大语言模型融合技术综述：从原理、实践到未来展望

摘要

大语言模型（LLM）的兴起推动了人工智能应用的普及，但其高昂的训练成本、巨大的算力需求以及数据隐私风险也成为行业面临的关键挑战[1]。模型融合作为一种高效的技术范式，通过在参数空间中对多个预训练或微调后的模型进行合并，能够无需访问原始训练数据，也无需进行大规模再训练，即可将不同模型的优势整合到一个单一模型中。这不仅显著提高了资源利用率，也为快速迭代和部署提供了一种经济高效的替代方案[3]。本报告对现有的大模型融合方法进行了系统性的梳理与分类，深入剖析了各类技术的原理、优缺点，并结合主流开源工具和实证研究，讨论了该领域面临的挑战与未来发展方向。

1. 引言：大模型融合的背景、定义与核心价值

1.1 大模型时代的挑战：高效训练与部署的瓶颈

当前，大语言模型正以惊人的速度发展，其参数规模持续膨胀，但随之而来的挑战也日益凸显。首先是能源消耗与算力成本的急剧攀升，这不仅对企业的财务状况构成严峻考验，也引发了对“绿色AI”发展的深层思考[1]。其次，大规模模型的训练和微调需要海量的原始训练数据，这在许多特定领域，尤其是涉及用户敏感数据的场景中，会引发严重的数据隐私与安全问题[1]。

在这一背景下，业界迫切需要一种能够打破现有瓶颈、实现高效赋能的技术。模型融合正是对这一需求的直接响应，它提供了一种在模型参数层面进行创新的解决方案[6]。与需要运行多个模型、增加推理开销的传统集成（Ensembling）方法不同，模型融合将多个模型的知识“压缩”进一个单一模型，从而维持与单个模型相同的推理成本，同时实现多模型能力的整合[4]。

这种技术范式的兴起，从根本上体现了人工智能发展中，技术选择开始越来越多地受到经济效益和可持续性因素的驱动。当高昂的训练成本和算力消耗成为技术普及的瓶颈时，像模型融合这样“无需再训练”的高效方案，自然成为了解决之道，它标志着AI创新正在从单纯的“规模竞赛”转向

对“效率与价值”的深度挖掘。

1.2 模型融合的定义与独特优势

模型融合，也称为模型合并或模型汤（Model Soups），是一种将两个或多个预训练或微调后的模型权重合并成一个新模型的策略[3]。其核心目标是在不进行额外训练的情况下，提升新模型在特定任务上的性能或整合多项专业能力。

模型融合的独特优势主要体现在以下几个方面：

- 成本效益与资源利用率：模型融合能够有效利用“失败的实验”或多个在不同超参数、数据集上训练的模型检查点，将其组合起来，从而减少实验浪费[3]。它提供了一种经济高效的方案，允许开发者在无需昂贵计算资源（甚至仅使用CPU或少量显存）的情况下，创建出具有竞争力的模型[4]。
- 能力组合与知识迁移：模型融合能够将不同专家模型（如专注于代码、数学、多语言能力的模型）的特定能力融合到一个单一的通用模型中[4]。这种方法的核心在于，它能够在不访问原始训练数据的情况下，实现模型间的知识和能力的迁移[4]。
- 性能提升与稳健性增强：通过整合多个模型，融合后的模型往往能展现出比单一最佳模型更高的性能和更强的稳健性[7]。在某些情况下，模型融合甚至能作为一种有效的正则化手段，通过平均化权重来减少方差，从而防止过拟合[9]。

2. 大模型融合技术分类与综述

根据其技术原理和应用阶段，大模型融合方法可以被系统地划分为两大类：预融合技术和融合中技术[6]。预融合技术旨在为模型融合创造有利条件，而融合中技术则是核心的算法操作。

2.1 预融合技术：为成功融合做准备

预融合阶段的核心理念是，模型的融合效果与其在参数空间中的位置密切相关[6]。如果待融合的模型位于相似的“损失盆地”（Loss Basin）中，即它们的权重向量在优化后的参数空间中距离较近，那么简单的融合操作也更可能成功[8]。

损失盆地对齐（**Loss Basin Alignment**）是该阶段的关键概念。实践中，最常见的做法是使用同一

个预训练的基座模型，然后在其上通过不同的微调任务、超参数或数据集创建多个变体[8]。由于这些模型从相同的起点开始微调，它们的权重向量通常会停留在同一个损失盆地内，从而保证了它们之间的"同构性"和可合并性[8]。

这一原理解释了为何像模型汤（Model Soup）这样的简单加权平均方法在实践中能取得成功：它并非对任意模型都有效，而是严格依赖于待融合模型是否具有同源性[8]。这强调了模型融合并非"无中生有"的炼金术，其成功的前提在于"输入"的质量和兼容性。

2.2 融合中技术：核心融合算法

融合中技术是实现参数合并的核心算法，它们在处理模型权重、任务向量或内部激活模式时采用了不同的策略。

2.2.1 基于参数空间的线性/非线性插值融合

这类方法通过在模型权重空间中进行直接的数学插值操作来生成新模型。

- 简单加权平均（**Model Soup**）：
  - 原理：这是最直观的方法，通过对多个模型的权重进行简单的算术平均或加权平均来获得最终模型[8]。
  - 优点：实现起来极为简单，如果待融合的模型都由同一个基座模型微调而来，该方法能有效提升模型的性能稳健性并防止过拟合[8]。
  - 缺点：对于差异较大的模型，简单平均可能导致性能灾难性下降，因为其没有考虑权重向量之间的复杂几何关系[8]。
- 球面线性插值（**SLERP**）：
  - 原理：SLERP 是一种"更智能"的插值方法[3]。它在高维向量空间中，沿球面最短路径进行插值，而非简单的线性插值[3]。这种方法能够更好地保持权重向量的方向和幅度，通常比线性插值效果更好[10]。
  - 优点：能更好地处理高维空间中的插值问题，在融合两个模型时表现出色[3]。
  - 缺点：SLERP 严格意义上仅适用于两个模型之间的融合，如果要融合多个模型，需要进行分层或链式操作，这增加了方法的复杂性[10]。

2.2.2 基于任务向量的算术融合

这类方法将模型微调的过程抽象为“任务向量”，然后对这些向量进行算术运算，其概念更具可解释性。

- 任务算术（**Task Arithmetic**）：
  - 原理：该方法将模型的“任务向量”定义为微调后的模型参数与预训练模型参数之间的差值[6]。通过对这些任务向量进行加减法运算，可以实现不同任务能力的组合、知识遗忘或任务类比[6]。例如，通过将多个任务向量相加，可以实现多任务学习[6]。
  - 优点：概念直观，具有很强的可解释性，允许用户像“乐高积木”一样灵活组合模型能力[6]。
  - 缺点：实证结果表明，其效果中等，并且当不同任务的参数更新存在冲突时，可能会导致任务间的负面干扰[6]。

2.2.3 基于稀疏化与子空间对齐的融合

这类方法旨在通过识别和利用参数更新中的稀疏性来减少任务间的干扰。

- **TIES (Trim, Elect Sign, and Merge)**：
  - 原理：TIES 旨在解决多模型融合中参数更新的“符号干扰”问题[3]。它包含三个核心步骤：修剪（**Trim**），丢弃不重要的、幅度较小的参数更新；多数表决（**Elect Sign**），根据剩下的参数在每个位置上的符号进行多数投票，确定最终的符号方向；最后，合并（**Merge**），仅保留与多数符号一致的参数并进行合并[3]。
  - 优点：能够有效处理多模型融合中的参数冲突问题，在实证测试中表现良好[3]。
  - 缺点：算法比简单方法更复杂，需要更多的计算步骤。
- **DARE (Drop and REScale)**：
  - 原理：该方法通过对任务向量进行**丢弃（Drop）和重新缩放（Rescale）**来融合模型[7]。丢弃操作引入稀疏性，以减少不重要的参数更新所带来的干扰；重新缩放则旨在调整权重，以保持合并后模型的预期性能[7]。
  - 优点：是一种有效的参数稀疏化方法，在长到短（Long-to-Short）推理等任务中表现突出[14]。
  - 缺点：需要额外的超参数（如密度 density）进行调优，以平衡稀疏性与性能。

2.2.4 其他新兴与高级融合方法

- 基于激活的融合方法（**Activation-based Merging**）：
  - 原理：这类方法不仅考虑模型权重，还参考模型在特定输入下的内部激活模式。融合的目标是通过匹配不同模型在中间层的激活分布来实现能力合并[14]。
  - 优点：在长到短推理等任务上，其性能表现令人印象深刻，被认为是未来的重要研究方向

[14]。
    - 缺点：理论和实现都更为复杂，目前相关工作尚处于早期研究阶段。
- 基于**SVD**的方法（**SVD-based Merging Methods**）：
    - 原理：主要应用于参数高效微调（PEFT）的场景，利用奇异值分解（SVD）来近似 LoRA 适配器，实现参数的降维和融合 [13]。
    - 优点：能够处理不同秩（rank）的LoRA适配器 [13]。
    - 缺点：实证结果表明，此类方法在通用基准测试中表现普遍不佳，效果不甚理想 [14]。

从技术发展路径来看，大模型融合方法正从最初基于简单数学原理（如线性/SLERP插值）的方法，演进为更具经验启发式（如 TIES、DARE）的方法。这表明在实践中，纯粹的数学优美性往往让位于能够有效解决实际痛点（如多任务冲突）的经验性解决方案。未来的研究方向很可能会继续沿着这一路径，利用更复杂的信息（如激活值、注意力模式）来指导融合过程，以期获得更好的性能。

表**1**：主要大模型融合方法分类与优缺点概览

融合方法	技术分类	核心原理	优点	缺点
简单加权平均（Model Soup）	参数空间插值	对多个模型权重进行算术平均或加权平均	简单易实现；在同源模型上能提高稳健性	对模型同构性要求高；效果不稳定
球面线性插值（SLERP）	参数空间插值	沿高维球体最短路径插值，保持方向和幅度	插值效果优于线性平均，保持模型能力	严格限制为两个模型融合；多模型融合复杂
任务算术（Task Arithmetic）	任务向量算术	对微调模型与基座模型的差值（任务向量）进行加减运算	概念直观，可解释性强；灵活组合能力	效果中等，存在任务间负面干扰
TIES (Trim, Elect Sign, and Merge)	稀疏化/子空间对齐	修剪不重要参数，通过多数表决确定符号后合并	有效解决多模型融合中的符号干扰问题	算法实现比简单方法更复杂
DARE (Drop and Rescale)	稀疏化/子空间对齐	对任务向量进行丢弃和重新缩放	有效引入稀疏性，减少任务干扰	需要额外的超参数进行调优
基于激活的融	新兴高级方法	参照模型内部激活模式，匹配	实验效果卓越，在长到短推理	处于早期研究阶段；实现和理

合		不同模型的激活分布	任务中表现出色	论理解更复杂

3. 核心开源工具与实践指南

模型融合技术的快速发展，离不开活跃的开源社区和易用的工具。它们将复杂的理论算法封装成简单的接口，极大地降低了技术门槛，使得非专业研究者也能进行高效的模型融合实验。

3.1 MergeKit：大模型融合的瑞士军刀

MergeKit 是由 Arcee AI 开发的一个专用于大语言模型融合的开源工具包 [3]。其核心设计理念是在资源受限的环境下，实现复杂的模型融合操作。

- 核心功能：该工具采用 out-of-core（核外）方法，能够将大型模型分块加载到内存中进行处理，因此即使在仅有CPU或8GB显存的设备上，也能执行复杂的合并任务 [4]。
- 支持方法广泛：MergeKit 支持多种融合方法，包括但不限于 slerp, linear, ties, dare_linear, task_arithmetic 等 [4]。
- 灵活的配置：用户通过简单的 YAML 配置文件即可指定融合策略，包括选择模型、定义融合方法、设置权重和密度参数，甚至可以进行分层融合（Frankenmerging）或逐层精细控制 [4]。
- 开源地址：https://github.com/arcee-ai/mergekit [4]

3.2 Hugging Face PEFT：参数高效微调与融合

Hugging Face 的 PEFT（Parameter-Efficient Fine-Tuning）库是参数高效微调领域的领导者，它也提供了用于合并 LoRA 适配器的实用工具 [13]。

- 核心功能：PEFT 的 peft.utils.merge_utils 模块提供了一系列用于合并 LoRA 适配器的工具函数 [13]。这是实践中最常见的模型融合形式之一，因为 LoRA 适配器（Adapter）本身就代表了在基座模型上学习到的特定任务知识。
- 实用工具：该模块提供了 task_arithmetic, ties, dare_linear, dare_ties 等多种融合方法的实现，允许开发者在自己的代码中轻松调用，实现 LoRA 适配器的合并 [13]。
- 开源地址：https://huggingface.co/docs/peft/ [16]

开源工具的普及是模型融合技术得以迅速发展和普及的催化剂。理论研究的复杂算法，通过像 MergeKit 这样的工具化封装，降低了其应用门槛，使得模型融合不再局限于少数研究机构，而是成为广大开发者和社区可以轻松尝试的有效手段。这种理论研究与工程实践的紧密结合，共同推动了该领域的创新。

4. 融合方法性能实证分析

为了客观评估不同融合方法的有效性，研究者们提出了专门的基准测试，如 MergeBench，以在统一的标准下对融合模型进行评估 [15]。

4.1 核心评估指标

- 多任务性能：衡量融合模型在多个目标任务上的综合表现 [15]。
- 知识保留（**Forgetting**）：评估融合过程对基座模型通用知识的损害程度 [15]。
- 运行时效率：特别关注模型的输出长度，例如在长到短推理（Long-to-Short reasoning）任务中，融合模型能否有效减少冗余输出，同时保持性能 [14]。

4.2 基准测试结果综合对比

《解锁高效长到短LLM推理与模型融合》等研究对不同融合方法在7B模型上的性能进行了详尽的实证分析 [14]。

- 基于任务向量的方法：Task Arithmetic 和 TIES-Merging 表现出中等效果，在减少响应长度的同时能够保持甚至略微提升准确性。在7B模型上，这些方法能将平均响应长度减少约50%，同时在多项数学和推理任务上保持了与基线模型相当的性能 [14]。
- 基于**SVD**的方法：这些方法普遍表现不佳，被认为效果不尽人意，仅在任务向量本身具有低秩谱特征时，才可作为一种可行的备选方案 [14]。
- 基于激活的方法：该类方法在实证测试中展现出卓越潜力，在推理准确率和响应长度压缩比方面均有亮眼表现，甚至被认为是未来重要的研究方向 [14]。

4.3 优缺点综合讨论

实证分析揭示了模型融合的一个重要趋势：它的有效性并非普适，而是与模型规模和基座模型的质量密切相关[14]。实验表明，在更强大的基座模型（Stronger Base Models）上进行融合，通常能获得更好的效果[15]。相比之下，较小的模型（如1.5B规模）本身通用能力和任务学习能力有限，通过参数融合很难获得复杂的推理能力[14]。

这个发现表明，模型融合的成功并非"无中生有"，它的本质是知识的加法或组合。如果基座模型本身通用能力（即预训练知识）不足，或者微调后的任务向量所代表的能力不扎实，那么即使是再"聪明"的融合算法也难以创造出新的、更强的能力。因此，对于实践者而言，选择一个强大的基座模型是成功进行模型融合的首要前提。这也解释了为什么MergeBench等基准测试选择 Llama 和 Gemma 等顶尖开源模型作为测试对象[15]。

表**2**：不同融合方法在**7B**模型上的实证性能对比

方法	总体平均得分（Avg.）	响应长度减少（%）	综合表现	适用场景
Average Merging	48.6	-34.6%	中等，性能提升有限	同源模型融合
Task Arithmetic	53.5	-48.7%	良好，有明显提升	多任务能力组合
Ties-Merging	54.8	-53.0%	优秀，尤其在多模型融合	解决参数冲突
DARE-Ties	52.4	-50.4%	良好，兼顾性能和效率	需引入稀疏性
SVD-based	50.5-51.6	-35.8%至-41.6%	效果不佳，表现不稳定	特定低秩任务
Activation-based	55.0-56.4	-49.8%至-55.3%	卓越，展现未来潜力	高级任务，如长到短推理
*数据来源：基于Unlocking Efficient				

Long-to-Short LLM Reasoning等研究中7B模型基准测试结果的综合归纳[14]。				

5. 挑战、发展趋势与未来展望

5.1 现有挑战

尽管模型融合技术取得了显著进展，但仍面临一些挑战：

- 性能上限：在某些特定任务上，通过模型融合得到的模型性能，可能仍略逊于经过精心调优的多任务联合训练模型[15]。
- 计算成本：尽管融合过程无需大规模训练，但对于超大规模的模型，加载和合并参数的过程仍然需要巨大的计算资源和内存，并非完全无开销[15]。
- 同构性要求：目前大多数融合方法都要求待融合的模型具有相同的架构，这限制了其在异构模型间的应用[6]。

5.2 发展趋势与未来展望

- 跨模态融合：模型融合的理念正在从单一的语言模型领域拓展到多模态领域。未来的研究方向将包括将大语言模型与语音识别、图像生成、视频理解等其他模态进行融合，以构建更强大的多模态大模型（如将LLM与语音识别模型融合以提升ASR准确率）[11]。
- 动态与路由融合：现有的大模型效率远低于人脑，因为它们在处理任何问题时都可能调动所有参数[2]。相比之下，人脑在处理复杂问题时仅激活一小部分神经元，实现了极高的效率[2]。未来的模型融合可能会借鉴这一生物智能模式，探索基于输入内容动态选择或路由不同专家模型的方法，即所谓的稀疏激活或专家混合模型（MoE）[2]。
- 与边缘智能的结合：大模型融合所提供的“单体”模型特性，使其非常适合在边缘和设备终端进行部署[2]。未来的研究将重点关注如何进一步降低融合模型的尺寸和功耗，使其能够在低成本、低功耗的边缘设备上实现高效、低延时的推理[2]。

模型融合的最终发展目标可能不是简单地将所有知识“揉”在一起，而是构建一个“智能调度中心”，根据任务需求，在众多专家能力中进行高效、动态的路由和组合。这使得模型融合技术与AI领域对“效率”和“可持续性”的追求形成了完美的共振。

6. 结论

大语言模型融合是一种在参数空间中进行创新的高效赋能技术，它通过整合多个模型的权重，在无需原始数据和昂贵训练资源的前提下，实现了能力组合、知识迁移和性能提升 [3]。本报告系统地梳理了从简单的参数插值到复杂的基于稀疏化和激活的方法，并讨论了开源工具（如MergeKit和PEFT）在技术普及中的关键作用 [4]。

实证研究表明，融合方法的有效性与基座模型的质量密切相关，在强大的基座模型上进行融合更容易取得成功 [14]。尽管当前技术仍面临性能上限和同构性要求等挑战，但其向跨模态、动态路由和边缘部署方向的演进，预示着模型融合将在未来的AI生态中扮演更加重要的角色。它不仅是应对当前高成本、高能耗挑战的实用方案，更是通往更高效、更具可持续性的AI未来的关键路径之一。

引用的著作

1. 人工智能大模型行业高质量发展的挑战、趋势与未来展望- 中国日报网 - 天下专栏, 访问时间为 九月 15, 2025, https://column.chinadaily.com.cn/a/202412/24/WS676a5d7fa310b59111daa91e.html
2. 【大模型及其发展趋势】 人工智能的发展、大模型，最重要的基石是数字化。ChatGPT 的出现代表, 访问时间为 九月 15, 2025, https://air.tsinghua.edu.cn/__local/A/F3/79/CC9A0C81875F8B35A4733E36A57_BD4E1211_324F1.pdf
3. LLM 模型合并入门指南- NVIDIA 技术博客, 访问时间为 九月 15, 2025, https://developer.nvidia.com/zh-cn/blog/an-introduction-to-model-merging-for-llms/
4. arcee-ai/mergekit: Tools for merging pretrained large ... - GitHub, 访问时间为 九月 15, 2025, https://github.com/arcee-ai/mergekit
5. LLM Trends 2025: A Deep Dive into the Future of Large Language Models | by PrajnaAI, 访问时间为 九月 15, 2025, https://prajnaaiwisdom.medium.com/llm-trends-2025-a-deep-dive-into-the-future-of-large-language-models-bff23aa7cdbc
6. Model Merging in LLMs, MLLMs, and Beyond: Methods, Theories, Applications and Opportunities - arXiv, 访问时间为 九月 15, 2025, https://arxiv.org/html/2408.07666v4
7. Merge Large Language Models. Combine Mistral, WizardMath and... | by Sergei

Savvov - Medium, 访问时间为 九月 15, 2025, https://slgero.medium.com/merge-large-language-models-29897aeb1d1a
8. Model Soup 的Embedding 食譜 - Jina AI, 访问时间为 九月 15, 2025, https://jina.ai/zh-TW/news/model-soups-recipe-for-embeddings/
9. Model Soup 的Embedding 配方 - Jina AI, 访问时间为 九月 15, 2025, https://jina.ai/zh-CN/news/model-soups-recipe-for-embeddings/
10. Merge Large Language Models with mergekit - Hugging Face, 访问时间为 九月 15, 2025, https://huggingface.co/blog/mlabonne/merge-models
11. EnnengYang/Awesome-Model-Merging-Methods-Theories-Applications - GitHub, 访问时间为 九月 15, 2025, https://github.com/EnnengYang/Awesome-Model-Merging-Methods-Theories-Applications
12. [论文审查] Task Arithmetic for Language Expansion in Speech Translation - Moonlight, 访问时间为 九月 15, 2025, https://www.themoonlight.io/zh/review/task-arithmetic-for-language-expansion-in-speech-translation
13. PEFT welcomes new merging methods - Hugging Face, 访问时间为 九月 15, 2025, https://huggingface.co/blog/peft_merging
14. Unlocking Efficient Long-to-Short LLM Reasoning with Model Merging - arXiv, 访问时间为 九月 15, 2025, https://arxiv.org/html/2503.20641v1
15. MergeBench: A Benchmark for Merging Domain-Specialized LLMs - arXiv, 访问时间为 九月 15, 2025, https://arxiv.org/html/2505.10833v1
16. Model merge - Hugging Face, 访问时间为 九月 15, 2025, https://huggingface.co/docs/peft/package_reference/merge_utils
17. [2505.10833] MergeBench: A Benchmark for Merging Domain-Specialized LLMs - arXiv, 访问时间为 九月 15, 2025, https://arxiv.org/abs/2505.10833
18. 语音识别与大语言模型融合技术研究综述, 访问时间为 九月 15, 2025, https://lib.zjsru.edu.cn/25-5.6-3.pdf