2017。用于卷积神经网络的简单高效架构搜索。*arXiv*预印本 *arXiv:1711.04528*（2017）。[9] Pengsheng Guo、Chen-Yu Lee 和 Daniel Ulbricht。2020。学习如何分支以实现多任务学习。载于国际机器学习会议。PMLR，3854–3863。[10] Guy Hadash、Oren Sar Shalom 和 Rita Osadchy。2018。排序与评分：面向推荐系统的多任务学习。载于第*12*届*ACM*推荐系统会议论文集。451–454。[11] Kazuma Hashimoto、Caiming Xiong、Yoshimasa Tsuruoka 和 Richard Socher。2016。一种联合多任务模型：为多个自然语言处理任务扩展神经网络。*arXiv*预印本 *arXiv:1611.01587*（2016）。[12] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。2016。用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议论文集。770–778。[13] Xinran He、Junfeng Pan、Ou Jin、Tianbing Xu、Bo Liu、Tao Xu、Yanxin Shi、Antoine Atallah、Ralf Herbrich、Stuart Bowers 等。2014。从 Facebook 广告点击预测中获得的实践经验。载于第八届在线广告数据挖掘国际研讨会论文集。1–9。[14] Adrián Javaloy 和 Isabel Valera。2021。RotoGrad：多任务学习中的梯度同质化。*arXiv*预印本 *arXiv:2103.02631*（2021）。[15] Alex Kendall、Yarin Gal 和 Roberto Cipolla。2018。利用不确定性对场景几何与语义损失加权的多任务学习。载于*IEEE*计算机视觉与模式识别会议论文集。7482–7491。[16] Iasonas Kokkinos。2017。Ubernet：利用多样化数据集和有限内存训练用于低层、中层和高层视觉任务的通用卷积神经网络。载于*IEEE*计算机视觉与模式识别会议论文集。6129–6138。[17] Chenxi Liu、Barret Zoph、Maxim Neumann、Jonathon Shlens、Wei Hua、Li-Jia Li、Li Fei-Fei、Alan Yuille、Jonathan Huang 和 Kevin Murphy。2018。渐进式神经架构搜索。载于欧洲计算机视觉会议（*ECCV*）论文集。19–34。[18] Hanxiao Liu、Karen Simonyan 和 Yiming Yang。2018。Darts：可微分架构搜索。*arXiv*预印本 *arXiv:1806.09055*（2018）。[19] Shikun Liu、EdwardJohns和Andrew J Davison。2019。基于注意力的端到端多任务学习。在*IEEE/CVF*计算机视觉与模式识别会议论文集中。1871–1880。[20] Mingsheng Long、Zhangjie Cao、Jianmin Wang 和 Philip S Yu。2017。利用多线性关系网络学习多个任务。神经信息处理系统进展 30（2017）。[21] Jiaqi Ma、Zhe Zhao、Jilin Chen、AngLi、Lichan Hong 和 Ed HChi。2019。SNR：用于多任务学习中灵活参数共享的子网络路由。在*AAAI*人工智能会议论文集，第33卷。216–223。[22] Jiaqi Ma、Zhe Zhao、Xinyang Yi、JilinChen、LichanHong 和 Ed H Chi。2018。利用多门混合专家模型对多任务学习中的任务关系进行建模。在第*24*届*ACM*知识发现与数据挖掘国际会议论文集中。1930–1939。[23] Elliot Meyerson和 Risto Miikkulainen。2017。超越共享层级：通过软层排序实现深度多任务学习。*arXiv*预印本 *arXiv:1711.00108*（2017）。[24] Ishan Misra、Abhinav Shrivastava、Abhinav Gupta和Martial Hebert。2016。用于多任务学习的交叉缝合网络。在*IEEE*计算机视觉与模式识别会议论文集中。3994–4003。[25] Esteban Real、Alok Aggarwal、YanpingHuang和Quoc V Le。2019。用于图像分类器架构搜索的正则化演化。在*AAAI*人工智能会议论文集，第33卷。4780–4789。[26] Sebastian Ruder、Joachim Bingel、IsabelleAugenstein 和 Anders Søgaard。2019。潜在多任务架构学习。在*AAAI*人工智能会议论文集，第33卷。4822–4829。[27] Ozan Sener 和 Vladlen Koltun。2018。将多任务学习视为多目标优化。神经信息处理系统进展 31（2018）。[28] Ximeng Sun、Rameswar Panda、Rogerio Feris 和 Kate Saenko。2020。AdaShare：学习共享哪些内容，以实现高效的深度多任务学习。神经信息处理系统进展33（2020），8728–8740。[29] Hongyan Tang、JunningLiu、MingZhao 和 Xudong Gong。2020。渐进式分层提取（PLE）：一种用于个性化推荐的新型多任务学习（MTL）模型。在第十四届*ACM*推荐系统会议中。269–278。[30] Simon Vandenhende、Stamatios Georgoulis、Bert DeBrabandere 和 LucVan Gool。2019。分支式多任务网络：决定共享哪些层。*arXiv*预印本 *arXiv:1904.02920*（2019）。[31] Yuyan Wang、Zhe Zhao、BoDai、Christopher Fifty、Dong Lin、LichanHong、Li Wei和 EdH Chi。2022。小型头部能有所帮助吗？理解并改进多任务泛化。在*2022*年*ACM Web*会议论文集中。3009– 3019。[32] Christopher Williams、Stefan Klanke、Sethu Vijayakumar 和 Kian Chai。2008。机器人逆动力学的多任务高斯过程学习。神经信息处理系统进展21（2008）。[33] TianheYu、Saurabh Kumar、Abhishek Gupta、Sergey Levine、Karol Hausman和 Chelsea Finn。2020。多任务学习的梯度手术。神经信息处理系统进展 33（2020），5824–5836。[34] Zhanpeng Zhang、Ping Luo、Chen Change Loy和Xiaoou Tang。2014。通过深度多任务学习进行人脸标志点检测。在欧洲计算机视觉会议中。Springer，94–108。[35] Zhe Zhao、Lichan Hong、Li Wei、Jilin Chen、Aniruddh Nath、Shawn Andrews、Aditee Kumthekar、Maheswaran Sathiamoorthy、Xinyang Yi 和 Ed Chi。2019。推荐接下来观看的视频：一个多任务排序系统。在第*13*届*ACM*推荐系统会议论文集中。43–51。[36] BarretZoph和 Quoc V Le。2016。利用强化学习进行神经架构搜索。*arXiv*预印本 *arXiv:1611.01578*（2016）。