在战场物联网中边缘设备上的数据库自然语言交互

Christopher D. Molek*†, Roberto Fronteddu†, K. Brent Venable*†, Niranjan Suri*†‡
* Department of Intelligent Systems & Robotics - The University of West Florida (UWF), Pensacola, FL, USA
† Florida Institute for Human and Machine Cognition (IHMC), Pensacola, FL, USA
{ cmolek, rfronteddu, bvenable, nsuri } @ihmc.org
‡ US Army DEVCOM Army Research Laboratory (ARL), Adelphi, MD, USA
niranjan.suri.civ@mail.mil

***Abstract*—Internet of Things (IoT) 在战场上的扩展，Internet of Battlefield Things (IoBT)，为增强态势感知带来了新的机会。为了提高 IoBT 在关键决策中增强态势感知的潜力，必须将这些设备的数据处理成满足消费者需求的信息对象，并根据需求提供给消费者。为了解决这一挑战，我们提出了一种使用自然语言处理（NLP）查询数据库技术并以自然语言返回响应的工作流。我们的解决方案利用适合边缘设备进行 NLP 的 Large Language Models (LLMs)，以及适合动态连接网络的图形数据库，这些网络广泛应用于 IoBT。我们的架构利用大型语言模型（LLM）将自然语言的问题映射到 Cypher 数据库查询，并总结数据库输出以自然语言反馈给用户。我们在代表美国陆军在新墨西哥州拉斯克鲁塞斯的约纳达靶场的多功能传感区域（MSA）公开数据的数据库上评估了若干中等大小的 LLM，用于这两项任务。我们观察到，与所考虑的所有指标相比，Llama 3.1（80 亿参数）表现优于其他模型。最重要的是，我们注意到，与当前的方法不同，我们的两步方法允许放宽对生成的 Cypher 查询与真实代码的 Exact Match (EM) 要求，从而实现了准确性提高 19.4 %。我们的工作流为在边缘设备上部署 LLM 奠定了基础，以便与包含关键决策信息对象的数据库进行自然语言交互。**

***Index Terms*—Large Language Model, Internet of Things, Internet of Battlefield Things, Graphical Database, Cypher Query Language**

大量的研究越来越关注 IoT ——一个从温度传感器到移动设备的互联设备网络。它们数量的不断增加产生了大量数据集，这些数据集需要强大的存储、处理和分析。从这些数据中得出的洞察提升了对环境的态势感知，并支持改进决策 [1] 。实时数据处理和分析在军事环境中尤为关键，尤其是对于 IoBT 网络，在这些网络中，间歇性或有限的连接性降低了对云计算能力的访问。边缘设备通过在本地处理和分析数据以供消费者使用，解决了这一限制 [2] 。

LLMs 在其他研究领域的一个显著能力是其从庞大数据集中提取有用信息的有效性 [2]–[4] 。LLMs 也被用于根据自然语言问题生成特定的代码输出 [5] 。这一点特别有用，尤其是在考虑从自然语言提示生成数据库查询的任务时。

本研究的目的是确定 LLMs 在设备上从大量由 IoBT 生成的文献中提取数据的能力和局限性。首先，我们使用图数据库探查模拟的 IoBT 结构，图数据库在此背景中优于 SQL 数据库。图数据库实现动态更新，因为节点可以加入或离开网络，并有效管理深度连接的多重关系数据。它们的可扩展性允许查询仅定位相关的图段，而 SQL 数据库由于随着数据量增加或增加关系而导致昂贵的连接操作而产生性能下降。此外，图数据库更有效地处理层次结构数据关系。

在众多可用的图形数据库中，我们选择专注于 Neo4j 的数据库及其查询语言 Cypher [6] 。Cypher Query Language (CQL) 是一种专门为图结构设计的成熟查询语言。有大量研究集中在使用 LLMs 将自然语言映射到 Cypher 代码 [7]–[11] ，事实上，大多数当前的 LLMs 确实对 CQL 有所涉及。

这项工作的核心目标是利用 LLMs 作为人类与数据库之间的接口，用于检索 IoBT 数据。该接口基于两步方法：1）将自然语言问题转换为数据库查询，2）将数据库输出和用户的原始问题生成自然语言响应。我们的方法设想是将实时 IoBT 数据收集到数据库中，可以是在设备上或附近的网络设备上。一旦存储在数据库中，现场人员可以通过我们的系统使用自然语言询问数据。Large Language Model (LLM) 首先将问题翻译为 Cypher 查询以查询数据库。返回的数据库输出与原始用户的问题结合，处理生成自然语言响应返回给用户。这个过程简化了不熟悉数据库查询语言的用户对 IoBT 数据的访问。

在第二步中使用 LLM 允许松弛为 Cypher 查询生成中预测的 Cypher 输出到真实值的 EM 要求。这种松弛在 Cypher 查询从数据库中提取正确的信息加上辅助信息的情况下是可行的。由于正确的「内容」在数据库响应中，第二次 LLM 调用有机会提取适当的信息来回答用户的问题。

我们的方法研究了一些当前最先进的模型，这些模型小到可以在现场设备上使用，以回答有关数据和/或环境的问题。通过这种方式，我们为用户提供了一个自然语言界面，无需依赖云资源和查询语言的知识。

本研究的主要贡献包括：

- 设计、实现和评估一个管道，使其能够在需要设备端计算的环境中对动态数据库进行自然语言查询；
- 评估 LLMs 生成 Cypher 代码以提取数据的能力；
- 研究 LLMs 将数据库对查询的回答重构为自然语言句子的能力；
- 设计一个实验评估框架，以评估 LLMs 与图形数据库的交互。

我们的方法使用较小的模型来生成 Cypher 查询，仅通过提供关于数据库模式的信息和一个查询的示例进行提示。此外，我们评估了使用零样本模式，以便与 IoBT 中资源受限的场景相一致，例如设备电力限制。

Christopher Molek and K. Brent Venable were supported by NSF award NSF-RI-2008011.

I. 相关工作

自然语言查询数据库是研究文献中一个成熟的研究领域。例如，Bukhari 等人 [12] 发表了一篇关于该主题的文献综述，在他们回顾的 47 个框架中，他们确定 70 % 关注于 Structured Query Language (SQL)，另外 30 % 则专注于 “Not only” Structured Query Language (NoSQL) 特定语言，而在这些语言中，只有 10 % 使用 Cypher。由于图数据库的使用日益增多，作者 “敦促” 研究人员专注于开发自然语言到 NoSQL 图数据库（如 Cypher）框架的开发，这是我们在此工作中要解决的一个空白。

随着 LLMs 的兴起，它们在自然语言处理中的使用以生成数据库查询已经成为前沿研究领域 [13]–[16] 。它们用于生成 SQL 查询的研究也在积极展开。相比之下，虽然关于将大型语言模型用于 NoSQL 数据库的研究也在进行，但历史上一直不太广泛。

近年来，研究已经转向利用知识图谱中的 LLMs 进行知识提取 [8] 。许多知识图谱表示存储在像 Neo4j 这样的 NoSQL 数据库中，该数据库采用 Cypher [6] 。Cypher 查询语言具有相当的复杂性，用户学习起来具有挑战性。这推动了进一步研究，使用 LLMs 进行 Cypher 查询生成，特别是针对非专业人士希望从数据库中提取数据的情况。

Hornsteiner 等人创建了一个聊天框架，利用三个独立的 LLMs ，使用户可以用自然语言与 Cypher 数据库进行交互 [7] 。在这个框架中，用户提出一个问题，如果聊天记录中无法回答，则会发送给 “Chat LLM” 以生成一个 Cypher 查询。然后将 Cypher 查询发送回用户，以评估错误，如获批准便执行。虽然此工作有聊天功能，但需要用户对 Cypher 查询质量做出决定，这为非专业用户设立了障碍。在我们的框架中，Cypher 查询的使用完全从用户处抽象化，消除了任何对查询语言的事先熟悉需求。

在类似的研究方向中，Mandilara 等人定义了一个框架，用于系统地评估在自然语言到 Cypher 查询转换中的 LLMs 效率 [8] 。该框架引入了一种度量设计，旨在促进 LLMs 以及模式感知提示工程方法的比较。虽然相关，但他们的重点是仅利用 LLMs 进行从文本到 Cypher 查询转换，而我们使用 LLMs 进行一个额外步骤，即以自然语言汇总查询输出。从这个意义上说，我们提高从图形数据库进行问题回答准确性的方式与他们的方法相互独立。他们集中优化 Cypher 查询生成，而我们提出一个新颖的管道，依次使用两个 LLMs，其中第二个 LLM 的作用是减轻第一个生成的 Cypher 查询的次优性。此外，我们的重点是资源受限的战术环境，与如小型笔记本电脑、平板或配备硬件加速器节点的边缘设备的能力相一致 [17] 。相比之下，他们关注于选择用于知识驱动 AI 的最可靠的 LLMs，通常导致模型资源需求超出 IoBT 限制。

已经设计了使用 LLMs 微调的更复杂的框架来提高 EM 分数。Tran 等人. [11] 开发了一个使用 BERT、GraphSAGE 和 T5 Transformer (CoBGT) 模型组合构建的鲁棒文本到 Cypher 框架。通过这个广泛的框架，作者发现 EM 指标的性能相比 T5 Transformer 提高了 39.04 %，较 GPT2 提高了 35.81 %。该框架还有令人印象深刻的 EM 值为 87.1 % [11] 。作者讨论了训练数据中缺乏高质量（question,Cypher query）对的问题，这一缺点也被其他研究人员识别出，并通过公开可用的 “Text2Cypher” 数据集部分解决了此问题 [8]–[11] 。Zhong 等人，则专注于合成数据集生成以进行有监督的微调。他们的方法在具有 80 亿参数的 Llama 3-instruct 模型上提升了输出精度多达 12 % [10] 。

这些工作只涉及我们系统中 LLMs 的一个角色，即将自然语言翻译成 Cypher 查询。相比之下，我们还评估了 LLMs 在用自然语言总结查询结果的能力。此外，我们采用零样本的方法，避免微调，因为在资源受限的战术环境中，收集训练数据和确保必要的计算资源可能不切实际。

在本节中，我们描述了我们的方法，使用户可以用自然语言查询图形数据库，而无需了解数据库查询语言。

我们提出的流程，如图 1 所示，包含两个主要任务：

- 任务 1：将自然语言用户问题翻译成 Cypher 查询。
- 任务 2：将 Cypher 查询的结果总结为自然语言答案。

我们建议在两个任务中使用一个 LLM，而不要求它们相同。

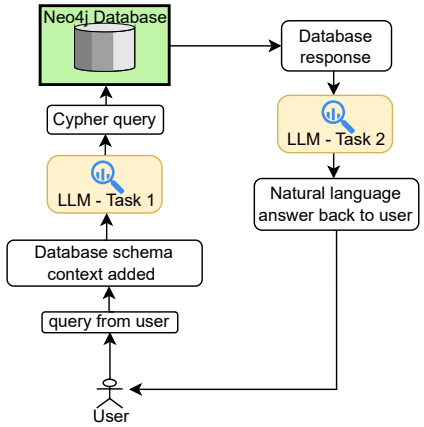

Fig. 1. 数据库自然语言交互工作流程。

我们设想的流程始于用户输入到系统中的自然语言问题。任务 1 中 LLM 的输入由这个问题加上一些额外的上下文组成。该任务的输出是一个 Cypher 查询，当运行时，应从数据库中提取出回答问题所需的数据。

随后，生成一个新提示，其中包含数据库输出、原始问题以及总结请求。该提示被输入到用于任务 2 的 LLM 中，目的是用自然语言向用户提供正确答案。

图 2 展示了任务 1 的一个提示示例。加粗字体表示用户提供的问题示例。它前面有一个序言，提示 LLM 使用其 Cypher 技能并描述步骤以指导其推理生成 Cypher 代码。问题之后是图形数据库架构的描述和 Cypher 代码中的关系。图 2 中提供的示例涉及一个传感器 s 与其所在塔的关系。

更具体来说，根据我们正在进行的示例，任务 1 的理想输出将是一个允许从数据库中提取正确数据的 Cypher 查询，即：

`MATCH (t:Tower Tower: 4) RETURN t.Lat AS Lat, t.Long AS Long` 。

这样的查询可以包含在对 Neo4j 数据库的 API 调用中，并返回：

`[ < Record Lat=32.58088351 Long = -106.753 3307 > ]` 。

鉴于此，任务 2 LLM 的输入提示如图 3 所示。

Task 1 Prompt:

You are an AI agent that is an expert with Cypher query language and relational databases that thinks before answering.
Example:
{
"step1": "To solve this problem, I need to identify the correct nodes and labels...",
"step2": "Based on results from step1, I need to identify the name of the relationship between nodes...",
...
"stepn": "We can get final result by outputing the correct connections between nodes based on the question...",
"Cypher output": Cypher query
}
Your question is:

What is the location of tower 4?

In this database there are nodes with the following labels: "Sensor" and "Tower". The "Sensor" node has the following property keys: "sensorType" and "Tower". The "Tower" node has the following property keys: "Tower", "Lat" and "Long". The property of Tower called "Tower" is an integer all other properties are strings.

Here is an example of the relationship:
(s:Sensor)-[:is_located_on]->(t:Tower {Tower: 8})

Fig. 2. 用于任务 1 的示例提示。

Task 2 Prompt:

Please summarize the following back to the user:
your question

What is the location of tower 4?

with a result of
[<Record Lat = 32.58088351 Long = -106.7533307>]

Fig. 3. 用于任务 2 的示例提示。

对于这种情况，任务 2 的正确输出将是类似以下的句子：
塔 4 的位置在纬度 32.58088351° 和经度-106.7533307°。
我们注意到，此示例说明了在我们的工作流程中使用 Llama3.1:8b 模型获得的实际结果。

II. 实验设计

我们现在概述我们设计的实验评估框架的关键组成部分，以评估在工作流程中使用不同的 LLMs。

A. 物联网数据库

数据库的设立是为了代表公开可用的美国陆军的 Multi-Purpose Sensing Area (MSA) 系统，该系统由相隔三英里的 13 座塔组成。每个塔上连接着多个传感器。这些传感器主要用于收集当地的天气状况，包括：温度、湿度、风速、降水量、气压、土壤状况和网络状况。传感器通过网络连接在一起，创建一个具有数据流功能的实时 IoT 。用户可以捕获数据的实时流以原始形式存储，或者将其处理并存储为消费者准备好的信息对象。在这项工作中进行的当前实验涉及探测代表 MSA 系统的塔和传感器设置的数据库。数据库的范围包括 135 个节点和 121 个关系，以及 11 种属性类型。模式示例可以在图 2 的提示的第二部分中看到。

B. 问题

为了评估我们架构的问答性能，我们生成了几个自然语言问题，如表 I 所示。注意，其中一个问题是一个关于塔 22 的陷阱题，因为塔 22 并不存在。
这些问题被选中用来展示找到答案所需的不同推理深度，并探测数据库模式的不同部分。

TABLE I
关于 MSA 系统塔和传感器设置的七个问题。

Questions
What sensors are attached to tower 0?
What sensors are attached to tower 12?
What sensors are attached to tower 22?
What is the location of tower 4?
Which towers are closest to each other?
How many sensors are on tower 8?
How many towers are there?

如 **??** 节所述，任务 1 的提示除了包括用户的问题，还包括关于 LLM 如何表现的附加背景信息以及数据库模式。该提示如图 2 所示，是旨在优化任务 1 所用模型输出的提示工程努力的结果。任务 1 中的所有模型在给定问题时都收到相同的提示。

C. 度量标准

我们现在提供一个直观和正式的描述，说明我们用来评估工作流程中 LLMs 性能的指标。对于任务 1，我们考虑以下内容：

- 精确匹配（EM）分数：衡量专家基础 Cypher 查询（真实值）与 LLM 产生的 Cypher 查询之间的精确匹配次数，并对预测的查询总数进行标准化。
- 内容评分：由反馈到数据库以获取回答用户问题所需信息的 Cypher 查询数量构成，并且根据预测查询的总数进行标准化。这可以通过解析返回的数据来验证。
- 内容长度：内容长度衡量正确数据库响应的字符串长度：EM 的结果产生最短的内容长度，而仍然包含正确信息的次优 Cypher 查询则产生较长的内容长度。
- 错误信息评分：语法正确但语义错误的 Cypher 查询的数量，经过归一化处理后相对于预测查询总数。例如，这样的查询可能是检索关于塔楼 8 的信息，而用户实际上询问的是塔楼 3，这可能是由于生成示例中提到塔楼 8 而导致的混淆，如图 2 所示。

更正式地说，EM 分数是根据 Tran 等人计算的。对于单个问题，每个由 LLM 生成的查询 $\hat{Y}$ 是与真实查询 Y 的 EM ，赋予一个值，用 $Score_{LF}$ 表示，为 1，否则为 0。
然后从 N 个问题中的得分计算出 EM 分数为：

$$EM = \frac{1}{N}\sum_{n=1}^{N} Score_{LF}(\hat{Y}_n, Y_n) \tag{1}$$

由于 LLMs 的特性，预测的 Cypher 查询不总是 EM 。这些查询可能拥有正确的语法，但仍然产生次优的响应，

其中包含正确的数据以及与问题无关的信息。为了处理这种类型的结果，我们使用内容评分。实际上，额外信息量的变动使得很难通过编程自动地去除它。此外，数据库的输出并不符合用户需求。在这种情况下，任务 2 中 LLM 的角色尤为关键，因为它必须区分相关的信息与无关或附带的数据，并以自然语言回复用户。

这里证明的一个关键点是，可以利用一个 LLMs 团队来提高用户的输出效果，而无需进行专业培训或微调。这对于在 IoBT 中的边缘设备上部署尤为重要，因为由于计算限制，可能无法进行额外的培训。

任务 2 中使用的 LLM 还根据一个附加指标进行评分，即输出评分：

- 输出分数：此分数表示以给定数据库结果作为输入时，任务 2 的正确响应数量。数据库的输出可能有以下四种情况之一：1）它包含正确回答用户问题所需的数据；2）为空列表 "[]"，意味着预测的 Cypher 查询中的语义关系不正确或搜索的数据在数据库中不存在；3）为 " *nan* "，表示返回了一种由不正确的 Cypher 查询预测引起的语法错误；4）有返回响应，但不包含回答问题所需的数据。这发生在预测的查询结构上有效但语义上失败时，导致提取了错误的数据。

整体工作流程根据绝对评分进行评分：

- 绝对分数：此全局指标是系统产生的正确响应数量，并在总输入数量上进行归一化。

III. 结果与讨论

我们在此工作流程中评估多个 LLMs ，以了解它们在生成 Cypher 代码（任务 1）和将数据库查询的答案重新表述为自然语言（任务 2）方面的能力和限制。

鉴于我们专注于边缘设备，我们仅考虑中等大小的 LLMs ，如 [18] 中定义的，即参数在 1b 到 10b 之间。我们在该尺寸范围的两端附近测试模型，以确保与在内存和功率方面具有不同能力的边缘设备的兼容性。

用于这些实验的软件包括：Neo4j 数据库桌面版本 1.6.1，Ollama LLM 平台版本 0.5.4，以及 Python 版本 3.12。与 Ollama 一起使用的模型包括：Gemma2（2b 参数）、Llama3.2（3b 参数）、Llama3.1（8b 参数）和 Deepseek-coder（6.7b 参数）。所有实验均在戴尔 XPS 15 笔记本电脑上运行，该电脑配备第 13 代 Intel i9 2.60 GHz 处理器，32 Gb 内存和带有 8 Gb 内存的 Nvidia GeForce TRX 4060 笔记本 GPU。

给定表格 I 中显示的七个问题，工作流生成的输出被评估其 EM、内容、输出和绝对分数，结果显示在表格 II 中。在本研究中，我们在任务 1 和任务 2 中使用了相同的 LLM。鉴于任务的性质，所有模型的温度设置均为零。

在任务 1 中使用的所有 LLM 都取得了高内容评分（超过 71 %），这表明它们对用户问题（即 Cypher 查询）的回答大多数时间能够产生正确的数据库结果。然而，这些查询通常并不完全匹配（见表 II ，第一列）。在任务 2 中，LLM 的工作是从数据库结果中提取内容，并向用户返回正确的响应。对于任务 2 中使用的所有模型，输出得分都超过 71 %，并且对所有模型来说都高于内容评分，除了 Deepseek-coder:6.7b。这表明以自然语言总结数据输出的任务比将用户问题翻译成 Cypher 查询更容易。

TABLE II
使用 7 个问题评估的四个 LLM 的性能。

Model	EM score Task 1	Content Score Task 1	Output Score Task 2	Absolute Score
Gemma2:2b	14.3 %	71.4 %	85.7 %	57.1 %
Llama3.2:3b	28.6 %	71.4 %	100 %	71.4 %
Llama3.1:8b	42.9 %	85.7 %	100 %	85.7 %
Deepseek-coder:6.7b	28.6 %	85.7 %	71.4 %	57.1 %

一般来说，每个模型对相同输入生成的 Cypher 查询各不相同，导致数据库返回的输出各异。我们在表 III 中阐述了这一点，问题是 "塔 4 的位置在哪里？"。虽然所有答案都包含了正确的信息，但只有 Llama3.1:8b 有一个 EM ，其内容长度最小，仅为 44 个字符，如图 3 所示。而 Llama3.2:3b 模型生成了最长的内容长度，为 2171 个字符。在表 III 中报告的可变性，以及表 II 和 IV 中的内容得分，说明了由预测的 Cypher 查询生成的输出的可变性，这些输出常常包括多余的数据，同时展示了大型语言模型从嘈杂或无关输入中提取有意义信息的能力。

TABLE III
返回的数据库查询内容长度变化。

Model	User Question	Content Length of DB output
Gemma2:2b	What is the location of tower 4?	167
Llama3.2:3b	What is the location of tower 4?	2171
Llama3.1:8b	What is the location of tower 4?	44
Deepseek-coder:6.7b	What is the location of tower 4?	624

我们使用 Llama3.1:8b 模型将每个原始问题重新表述 10 次，并手动检查其正确性，以评估系统对不同提问方式的鲁棒性。77 个问题的 EM、内容、输出和绝对分数如表 IV 所示。如预期，大多数分数有所下降；然而，主要趋势仍然存在：内容分数比 EM 分数高，输出分数超过内容分数。这支持了我们的假设，即完全匹配（EM）可能是一个过于严格的指标，因为正确的信息往往嵌入在多余内容中，而大型语言模型通常能够有效地提取并以自然语言呈现这些信息。

TABLE IV
对四个大型语言模型使用 77 个问题进行评估的表现。

Model	EM score Task 1	Content Score Task 1	Output Score Task 2	Absolute Score
Gemma2:2b	15.6 %	51.9 %	76.6 %	33.8 %
Llama3.2:3b	15.6 %	57.1 %	85.7 %	44.2 %
Llama3.1:8b	37.7 %	61.0 %	96.1 %	57.1 %
Deepseek-coder:6.7b	20.78 %	42.9 %	74.0 %	31.2 %

我们可以通过将仅对应于 EM 案例的绝对分数（仅 EM）与之前定义的绝对分数（考虑任务 1 的内容分数）进行比较，进一步评估我们的工作流程，如表 V 所示。我们观察到绝对分数显著提高，其中 Llama3.1:8b 得分最高，达到 57.1 %，其他模型的绝对分数也翻了一倍多。

对于所有模型，当任务 1 的结果中存在语法、语义或错误信息时，或任务 2 未能提取可用内容时，就会出现不正确的答案。错误信息尤其微妙，因为它将错误信息提供给任务 2，并传播给用户。该错误被量化为错误信息分数，测量对用户问题给出错误答案的频率（见表 VI ）。值得注意的是，Llama3.1:8b 的错误信息率非常低，仅为 5.2 %。

TABLE V
仅 EM 查询的绝对得分与 77 个问题的绝对得分。

Model	Absolute Score (EM Only)	Absolute Score
Gemma2:2b	14.3 %	33.8 %
Llama3.2:3b	14.3 %	44.2 %
Llama3.1:8b	37.7 %	57.1 %
Deepseek-coder:6.7b	14.3 %	31.2 %

TABLE VI
77 个问题的错误信息评分。

Model	Misinformation Score
Gemma2:2b	35.1 %
Llama3.2:3b	7.8 %
Llama3.1:8b	5.2 %
Deepseek-coder:6.7b	26.0 %

人工分析表明，Cypher 查询中的一致性错误源自错误的塔号（经常使用工程化提示的示例而非用户输入），而非对问题本身的误解。这个洞察可以指导未来对任务 1 提示的改进。

总体而言，我们注意到，在所有的评估和指标中，Llama3.1 是该工作流程中的顶级模型，获得最高的绝对分数（57.1 % ）和最低的错误信息分数（5.2 % ），这使其成为一个有希望的候选者，值得进一步测试以实现其在边缘设备上的部署。

IV. 未来工作

我们未来的研究计划包括：在动态条件下使用来自 MSA 系统的实时数据测试我们的工作流程；调查具有推理能力的其他模型（例如，Deepseek-r1 和 Gemma 3）；并研究上下文学习以及将此工作流程与轻量模型微调相结合，以提高整体性能。

References

[1] D. Tsiolis and A. Leivadeas, "A Smart City Assisted Internet of Military Defense Things Architecture," *IEEE Internet of Things Magazine*, vol. 8, no. 2, pp. 18–24, Mar. 2025.

[2] S. Bhardwaj, P. Singh, and M. K. Pandit, "A Survey on the Integration and Optimization of Large Language Models in Edge Computing Environments," in *2024 16th International Conference on Computer and Automation Engineering (ICCAE)*, Mar. 2024, pp. 168–172.

[3] X. Luo, A. Rechardt, G. Sun, K. K. Nejad, F. Yáñez, B. Yilmaz, K. Lee, A. O. Cohen, V. Borghesani, A. Pashkov, D. Marinazzo, J. Nicholas, A. Salatiello, I. Sucholutsky, P. Minervini, S. Razavi, R. Rocca, E. Yusifov, T. Okalova, N. Gu, M. Ferianc, M. Khona, K. R. Patil, P.-S. Lee, R. Mata, N. E. Myers, J. K. Bizley, S. Musslick, I. P. Bilgin, G. Niso, J. M. Ales, M. Gaebler, N. A. Ratan Murty, L. Loued-Khenissi, A. Behler, C. M. Hall, J. Dafflon, S. D. Bao, and B. C. Love, "Large language models surpass human experts in predicting neuroscience results," *Nature Human Behaviour*, vol. 9, no. 2, pp. 305–315, Feb. 2025.

[4] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, H. Chen, Z. Liu, Z. Dou, and J.-R. Wen, "Large Language Models for Information Retrieval: A Survey," Sep. 2024, *arXiv preprint arXiv:2308.07107*.

[5] J. Wang and Y. Chen, "A Review on Code Generation with LLMs: Application and Evaluation," in *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, Nov. 2023, pp. 284–289.

[6] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, and A. Taylor, "Cypher: An Evolving Query Language for Property Graphs," in *Proceedings of the 2018 International Conference on Management of Data*, ser. SIGMOD '18. New York, NY, USA: Association for Computing Machinery, May 2018, pp. 1433–1445.

[7] M. Hornsteiner, M. Kreussel, C. Steindl, F. Ebner, P. Empl, and S. Schönig, "Real-Time Text-to-Cypher Query Generation with Large Language Models for Graph Databases," *Future Internet*, vol. 16, no. 12, p. 438, Nov. 2024.

[8] I. Mandilara, C. Maria Androna, E. Fotopoulou, A. Zafeiropoulos, and S. Papavassiliou, "Decoding the Mystery: How Can LLMs Turn Text Into Cypher in Complex Knowledge Graphs?" *IEEE Access*, vol. 13, pp. 80 981–81 001, 2025.

[9] M. G. Ozsoy, L. Messallem, J. Besga, and G. Minneci, "Text2Cypher: Bridging Natural Language and Graph Databases," Dec. 2024, *arXiv preprint arXiv:2412.10064*.

[10] Z. Zhong, L. Zhong, Z. Sun, Q. Jin, Z. Qin, and X. Zhang, "SyntheT2C: Generating Synthetic Data for Fine-Tuning Large Language Models on the Text2Cypher Task," Jan. 2025, *arXiv preprint arXiv:2406.10710*.

[11] Q.-B.-H. Tran, A. A. Waheed, and S.-T. Chung, "Robust Text-to-Cypher Using Combination of BERT, GraphSAGE, and Transformer (CoBGT) Model," *Applied Sciences*, vol. 14, no. 17, p. 7881, Jan. 2024.

[12] S. A. C. Bukhari, H. S. Dar, M. I. Lali, F. Keshtkar, K. M. Malik, and S. Kadry, "Frameworks for Querying Databases Using Natural Language: A Literature Review –NLP-to-DB Querying Frameworks," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 17, no. 2, pp. 21–38, 2021, publisher: IGI Global.

[13] V. Lamas, M. R. Luaces, and D. Garcia-Gonzalez, "DSL-Xpert: LLM-driven Generic DSL Code Generation," in *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*. Linz Austria: ACM, Sep. 2024, pp. 16–20.

[14] M. M. Mosthaf and A. Wasowski, "From a Natural to a Formal Language with DSL Assistant," in *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*. Linz Austria: ACM, Sep. 2024, pp. 541–549.

[15] B. Bogin, S. Gupta, P. Clark, and A. Sabharwal, "Leveraging Code to Improve In-context Learning for Semantic Parsing," Mar. 2024, *arXiv preprint arXiv:2311.09519*.

[16] S. Pimparkhede, M. Kammakomati, S. Tamilselvam, P. Kumar, A. P. Kumar, and P. Bhattacharyya, "DocCGen: Document-based Controlled Code Generation," Jul. 2024, *arXiv preprint arXiv:2406.11925*.

[17] N. Ng, A. Souza, S. Diggavi, N. Suri, T. Abdelzaher, D. Towsley, and P. Shenoy, "Collaborative Inference in Resource-Constrained Edge Networks: Challenges and Opportunities," in *MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM)*, Oct. 2024, pp. 1–6.

[18] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large Language Models: A Survey," Feb. 2024, *arXiv preprint arXiv:2402.06196*.