利用 RAFT 提升大型语言模型驱动的 EDA 助手

Luyao Shi
IBM Research
San Jose, CA
luyao.shi@ibm.com

Michael Kazda
IBM Infrastructure
Poughkeepsie, NY
kazda@us.ibm.com

Charles Schmitter
IBM Infrastructure
Poughkeepsie, NY
charles.schmitter@ibm.com

Hemlata Gupta
IBM Infrastructure
Poughkeepsie, NY
guptah@us.ibm.com

***Abstract*—电子设计工程师在进行设计验证和技术开发等任务时通常难以有效获取相关信息。虽然大型语言模型（LLM）可以作为对话代理来提高生产力，但预训练的开源 LLM 缺乏电子设计自动化（EDA）的领域特定知识。在检索增强生成（RAG）环境中，LLM 依赖外部上下文，但可能仍会产生不准确的响应。检索增强微调（RAFT）可以提高 LLM 的性能，但在 EDA 领域获取标记的问题/答案（Q/A）数据很困难。为了解决这个问题，我们提出使用合成 Q/A 数据集来通过 RAFT 增强 LLM。我们的结果表明，使用合成数据的 RAFT 显著提升了 LLM 在基于 RAG 的 EDA 任务中的表现。我们还研究了使用真实用户问题作为检索增强小样本（RAFS）示例来生成合成数据的影响。此外，我们实施了安全访问控制，以确保敏感信息仅对授权人员可见。最后，我们评估了在使用合成数据进行微调时的数据泄漏和意外记忆化风险，并提供了实际的见解。**

***Index Terms*—EDA, LLM, fine-tuning, RAG, RAFT**

I. 引言

设计工程师在处理复杂的工作流程时常常面临巨大挑战，尤其是在设计建造和验证方面。在大型组织中，一个主要障碍是如何高效地获取相关文档或找到合适的领域专家以获得指导。文档可能以多种冗余版本存在，或者分散在不同地点，使得检索变得困难。对于新员工或者当需要整合不熟悉的工具时，这些困难尤为明显，因为专业术语、行话和缩略语会造成额外的障碍。一个智能的、随时可用的助手能够提供及时和准确的信息，这将极大地提升工程效率和决策能力。

大型语言模型（LLMs）[1] 已成为在各种任务中生成类似人类响应的强大工具。然而，它们的可靠性受到过时的训练数据以及缺乏特定领域或专有知识的限制。此外，LLMs 经常出现幻觉，在面对不熟悉的话题时自信地产生错误的响应——这一问题在需要精确性的工程应用中尤为严重。检索增强生成（RAG）[2] 通过将外部知识检索整合到响应生成中，解决了这一挑战，确保输出准确且具备上下文相关性。在设计环境中的显著应用包括 ChipNemo [3]，它是一个利用 RAG 增强精准性并减少幻觉的面向 EDA 的特定领域 LLM，以及 Ask-EDA [4]，它结合了混合 RAG 和缩写去幻觉（ADH），以提供更准确且上下文相关的响应。

在使用大型语言模型时，确保数据安全是一个关键问题，特别是在芯片设计等专业领域，通过第三方 API 交互可能会暴露专有信息。通过开发特定领域的内部模型，组织可以在降低这些风险的同时，在专业任务上取得更好的性能。研究表明，为特定领域量身定制的大型语言模型不仅增强了安全性，还提高了准确性和可靠性。最近的努力，如 ChipNemo [3]、ChatEDA [5] 和 OpenROAD-Assistant [6]，已证明通过电子设计自动化（EDA）特定数据微调大型语言模型的好处，从而在该领域产生更有效的结果。

领域特定的微调可以改善 LLM 的性能，但也面临着一些挑战，例如过时的知识、大量的数据需求和过拟合。虽然 RAG 可以通过让模型检索外部信息而不是仅仅依赖静态参数来解决这些限制，但它也有自身的缺点。通用 LLM 可能在将检索到的领域专有信息进行连贯整合时遇到困难，导致响应中出现不一致或误解。此外，RAG 依赖于检索文档的质量和相关性，检索失败仍可能导致幻觉。检索增强微调（RAFT）[7] 通过专门为 RAG 优化模型来缓解这些问题。RAFT 微调 LLM 以更好地将检索到的上下文整合到其生成过程中，适应模型动态检索和根据来自知识库的实时数据调整其响应。虽然 RAFT 在 EDA 领域 [6] 中表现出了良好的效果，但它仍依赖于标注的问答对 [8]，而这些问答对在许多技术领域中往往稀缺或不可用。

在本文中，我们提出使用合成问答数据集通过 RAFT 增强 LLMs，展示了对基于 RAG 的 EDA 助手的显著改进。合成数据可以利用广泛可用的跨越各种技术领域的未标记文档，从而减少对稀缺的人工标记数据集的依赖。我们还探讨了在合成数据生成中结合真实用户问题作为检索增强小样本（RAFS）示例的影响。此外，我们在我们的 EDA-LLM 助手中实施了安全访问控制，确保只有授权人员可以通过基于检索的上下文管理访问敏感信息。为了评估潜在风险，我们调查了在合成数据上进行检索增强微调（RAFT）是否会导致意外记忆和数据泄漏，并根据我们的研究结果提供实际的考虑。

我们强调我们的主要贡献如下：

- 我们建议使用合成的问答数据集，通过 RAFT 增强 LLMs，利用广泛可用的无标签文档以减少对稀缺标签数据的依赖，并提高基于 RAG 的 EDA 助理的性能。
- 我们建议在合成数据生成中将真实用户问题作为检索增强少样本（RAFS）示例纳入，评估其在提高大型语言模型（LLM）性能方面的益处。
- 我们实施了一种基于检索的访问控制机制，确保敏感信息仅供授权人员访问，同时保持可用性。
- 我们研究了基于合成数据进行 RAFT 微调是否会导致非预期的记忆化和数据泄露，并提供了一些实用的考量来减轻这些风险。

II. 方法

大型语言模型驱动的 EDA 助手（EDA-LLM）的微调工作流程如图 1 所示。

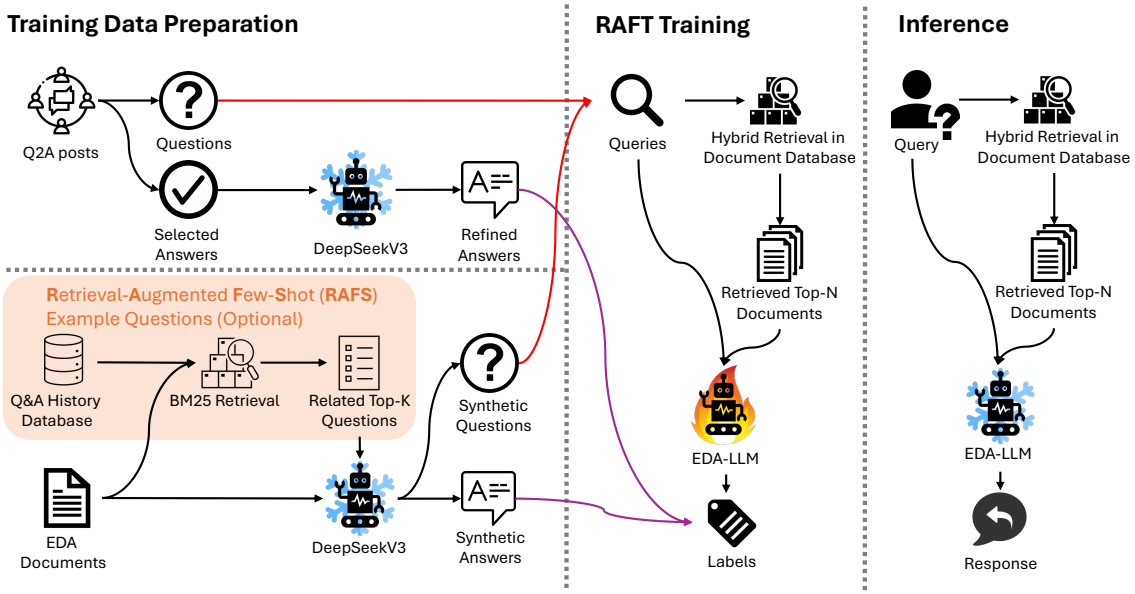

Fig. 1. 训练数据准备、RAFT 训练和推理的工作流程概述。

A. 训练数据准备

我们利用了两种数据来源，并准备了两种类型的训练集用于微调 EDA-LLM。

Q2A 帖子：我们有一个类似于 Stack Overflow 的系统，工程师可以在其中提出问题，领域专家（SMEs）提供答案和评论。投票系统允许将最佳答案标记。这些专家的回应形成了一个带有真实标签数据的集体知识库。然而，人类撰写的答案往往是自由格式的，并包含不必要的元素，如问候语（例如，“嗨，约翰，回答你的问题……”）。为了增强清晰度、语法和连贯性，我们使用 DeepSeek-V3 来优化这些答案。经过优化的问题答案数据集被纳入训练集中，以提高 EDA-LLM 的响应结构和清晰度。用于优化的提示如图 2 所示。

> You are a helpful assistant. Given a question and its corresponding answer, improve the answer's phrasing for better grammar, clarity, and coherence while preserving its original meaning. Additionally, remove any irrelevant personal references such as names or greetings. Output only the revised answer and nothing else.
>
> Question: **<Raw Question>**;
> Answer: **<Raw Answer>**

Fig. 2. 用于在 Q2A 帖子中改进答案的提示，以提高语法、清晰度和连贯性。

合成问答：尽管 Q2A 帖子提供专业的回复，但其数量往往有限。此外，在用 RAFT 训练 EDA-LLM 时，无法保证相关文件会出现在检索的上下文中。为了解决这一问题，我们利用更大的未标记 EDA 文档库，并采用 DeepSeek-V3 根据这些文档生成合成问题及相应的答案。用于从给定文档生成合成问答对的提示如图 3 所示。我们观察到某些文档，例如设计指南，比其他文档（如命令参考）更复杂。因此，在提示中，我们鼓励 LLM 先分析文档并评估其复杂性：如果文档包含复杂概念，我们要求 LLM 生成一个需要更深理解和推理的挑战性问题；如果文档较简单或更直接，我们要求 LLM 生成一个测试用户对材料基本理解的一般问题。我们在图 4 中提供了一些合成的 Q & A 示例。可以看到，从设计指南文档生成的合成问题更为复杂，需要较深的理解和推理，而基于命令参考文档的问题更为简单。在这两种情况下，合成的答案都有效地解决了相应的问题。

我们还保留了一系列来自 EDA 助理部署过程中的用户问题和相应的 LLM 生成的回答。这些问题可以作为检索增强的少样本（RAFS）示例，指导 Q & A 生成 LLM（此处为 DeepSeek-V3）生成更符合人类在复杂性和风格上真实询问的问题。在我们的实现中，我们首先将每个问题与其对应的 LLM 回答连接在 Q & A 历史数据库中。这种连接为后续的相关搜索提供了额外的上下文。然后，针对目标 EDA 文档，我们使用 BM25 搜索识别出前 K（K=5）个最相关的连接 Q & A 对。这些前 K 的问题被用作提示中的少样本示例（在图 2 中的虚线框中突出显示）用于合成 Q & A 生成（如图 1 的橙色阴影部分所示）。

B. 检索增强微调 (RAFT)

的 Q2A 帖子和合成 QA 数据集用于使用 RAFT 微调 EDA-LLM。如图 1 所示，给定查询，我们首先执行混合搜索以从我们的文档数据库中检索前 N 个相关的文档片段。根据 [4]，我们采用互反排序融合（RRF）将语义搜索结果（使用句子转换器 [9] 嵌入）与词汇搜索结果（使用 BM25 [10]）结合起来。然后将查询和前 N 个检索到的文档片段合并到提示中，训练 EDA-LLM 生成真实答案。同样地，在推理时，我们保持 EDA-LLM 冻结，并要求 EDA-LLM 基于提供的查询和检索到的上下文生成响应。用于 RAFT 训练和基于 RAG 推理的提示如图 5 所示。

C. 安全访问控制

在构建基于 LLM 的 EDA 助手时，一个关键考虑因素是对摄取的数据实施安全访问控制。一些文档，如受限设计指南，包含敏感信息，应仅限于授权人员访问，而像命

You are an expert AI specializing in technical content generation.
Your task is to generate a synthetic question and its corresponding answer based on a given technical document.

Reference Examples:
Below are five example questions. Use them as inspiration for structuring your response in terms of complexity, style, and relevance.
<Retrieved Questions>

Instructions:
- Analyze the document carefully and assess its complexity.
- If the document contains complex concepts, generate a challenging question that requires deeper understanding and reasoning.
The question should be specific, focus on nuanced details,
and test someone's ability to analyze or synthesize information.
- If the document is simpler or more straightforward, generate a general question that tests the user's basic comprehension of the material.
- The question should always be clear, precise, and end with a question mark.
- The answer should be directly derived from the document and be accurate, concise, and informative.
- If the document is complex, the answer should also provide reasoning or justification where appropriate. For simpler documents, the answer should be straightforward but still correct and relevant to the question.
- Do not add information beyond what is contained in the document, but feel free to explain any nuanced details or connections between concepts where applicable.

Output Format:
Question: [Generated question]
Answer: [Generated answer]

Now, analyze the following document and generate the output accordingly, adjusting the difficulty of the question based on the document's complexity.

Document:
<Target Document>

Fig. 3. 从目标文档生成合成问答对的提示。作为少样本示例使用的五个检索增强示例问题（在虚线框中突出显示）是可选的。

Document Category: Design Guide
Synthetic Question: How does the concept of maximum skew impact the functional correctness of an asynchronous clock domain crossing, and what formula determines the tolerable skew for a given circuit?
Synthetic Answer: Maximum skew is the maximum difference in arrival times of transitions from a given source clock domain at a given set of receive latches/flip-flops. It is critical to functional correctness because excessive skew can cause the circuit to malfunction, as demonstrated in the example where data might still be toggling during the sample window of the receive latch, leading to unreliable sampling. The tolerable skew for a given circuit is determined by the formula: (2 * (receive clock period)) - guardband.

Document Category: Command Reference
Synthetic Question: What is the purpose of the `msg::group_set_options` command, and which option is required when using it?
Synthetic Answer: The `msg::group_set_options` command is used to set all group options in one command, ensuring it is thread safe. The required option when using this command is `-group string` (or `group=string` in Python), which specifies the group to set the output file on.

Fig. 4. 合成 Q & A 对的示例。来自设计指南的问题更复杂，需要更深入的推理，而那些来自命令参考的问题更为直接。

令参考这样的通用 EDA 文档在组织内的所有人员都可以访问。

为了实施访问控制，我们在导入过程中将每个文档块与其对应的访问组相关联。当用户提交查询时，我们会检索其访问组信息，并使用 Elasticsearch [11] 来过滤出他们被授权访问的文档。然后，这些经过筛选的文档被整合到提

You are an advanced AI assistant designed for Retrieval-Augmented Generation (RAG).
Your primary task is to generate well-informed, accurate, and contextually relevant responses based on retrieved documents.

Instructions:
-Use Only the Provided Context:
Base your answer strictly on the retrieved documents. Do not rely on external knowledge or make assumptions.
-Synthesize Information:
If multiple sources provide relevant details, combine them into a coherent and comprehensive response.
-Handle Conflicting Information:
If sources contradict each other, acknowledge the discrepancy rather than assuming correctness.
-Be Clear and Concise:
Provide direct answers while maintaining completeness and readability.
-Indicate Uncertainty:
If the retrieved documents do not contain enough information, state this explicitly rather than guessing.
-Return the Most Relevant Command (if applicable):
If the question asks for a command, return the first relevant one.

Retrieved Documents:
The following excerpts have been retrieved from multiple sources:
<Retrieved Documents>

Question:
<Query>

Fig. 5. 用于 RAFT 训练和基于 RAG 推理的提示。

示中，并发送给 LLM 用于生成响应。

然而，在这种情况下，确保 EDA-LLM 在 RAFT 期间不记忆敏感信息至关重要，因为这可能导致意外的数据泄露给未经授权的人员。为了验证这一点，我们创建了一个测试集 SynthQA_MissingContext，该测试集来源于 Synthetic QA 训练集。

在此次评估中，我们修改了提示，移除了任何来自用于生成合成 Q & A 的目标 EDA 文档的检索文档块。这防止模型在回答问题时依赖这些来源。我们的目标是确定精调 LLM 是否记住了训练中的信息，并且在缺少相关检索文档的情况下仍然能够生成答案。此外，我们调查了是否在缺少上下文的情况下包含一些训练样本能帮助模型在缺乏足够信息时适当地拒绝生成答复。

III. 评估

我们收集了 749 个 Q2A 帖子，如第 II 节所述。我们过滤掉了没有标记最佳答案的帖子，以及那些答案仅由参考链接（例如，网页链接）组成的帖子。剩下的 405 个 Q & A 帖子使用 DeepSeek-V3 进行了优化，然后随机分为 305 个样本的训练集和 100 个样本的测试集。我们在整篇论文中将此测试集称为 Q2A。

对于合成问答生成，我们首先移除了非常短的 EDA 文档（少于 1,000 个字符），并抽取了 1,000 篇文档。其中，较长的文档被截断以仅保留前 10,000 个字符。文档类别列在表格 I 中。在每个类别中，我们在样本量允许的情况下，尽量按 90% /10% 的比例划分样本。相应的数量也在表格 I 中显示。按照 II .A 中的步骤，我们使用 DeepSeek-V3 为每个 EDA 文档生成一对合成问答。我们在整篇论文中称这一测试集为 SynthQA。对于 Q2A 和 SynthQA 问题，我们使用混合搜索加上 RRF 检索出排名前 N（N=10）的

文档块，将查询和前 N 检索到的块一起包括在每个样本的提示中。每个块长度为 2,000 字符，重叠部分为 200 字符。请注意，检索空间由我们整个文档集合组成，包括大约 200,000 个块，来自 14,000 篇文档。

如在第 II .C 节所述，我们进一步从 SynthQA 训练集抽取了 100 个人工合成的 Q & A 对，并从提示中移除相关的检索到的文档片段以模拟缺失的上下文。我们在整篇论文中称这 100 个测试样本为 SynthQA_MissingContext。

TABLE I
文档类别及其对应的训练/测试拆分用于合成 QA 数据集。

Category/Number	Train	Test
Parameter Reference	1	1
Timing	24	3
DevOps	237	26
Design Guide	295	33
Command Reference	343	37
Total	900	100

A. 实现细节

DeepSeek-V3 [12] 是一个强大的专家融合（MoE）语言模型，总参数量为 6710 亿，每个 token 激活 37 亿参数。在训练数据准备中，我们利用 DeepSeek-V3 [1] 来生成高质量的精炼答案和合成问答对，如图 1 所示。在我们的实验中，我们使用 Llama-3.1-8B-Instruct [2] [13] 作为我们的 EDA-LLM 助手并进行微调。此外，我们将其表现与几个基准 LLM 模型进行比较，包括 Mistral-7B-Instruct-v0.3 [3] 、Mistral-Nemo-Instruct-2407 [4] （一个 12B 模型）、Mistral-Small-Instruct-2409 [5]（一个 22B 模型）和 Llama-3.1-70B-Instruct [6] 。

在我们的实验中，我们使用 Unsloth [14] 以实现快速训练和低内存使用。EDA-LLM 模型经过参数高效微调（PEFT）和低秩适应（LoRA）[15] 的微调，应用 LoRA 秩为 128 的低秩适应，$\alpha = 32$ ，并设置丢弃率为 0。所有模型都以 8192 个 tokens 的最大序列长度、4 比特量化和梯度检检查点进行训练。训练在 8 个批大小、2 次梯度累积步骤、学习率 2×10^{-5} 中进行，共进行 5 个周期，热身比率为 0.1，权重衰减为 0，并采用余弦学习率调度器。所有训练均在单个 Nvidia A100 80GB GPU 上进行。请注意，为了公平对比，所有基线模型的推理也使用 Unsloth 进行 4 比特量化（除了用于高质量 Q & A 生成的 DeepSeek-V3）。

对于混合检索，我们使用互惠等级融合（RRF）将语义搜索结果（来自句子转换器嵌入）与词法搜索结果（使用 BM25）结合起来。我们的语义搜索是通过我们内部的双编码器句子转换器模型 IBM Slate 125m [16] 进行的，尽管我们的方法对嵌入模型的选择是没有限制的。

[1]https://huggingface.co/deepseek-ai/DeepSeek-V3
[2]https://huggingface.co/unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit
[3]https://huggingface.co/unsloth/mistral-7b-instruct-v0.3-bnb-4bit
[4]https://huggingface.co/unsloth/Mistral-Nemo-Instruct-2407
[5]https://huggingface.co/unsloth/Mistral-Small-Instruct-2409
[6]https://huggingface.co/unsloth/Meta-Llama-3.1-70B-Instruct

B. 度量标准

BERTScore [17] 和 BARTScore [18] 是评估生成文本质量的度量指标，它们衡量的不仅仅是精确的词汇重叠。BERTScore 从类似 BERT 的模型中计算词级别的嵌入，以一种上下文敏感的方式测量语义相似性。具体来说，我们使用”microsoft/deberta-xlarge-mnli” 作为 BERT 评分模型，这是表现最佳的模型之一。

BARTScore 测量在给定一个文本序列的情况下生成另一个文本序列的对数似然。它使用 BART 模型分配的概率来评估生成文本与参考文本之间的语义和上下文对齐。注意，BARTScore 的原始分数是难以解释的负对数概率。因此，我们使用归一化的 BART 分数来计算精度和召回率，公式如下 [7] ：

$$\text{Precision} = \frac{\text{BARTScore}(Ref, Ref)}{\text{BARTScore}(Ref, Pred)} \tag{1}$$

$$\text{Recall} = \frac{\text{BARTScore}(Ref, Ref)}{\text{BARTScore}(Pred, Ref)} \tag{2}$$

在这些方程中，Pred 指的是预测文本，Ref 指的是参考文本。如方程 1 和 2 所示，基于 BARTScore 的精确率和召回率通过 Ref 的自相似性进行归一化，因此分数范围在 0 到 1 之间。根据 [18] ，我们还加入了提示，以指导 BART 模型对任务的理解。具体而言，我们在 BARTScore(x, y) 中的 y 之前加入提示 z ，其中 z $\in$ { “也就是说，”，“换句话说，”，“重述一下，”，“即，” } ，并报告 4 个提示的平均分数。最后，F1 分数计算为精确率和召回率的算术平均。我们使用在 “ParaBank2” 上微调的 “facebook/bart-large-cnn” 模型作为我们的 BART 评分器。

C. 结果

在 Q2A 和 SynthQA 测试集上的主要评估结果如表 II 所示。对于使用 RAFT 微调的 EDA-Llama-8B 模型，“D1” 指的是在单一数据源上的训练，即包含 305 个训练样本的 Q & A 帖子。“D2” 是通过加入来自 SynthQA 的额外 900 个训练样本（总共 1,205 个训练样本）来使用两个数据源。“D2-RAFS” 也使用两个训练数据源，但利用了带有检索增强少样例问题的 SynthQA（也是 1,205 个训练样本）。值得一提的是，“D2” 和 “D2-RAFS” 中的合成问题和答案是不同的。还请注意，100 个 SynthQA 测试例子是在没有 RAFS 问题例子的情况下生成的。

如表 II 所示，与基线 LLM 模型相比，使用 EDA 领域特定数据（EDA-Llama-8B-D1）的检索增强微调（RAFT）显著提高了 Q2A 测试集上的性能。然而，在 SynthQA 测试集中，仅精确度有所提高，而召回率没有改善。这表明在有限的 Q2A 数据集上进行训练有助于 LLM 生成更简洁的回答，但较低的召回率可能源于 SynthQA 问题的复杂性和多样化风格的增加。值得注意的是，Q2A 问题是人类撰写的，通常比较宽泛，而 SynthQA 由特定文档衍生的合成问题组成，需要更深入的理解和对技术细节的推理。

在加入 SynthQA 训练数据（D2）后，模型在 Q2A 和 SynthQA 测试集上的表现都进一步提升，尤其是在 SynthQA 测试结果上取得了显著的进步。

[7]https://huggingface.co/datasets/tau/scrolls/blame/7d4a9140e65653bf8bc2aece85c3b774cd1ff7c8/metrics/bart_score.py

TABLE II
Q2A 和 SynthQA 测试集的主要评估结果。每个数据集每个指标的最高分用粗体显示。

Models/Metrics	Q2A						SynthQA					
	BERT Score (%)			BART Score (%)			BERT Score (%)			BART Score (%)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Mistral-7B-Instruct-v0.3	78.60	81.00	79.66	32.73	36.68	34.71	79.74	81.87	80.72	26.12	30.44	28.28
Mistral-Nemo-Instruct-2407	77.55	79.23	78.22	29.51	34.22	31.86	75.03	80.09	77.40	21.27	29.52	25.39
Mistral-Small-Instruct-2409	80.99	82.33	81.56	35.28	39.52	37.40	78.70	81.74	80.11	25.17	31.72	28.44
Llama-3.1-8B-Instruct	79.61	80.28	79.84	32.73	36.73	34.73	75.73	80.13	77.80	21.06	29.89	25.48
Llama-3.1-70B-Instruct	82.52	81.41	81.85	38.21	38.51	38.36	79.74	81.54	80.56	25.90	30.84	28.37
EDA-Llama-8B-D1	84.28	84.38	84.26	41.40	42.50	41.95	78.25	79.93	79.01	24.13	27.90	26.02
EDA-Llama-8B-D2	84.35	84.75	84.47	41.05	43.50	42.28	83.47	82.76	83.07	31.45	31.48	31.47
EDA-Llama-8B-D2-RAFS	84.95	85.52	85.17	43.69	45.48	44.59	82.86	82.77	82.78	29.99	31.50	30.74

在 SynthQA 训练的提示中引入检索增强少样本(RAFS)问题示例（D2-RAFS）进一步提升了 Q2A 测试集上的表现，但对 SynthQA 没有显著提升。这可能是因为少样本示例是从过去的用户问题数据库中检索出来的，能够更好地与 Q2A 测试问题的风格对齐。这个发现表明，对于真实世界的 EDA 助手部署，当模型与人类用户互动时，在 SynthQA 训练过程中加入 RAFS 问题示例可能会更有益。

表 III 呈现了在 RAG 提示中处理缺失上下文的探索性结果。具体来说，我们评估了“SynthQA_MissingContext”测试集（见章节 III .A），以评估微调模型在未提供任何源上下文的情况下记忆真实答案和生成响应的程度。这对于在 RAG 中实施安全访问控制是一个关键考虑因素，确保 EDA-LLM 助手不会无意中向未授权用户披露敏感信息。理想情况下，在“SynthQA_MissingContext”上的召回应该尽可能低，而在 SynthQA 上则应保持较高。然而，与基础的 Llama-8B 模型相比，微调的 D2 模型显示出增加的召回率，表明 RAFT 微调使 LLM 在某种程度上记住了答案，即使相关文件在 RAG 上下文中缺失。同时也值得注意的是，基础模型在“SynthQA_MissingContext”上的召回分数并不是特别低。这可能是因为，即使没有在 EDA 数据上进行微调，基础模型也尝试从提取的片段中提取相关信息并生成响应。我们手动验证了基础模型在 6 个问题上以“我不知道”类型的回答进行回应（在表 III 中为 # IDK）。尽管这些回答在技术上可能不准确，但它们仍可能展示出与真实情况中等程度的语义相似性。此外，虽然我们移除了用于生成合成 Q & As 的来源文件，但不能保证这些是唯一包含必要信息的文件。模型的响应可能基于其他我们无法识别和从提示中排除的检索到的相关文件。

我们试图重用训练样本中的 10 个% 和 20 个%（确保与测试集无重叠）并从提示中移除相关上下文，以创建“缺失上下文”训练样本，并将“我不知道”（IDK）响应指定为训练标签。然后将这些样本纳入训练集，形成模型 D3-MC10 % 和 D3-MC20 %。如表 III 所示，这些模型确实在 SynthQA_MissingContext 测试集上产生了较低的召回率分数，并更可能生成 IDK 类型的响应。然而，正如在 SynthQA 测试集上的召回率分数减少和 # IDK 事件增加所表明的，即使在提示中提供了源信息，它们也倾向于产生 IDK 响应。

防止使用 RAFT 进行 LLM 微调时记住和泄露敏感信息仍然是一个挑战。在开发出有效的解决方案之前，我们建议将 RAFT 微调限制在非敏感数据上，或者为不同用户群体维持独立的 EDA-LLM 助手或不同的 LoRA 适配器。

IV. 结论与讨论

本文提出使用合成的问答数据集，通过检索增强微调（RAFT）来提高 LLM 在 EDA 任务中的性能。通过利用广泛可用的无标签文档，我们减少了对稀缺标记数据的依赖，提高了 LLM 在 RAG 上下文中的准确性和功能。我们将真实用户问题作为检索增强少样本（RAFS）示例融入合成数据生成中进行研究，证明这种方法进一步改善了模型性能。我们还通过实施基于检索的访问控制机制解决了重要的安全问题，确保敏感信息仅限于授权人员访问。此外，我们评估了合成数据微调过程中无意记忆和数据泄漏的潜在风险，并提供了宝贵的见解和实际建议来减轻这些问题。尽管用于评估的 SynthQA 测试集中包含尚未经人类验证的合成答案，但 Q2A 集合的结果显示出明确且有希望的趋势，突显了我们方法的有效性。未来的工作将重点利用更大和更多样化的合成問答集，以进一步改善训练。尽管存在限制，我们的结果强调了使用合成数据的 RAFT 在构建可靠且安全的 EDA-LLM 助手方面的有效性，为领域特定 LLM 应用的未来进步铺平了道路。

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[3] M. Liu, T.-D. Ene, R. Kirby, C. Cheng, N. Pinckney, R. Liang, J. Alben, H. Anand, S. Banerjee, I. Bayraktaroglu *et al.*, “Chipnemo: Domain-adapted llms for chip design,” *arXiv preprint arXiv:2311.00176*, 2023.

[4] L. Shi, M. Kazda, B. Sears, N. Shropshire, and R. Puri, “Ask-eda: A design assistant empowered by llm, hybrid rag and abbreviation de-hallucination,” in *2024 IEEE LLM Aided Design Workshop (LAD)*. IEEE, 2024, pp. 1–5.

[5] H. Wu, Z. He, X. Zhang, X. Yao, S. Zheng, H. Zheng, and B. Yu, “Chateda: A large language model powered autonomous agent for eda,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.

TABLE III
关于在 RAG 提示中处理缺失上下文的探索性结果。在 SynthQA_MissingContext 上较低的召回率表明在相关上下文缺失时较少的记忆化，而在 SynthQA 上当相关上下文存在时召回率应保持高水平。更多 "I don't know" (# IDK) 的回答应出现在 SynthQA_MissingContext 上，而在 SynthQA 上应更少。

Models/Metrics	SynthQA_MissingContext			SynthQA		
	BERT Recall (%)	BART Recall (%)	# IDK	BERT Recall (%)	BART Score (%)	# IDK
Llama-3.1-8B-Instruct	74.86	20.73	6	80.13	29.89	0
EDA-Llama-8B-D2	76.57	21.01	0	82.76	31.48	0
EDA-Llama-8B-D3-MC10 %	66.78	16.72	64	81.11	29.72	8
EDA-Llama-8B-D3-MC20 %	65.84	26.23	70	79.67	28.91	12

[6] U. Sharma, B.-Y. Wu, S. R. D. Kankipati, V. A. Chhabria, and A. Rovinski, "Openroad-assistant: An open-source large language model for physical design tasks," in *Proceedings of the 2024 ACM/IEEE International Symposium on Machine Learning for CAD*, 2024, pp. 1–7.

[7] T. Zhang, S. G. Patil, N. Jain, S. Shen, M. Zaharia, I. Stoica, and J. E. Gonzalez, "Raft: Adapting language model to domain specific rag," in *First Conference on Language Modeling*, 2024.

[8] B.-Y. Wu, U. Sharma, S. R. D. Kankipati, A. Yadav, B. K. George, S. R. Guntupalli, A. Rovinski, and V. A. Chhabria, "Eda corpus: A large language model dataset for enhanced interaction with openroad," IEEE, New York, NY, June 2024.

[9] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[10] S. Robertson, H. Zaragoza *et al.*, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[11] Elastic N.V., "Elasticsearch," 2024. [Online]. Available: https://www.elastic.co/elasticsearch

[12] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[13] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[14] M. H. Daniel Han and U. team, "Unsloth," 2023. [Online]. Available: http://github.com/unslothai/unsloth

[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[16] IBM, "IBM SLATE-125M English Retriever v2 Model Card," 2024, accessed: 2025-05-22. [Online]. Available: https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/fm-slate-125m-english-rtrvr-v2-model-card.html?context=wx

[17] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[18] W. Yuan, G. Neubig, and P. Liu, "Bartscore: Evaluating generated text as text generation," *Advances in neural information processing systems*, vol. 34, pp. 27 263–27 277, 2021.