推理过程是大模型推理的关键！

詹佳豪
计算机科学与技术学院
22307140116@m.fudan.edu.cn

杨淳瑜
计算机科学与技术学院
22307140114@m.fudan.edu.cn

杨永卓
计算机科学与技术学院
22307140078@m.fudan.edu.cn

王海天
计算机科学与技术学院
22307140113@m.fudan.edu.cn

摘要

当下，大模型飞速发展，预训练的大模型在下游具体任务上进行微调以获得出色的表现结果已经成了主流范式。目前的微调模型虽然能将预训练过程获取的知识迁移到下游任务，但在较长逻辑链推理如数学问题中表现不佳，由于推理时的累计误差导致模型最终推断错误。在本研究中，我们试图解决这一问题，针对 Qwen-0.5B 模型，基于 MATH 和 GSM8K 数据集进行训练和测试，我们在微调全流程（数据、训练、评测）提出了一系列创新方法。具体而言，我们进行了全局微调和基于 LoRA 的局部微调策略，以此作为基线。在数据层面，我们对数据集进行聚类以优化任务分配，同时借助 Prompt Engineering 实现了链式思维（CoT）推理。在训练层面，我们提出了 LLM-Dagger，通过修正错误的思维链，来进一步提升模型能力。实验结果表明，与未经微调的 Qwen-0.5B 模型预测结果相比，我们提出的上述方法显著提升了模型的数学推理能力，从多维度验证了所提优化策略的有效性和实用性。最后，在评测层面，我们提出了新的数学题评测指标——Process Inference Score，并通过实验验证了该指标的有效性。

1 引言

在过去两年中，大语言模型（LLM）在自然语言处理领域取得了显著进展。特别是自回归模型，在翻译、摘要以及情感分析等多种下游任务中展现了卓越的性能。这些由数据驱动的模型从海量训练实例中学习，逐步展现出涌现能力，其在语言处理上的表现日趋接近人类智能水平。

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

已有研究表明，大语言模型不仅具有语言处理能力，还具备对世界进行深度推理和理解的潜力。例如，Yao 等人（2022）[15] 展示了通过迭代推理和生成式回答，大语言模型可以展现出接近人类的智能水平。进一步地，Park 等人（2023）[8] 构建了基于大语言模型的多代理系统，以探索大语言模型模拟社会动态的能力。类似地，Wang 等人（2023）[11] 推出了 Voyager，这是一个与现实世界交互的嵌入式大语言模型代理，证明了它们与现实世界互动的能力。

尽管如此，大语言模型在逻辑推理，特别是数学推理能力上仍存在较大提升空间。数学问题通常需要严格的分析和长程逻辑思考，我们通过分析大模型生成的结果发现，大模型往往会因为中间一步的错误，导致后面所有推理出错，最后得到错误答案。这样的累计误差我们认为是当下大模型主要的推理瓶颈。我们从数据、训练以及指标上入手解决。

首先，从数据上，我们针对 GSM8K 原本没有思维链的训练数据进行了数据标注，通过将文本嵌入到向量空间，进行聚类，对每一个类别添加相同的思维链，从而实现 task-specific 的数据增强，同时，细化的每一步推理过程也使得微调后的大模型掌握了更强、更可靠的逻辑推理能力。

其次，在训练上，我们从 Imitation Learning 中的 Dagger 获取灵感，提出了 LLM-Dagger，即在一轮训练后，把做错的推理样本挑出来，请人类专家或者调用更强的大模型进行纠正，将纠正的结果作为训练的数据再进行一次训练，这样可以有效扩充大模型在数学推理上的搜索空间，避免因为累计误差到达分布外区域，从而得到错误结果。

最后，我们认为当下基于正则表达式的 hard-match 评测指标有不完善的地方，只关注于最终答案的正确与否会导致评测信号过于稀疏，在这种评测模式下，过程的正确与否都无法得到体现，就导致不能有效反映大模型在微调后的提升程度。我们提出了 Process Inference Score，基于 Qwen 大模型，通过精心撰写的 prompt 引导 Qwen 大模型对输出结果的过程以及答案联合打分，这种给“过程分”的方式更能细粒度地衡量大模型的能力。

2 相关工作

2.1 数学推理任务

大语言模型已被应用于多种数学推理任务，从基础的数学单词问题到复杂的几何问题。Cobbe 等人（2021）[2] 构建了 GSM8K 数据集，收录了 8,500 个涵盖小学数学知识的高质量问题，成为数学推理研究的一个重要基准。针对更高难度的任务，Dan 等人（2021）[3] 推出了 MATH 数据集，该数据集包含更具挑战性的数学问题及其详细解答步骤。此外，几何问题因其对空间想象力和抽象推理能力的要求，对大语言模型提出了巨大的挑战。Trinh 等人（2021）[10] 提出了 AlphaGeometry 模型，用于证明欧氏空间下的几何问题，该模型在人类专家评估下解决了 2000 年和 2015 年国际数学奥林匹克（IMO）中的所有几何问题，显示出与人类接近的智能水平。

2.2 提示词工程（Prompt Engineering）

提示词工程是提升大语言模型数学推理能力的关键方法之一。通过提供精心设计的指令和上下文信息，提示词工程可以有效引导模型执行特定任务（Brown 等人，2020）[1]。一些提示词工程方法旨在加强大语言模型的的长程推理能力。其中一种方法“思维链”（chain of thought, COT）尤为重要。它通过逐步引导模型生成中间推理步骤，从而处理复杂问题（Wei 等人，2022）[12]。例如，在 GSM8K 数据集的基准测试中，仅用 8 个 CoT 范例提示，一个拥有 5400 亿参数的语言模型便达到了最高水平的的准确度。在此基础上，Yao 等人（2023）[16] 提出了“思维树”（Tree of Thought）方法。这是思维链的一个增强版本，通过多路径推理和自我评估选择来确定下一步的推理方向，从而进行更精确的细粒度决策。这两种方法都具有很强的适应性，可以在各种任务中轻松实施。这些技术也为提高模型的数学推理能力奠定了重要基础。

2.3 大模型微调

上述的几项工作不会改变大语言模型的基本参数，这意味着它们对提高模型数学推理能力的影响是有限的。然而，利用专业数据集对大语言模型进行微调或训练额外的模型，可以有效地增强其特定领域能力，使其能力突破原来的极限。例如，Cobbe（2021）[2] 训练了一个验证模型来对大语言模型的输出进行评分，选择评分最高的答案，从而显著提高了该模型在 GSM8K 基准测试中的性能。同样，Lightman（2023）[6] 提出利用包含多个推理步骤的训练数据对大型模型进行监督微调。他的过程监督模型在 MATH 测试集的一个代表子集上实现了 78% 的求解率。这项工作也为开发 OpenAI 的博士级模型 O1 奠定了基础。在另一种方法中，Lu（2022）[7] 通过训练 PromptPG（一种小型模型，旨在生成最佳提示，以增强大语言模型对特定任务的理解），进一步优化了大语言模型对任务的理解。这些研究表明，微调与提示词工程可以形成互补，共同提升模型性能。得益于这些进展，大模型在小学水平的数学问题上几乎达到了完美的程度。

3 方法

基于相关工作，本文尝试多种方法对 Qwen-0.5B 模型进行数学推理能力的提升。3.1中，我们介绍了基线微调方法的具体做法。3.2中，我们将介绍我们是如何对 GSM8K 中的问题进行聚类，并且加入思维链增强模型推理能力的。3.4中，我们介绍了我们提出的新指标 Process Inference Score。

3.1 Lora 微调

LoRA(Low-Rank Adaptation) LoRA 微调是一种高效的模型微调方法，通过在预训练模型的权重矩阵中引入低秩矩阵来进行适应性调整，达到高效微调的目的。其核心思想是在不修改原始模型权重的情况下，通过添加一个可训练的低秩矩阵来适应下游任务。LoRA 会将模型中的某些层（如注意力层）的权重分解为低秩矩阵的和，即将原始的高维权重矩阵 W 分解为

$$W + \Delta W$$

其中 ΔW 是低秩矩阵。LoRA 微调的优点在于，模型的预训练权重不需要进行大规模修改，且低秩矩阵的维度较小，微调过程更加高效，节省了显存和计算资源，特别适用于大规模预训练模型的微调。通过这种方法，能够在少量任务特定数据上进行高效的微调，保持了原始模型的能力，同时适应新任务 [4]。

Llama-factory 实现简单微调 使用 Llama-factory 进行微调的优势在于其高效的模块化设计和优化的训练流程上。Llama-factory 提供了灵活的接口和丰富的预训练模型，使得微调过程更加简化和高效，同时也解决了训练过程因代码问题而出现效果不佳的问题。以下为参数设置：初始学习率设置为 5×10^{-5}，总共进行 3 个训练轮次，最大梯度裁剪设置为 1.0，每个数据集的最大样本数设置为 100000，使用 bf16 精度进行混合精度训练，最大输入序列长度为 2048 个标记（tokens），每个 GPU 上的批量大小设置为 1，梯度累积步数设置为 8，设置验证集比例为 0，使用 cosine 学习率调度器。

3.2 数据增强

3.2.1 数据集构建

为提升小型语言模型在 GSM8K 和 MATH 等数学推理数据集上的表现，本研究采用指令微调方法，并在训练数据集的构建过程中融合多种优化策略。

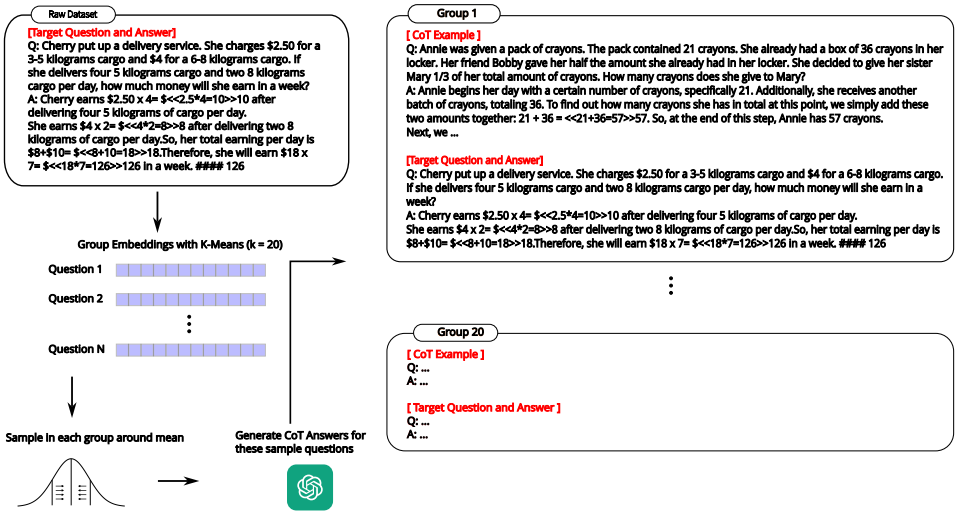

图 1: 数据集构建方法

引入思维链示例 考虑到为全部训练数据生成包含完整思维链的答案在计算资源上的限制，我们采用了一种创新的方案：在原有的问题-答案对前添加带有思维链推理（Chain of Thought，CoT）的示例。这种方法巧妙地结合了思维链学习和单样本学习（One-shot Learning）的优势。选择单样本学习的主要考虑是小型模型上下文窗口（Context Window）的限制，过多示例可能超出其处理能力。选择思维链学习的原因是原始数据集中没有详细的推理过程，而对于每一条数据构建带有 CoT 的回答显然是不现实的。

基于聚类的 CoT 示例选择策略 在选择思维链示例时，我们特别注重任务相似性原则。具体而言，对于数学问题中的不同领域（如数论、图论、物理、金融等），我们认为提供相似类型的问题解决范例能够更有效地指导模型学习。如图1所示，我们采用如下步骤：

1. 词嵌入：使用 all-distilroberta-v1 模型 [9] 提取训练集中所有问题的词嵌入表示。该模型作为 BERT 的轻量化版本，在保持性能的同时显著降低了计算开销。

2. 聚类：对获得的词嵌入进行 K-means 聚类（k=20），将训练集划分为 20 个类别。这使得每条训练数据都被分配到 [0,19] 范围内的某个类别。

3. 示例筛选：在每个类别中，选取 8 个与类别中心最接近的问题作为该类别的代表性样本。这样总共筛选出 160 条高质量示例数据。

4. 思维链生成：使用 GPT-4-mini 模型为这些示例生成包含详细推理步骤的答案。

这种基于语义相似度的系统化筛选方法，确保了示例数据的代表性和多样性，为后续的模型微调提供了高质量的训练材料。

3.3 LLM-Dagger

通过类比模仿学习，如果将大模型推理思维链的每一个语步作为 action，那么当下大模型在数学推理上出错的原因就和模仿学习中的累计误差导致 Out of Distribution 十分类似，由于中间的一个推理语步出错或者是推理到了不曾学习过的空间，就导致 Out of Distribution，从而导致后面依赖于这个语步的推理也出现错误。

为了解决这个问题，我们希望扩展大模型见过的搜索空间，我们提出了 LLM-Dagger，将大模型生成的错误样本进行纠正，再重新交给大模型进行微调，从而像一个老师一样，手把手为学生纠错，使得大模型学习到，当到达当前推理空间时应该如何继续推理到达最后答案。通过学习之前做错的样本，其实是可以大大扩展大模型在数学推理中的搜索空间。

由于资源限制，我们在实验中并没有使用人类专家，而是使用了较强的 gpt-4o 的 api，将大模型错误的推理以及正确的 ground truth 数据交给 gpt-4o，要求 gpt-4o 基于大模型错误的输出并且参考 ground truth 修正错误的推理语步，从而带给大模型更强的泛化性。

3.4 过程评测

在硬匹配评测方式中，模型生成的解题过程和答案即便逻辑正确且准确无误，仍可能因最终输出答案的格式与标准答案不完全匹配而被判定为错误。这种严格的格式约束在一定程度上限制了对模型真实能力的全面评估。为解决这一问题，我们提出了一种新的评分方法——Process Inference Score。

过程评测借助更为强大的大模型（如 Qwen-plus 等）的辅助，通过多维度对模型生成结果进行细粒度的分析与评分，包括解题步骤的完整性、逻辑性的严密程度、计算的正

确性以及最终答案的准确性等。该方法不再仅以格式匹配为评价标准，而是对模型生成内容的实际质量进行了更为宽松且全面的考量，从而更好地反映出模型在不同任务中的思维方式、推理能力及计算精度。

具体而言，过程评测旨在通过引入语义理解能力更强的大模型来验证和评分，降低因格式不匹配导致的误判问题，为模型的输出提供了更具鲁棒性和科学性的评估方式。与此同时，这种方法也对模型本身提出了更高的要求，即不仅需展现出良好的逻辑推理能力，还需在逐步计算和多步推导中保持一致性与准确性。

4 实验与结果分析

数据集	GSM8K	MATH
无微调	0	0
0.5B-Instruct	0	0
LoRa 微调	32.23	30.34
数据增强（COT）	39.74	–
LLM-Dagger	–	35.48

表 1: 注意到前两种情况下准确率都为 0，这是因为计算准确率时采用硬匹配（hard match），而这两种情况模型的输出格式都和标准格式完全不同，因此准确率为 0。

准确率（hard match） 实验结果如表 1所示，我们在 GSM8K 和 MATH 两个数据集上分别比较了以下四种情况下的模型性能：无微调的 Qwen-0.5B 模型、0.5B-Instruct 模型、LoRa 微调模型以及采用数据增强策略的模型。表中数据表明，与基线模型（无微调及 0.5B-Instruct）的表现相比，经过微调后的模型性能有了显著提升。（由于 MATH 数据集的训练集已经是进行过 COT 数据增强的了，因此无需进行性能评估。）

具体而言，LoRa 微调模型在 GSM8K 数据集上达到了 32.23% 的准确率，而在 MATH 数据集上也取得了 30.34% 的准确率，显示出 LoRa 技术在多样化任务优化中的有效性。同时，数据增强策略进一步提高了模型在 GSM8K 数据集上的表现。通过对数据进行聚类分析、匹配相关示例，并借助更强大的生成模型（GPT-4-mini）生成详细的解题步骤作为思维链引导模型进行训练，强化了模型的推理能力，最终将准确率提升至 35.89%。值得注意的是，由于计算准确率时采用了硬匹配（hard match），基线模型的输出格式与目标标准格式存在显著差异，这导致了基线模型的准确率为零，但也进一步证明了经过微调的模型在特定下游任务上的能力。

微调 为了更直观地展现微调对模型能力的影响，我们在图 2中对比了微调前后模型在同一问题上的输出表现。从图中可以观察到，在未经微调的情况下，模型不仅难以生成符合特定格式的答案，且无法提供正确的解答；而经过微调后，模型不仅能够依据预定格式推理与输出，还能生成准确的答案。这表明，微调策略能够显著改善模型的任务表现，尤其在需要结构化输出和复杂推理的情境下更为明显。

过程评测 我们在 Lora 微调和数据增强后的模型输出上分别应用了新提出的过程评测方法，结果如表 2所示。从数据中可以观察到，经过数据增强后，模型在 GSM8K 数据

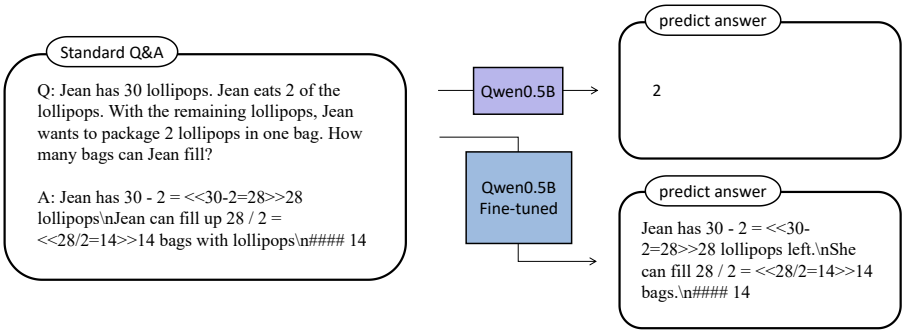

图 2: 模型微调输出展示

数据集	GSM8K	MATH
LoRa 微调	54.39	65.22
数据增强（COT）	56.25	–

表 2: 过程评测对模型生成回答质量评分，评分维度包括解题步骤的完整性、逻辑性与表述清晰度。

集上的得分有显著提升，这一结果与硬匹配评测方式中的准确率提升基本吻合，进一步验证了过程评测方法（Process Inference Score）的合理性和有效性，而其更细化和多维度的能力测量也进一步拓展了评估大语言模型表现的视角。

LLM-Dagger 我们对 MATH 数据集的输出结果进行了 LLM-Dagger 的实验，引入更强大的大模型 GPT-4o，对微调后模型的输出结果进行过程纠正，从而提升模型的性能。首先，我们找到所有的错误输出，将其加上原问题和标准答案后交给 GPT-4o，生成更符合逻辑且表达清晰的 dagger 版本。然后，在已经微调后的模型基础上使用 dagger 后的数据再次进行微调。如图 3所示：

实验结果表明，经过 LLM-Dagger 的校正和优化后，模型在 MATH 数据集上的最终准确率从原来的 30.34% 提升至 35.48%，这一增幅体现了该方法在处理数学推理任务上的有效性和适用性。此外，这一方法的改进不依赖于大规模额外的标注数据，而是充分利用已有模型和大模型的交互式校正能力，具有较高的实际应用价值。

5 结论

本工作希望优化大模型在数学推理任务上的表现，针对大模型微调的全流程，我们分别在数据、训练、评测三个维度提出了创新。我们提出的数据增强方法创新性地采用了聚类增强，有效地提升了数据中蕴含的多步推理模式。我们在训练中提出的 LLM-Dagger，有效地扩大的大模型的搜索空间，扩展了大模型的搜索树。而最后，针对当下 hard-match 的评分方法，我们提出了自己的指标——Process Inference Score，可以更细粒度地反映大模型在推理任务上的表现效果。

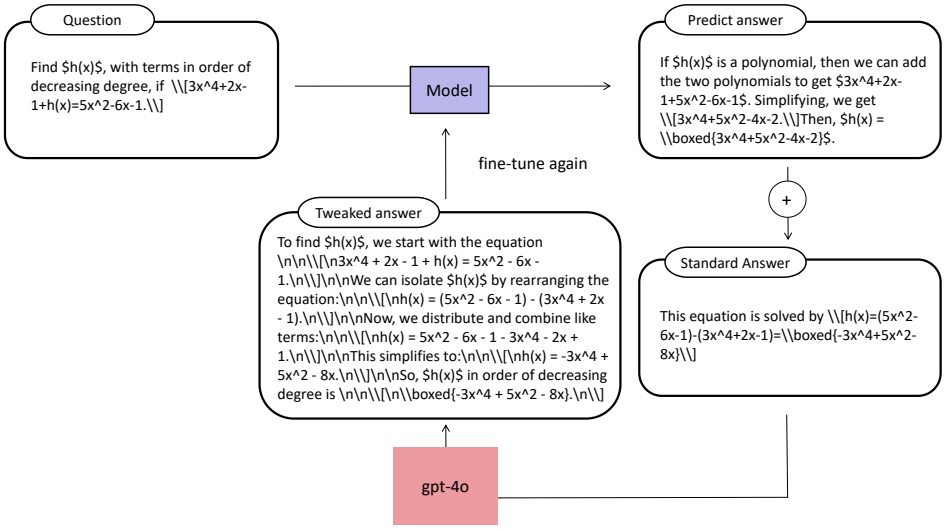

图 3: LLM-Dagger

6 限制与讨论

本文的工作还有许多值得探索的地方，当大模型的能力不断趋于人类，我们不得不思考的问题就是，还有谁能作为专家去纠正大模型的错误从而使得大模型超越人类。而具体到我们提出的 LLM-Dagger，很值得尝试的是采用模型本身去对错误进行纠正，由于评判与生成是两个任务，所以很有希望模型可能在评判过程中展现出生成时不具备的能力，因此值得探索的时能否通过自我纠正的方式来训练出超越人类的大模型。

另外，本文工作在 GSM8K 与 MATH 上进行了微调，并且进行评测，但分数的提升能否证明一定时模型推理能力的提升是要打一个问号的，应该需要更多的实验证明微调后的模型没有过拟合到某个数据集上，而是实打实地获得了推理能力的提升。

7 成员贡献

詹佳豪：项目负责人，负责构思、组织讨论本工作框架，提出了 LLM-Dagger，参与了 Process Inference Score 的实现、lora 微调的调试与训练，以及论文 Introduction、Abstract 部分的撰写。

杨永卓：训练部分主要负责人，负责了 llama-factory 进行微调的实验部分，在实验结果上做出了突出贡献，参与了论文撰写。

杨淳瑜：数据部分主要负责人，负责了数据增强部分全链路的设计与实现，并且负责了 LLM-Dagger 的数据修正，参与了论文撰写。

王海天：论文部分主要负责人，负责了论文主体的整合与撰写，负责了 Process Inference Score 的实验部分。

8 附录

8.0.1 数据增强相关工作

上下文学习对于提升大语言模型在特定任务上的表现具有显著作用。根据对任务数据集的依赖程度，我们可以将上下文学习方法大致分为两大类：基于提示词工程的方法（无需模型微调）和基于微调的方法。我们首先系统地介绍这些方法的特点与应用场景。

基于提示词工程的方法主要有如下几种：

少样本学习（Few Shot Learning） 尽管大语言模型由于在与训练阶段展现了展现出了强大的泛化能力和显著的零样本学习能力 [5]，但在面对复杂任务（如数学推理）或规模较小的模型（如 Qwen 0.5B）时，其表现可能仍有提升空间。少样本学习通过在提示词中引入有限数量的示例来增强模型的任务理解能力。这些示例通过类似条件概率的机制影响模型的输出分布，从而提高生成质量。实验证明，少样本提示显著提升了模型的输出质量 [1]。

思维链学习（Chain-of-Thought Learning） 思维链学习 [13] 通过引导模型逐步推理，提升其准确性，主要分为两种实现方式：

1. **少样本提示**：在提示词中嵌入带推理解释的示例，随后添加实际任务指令，帮助模型通过案例建立推理逻辑。

2. **零样本提示**：通过引导语（如 Let's think step by step），将推理过程分为两步：先生成详细推理步骤，再从中提取答案。这种方法通常在第一步已得出正确结果，第二步主要用于验证和总结 [17]。

思维链推理方法显著提升了大型语言模型在复杂认知任务中的推理能力和表现 [13]。

上述方法虽能有效提升模型性能，但均未涉及模型参数的调整。为进一步提升模型表现，研究者提出了基于参数微调的方法。

传统微调（Vanilla Fine-tuning） 该方法在预训练模型的基础上，利用特定任务的标注数据集直接更新模型参数。这种方法在多个评估基准上展现出良好的性能 [1]。然而，其主要局限在于：首先，每个新任务都需要大规模（通常是数万级别）的标注数据；其次，模型容易出现过拟合现象，导致泛化能力受限。

指令微调（Instruction Fine-tuning） 这种方法采用多任务学习范式，通过在预训练模型上使用多个不同任务（如任务 B、C、D）的指令-答案对进行微调，来提升模型的通用能力。关键在于，评估时所用的目标任务（任务 A）必须与训练集中的任务不同。这种方法的核心目标是增强模型在未见任务上的零样本学习能力。研究表明，指令微调能显著提升模型的零样本泛化性能 [14]。

References

[1] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[2] Karl Cobbe et al. *Training Verifiers to Solve Math Word Problems*. 2021. arXiv: 2110.14168 [cs.LG]. URL: https://arxiv.org/abs/2110.14168.

[3] Dan Hendrycks et al. *Measuring Mathematical Problem Solving With the MATH Dataset*. 2021. arXiv: 2103.03874 [cs.LG]. URL: https://arxiv.org/abs/2103.03874.

[4] Edward J Hu et al. "Lora: Low-rank adaptation of large language models". In: *arXiv preprint arXiv:2106.09685* (2021).

[5] Takeshi Kojima et al. *Large Language Models Are Zero-Shot Reasoners*. Jan. 29, 2023. DOI: 10.48550/arXiv.2205.11916. arXiv: 2205.11916 [cs]. URL: http://arxiv.org/abs/2205.11916 (visited on 12/13/2024). Pre-published.

[6] Hunter Lightman et al. *Let's Verify Step by Step*. 2023. arXiv: 2305.20050 [cs.LG]. URL: https://arxiv.org/abs/2305.20050.

[7] Pan Lu et al. *Dynamic Prompt Learning via Policy Gradient for Semi-structured Mathematical Reasoning*. 2023. arXiv: 2209.14610 [cs.LG]. URL: https://arxiv.org/abs/2209.14610.

[8] Joon Sung Park et al. *Generative Agents: Interactive Simulacra of Human Behavior*. 2023. arXiv: 2304.03442 [cs.HC]. URL: https://arxiv.org/abs/2304.03442.

[9] Victor Sanh et al. *DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter*. Mar. 1, 2020. DOI: 10.48550/arXiv.1910.01108. arXiv: 1910.01108 [cs]. URL: http://arxiv.org/abs/1910.01108 (visited on 12/13/2024). Pre-published.

[10] Trieu H. Trinh, Yi Wu, Quoc V. Le, et al. "Solving olympiad geometry without human demonstrations". In: *Nature* 625 (2024), pp. 476–482. DOI: 10.1038/s41586-023-06747-5. URL: https://doi.org/10.1038/s41586-023-06747-5.

[11] Guanzhi Wang et al. *Voyager: An Open-Ended Embodied Agent with Large Language Models*. 2023. arXiv: 2305.16291 [cs.AI]. URL: https://arxiv.org/abs/2305.16291.

[12] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL]. URL: https://arxiv.org/abs/2201.11903.

[13] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. Jan. 10, 2023. DOI: 10.48550/arXiv.2201.11903. arXiv: 2201.11903 [cs]. URL: http://arxiv.org/abs/2201.11903 (visited on 12/13/2024). Pre-published.

[14] Jason Wei et al. *Finetuned Language Models Are Zero-Shot Learners*. Feb. 8, 2022. DOI: 10.48550/arXiv.2109.01652. arXiv: 2109.01652 [cs]. URL: http://arxiv.org/abs/2109.01652 (visited on 12/13/2024). Pre-published.

[15] Shunyu Yao et al. *ReAct: Synergizing Reasoning and Acting in Language Models*. 2023. arXiv: 2210.03629 [cs.CL]. URL: https://arxiv.org/abs/2210.03629.

[16] Shunyu Yao et al. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. 2023. arXiv: 2305.10601 [cs.CL]. URL: https://arxiv.org/abs/2305.10601.

[17] Zhuosheng Zhang et al. *Automatic Chain of Thought Prompting in Large Language Models*. Oct. 7, 2022. DOI: 10.48550/arXiv.2210.03493. arXiv: 2210.03493 [cs]. URL: http://arxiv.org/abs/2210.03493 (visited on 12/13/2024). Pre-published.