Lang2Graph: Towards Leveraging Human Language for Indoor Topology Inference Using LLMs

Moamin Ibrahim, Yaqoob Ansari, Khaled A. Harras, Eduardo Feo-Flushing
Carnegie Mellon University
{meibrahi, yansari, kharras, efeoflus}@andrew.cmu.edu

***Abstract*—Edge-deployed systems such as autonomous robots, AR/XR devices, and emergency-response handhelds require accurate indoor topological representations, yet existing sensor-based and expert-curated mapping methods are impractical for crowd-sourced, resource-constrained deployment. Additionally, current large language model (LLM) approaches to indoor topology inference lack a systematic framework for evaluating the factors that govern performance. We present Lang2Graph, an experimental framework for indoor topological graph inference from natural-language navigational instructions that isolates four governing factors: instruction structure, metadata clarity, prompting strategy, and model size and reasoning capability. We propose the Independent Prompt Executor (IPE), a prompting strategy that decomposes graph construction into independent per-instruction reasoning steps, preventing error propagation. To support factor-level evaluation, we develop a fully synthetic dataset and an augmented benchmark (R2R-AUG) covering idealized and real-world conditions. Our evaluation across multiple model families show that structured instructions, clear metadata, and IPE improve precision, recall, and F1 by 44%, 45%, and 47%, respectively. Reasoning-aligned open-source models of moderate scale (14B parameters) outperform larger proprietary models on the most challenging instruction categories, indicating that reliable indoor topology inference is achievable without cloud-scale dependencies and establishes a viable path toward on-device edge deployment.**

***Index Terms*—indoor mapping, topological inference, large language models, edge computing, spatial computing, navigational instructions, prompting strategies**

I. Introduction

Accurate indoor topological representations are a prerequisite for a growing class of systems that operate at the network edge: autonomous robots performing real-time path planning [1]–[3], emergency responders relying on mobile devices for situational awareness [4]–[7], augmented and extended reality (AR/XR) headsets anchoring virtual content to physical spaces [8]–[10], and smart building agents coordinating facility-level services [11]–[13]. These systems share a common constraint: they must reason about spatial structure under real-time latency and limited on-device resources, making round-trip dependence on cloud infrastructure impractical for many deployment scenarios [14]–[18]. Existing approaches to indoor map construction rely on specialized scanning equipment or expert-curated data models—Computer-Aided Design (CAD) drawings, Building Information Models (BIMs), or point clouds [19]—that yield rich geometric detail but demand domain expertise, physical site access, and significant cost [20]–[24]. Standardized indoor spatial formats such as CityGML [25], IndoorGML [25], and Apple's Indoor Mapping Data Format (IMDF) [26] offer comprehensive schema designs, yet their complexity limits accessibility for crowd-sourced, at-scale deployment in edge scenarios [27]. These limitations motivate the exploration of natural-language-based mapping, which offers a scalable, low-cost alternative that can leverage crowd-sourced descriptions without specialized hardware or expert intervention.

Motivated by these challenges, recent work on semantically rich indoor map creation pivoted towards leveraging Large Language Models (LLMs) *(Section II)*. The idea is to investigate ways in which natural-language navigational instructions can be converted into indoor topological graphs. This approach would provide a more natural, accessible, and easily crowd-sourced solution to such a critical problem. Recent solutions [28], [29] demonstrate that LLMs can parse descriptions like *"walk past the kitchen into the living room"* to generate topological spatial representations. Despite promising initial results, these preliminary natural-language indoor topological inference methods exhibit fundamental limitations. Ambiguous instructions, inconsistent region labeling, and monolithic prompting strategies introduce variability, reduce reproducibility, and obscure error sources. It also remains unclear whether reported performance gains stem from LLM-usage design choices or from the underlying scale and reasoning capabilities of the language models themselves.

In this paper, we introduce *Lang2Graph*, a structured indoor graph generation framework that enables the use of human language to construct indoor topologies using LLMs *(Section III)*. We argue that leveraging LLMs for indoor mapping requires formalization, explicitly defining how instructions, metadata, and prompting strategies interact. *Lang2Graph* isolates and studies the foundational factors that influence the performance of natural-language indoor topological inference. The first factor, ***instruction structure***, categorizes navigational instructions along dimensions such as direct vs. indirect and explicit vs. implicit, enabling controlled evaluation of linguistic ambiguity and reference difficulty. The second factor, ***metadata clarity***, requires uniquely identified regions to ensure reliable reference resolution and consistent edge extraction in complex environments. The third factor, ***prompting strategy***, argues for utilizing our proposed Independent Prompt Executor (IPE) that decomposes graph construction into smaller reasoning steps, improving interpretability and graph accuracy. The fourth factor, ***model size and reasoning capabilities***, evaluates how model scale and reasoning alignment affect spatial understand-

ing; this is critical in providing insight into whether natural-language indoor topological inference performance depends more on parameter count or explicit reasoning fine-tuning.

We evaluate *Lang2Graph* through extensive experimentation on three datasets *(Sections IV and V)*. The first is a fully synthetic dataset that we develop with controlled linguistic structure. The second is a real-world Room-to-Room (R2R) dataset [30] with unconstrained human instructions. The third dataset is an augmented version of R2R that we develop which introduces structured metadata and controlled variations. These datasets together cover both idealized conditions and realistic "noisy" scenarios in which instructions contain ambiguity, inconsistent phrasing, or incomplete spatial references. The experiments assess how the four formalized factors: instruction structure, metadata clarity, prompting strategy, and model size and reasoning capabilities, each shape performance, scalability, and robustness in natural-language indoor topological inference. Our results show that structured and unambiguous instructions, clear building metadata labeling, and independent prompting, through IPE, significantly improve graph inference precision, recall, and F1 scores by 44%, 45%, and 47% respectively. Moreover, the findings reveal that reasoning alignment outweighs raw model scale. This paper makes the following contributions:

- We provide a methodological framework that formalizes natural-language indoor topological inference along four factors: instruction structure, metadata clarity, prompting strategy, and model size and reasoning capability.
- We introduce a principled taxonomy of navigational instructions based on path structure (direct vs. indirect) and reference style (explicit vs. implicit), enabling controlled evaluation of linguistic ambiguity.
- We propose the *Independent Prompt Executor* (IPE), a prompt-decomposition workflow that processes each navigational instruction independently for improved interpretability, error localization, and scalability.
- We design a fully synthetic dataset and a structured augmentation of the R2R benchmark that support reproducible, factor-level analysis under both idealized and real-world conditions.
- We conduct extensive evaluation across different LLMs and using multiple datasets, including synthetic and augmented datasets developed in this work. Our results show the impact of the factors defined by our framework.
- We demonstrate that reasoning-aligned open-source models of moderate scale outperform larger proprietary models on challenging instruction categories, indicating that indoor location inference is achievable without cloud-scale models—a key enabler for edge deployment.

II. Related Work

The problem of constructing indoor spatial representations has been approached through robotic mapping [31] [32] and, more recently, through natural language processing. While early methods relied on the probabilistic grounding of language into sensor-based maps [33], recent work attempts to leverage LLMs by generating topological graphs directly from textual descriptions [28]. Despite these advancements, existing approaches lack a formalized framework for studying how instruction structure, metadata clarity, and prompting strategies fundamentally govern inference performance.

Prior to LLMs, language-guided mapping was largely a sensor-fusion problem. Early works, such as Walter et al. [33], proposed frameworks to learn semantic maps by fusing natural language descriptions with metric sensor data (e.g., LIDAR, odometry). These methods modeled the environment as a semantic graph, using probabilistic graphical models to ground spatial concepts like "gym" or "hallway" into the robot's metric observations. Although effective for robotic agents equipped with sensors, these approaches are computationally intensive and heavily dependent on physical exploration and low-level metric data. Consequently, they are unsuitable for generating maps solely from varied natural language sources without physical presence.

Recent work has shifted towards leveraging LLMs to infer indoor topological maps directly from text, bypassing the need for sensory data. Karkour et al. introduced TEXT2MAP (T2M), a methodology that utilizes off-the-shelf LLMs to convert navigational instructions into graph-based connectivity matrices [28]. TEXT2MAP demonstrated that LLMs could parse sequences like *"walk past the kitchen into the living room"* to reconstruct topological layouts using few-shot learning, effectively bridging the gap between unconstrained text and structured spatial data. Similarly, Deguchi et al. explored this domain by proposing methods for generating "implicit" (memory-based) versus "explicit" (node-edge) topological maps from textual paths [29]. Their work highlights the superiority of explicit map generation, where the LLM constructs a structured graph of nodes and edges rather than relying on the model's internal memory state for path planning. These studies collectively validate the potential of LLMs to serve as engines for spatial reasoning.

While groundbreaking, current natural-language indoor topological inference approaches exhibit key shortcomings directly tied to the factors that govern reliable graph generation. First, the absence of a defined instruction structure results in inconsistent and ambiguous spatial descriptions, preventing controlled evaluation of how linguistic form influences model reasoning. Second, the lack of metadata clarity, with non-unique or inconsistently labeled regions, introduces reference ambiguity that limits reproducibility and interpretability. Third, the use of monolithic prompting obscures the reasoning process, making it difficult to identify whether errors arise from the model itself or from how it is instructed. Finally, the effect of model size and reasoning capability remains unquantified; current work does not distinguish whether improved mapping stems from increased model scale or reasoning-oriented alignment.

Our framework builds upon this emerging paradigm by introducing formalization as a core design principle. We argue that to meaningfully leverage LLMs for spatial reasoning, one must explicitly define and control these factors: how

instruction structure shapes linguistic clarity, how metadata conventions ensure unambiguous reference resolution, how prompting strategy governs reasoning granularity, and how model scale and reasoning specialization determine compositional understanding and scalability.

By unifying these components under a formal framework, *Lang2Graph* reformulates the task as a structured, analyzable process amenable to controlled study, supporting reproducible experimentation, factor-level attribution, and graph generation across a range of building scales.

The choice of which LLM to deploy for indoor topology inference is not purely an accuracy question but also a deployment question. Recent advances in model compression and on-device inference have demonstrated that large language models can operate under the memory and compute constraints of edge hardware. Post-training quantization methods such as GPTQ [34] and AWQ [35] reduce model precision to 4-bit or lower with minimal accuracy loss, enabling billion-parameter models to fit within the VRAM budgets of mobile GPUs and edge accelerators. Knowledge distillation approaches such as TinyLLM [36] transfer reasoning capabilities from large teacher models to compact student models suitable for on-device deployment. Complementary work on split inference and mobile LLM evaluation [37] characterizes the latency and throughput trade-offs of running transformer models across heterogeneous edge-cloud configurations, while recent surveys [38] synthesize these threads into a broader landscape of efficient LLM inference. These developments motivate our investigation of model scale versus reasoning alignment: the tension between parameter count and reasoning capability is the central design question for any system that must execute the inference pipeline on hardware with bounded memory and compute. Lang2Graph's model-scale experiments contribute empirical evidence to this question in the context of spatial reasoning tasks.

III. Lang2Graph

In this section, we present *Lang2Graph* for natural-language indoor topological inference. We begin by formally defining the task and then introduce the four key factors, instruction structure, metadata clarity, prompting strategy, and model size and reasoning alignment, that govern its performance. Figure 1 illustrates an overview of the framework from navigational instructions to topological graphs.

Formal Task Definition: Let $\mathcal{N} = \{n_1, n_2, \ldots, n_L\}$ be a set of natural-language navigational instructions. Each instruction n_i describes a path between two region-like areas in the indoor environment and may reference intermediate regions encountered along the path. The instruction set is assumed to collectively cover all regions in the building at least once.

Let $\mathcal{M} = \{m_1, m_2, \ldots, m_K\}$ be a set of metadata strings, where each entry takes the form: **"region X is a Y"**, with X denoting a unique region identifier (e.g., "01", "21") and Y denoting its semantic label (e.g., "bathroom", "kitchen"). Each

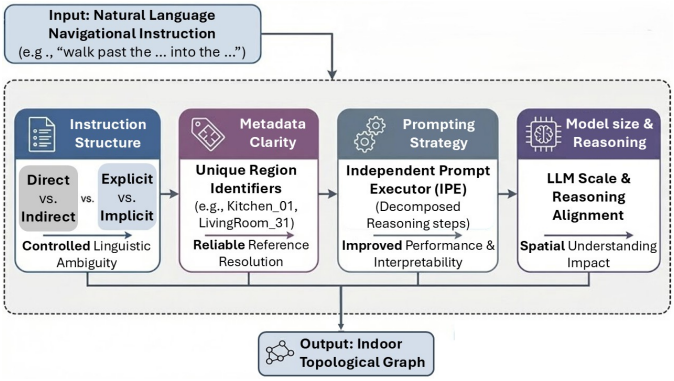

Fig. 1: Lang2Graph overview.

identifier corresponds to a single room-like region, ensuring unambiguous reference resolution.

The goal is to construct an undirected connectivity graph $G = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ corresponds to a region identifier in $\mathcal{M}$, and each edge $(u, v) \in \mathcal{E}$ indicates that regions u and v are directly connected. The output representation of G is an adjacency list:

$$\mathcal{A}(u) = \{\, v \mid (u, v) \in \mathcal{E} \,\}, \qquad \forall u \in \mathcal{V}.$$

Formally, the task is to infer $\mathcal{E}$ from the union of: (1) region-to-region transitions explicitly or implicitly described in the instructions $\mathcal{N}$, and (2) region identities specified in $\mathcal{M}$. The final output is the adjacency-list representation $\mathcal{A}$ encoding the building's topological connectivity.

A. Instruction Structure

A *structured navigational instruction* is defined as a description of a path from one room-like region to another, where all directional transitions are clearly stated, and every room involved in the path is either explicitly or implicitly referenced. We classify instructions along two attributes: (1) the structure of the path being described, and (2) the manner in which regions are referenced. These attributes specify how movement through the environment is expressed and how the involved regions are identified. Table I illustrates the different structure and reference types of instructions.

By formalizing these attributes, the framework enables systematic benchmarking of natural-language indoor topological inference methods and supports controlled dataset generation when needed.

1) Path Structure: Path structure indicates whether an instruction describes a single transition between two regions or a multi-step sequence that includes intermediate regions. An instruction may combine any path structure with either reference type, as illustrated in Table I.

a) Direct Instructions: These describe a path from a source room to a destination room without passing through any intermediate rooms.

> *"Exit the bathroom and move toward the bedroom. As you step in, you are now in the bedroom."*

Instruction	Structure	Edge/s	Type
Exit the **bedroom** and proceed into the **hallway**.	Direct	A→B	Explicit
Move past the bed, turn where the rug meets the floor, and continue forward until the open passage.	Direct	A→B	Implicit
Step past the sink in the **bathroom**, move toward the **bedroom**, exit the **bedroom**, and continue forward until the **dining** area.	Indirect	A→B→C	Explicit
Step past the sink, move toward the wooden dresser, then turn near the wardrobe before continuing to the table.	Indirect	A→B→C	Implicit

TABLE I: Examples of natural-language navigation instructions with varying path structures and room reference types.

b) Indirect Instructions: These describe a path that includes at least one intermediate room between the source and destination.

> *"Step past the sink in the bathroom, move toward the bedroom, exit the bedroom, and continue forward until the dining area."*

2) Room Reference Type: Room reference type specifies how regions are identified within an instruction, distinguishing between explicit mentions of room labels and implicit references based on contextual or functional cues. This attribute captures whether the model can directly ground a reference in metadata or must infer it from surrounding descriptions.

a) Explicit Instructions: These mention the specific labels or names of the rooms involved in the path. These are represented by the underlined words in the sample instruction shared above.

b) Implicit Instructions: These refer to rooms without naming them directly, instead using contextual or functional cues. For example, a room might be identified by an object typically found within it.

> *"walk along the framed pictures, continue through the passage, and enter where the sink is"*

B. Metadata Clarity

In addition to structured instructions, *Lang2Graph* identifies clear building *metadata* as a critical factor of LLM inference accuracy. Clear building metadata represents the descriptive information associated with each region (or room-like node) in the building, which is essential for both instruction interpretation and graph construction.

Each region in the building must be assigned a clear label that reflects its functional identity, such as *'kitchen'*, *'bedroom'*, *'bathroom'*, etc. These labels serve as anchors for both explicit and implicit references in navigational instructions.

Algorithm 1 Independent Prompt Executor (IPE).

Input: Building instructions $\mathcal{I}$, Building Metadata $\mathcal{M}$
Output: The adjacency-list representation $\mathcal{A}$ of the topology graph G
function PROCESSINSTRUCTION(i, $\mathcal{M}$)
    $prompt \leftarrow prompter(i, \mathcal{M})$
    $edges \leftarrow LLM(prompt)$
    return $edges$
end function
function CONSTRUCTGRAPH($\mathcal{E}$)
    $\mathcal{G} \leftarrow \{\}$
    for each $(e_1, e_2) \in \mathcal{E}$ **do**
        if $e_1 \in \mathcal{G}$ **then**
            $edges_{e_1} \leftarrow \mathcal{G}[e_1]$
            $\mathcal{G}[e_1] \leftarrow edges_{e_1} \cup [e_2]$
            if $e_2 \in \mathcal{G}$ **then**
                $edges_{e_2} \leftarrow \mathcal{G}[e_2]$
                $\mathcal{G}[e_2] \leftarrow edges_{e_2} \cup [e_1]$
            else
                $\mathcal{G}[e_2] \leftarrow [e_1]$
        else
            $\mathcal{G}[e_1] \leftarrow [e_2]$
    return $\mathcal{G}$
end function
$E \leftarrow [\ ]$
for $i \in \mathcal{I}$
    $instructionEdges \leftarrow processInstruction(i,\ \mathcal{M})$
    $E \leftarrow E\ \cup\ instructionEdges$
$\mathcal{G} \leftarrow constructGraph(E)$
return $\mathcal{G}$

To support scalability and avoid ambiguity in multi-room or multi-level environments, *Lang2Graph* requires that duplicate room types within the same level must be uniquely identified. This ensures that each room can be distinctly referenced, even if it shares a functional label with others.
This structured metadata should improve the interpretability of instructions and facilitate the generation of consistent graph representations. It helps LLMs resolve references accurately, whether they are explicit (e.g., *"go to the kitchen"*) or implicit (e.g., *"enter the room with the stove"*).

C. Prompting Strategy

We propose a structured prompting strategy, termed the *Independent Prompt Executor* (IPE), that treats indoor topology inference as a composition of independent edge-extraction problems. As shown in Algorithm 1, IPE decomposes the global graph construction task into a sequence of localized inference steps, each conditioned on a single navigational instruction and shared building metadata.

Given a fixed metadata specification describing all regions, IPE processes each instruction independently. For an instruction i, a prompt is constructed by pairing the instruction with the metadata, and the LLM is tasked with inferring only the region-to-region transitions implied by that instruction. The model output is a small set of edges, representing a partial topological graph. This procedure is repeated for all instructions, yielding a collection of instruction-level edge sets.

The final indoor topology is obtained by aggregating these independently inferred edges into a unified connectivity graph. By isolating inference at the instruction level, IPE decouples reference resolution, spatial reasoning, and edge extraction across instructions, reducing cross-instruction interference and

error propagation. This formulation treats the inference task as a compositional process and supports interpretable reasoning, per-instruction error analysis, and predictable scaling with environment size.

D. Model Size and Reasoning Capabilities

The fourth factor in our framework examines how model scale and reasoning specialization influence the process of natural-language indoor topology inference. This factor explores whether improvements in spatial reasoning arise primarily from parameter count or reasoning-oriented alignment. Reasoning alignment [39] plays a central role in enabling LLMs to perform structured, multi-step inference. In the DeepSeek [39] family of models DeepSeek-R1's distilled variants retain much of the parent model's reasoning capabilities despite their smaller parameter counts.

This training paradigm highlights two complementary scaling dimensions:

- **Parameter Scaling**: Larger models generally offer greater representational capacity.
- **Reasoning Alignment**: RL and distillation-based alignment enables smaller models to inherit structured problem-solving behavior without proportional increases in model size.

In the evaluation of Lang2Graph, we leverage a 14B reasoning-distilled open-source model and an 8B reasoning-distilled open-source model (introduced in Section V-A) as representative reasoning-oriented models, alongside non-reasoning baselines of comparable scale and proprietary models of larger scale. This enables a controlled analysis of how reasoning finetuning and parameter count interact with instruction structure, metadata clarity, and prompting strategy. By isolating this factor, we investigate whether robust spatial reasoning emerges from reasoning specialization rather than sheer model size, offering insight into the design of scalable and efficient LLM-based indoor topological inference systems.

IV. Datasets and Augmentation

To systematically evaluate how the four formalized factors of *Lang2Graph* influence inference, careful dataset design is essential. Existing benchmarks alone are insufficient to isolate these factors, as they contain unconstrained language, ambiguous references, and inconsistent metadata. We therefore evaluate our framework using a combination of synthetic and real-world datasets, complemented by a structured augmentation process that enables controlled experimentation under realistic conditions.

A. Dataset Preparation

We employ two primary data sources that serve complementary roles in our evaluation: a fully synthetic dataset designed for controlled analysis, and a real-world dataset containing naturally occurring navigational instructions.

Num. of regions	Direct	Indirect	Total Instructions
5	20	16	36
10	24	10	34
15	40	16	56
Total	84	42	126

TABLE II: Breakdown of the synthetic dataset, showing the number of generated instructions by building size and path type (direct vs. indirect).

a) Synthetic Dataset: The synthetic dataset is designed to reflect the ideal application of our framework. It is generated entirely using GPT-4 [40], guided by carefully crafted prompts and manually verified to ensure full adherence to the instruction structure defined in Section III-B. The breakdown of the synthetic dataset are presented in Table II.

This dataset serves two primary purposes. First, it provides an upper bound on the performance of our prompting strategies and graph construction methods under fully controlled conditions. Second, it enables isolation of how different instruction types (direct vs. indirect and explicit vs. implicit) affect an LLM's ability to extract accurate graph edges. By eliminating real-world linguistic noise, this dataset highlights the intrinsic difficulty of different instruction structures before introducing more complex scenarios.

b) R2R Dataset: To evaluate our framework under realistic, unconstrained conditions, we use the Room-to-Room (R2R) dataset [30], a widely adopted benchmark in Vision-Language Navigation research. R2R is built on top of the Matterport3D dataset [41], which spans 90 diverse buildings and provides rich semantic and topological information. These environments include houses, apartments, hotels, offices, and churches, covering a wide range of indoor layouts.

R2R contains 21,567 crowd-sourced navigation instructions, each describing a path from a starting location to a destination within a building. Instructions are open-vocabulary, average 29 words in length, and often span multiple room-like regions. While R2R offers realistic linguistic variability and complex spatial layouts, its instructions lack explicit region identifiers and consistent metadata. As a result, it does not natively support controlled evaluation of reference ambiguity or instruction structure, motivating the augmentation process described next.

B. Dataset Augmentation: R2R-AUG

To bridge the gap between idealized synthetic data and unconstrained real-world instructions, we introduce a structured augmentation of the R2R dataset, referred to as *R2R-AUG*. This augmentation enforces metadata clarity and systematically controls how regions are referenced in natural-language instructions, while preserving the underlying spatial layouts and semantic content of the original dataset.

In its original form, R2R exhibits ambiguous room references, inconsistent naming conventions, and missing region identifiers, preventing precise attribution of inference errors to linguistic structure, metadata quality, or prompting strat-

Num. of regions	Frequency	Direct	Indirect	Total Instructions
1–10	13	78	524	602
11–20	28	414	6858	7272
21–30	18	282	5998	6280
31–40	8	132	2512	2644
41–58	3	62	1232	1294
Total	70	968	17124	18092

TABLE III: Distribution of instructions in the augmented R2R dataset, categorized by region count and instruction type.

egy. Without augmentation, it is very difficult to disentangle whether failures arise from unclear instructions, insufficient metadata, or limitations of the model itself. R2R-AUG addresses these issues by explicitly aligning the dataset with the factors formalized in Section III.

a) Metadata Annotation: Each region in the R2R environments is assigned a unique identifier following the conventions outlined in Section III-C. Duplicate room types within the same level are explicitly disambiguated using indexed suffixes (e.g., *bedroom 00*, *bedroom 01*). This ensures that all room references are uniquely identifiable and consistently interpretable by both humans and models.

b) Instruction Augmentation: For each original R2R instruction, we generate two structured variants; **Explicit instructions**, in which all room references are replaced with their corresponding annotated identifiers. **Implicit instructions**, in which room names are replaced with functional or contextual descriptions (e.g., *"the room with the stove"* instead of *"kitchen"*). This process enables controlled comparison between explicit and implicit reference types while preserving the original navigational intent.

Table III presents the breakdown of the augmented dataset and reports the distribution of instructions by building size and instruction type.

The augmentation procedure is formalized in Algorithm 2, which outlines the metadata annotation process and the generation of both instruction variants. Unlike the synthetic dataset, the LLM used for augmentation is *DeepSeek-R1-Distill-Qwen-14B* [39], chosen for its reasoning capability in resolving object–room associations required for implicit references.

Together with the synthetic dataset, R2R-AUG enables evaluation across a spectrum of linguistic structure and metadata clarity, ranging from fully controlled conditions to realistic, noisy environments.

V. Experimental Evaluation

We describe the experimental setup—LLMs, evaluation metrics, and implementation details—and then present a comparative analysis across prompting strategies, instruction configurations, and model types. The evaluation isolates the contribution of each of the four formalized factors to the reliability, interpretability, and scalability of this task. Figure 2 outlines the overall experimentation process.

Algorithm 2 Building annotation algorithm.

Input: Building instructions $\mathcal{I}$, Building Metadata $\mathcal{M}$
Output: Two Sets of instructions $\mathcal{I}_{imp}$, $\mathcal{I}_{exp}$, and annotated metadata $\mathcal{M}'$
function ANNOTATELEVEL($\mathcal{L}$)
  $LabelIndex \leftarrow \{\ label : 0\ \}$ **for** $label \in \mathcal{L}[r];\ r \in \mathcal{L}$
  $\mathcal{L}' \leftarrow \{\}$
  for each $r_i \in \mathcal{L}$ **do**
    $label_{r_i} \leftarrow \mathcal{L}[r_i]$
    for each $r_j \in \mathcal{L}$ **do**
      $label_{r_j} \leftarrow \mathcal{L}[r_j]$
      if $label_{r_i} == label_{r_j}$ **and** $r_i \neq r_j$ **then**
        $\mathcal{L}'[r_i] \leftarrow append(label_{r_i},\ LabelIndex[label_{r_i}])$
        $increment(LabelIndex[label_{r_i}])$
  return $\mathcal{L}'$
end function
$\mathcal{M}' \leftarrow [\ ]$
$\mathcal{I}_{imp} \leftarrow \{\}$
$\mathcal{I}_{exp} \leftarrow \{\}$
for $l \in \mathcal{M}$
  $l' \leftarrow annotateLevel(l)$
  $\mathcal{M}' \leftarrow \mathcal{M}' \cup l'$
for $i \in \mathcal{I}$
  $\mathcal{I}_{imp}[i] \leftarrow annotateImplicit(i, \mathcal{M}')$
  $\mathcal{I}_{exp}[i] \leftarrow annotateExplicit(i, \mathcal{M}')$

yield $\mathcal{I}_{imp}$, $\mathcal{I}_{exp}$, $\mathcal{M}'$

A. Experimental Setup

1) LLMs: To evaluate the effects of the LLM size and reasoning alignment on the accuracy of inference, we select five LLMs that vary along three key dimensions: openness (proprietary vs. open-source), scale (large vs. medium vs. small), and reasoning capability. This diversity enables controlled comparisons of performance, reasoning robustness, and scalability across different model families.

a) Proprietary Models: We include *GPT-4o* and *GPT-4o-mini*, representing state-of-the-art proprietary LLMs. *GPT-4o* serves as the large-scale benchmark, while *GPT-4o-mini* provides a compact baseline to assess performance trade-offs under reduced capacity. Both are used to establish upper-bound performance under ideal synthetic conditions.

b) Open-Source Models: Three open-source models are selected to ensure transparency, reproducibility, and insight into architecture-driven variability.

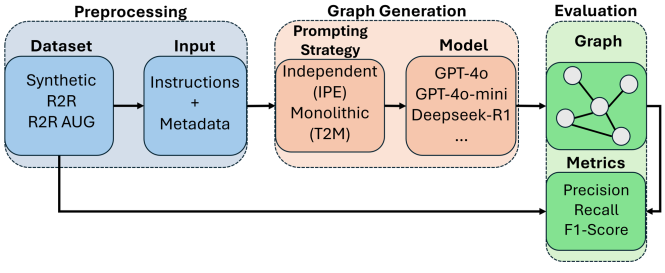

Fig. 2: Overall experimental process: We first select the dataset, then the prompting strategy, then the LLM, and finally evaluate the generated topological graph.

TABLE IV: Comparison of models with properties and performance across IPE and T2M.

Model	Size	Open-Src	Reason	IPE								T2M							
				IMP-DIR		IMP-IND		EXP-DIR		EXP-IND		IMP-DIR		IMP-IND		EXP-DIR		EXP-IND	
				Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
gpt-4o	200B	x	x	0.88	0.77	0.71	0.77	1.00	0.95	0.90	1.00	1.00	0.67	0.55	0.22	1.00	0.67	0.75	0.55
gpt-4o-mini	8B	x	x	0.94	0.92	0.61	0.61	1.00	1.00	0.85	1.00	0.93	0.79	0.51	0.41	0.94	0.72	0.79	0.74
Deepseek Distill Qwen	14B	✓	✓	1.00	1.00	0.81	0.73	1.00	1.00	0.78	0.80	0.93	0.66	0.65	0.54	0.83	0.65	0.66	0.53
Deepseek Distill Llama	8B	✓	✓	0.77	0.67	0.66	0.72	0.80	0.75	0.67	0.81	0.40	0.76	0.25	0.50	0.50	0.77	0.45	0.61
Llama 3.1 Instruct	8B	✓	x	0.92	0.68	0.47	0.59	0.95	0.95	0.73	0.92	0.20	0.30	0.20	0.46	0.73	0.76	0.20	0.23

- *DeepSeek-R1-Distill-Qwen-14B*, a large reasoning-oriented model used as the primary diagnostic engine in structured and real-world experiments. Its explicit step-by-step reasoning makes it well-suited for analyzing instruction ambiguity and metadata vagueness.
- *DeepSeek-R1-Distill-Llama-8B*, a small reasoning-tuned model used to evaluate scale and performance consistency among open-source reasoning LLMs.
- *Meta-Llama-3.1-Instruct-8B*, a non-reasoning model that serves as a baseline for assessing how reasoning alignment affects spatial understanding and prompt execution.

This model suite spans both reasoning paradigms and parameter scales, enabling a systematic investigation of how model capacity and reasoning depth interact with structured prompting on this task. The five models implement a deliberate 2×2 experimental design crossing reasoning alignment (reasoning-distilled vs. instruction-tuned only) with openness (open-source vs. proprietary), with parameter scale as a third continuous dimension. This factorial structure enables controlled attribution of performance differences to reasoning alignment rather than model family or training corpus, a comparison that would be confounded if all models shared the same training paradigm. While additional models exist (e.g., Gemma, Mistral, GPT-o3), expanding the suite would not alter the design's ability to isolate the reasoning alignment factor, which is the primary variable of interest. GPT-4o represented the state-of-the-art proprietary model at the time of experimentation, and the DeepSeek-R1 distillation family represented the most capable openly available reasoning-aligned models at the 8B–14B scale class.

2) Evaluation Metrics: To assess the effectiveness and quality of the generated indoor topological graphs, we employ three standard evaluation metrics: Precision, Recall, and F1 Score. Let $\mathcal{E}_{\text{pred}}$ denote the set of edges in the predicted graph and $\mathcal{E}_{\text{gt}}$ denote the set of edges in the ground-truth graph. These metrics quantify how well the predicted graph edges align with the ground-truth connectivity of the environment. Since G is undirected, edges are treated as unordered pairs, so (u, v) and (v, u) contribute as a single edge when computing the intersection $|\mathcal{E}_{\text{pred}} \cap \mathcal{E}_{\text{gt}}|$.

a) Precision: Precision reflects the model's ability to avoid false positives, i.e., predicted connections that do not exist in the actual building layout:

$$\text{Precision} = \frac{|\mathcal{E}_{\text{pred}} \cap \mathcal{E}_{\text{gt}}|}{|\mathcal{E}_{\text{pred}}|} \tag{1}$$

b) Recall: Recall captures the model's ability to identify all relevant connections, minimizing false negatives:

$$\text{Recall} = \frac{|\mathcal{E}_{\text{pred}} \cap \mathcal{E}_{\text{gt}}|}{|\mathcal{E}_{\text{gt}}|} \tag{2}$$

c) F1 Score: The F1 Score is the harmonic mean of Precision and Recall, providing a single value that balances both concerns when they are in tension:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

3) Implementation Details: All experiments are carried out on a Linux-based high-performance system equipped with two NVIDIA A6000 GPUs, an Intel Core i7 CPU, and 128 GB of RAM.

B. Results

We report results across all datasets and models, organized by the four formalized factors. Each factor is analyzed across multiple datasets and LLMs to reveal its effect on graph generation performance, measured through Precision, Recall, and F1-score. Together, these results show how controlled linguistic structure, unique metadata, and independent prompting jointly affect the reliability and scalability of the task.

*1) **Impact of Instruction Structure**:* To evaluate the impact of instruction structure, we first evaluate the performance of different combinations of instruction types, direct vs. indirect and implicit vs. explicit using IPE and T2M on the synthetic dataset across all five models.

The synthetic dataset experiment's results in Table IV show that instruction structure has a direct and measurable effect on graph reconstruction accuracy. There is a clear performance gap between explicit and implicit instructions, as well as between direct and indirect paths. Implicit instructions yield substantially lower scores because the model must infer the correct region label from contextual cues and align it with metadata. For example, across all models, implicit-indirect combinations consistently underperform explicit-direct ones: *GPT-4o* records precision / recall of $0.71/0.77$ for implicit-indirect versus $1.00/0.95$ for explicit-direct, while *DeepSeek-R1-Distill-Qwen-14B* achieves $0.81/0.73$ for implicit-indirect and $1.00/1.00$ for explicit-direct. Similarly, indirect instructions are more challenging than direct ones even within the same reference type, as they require recognizing intermediate transitions before identifying the final destination.

Solution	Direct	Indirect	Precision	Recall	F1-Score
T2M	✓	**X**	0.61	0.84	0.7
	X	✓	0.20	0.27	0.22
	✓	✓	0.24	0.28	0.26
IPE	✓	**X**	0.10	0.71	0.17
	X	✓	0.26	0.64	0.37
	✓	✓	0.20	0.48	0.28

TABLE V: Comparison of T2M and IPE performance on the R2R dataset, evaluated on direct and indirect instructions using precision, recall, and F1 score.

Vagueness	Direct	Indirect	Implicit	Explicit	Precision	Recall	F1-Score
50%	✓	**X**	✓	**X**	1.00	1.00	1.00
	✓	**X**	**X**	✓	0.89	0.88	0.88
	X	✓	✓	**X**	0.80	1.00	0.89
	X	✓	**X**	✓	0.70	0.98	0.81
100%	✓	**X**	✓	**X**	1.00	1.00	1.00
	✓	**X**	**X**	✓	0.89	0.98	0.93
	X	✓	✓	**X**	0.67	0.93	0.78
	X	✓	**X**	✓	0.58	0.72	0.64

TABLE VI: Performance of IPE on the synthetic dataset under metadata vagueness (50% and 100%).

We also evaluate the performance of unstructured instructions on the R2R dataset using IPE and T2M. This experiment provides a baseline diagnosis on unstructured data using a large reasoning model under identical conditions.

The results of the R2R dataset experiment in Table V show that the pattern persists under conditions where natural crowd-sourced instructions introduce uncontrolled phrasing and reference styles. This decline indicates that instruction clarity and structure contribute substantially to graph accuracy alongside model capacity.

This performance gap provides a key insight: LLMs' compositional reasoning is greatly improved by the structure of the linguistic input. The framework's categorization of instructions into direct/indirect and explicit/implicit quantifies these differences and enables systematic benchmarking. This allows future dataset designers to measure and control instruction quality, bridging the gap between crowd-sourced variability and formal spatial reasoning tasks.

*2) **Impact of Model size and Reasoning Capability**:* We assess how model scale and reasoning specialization interact with the formalized factors, instruction structure, metadata clarity, and prompting strategy. Results in Table IV indicate that reasoning alignment contributes more to performance than raw parameter count, especially for complex or ambiguous instructions.

Proprietary models (*GPT-4o*, *GPT-4o-mini*) perform near-perfectly on explicit-direct instructions (Precision/Recall $\approx$ 1.00/0.95 and 1.00/1.00), but degrade on implicit-indirect cases. The reasoning-tuned *DeepSeek-R1-Distill-Qwen-14B* remains more robust under these challenging conditions, achieving 0.81/0.73, surpassing *GPT-4o* despite being significantly smaller (14B vs. 200B parameters). The smaller reasoning model, *DeepSeek-Distill-LLaMA-8B*, shows a modest decline (Precision/Recall 0.66/0.72), indicating that while reasoning finetuning is important, model capacity still matters. The non-reasoning *LLaMA-3.1-Instruct-8B* lags behind comparably sized reasoning models, further reinforcing the role of reasoning alignment. These findings carry implications for resource-constrained deployment: because reasoning alignment outweighs raw parameter count on the most challenging instruction categories, moderate-scale open-source models can achieve reliable indoor topology inference without requiring cloud-scale proprietary infrastructure. This suggests that edge deployment of LLM-based indoor mapping is feasible when the inference pipeline is paired with appropriate prompting decomposition and structured input design.

*3) **Impact of Instruction Path Length**:* We evaluate the performance of different instruction path lengths using IPE on R2R AUG using the large open-source reasoning model. We define an instruction's path length as the number of regions that an instruction passes through from the start to the end regions. The results in Figure 3 show a gradual decline in F1-score as instruction path length increases. Instructions describing shorter paths are reconstructed more reliably, whereas longer paths exhibit reduced accuracy. This behavior suggests that additional intermediate regions introduce cumulative uncertainty in reference resolution and transition ordering, even when instructions are processed independently using IPE. Although the large open-source reasoning model remains robust for moderate path lengths, performance degradation becomes more noticeable as the number of intermediate regions grows. These results indicate that indirect instructions impose additional reasoning demands and remain more challenging than shorter, direct paths under otherwise identical conditions.

*4) **Impact of Metadata Clarity**:* To evaluate the impact of metadata clarity, we test IPE on the synthetic dataset using the large reasoning model under varying levels of metadata vagueness. Table VI presents the results of the experiment. We define vagueness as the proportion of regions in a building that have at least one duplicate region with the same label on the same level. For example, a building with 50% vagueness has half of its regions sharing labels with at least one other region on the same floor. This simulates real-world scenarios where

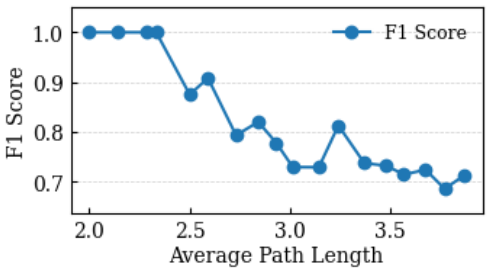

Fig. 3: F1 score vs average path length using IPE on the augmented dataset using the large open-source reasoning model.

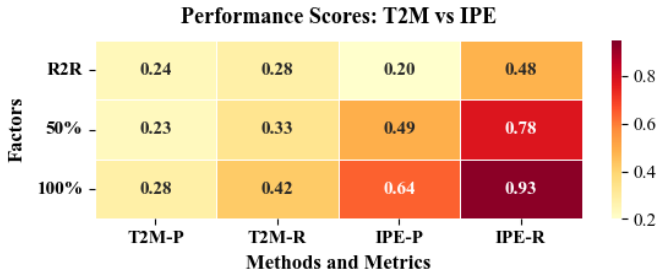

Fig. 4: Precision and recall comparison of IPE vs. T2M with different levels of dataset augmentation.

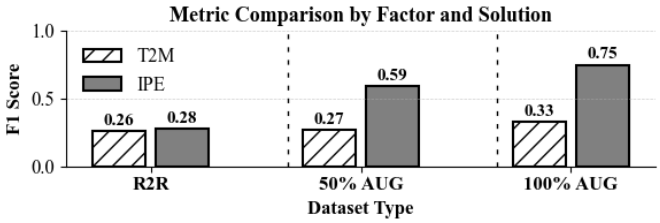

Fig. 5: F1 score comparison of T2M vs. IPE across augmentation levels on R2R (original), 50% augmentation, and 100% augmentation.

metadata may be ambiguous or insufficiently disambiguated. We experiment with two vagueness levels, 50% and 100%, and evaluate performance across direct, indirect, implicit, and explicit instruction combinations.

The results show that performance generally declines with increasing vagueness: at 50% overlap, IPE achieves an F1 of 0.89 on implicit-indirect cases, which drops to 0.78 at 100%.

The results highlight that LLMs depend on unambiguous metadata to ground linguistic references. Without disambiguation, language-only inference fails to resolve multiple candidate regions, leading to incorrect edge associations in the constructed graph.

*5) **Impact of Prompting Strategy**:* To evaluate how prompting strategies interact with varying instruction structures and levels of metadata clarity, we perform partial augmentation experiments on the R2R dataset. In this setup, 50% augmentation indicates that half of the instructions and their corresponding regions follow the structured design and metadata conventions defined in our framework, while the remaining half retain their original unstructured form. 100% augmentation corresponds to the fully structured R2R-AUG dataset, where all instructions and regions comply with the framework's guidelines.

The results, in Figures 4 and 5, show that both IPE and T2M degrade under unstructured conditions, but IPE exhibits strong recovery as structure is introduced. On the original R2R dataset, IPE achieves an F1 score of 0.28, rising to 0.59 at 50% augmentation and 0.75 at 100% augmentation. In contrast, T2M remains largely insensitive to added structure, improving only marginally from 0.26 to 0.33 under full augmentation. This demonstrates that IPE can exploit linguistic and metadata structure when present, while monolithic prompting dilutes these benefits by processing all instructions jointly. However, results in Figure 4 show that IPE still struggles with a moderate precision score of 0.64 in the fully augmented dataset. This shows that IPE is affected by the added complexity that the R2R description presents over the synthetic dataset. The augmentation process constitutes a diagnostic finding: the performance gap between R2R and R2R-AUG quantifies the structural distance between crowd-sourced navigational language and the linguistic properties required for reliable inference. This gap serves as a proxy measure that informs dataset designers and system deployers of exactly which instruction-level and metadata-level properties must be engineered to achieve robust graph generation from naturalistic input.

By decomposing map construction into localized reasoning steps, IPE reduces error propagation and confines uncertainty to individual instructions. This independent structure groups reasoning into per-instruction units, limiting how much an error in one instruction can affect others. In addition, IPE's modularity aids interpretability: each instruction produces a distinct subgraph, allowing analysts to trace specific edges, evaluate partial failures, and incrementally refine the inference process.

Collectively, these findings indicate that IPE decomposes language-to-graph generation into a more transparent, compositional process, with measurable improvements in accuracy, robustness, and scalability across the studied instruction and metadata conditions.

*6) **Impact of Building Scale and Instruction Density**:* To evaluate scalability, we examine two complementary aspects: building scale and instruction density. The building-scale experiments are conducted on three open-source models, *DeepSeek-R1-Distill-Qwen-14B*, *DeepSeek-Distill-LLaMA-8B*, and *LLaMA-3.1-Instruct-8B*, to assess the impact of building scale on performance across varying levels of LLM size and reasoning alignment. In contrast, the instruction-density experiments are performed exclusively on the large reasoning model *DeepSeek-R1-Distill-Qwen-14B* to isolate the effects of linguistic redundancy under fixed model and environment conditions.

Figure 6 depicts the F1 scores when using the three open-source models while varying the number of regions in order to reflect building scale. As building complexity increases, the performance of all models decreases, but the degradation rate varies sharply by prompting strategy. Across all scales, IPE maintains stable accuracy, while T2M deteriorates rapidly. For example, using *DeepSeek-R1-Distill-Qwen-14B*, IPE achieves near-saturated performance on small environments (F1 $\approx$ 0.94 for 1-10 regions) and drops slightly at 30-60 regions (F1 $\approx$ 0.68), while T2M collapses below 0.25 under identical conditions. Similar patterns hold across the smaller models: reasoning-oriented alignment sustains performance, while non-reasoning models exhibit steeper declines as the number of regions increases. These results confirm that reasoning capability, rather than parameter count alone, governs scalability in complex environments.

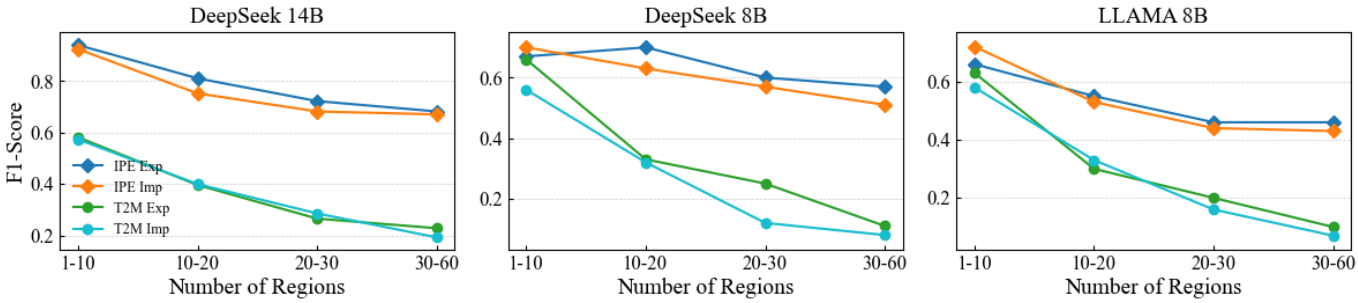

Fig. 6: F1 score vs. number of regions using IPE on the augmented dataset across three open-source models.

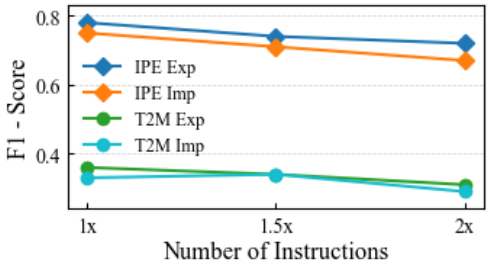

Fig. 7: F1 score vs number of instructions using IPE on the augmented dataset using the large open-source reasoning model.

Figure 7 analyzes how instruction density influences performance. We define instruction density as: $1\times$ indicates that each edge is represented once in the instructions, $1.5\times$ corresponds to an average of 1.5 instructions per edge, and $2\times$ means that two distinct instructions describe each edge. At $1\times$ density, IPE achieves F1 scores of 0.78 (explicit) and 0.75 (implicit), while T2M only achieves 0.36 and 0.33. As density increases to $1.5\times$ and $2\times$, both prompting strategies experience slight but consistent performance drops, IPE decreases to 0.72/0.67 and T2M to 0.31/0.29. Contrary to intuition, adding more instructions does not improve accuracy; instead, it introduces redundant or conflicting linguistic cues that increase the likelihood of generating spurious or hallucinated edges.

*7) **Implications for Edge Deployment**:* The experimental results carry direct implications for deploying LLM-based indoor topology inference on edge hardware. At FP16 precision, the 14B reasoning-distilled model requires approximately 28 GB of VRAM, while the 8B variant requires approximately 16 GB. Standard 4-bit post-training quantization techniques [34], [35] reduce these requirements by roughly $4\times$, bringing the 8B model to approximately 4 GB—within the memory budget of commodity edge AI accelerators such as the NVIDIA Jetson AGX Orin (64 GB unified memory) or comparable inference-class devices. Because the 14B reasoning-distilled model achieves the highest open-source performance across challenging instruction categories, and the 8B variant maintains reasonable accuracy, both represent viable candidates for on-device deployment.

IPE's per-instruction independence is architecturally well-suited to resource-constrained inference. Each navigational instruction is processed as a separate reasoning call with shared building metadata, eliminating the need to hold the full instruction set in context simultaneously. This design reduces peak memory consumption during inference and enables incremental graph construction—a property that benefits systems receiving instructions progressively from a crowd-sourcing pipeline or from a user navigating in real time.

Together, these results suggest that reliable indoor topology inference is achievable on edge-deployable hardware using reasoning-aligned open-source LLMs of moderate scale, without dependence on cloud-scale proprietary models.

VI. Conclusion

Lang2Graph provides a systematic experimental framework for indoor topological graph inference from natural-language navigational instructions, targeting edge-deployed systems that require accurate spatial representations under real-time constraints. By isolating four governing factors (instruction structure, metadata clarity, prompting strategy, and model size and reasoning capability), the framework demonstrates that structured instructions, clear building metadata, and independent prompting through IPE improve precision, recall, and F1 by 44%, 45%, and 47% respectively. IPE decomposes graph construction into independent per-instruction reasoning steps, preventing error propagation. The synthetic dataset and R2R-AUG benchmark support reproducible, factor-level evaluation under both idealized and real-world conditions. Reasoning-aligned open-source models of moderate scale achieve reliable topology inference without cloud-scale dependencies, establishing a viable path toward edge deployment. Future work will characterize the trade-off between post-training quantization and spatial-reasoning fidelity, measure end-to-end latency, memory, and energy on commodity edge accelerators (e.g., the NVIDIA Jetson family), broaden baseline comparisons to retrieval-augmented, multi-agent, and classical rule-based methods, and study finer-grained error categorization under noisy, contradictory, or incomplete instructions.

Acknowledgment

This research used GPT-4 [40] to generate the synthetic dataset (Section IV) and DeepSeek-R1-Distill-Qwen-14B [39] for the augmentation in Algorithm 2, under prompts verified by the authors. All experimental results were produced by the authors.

References

[1] A. Saeed, A. Abdelkader, M. Khan, A. Neishaboori, K. A. Harras, and A. Mohamed, "On realistic target coverage by autonomous drones," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 3, pp. 1–33, 2019.

[2] M. Khan, K. Heurtefeux, A. Mohamed, K. A. Harras, and M. M. Hassan, "Mobile target coverage and tracking on drone-be-gone uav cyber-physical testbed," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3485–3496, 2017.

[3] A. Saeed, A. Abdelkader, M. Khan, A. Neishaboori, K. A. Harras, and A. Mohamed, "Argus: realistic target coverage by drones," in *ACM/IEEE IPSN*, 2017.

[4] R. Kitchin and O. Dawkins, "Digital twins and deep maps," *Transactions of the Institute of British Geographers*, vol. 50, 07 2024.

[5] S. Mostafa, K. A. Harras, and M. Youssef, "Unicellular: An accurate and ubiquitous floor identification system using single cell tower information," in *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems (SIGSPACIAL)*, 2023.

[6] S. Mostafa, M. Youssef, and K. A. Harras, "Accurate and ubiquitous floor identification at the edge using a single cell tower," in *IEEE/ACM SEC*, 2024.

[7] O. Hashem, M. Youssef, and K. A. Harras, "Winar: Rtt-based sub-meter indoor localization using commercial devices," in *IEEE PerCom*, 2020, pp. 1–10.

[8] S. Puttinaovarat, S. Jutapruet, A. Saeliw, S. Pruitikanee, J. Kongcharoen, W. Jiamsawat, S. Limpasamanon, and M. Srirat, "Facility maintenance management system based on gis and indoor map," *International Journal of Electrical and Computer Engineering*, vol. 9, pp. 3323–3332, 08 2019.

[9] K. Alkiek, K. A. Harras, and M. Youssef, "EarGest: Hand gesture recognition with earables," in *Proc. 19th Annu. IEEE Int. Conf. Sens. Commun. Netw.*, 2022, pp. 91–99.

[10] K. Alkiek, M. Youssef, and K. A. Harras, "EarBender: Enabling rich imu-based natural hand-to-ear interaction in commodity earables," in *IEEE PerCom*, 2023, pp. 333–338.

[11] O. Hashem, K. A. Harras, and M. Youssef, "Deepnar: Robust time-based sub-meter indoor localization using deep learning," in *2020 17th Annual IEEE international conference on sensing, communication, and networking (SECON)*. IEEE, 2020, pp. 1–9.

[12] ——, "Accurate indoor positioning using ieee 802.11 mc round trip time," *Pervasive and Mobile Computing*, vol. 75, p. 101416, 2021.

[13] S. Mostafa, K. A. Harras, and M. Youssef, "A survey of indoor localization systems for multi-floor environments," *IEEE Access*, 2025.

[14] M. A. Shah, K. A. Harras, and B. Raj, "Sherlock: A crowd-sourced system for automatic tagging of indoor floor plans," in *IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2020.

[15] A. Essameldin, M. N. Hoque, and K. A. Harras, "More than the sum of its things: Resource sharing across iots at the edge," in *IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2020.

[16] H. Gedawy, K. A. Harras, K. Habak, and M. Hamdi, "Femtoclouds beyond the edge: The overlooked data centers," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 44–49, 2020.

[17] H. K. Gedawy, K. A. Harras, T. Tanveer, and T. Bui, "Bridging the chasm between ideal and realistic federated learning: A measurements study," in *IEEE CloudCom*, 2023.

[18] H. Gedawy, A. Elgazar, and K. A. Harras, "Maestro: Orchestrating computational offloading to multiple femtoclouds in various communication environments," *IEEE Access*, vol. 10, pp. 27 096–27 112, 2022.

[19] A. Abdullah *et al.*, "Scanning technologies to building information modelling: A review," *Infrastructures*, vol. 7, no. 4, p. 49, 2021.

[20] J. Chen and K. C. Clarke, "Indoor cartography," *Cartography and Geographic Information Science*, vol. 47, no. 2, pp. 95–109, 2020.

[21] "Problems in indoor mapping and modelling," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, pp. 63–68, 2013.

[22] M. A. Shah, B. Raj, and K. A. Harras, "Inferring room semantics using acoustic monitoring," in *IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2017.

[23] X. Song, X. Liang, and Z. Huaidong, "Semantic mapping techniques for indoor mobile robots: Review and prospect," *Measurement and Control*, vol. 58, no. 3, pp. 377–393, 2025. [Online]. Available: https://doi.org/10.1177/00202940241259903

[24] Y. Ansari, A. Karkour, E. F. Flushing, and K. A. Harras, "Tesseract: Unfolding navigable graph representations from low-semantic floor plans," in *ACM SIGSPACIAL*, 2025.

[25] H.-G. Ryoo, T. Kim, and K.-J. Li, "Comparison between two ogc standards for indoor space: Citygml and indoorgml," in *Proceedings of the Seventh ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, 2015, pp. 1–8.

[26] Open Geospatial Consortium, "Indoor mapping data format (imdf) 1.0.0," https://docs.ogc.org/cs/20-094/, 2021, oGC Community Standard, developed by Apple Inc.

[27] F. Noardo, K. Arroyo Ohori, F. Biljecki, C. Ellul, L. Harrie, T. Krijnen, H. Eriksson, J. van Liempt, M. Pla, A. Ruiz, D. Hintz, N. Krueger, C. Leoni, L. Leoz, D. Moraru, S. Vitalis, P. Willkomm, and J. Stoter, "Reference study of citygml software support: The geobim benchmark 2019—part ii," *Transactions in GIS*, vol. 25, no. 2, p. 842–868, Nov. 2020. [Online]. Available: http://dx.doi.org/10.1111/tgis.12710

[28] A. Karkour, K. A. Harras, and E. Feo-Flushing, "Text2map: From navigational instructions to graph-based indoor map representations using llms," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 1153–1160.

[29] H. Deguchi, K. Shibata, and S. Taguchi, "Language to map: Topological map generation from natural language path instructions," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2024, p. 9556–9562. [Online]. Available: http://dx.doi.org/10.1109/ICRA57147.2024.10611377

[30] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3674–3683.

[31] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE robotics & automation magazine*, vol. 13, 2006.

[32] Y. Cao, L. Hu, and L. Kneip, "Representations and benchmarking of modern visual slam systems," *Sensors*, vol. 20, no. 9, p. 2572, 2020. [Online]. Available: https://doi.org/10.3390/s20092572

[33] M. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller, "Learning semantic maps from natural language descriptions," 06 2013.

[34] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "Gptq: Accurate post-training quantization for generative pre-trained transformers," *arXiv preprint arXiv:2210.17323*, 2022.

[35] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, "Awq: Activation-aware weight quantization for on-device llm compression and acceleration," *Proceedings of machine learning and systems*, vol. 6, pp. 87–100, 2024.

[36] S. V. Kandala, P. Medaranga, and A. Varshney, "Tinyllm: A framework for training and deploying language models at the edge computers," *arXiv preprint arXiv:2412.15304*, 2024.

[37] S. Laskaridis, K. Katevas, L. Minto, and H. Haddadi, "Melting point: Mobile evaluation of language transformers," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 890–907.

[38] Z. Zhou, X. Ning, K. Hong, T. Fu, J. Xu, S. Li, Y. Lou, L. Wang, Z. Yuan, X. Li *et al.*, "A survey on efficient inference for large language models," *arXiv preprint arXiv:2404.14294*, 2024.

[39] DeepSeek-AI, D. Guo, D. Yang, and et. al, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[40] 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.

[41] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," in *2017 International Conference on 3D Vision (3DV)*, 2017, pp. 667–676.