

# Scaling 3D Reasoning with LMMs to Large Robot Mission Environments Using Datagraphs

W.J. Meijer\* A.C. Kemmeren\* E.H.J. Riemens<sup>†</sup> J.E. Fransman\* M. van Bekkum\* G.J. Burghouts\* J.D. van Mil\*

**Abstract**—This paper addresses the challenge of scaling Large Multimodal Models (LMMs) to expansive 3D environments. Solving this open problem is especially relevant for robot deployment in many first-responder scenarios, such as search-and-rescue missions that cover vast spaces. The use of LMMs in these settings is currently hampered by the strict context windows that limit the LMM’s input size. We therefore introduce a novel approach that utilizes a datagraph structure, which allows the LMM to iteratively query smaller sections of a large environment. Using the datagraph in conjunction with graph traversal algorithms, we can prioritize the most relevant locations to the query, thereby improving the scalability of 3D scene language tasks. We illustrate the datagraph using 3D scenes, but these can be easily substituted by other dense modalities that represent the environment, such as pointclouds or Gaussian splats. We demonstrate the potential to use the datagraph for two 3D scene language task use cases, in a search-and-rescue mission example.

## I. INTRODUCTION

This paper addresses the challenge of scaling the use of Large Multi-modal Models (LMMs) to large 3D environments. Current models can perform a wide range of 3D scene language tasks such as spatial and semantic reasoning (Figure 2, Hong et al. [1]), but the impressive capabilities of these models do not yet transfer to large environments that are generally encountered in search-and-rescue settings.

In these large environments, current 3D LMMs seem to face a similar limitation as many Vision Language Models (VLMs), namely the *resolution curse problem*<sup>1</sup>. The VLMs cannot properly encode the information in high-resolution input images due to the fixed and limited input size of the model [2]. Similarly, the context window size in transformer-based models limits the maximum number of tokens that can be used [3, 4]. This limits current 3D LMMs to be able to process only small areas. For example, current models have not yet been used on areas larger than 10 rooms [1, 5].

Recent developments tackle this issue by enabling transformer-based models to process contexts of infinite length flexibly [6]. However, it will take substantial effort to develop the models that are capable of performing all 3D scene tasks in Figure 2 in large 3D environments. These developments are slowed by a lack of large 3D scene datasets [7] and the vast hardware requirements for training new LMMs [6]. In the meantime, the approach in this paper will enable us to use *existing* 3D LMMs in large environments instead.

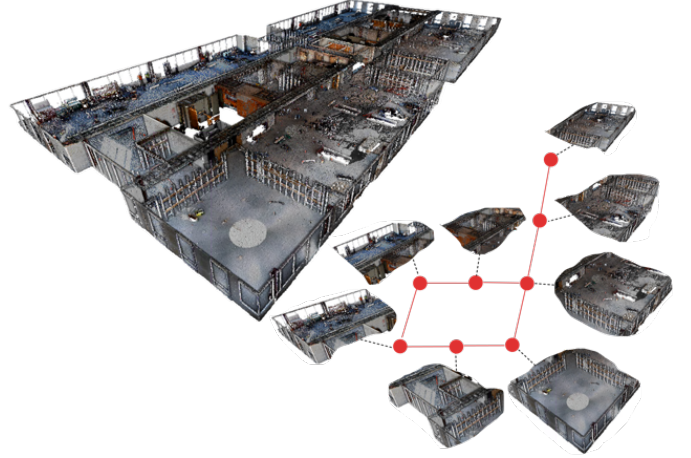


Fig. 1. On the left is a continuous multi-room 3D scene. On the right, a graph (red) is extended with 3D scenes at each node to form a datagraph. Instead of processing the whole multi-room 3D scene at once, existing LMMs can use the smaller scenes in the datagraph to iteratively cover large areas.

Our approach is inspired by how the resolution curse is often solved for 2D tasks, using *tiling*. In tiling, the input image is subsampled into several tiles of lower resolution. In more advanced works, strategies to dynamically choose these tiles are proposed, by internal dialogue with a Large Language Model and visual memory [8] or by exploiting the attention of the transformer [9]. Similarly, we divide large 3D environments into smaller areas by exploiting spatial structure and then choose the subset of areas that is potentially relevant to the task (Figure 1). This, for example, enables us to only consider the part of the environment that is near and reachable to a robot.

In this paper, we consider a scene representation that retains full 3D information, because we believe that having access to detailed information is crucial to search-and-rescue missions. For example, there is an important distinction between a person lying on the floor of the hallway and a person lying in bed. This is often not captured when using 3D scene graphs as representation, as context information is lost during the compression steps in the mapping procedure [10, 11]. Subsequent reasoning tasks then cannot access important contextual details that were initially captured by the sensors [12].

We propose to leverage a spatial graph structure to subdivide vast 3D scenes into smaller areas (Figure 1). With existing graph traversal algorithms, smaller chunks of 3D information can be fed iteratively into the LMMs. We demonstrate the potential of this approach using two algorithms. Algorithm 1

\*TNO, Autonomous Systems & Decision Support, Netherlands

<sup>†</sup>Delft University of Technology, Signal Processing Systems, Netherlands

<sup>1</sup><https://huggingface.co/blog/visheratin/vlm-resolution-curse>

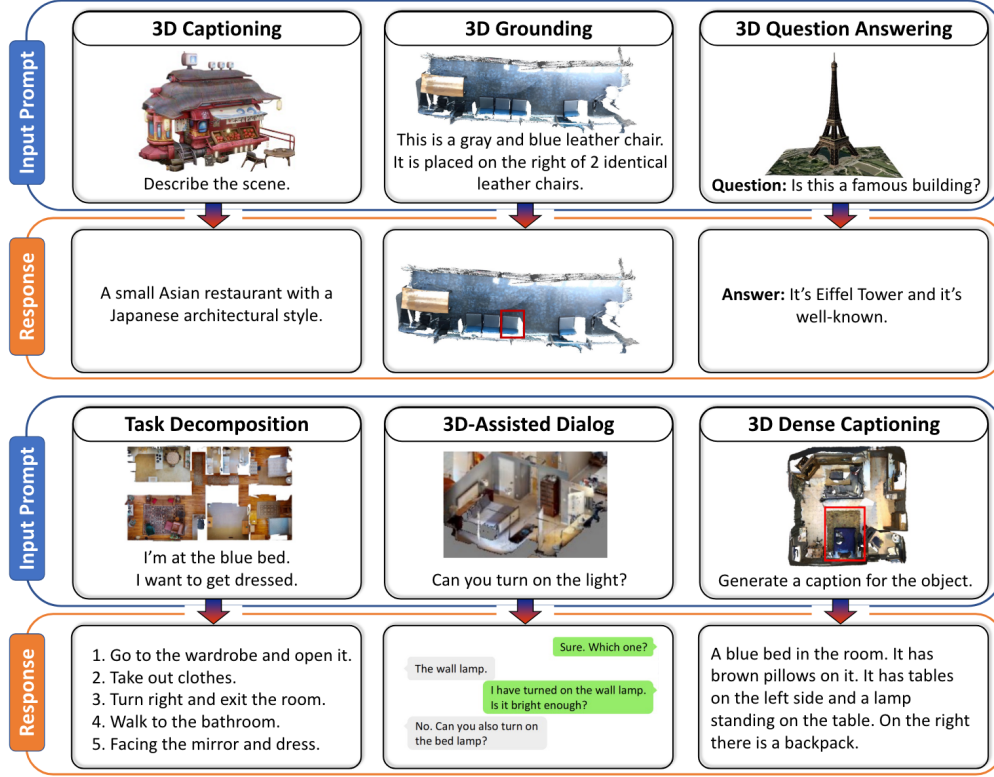


Fig. 2. There is a variety of 3D tasks we would like to perform with 3D-LMMs as illustrated in the work of Hong et al. [1]. We investigate how to perform such tasks over expansive environments with existing LMMs, which are limited in their context size.

starts querying locations closer to the robot, which is useful, for example, when searching for an object nearby. Alternatively, Algorithm 2 queries along a path, for example enabling an LMM to check for situations that make navigation along that path hazardous. These approaches bypass the context limitations of current LMMs, while allowing them to scale to vast environments. Moreover, the approach can be easily extended to store other types of information about the environment. The 3D scenes can be substituted with other modalities, e.g. when deploying LMMs that work with (*colored*) *pointclouds* [13, 14], *neural radiance fields* [15, 16], or *Gaussian splats* [17].

## II. APPROACH

This work considers robot deployments for time-critical first-responder scenarios, such as search-and-rescue missions. First responders operate in expansive and unknown environments, where it is important to quickly build situational awareness (SA) [18], and reason about complex questions such as the safety of an environment. For this, we use currently existing LMMs and combine them with a datagraph to bypass the LMM’s limited context size. The following sections provide more details on the datagraph and how existing graph traversal algorithms then allow for iterative prompting of the LMM.

### A. The datagraph

The datagraph is formalized as  $G = (V, E)$  with node set  $V$  and edge set  $E$ . Each node  $v \in V$  has a pose  $x$  and a

data snapshot  $s$ , which makes the tuple  $v = (x_v, s_v)$ . An edge  $e_{ij} = (v_i, v_j)$  connects two nodes for neighboring areas. One could consider additional requirements that an edge is only added if there is a traversable route between the two areas [19]. This would make  $G$  compatible with navigation or exploration graphs commonly used in autonomy stacks [20, 19, 21]. The  $s$  in the node can be data of any modality that can be used to query an LMM, including 3D scenes, Gaussian splats, or images.

### B. Graph traversal for iterative LMM prompting

Depending on the task that the LMM will execute, different parts of the environment could be interesting to prompt. The challenge for graph traversal thus is to prompt areas with highest priority first. We exemplify this with two graph traversal algorithms. The first algorithm (Algorithm 1) prioritizes the areas that are closest to the robot. The second algorithm (Algorithm 2) retrieves information along a path through the environment to enable, for example, safer navigation.

1) *3D scene language tasks prioritized by robot proximity:* Various situations may require a robot to first recall areas closest to its current situation and execute 3D scene tasks in these areas. A clear example in search-and-rescue missions is when searching for objects in the environment. A robot may need to go through a locked door and start to search in its

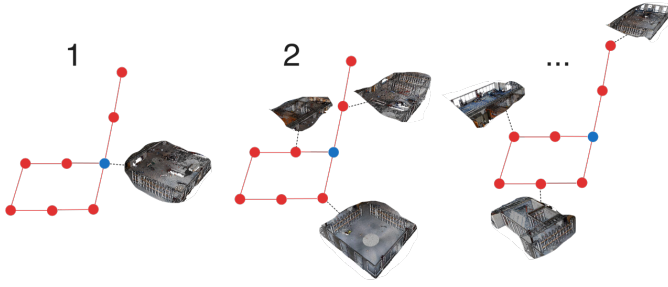


Fig. 3. Illustration of the iterative spatially grounded LMM querying algorithm. Search is started at the node the agent is at (blue), then it gradually expands outward to retrieve perception scenes that can be queried by the LMM.

memory (the datagraph) if it has seen tools that may help it to open or break this door. Finding the closest satisfactory tool then helps fulfilling the mission in the least amount of time. Another example of object search is when new information about risks in the environment becomes available during the mission, such as indicators of explosive or toxic materials. It is important for the agent to be aware of the locations of such materials, especially if the hazard is close.

Algorithm 1 starts with retrieving the nodes closest to the agent, in a breadth-first manner. It begins by initializing an empty set  $V_{visited}$  to track the visited nodes and a set  $V_{search}$  containing the starting node  $v_{agent}$ . For each node  $v$  in  $V_{search}$ , it retrieves the scene associated with  $v$  from the datagraph  $G$  and queries the LMM with this scene and the query  $q$ . Then, it marks  $v$  as visited by adding it to  $V_{visited}$  and updates  $V_{search}$  with the neighbors of  $v$  that have not yet been visited. This process repeats until the entire graph is traversed. The steps are summarized in Algorithm 1.

For specific tasks, one may consider altering this algorithm to stop when a satisfactory response is found. For example, there is little reason to continue searching for a tool to open a door after a suitable object is found. However, in the case of searching for hazards, one may naturally want to continue these queries even if the first hazardous area is found.

---

**Algorithm 1:** Iteratively query an LMM by expanding the datagraph spatially.

---

**Data:** datagraph  $G$ , query  $q$ , agent location  $v_{agent}$

**Result:** LLM responses  $R$  for each node  $v \in V$

$V_{visited} \leftarrow \{\};$

$V_{search} \leftarrow \{v_{agent}\};$

**for**  $v$  **in**  $V_{search}$  **do**

$scene = G[v];$

$response = LMM(scene, q);$

$R.append(response);$

$V_{visited} \leftarrow V_{visited} \cup v;$

$V_{search} \leftarrow neighbors(v) \text{ not in } V_{visited};$

**end**

**return**  $R;$

---



---

**Algorithm 2:** Query an LMM along a path.

---

**Data:** datagraph  $G$ , query  $q$ , path  $P$

**Result:** LMM responses  $R$  for each node in path  $P$

**for**  $v$  **in**  $P$  **do**

$scene = G[v];$

$response = LMM(scene, q);$

$R.append(response);$

**end**

**return**  $R;$

---

2) *3D scene language tasks along a navigation path:*

Consider a search-and-rescue mission, where a robot guides a victim to safety. Assessing the safety of traversing routes through the environment is vital. Therefore, we may want to query whether there are any hazards along the proposed routes.

Given a path in the datagraph, we can iteratively query the LMM about the safety of an area using the data in each of the nodes, as illustrated in Figure 4. The algorithm iterates through each node  $v$ , retrieves the scene associated with  $v$  from  $G$ , and queries the LMM with this scene context and query  $q$ . Responses are added to the list  $R$ . The algorithm is summarized in Algorithm 2.

Because our method only considers local scenes, it will require additional logic to answer certain global questions. For example, when querying how many times a certain object is present in all the scenes of the environment, we would need to aggregate the individual answers per scene (local node) over all scenes of the environment (all relevant nodes).

### III. DISCUSSION AND CONCLUSION

In this paper, we have introduced a novel datagraph structure that significantly expands the spatial context over which Large

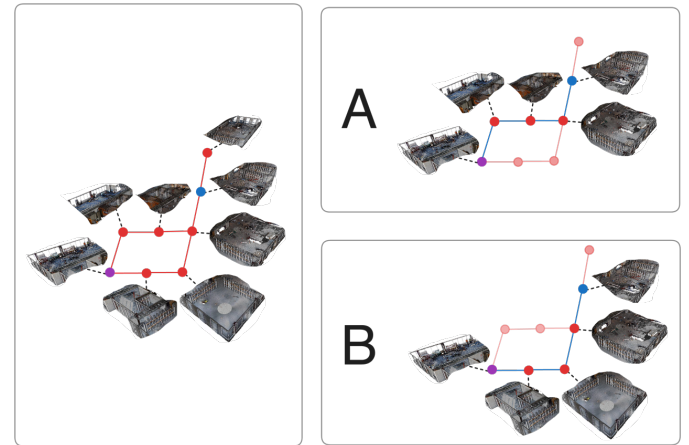


Fig. 4. Illustration of 3D scene language tasks along a navigation path. The purple node is the navigation goal and the blue node is where the agent is located. Two routes A and B are possible, highlighted with blue edges in the two figures on the right.

Multimodal Models (LMMs) can reason. The method was illustrated using 3D scenes, but in principle it is agnostic to the data type stored in the graph. The graph allows us to prioritize data from locations that are most relevant to the query, such as those closest to the agent. This prioritization improves the scalability of 3D scene language tasks, enabling their use in expansive robotic mission environments.

Looking ahead, there are several challenges and opportunities for future work. One such challenge arises at the boundaries between scenes. Relevant information might end up appearing in both scenes, for example, leading to duplicate detections, or scattered over distinct scenes, negatively affecting performance. Moreover, we see a need for experimental results to support our proposed approach. For instance, stitching the rooms of the SceneVerse dataset together could form large environments that would provide a robust testbed for our approach. Additionally, investigating methods of caching identical queries along a path for Algorithm 2 could further enhance the efficiency and effectiveness of our approach.

In conclusion, our work presents a relevant step forward in the application of LMMs in large environments. By leveraging the datagraph structure and prioritizing relevant locations, we have demonstrated a scalable solution for 3D scene language tasks in large robotic mission environments. As we continue to refine our approach and address the challenges identified, we are optimistic about the potential impact of our work on the future of robotic missions.

#### REFERENCES

- [1] Yining Hong et al. *3D-LLM: Injecting the 3D World into Large Language Models*. July 24, 2023. DOI: 10.48550/arXiv.2307.12981. arXiv: 2307.12981[cs].
- [2] Kai Han et al. “A Survey on Vision Transformer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (Jan. 2023). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 87–110. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2022.3152247.
- [3] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context*. June 2, 2019. DOI: 10.48550/arXiv.1901.02860. arXiv: 1901.02860[cs,stat].
- [4] Clayton Sanford, Daniel Hsu, and Matus Telgarsky. “Representational Strengths and Limitations of Transformers”. In: ().
- [5] William Chen, Siyi Hu, Rajat Talak, and Luca Carlone. *Leveraging Large (Visual) Language Models for Robot 3D Scene Understanding*. Nov. 8, 2023. DOI: 10.48550/arXiv.2209.05629. arXiv: 2209.05629[cs].
- [6] Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. *Leave No Context Behind: Efficient Infinite Context Transformers with Infinit-attention*. Apr. 10, 2024. DOI: 10.48550/arXiv.2404.07143. arXiv: 2404.07143[cs].
- [7] Baoxiong Jia et al. *SceneVerse: Scaling 3D Vision-Language Learning for Grounded Scene Understanding*. Mar. 6, 2024. DOI: 10.48550/arXiv.2401.09340. arXiv: 2401.09340[cs].
- [8] Penghao Wu and Saining Xie. *V\*: Guided Visual Search as a Core Mechanism in Multimodal LLMs*. Dec. 26, 2023. DOI: 10.48550/arXiv.2312.14135. arXiv: 2312.14135[cs].
- [9] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. *Towards Perceiving Small Visual Details in Zero-shot Visual Question Answering with Multimodal LLMs*. Feb. 12, 2024. DOI: 10.48550/arXiv.2310.16033. arXiv: 2310.16033[cs].
- [10] Iro Armeni et al. “3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South): IEEE, Oct. 2019, pp. 5663–5672. ISBN: 978-1-72814-803-8. DOI: 10.1109/ICCV.2019.00576.
- [11] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. *3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans*. June 16, 2020. DOI: 10.48550/arXiv.2002.06289. arXiv: 2002.06289[cs].
- [12] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. *SayPlan: Grounding Large Language Models using 3D Scene Graphs for Scalable Robot Task Planning*. Sept. 27, 2023. DOI: 10.48550/arXiv.2307.06135. arXiv: 2307.06135[cs].
- [13] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. *PointLLM: Empowering Large Language Models to Understand Point Clouds*. Dec. 1, 2023. arXiv: 2308.16911[cs].
- [14] Zhangyang Qi et al. *GPT4Point: A Unified Framework for Point-Language Understanding and Generation*. Dec. 5, 2023. DOI: 10.48550/arXiv.2312.02980. arXiv: 2312.02980[cs].
- [15] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. *LERF: Language Embedded Radiance Fields*. Mar. 16, 2023. DOI: 10.48550/arXiv.2303.09553. arXiv: 2303.09553[cs].
- [16] Leonard Bruns, Jun Zhang, and Patric Jensfelt. *Neural Graph Mapping for Dense SLAM with Efficient Loop Closure*. May 6, 2024. DOI: 10.48550/arXiv.2405.03633. arXiv: 2405.03633[cs].
- [17] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. *LangSplat: 3D Language Gaussian Splatting*. Mar. 31, 2024. arXiv: 2312.16084[cs].
- [18] Thomas F Sanquist, Brooke R Brisbois, and Matthew P Baucum. “Attention and Situational Awareness in First Responder Operations”. In: ().
- [19] W. J. Meijer, A. C. Kemmeren, J. M. van Bruggen, T. Haije, J. E. Fransman, and J. D. van Mil. *Situational Graphs for Robotic First Responders: an application to*

*dismantling drug labs*. Apr. 26, 2024. DOI: 10.48550/arXiv.2404.17395. arXiv: 2404.17395[cs].

- [20] Nicolas Hudson et al. “Heterogeneous Ground and Air Platforms, Homogeneous Sensing: Team CSIRO Data61’s Approach to the DARPA Subterranean Challenge”. In: *Field Robotics 2.1* (Mar. 10, 2022), pp. 595–636. ISSN: 27713989. DOI: 10.55417/fr.2022021. arXiv: 2104.09053[cs].
- [21] Marco Tranzatto et al. *CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge*. Jan. 18, 2022. arXiv: 2201.07067[cs].