Navigating Mazes with Large Language Models: The Impact of Spatial Representations and Frames of Reference

Valérie S. van Noesel
V.S.vanNoesel@student.tudelft.nl
Student Number: 4874579
Department of Cognitive Robotics

Defense date:
May 28^{th} 2026

Supervisor: Prof.dr.ir. J.C.F. de Winter

Examination committee:
Prof.dr.ir. J.C.F. de Winter (chair)
Dr. D. Dodou
R. Zhang, MSc

This thesis is submitted to obtain the degree of Master of Science in Robotics Engineering from Delft University of Technology (TU Delft), department of Cognitive Robotics.

Navigating Mazes with Large Language Models: The Impact of Spatial Representations and Frames of Reference

Valérie S. van Noesel
Department of Cognitive Robotics
Delft University of Technology
Mekelweg 2, Delft
V.S.vanNoesel@student.tudelft.nl
Student Number: 4874579

Abstract

Current evaluations of Large Language Model (LLM) spatial reasoning focus on several isolated competencies rather than a unified task, and use an array of different input formats. As a result, the influence of spatial representation and output Frame of Reference (FoR) on performance in navigation tasks remains unclear. This study asks: how do spatial representations and frames of reference influence LLMs' spatial reasoning capabilities, and which combinations are conducive to it?

This research investigates the spatial reasoning and navigation capabilities of one reasoning and one non-reasoning LLM. Using perfect mazes as a controlled testbed, this thesis examines how various input spatial representations, including visual (JPG and ASCII), grid-based (JSON and Tagged per-cell), and graph-based (Adjacency List) formats, interact with different output FoRs to influence model performance.

The methodology involves an evaluation using Gemini 2.5 Pro (reasoning) and Gemini 2.5 Flash-Lite (non-reasoning) across 11 spatial representations and three output FoRs: allocentric using absolute coordinates ("coordinates"), allocentric using absolute directions ("absolute directions"), and egocentric (relative directions). Performance is measured using two metrics: a "completion score", defined as the percentage of the path navigated correctly before the first error, and the mean number of output tokens generated, used as a proxy for efficiency.

The findings of this research indicate that performance is highest when mazes are expressed using structured graph-based spatial representations, particularly Adjacency List JSON (a graph-based representation formatted as a JSON file), across model types, while the choice of output FoR strongly shapes outcomes, with absolute coordinate responses yielding substantially better results than egocentric ones that require continuous relational analysis and state tracking and therefore lead to markedly lower completion scores, especially for the non-reasoning model. In addition, inspection of internal reasoning traces suggests that the use of formal graph-solving algorithms is positively correlated with success, while exclusive reliance on heuristics and unfounded declarations of confidence are negatively correlated with completion scores.

By systematically varying input spatial representation and output FoR this work provides the first integrated evaluation of these factors, addressing the lack of unified benchmarks and clarifying how methodological choices shape observed LLM spatial reasoning performance. The source code of this work is publicly available on GitHub.

1 Introduction

Large Language Models (LLMs) are a class of artificial intelligence systems designed for natural language processing and generation through probabilistic next-token prediction. They have emerged as powerful tools that can solve queries in a wide range of domains. In robotics, LLMs are increasingly used in embodied AI systems for autonomous navigation (Doula et al., 2025; Latif, 2024), where they function as path planners or produce suitable actions for autonomous agents (Lin et al., 2025), due to their ability to understand and generate natural language instructions (Tariq et al., 2025). Increasingly, as the deployment of autonomous robotic systems grows, the ability for LLMs to perform spatial reasoning is becoming indispensable for tasks such as autonomous navigation, robotic manipulation, and human-machine interaction through natural language (Fung et al., 2025; Shi et al., 2022). For these applications, achieving high navigational accuracy efficiently is essential.

A distinction exists between reasoning and non-reasoning LLMs, based on how the LLM behaves at inference. In this work, both classes of LLMs are evaluated on the same navigation task, despite generating answers through mechanistically different processes. Reasoning LLMs generate an internal reasoning trace in which self-correction or task exploration can occur prior to final answer generation. In contrast, non-reasoning LLMs lack a separate reasoning phase. Instead, they rely on a single generation pass to output task-appropriate solutions based on the input data. Therefore, while non-reasoning LLMs do not formally reason, they still process and incorporate spatial information to resolve navigational tasks. Given that these tasks traditionally necessitate spatial cognition in human agents, this thesis adopts the term "spatial reasoning" as a functional umbrella term. This definition encompasses both the explicit reasoning traces of reasoning LLMs and the statistical utilization of spatial data inputs by non-reasoning LLMs when addressing the same navigational objectives.

Puzzles are widely used to evaluate LLM reasoning because they provide controlled, rule-based environments with adjustable complexity and clearly verifiable solutions (Shojaee et al., 2025). Within the broader class of puzzles, mazes are particularly well-suited for studying spatial reasoning and navigational capabilities, as they simulate navigation as a constrained, logic-driven task that requires structured decision-making (Giadikiaroglou et al., 2024). For this reason, mazes are adopted as the primary testbed for evaluating spatial reasoning in LLMs in this thesis.
Maze solving and communicating the solution require several interrelated competencies: pathfinding as a form of sequential decision-making, relational analysis to infer how cells are positioned with respect to one another, and the translation of a discovered path into instructions within a specified Frame of Reference (FoR). For analytical clarity, the task is described in a manner that separates pathfinding from translation. This decomposition is borrowed from symbolic problem-solving and serves here as an analytical lens for diagnosis rather than a structural claim about how an LLM internally operates.

Existing benchmarks provide only a fragmented view of these three competencies, constituting the first of two key research gaps. For sequential decision-making, MazeEval (Einarsson, 2025) presented mazes from an agent's egocentric perspective and required the LLM to repeatedly choose the next step until completion, thereby testing its ability to track state (location + orientation) and decisions over time. Results from MazeEval and several other studies have shown that LLMs have limited ability for multi-step reasoning in spatial tasks (Einarsson, 2025; F. Li et al., 2024; Mane et al., 2023). For relational analysis, StepGame (Shi et al., 2022) supplied natural-language chain-like relational descriptions to LLMs and queried relative positions between the described entities. The authors concluded that reasoning over spatial relations is challenging for LLMs. For translation of a generated path into instructions, Premsri and Kordjamshidi (2025) examined how both reasoning and non-reasoning LLMs handle queries within (i.e. input and output are the same) and across (i.e. input and output are different) FoRs. The authors found that while individual FoRs are often processed correctly, transformations between them pose a significant challenge. Notably, each benchmark primarily targets a single competency of spatial reasoning. There exists no unified evaluation that requires an LLM to both find a path and translate that path into instructions in a specified FoR as a single task.

Additionally, a second research gap arises from the fragmentation of input spatial representations used to present mazes to LLMs. Prior work has used a wide range of representations that differ substantially in how spatial structure is encoded. In

this thesis, these variations are distinguished along two dimensions. The first dimension, termed *maze representation*, concerns the overall format used to describe the maze, consisting of three possible encodings: grid-based, graph-based, and visual. The second dimension, termed *wall encoding*, concerns how walls are represented within those maze representations, distinguishing between line-wall and occupancy grid.

Grid-based encodings describe the maze layout explicitly. For example, Deng et al. (2025) listed the locations of obstacles, start, and end, and presented them to an LLM in a structured JSON file, whereas Dao and Vu (2025) created a cell-wise textual description with tagged markup, which we will refer to as the "tagged per-cell" representation. Graph-based encodings instead abstract the maze into adjacency lists, representing connectivity without an explicit spatial layout. These are useful because unstructured input data can lead to reasoning hallucinations, which can be addressed by representing information as graph-structures (Fatemi et al., 2023). Graph-based encodings were employed in LLM maze-solving tasks by Ivanitskiy et al. (2023) and Spies et al. (2025). Visual representations, including images and ASCII art, resemble conventional maze depictions for humans. These representations are often used in LLM research to probe recognition or interpretation rather than navigational reasoning. For example, research by Hsu et al. (2025) supplied LLMs with maze images of varying levels of abstraction, to investigate which characteristics are critical for LLMs to recognize them as mazes. Other work investigated LLMs' ability to understand abstract visual inputs by means of classification tasks based on ASCII drawings (Luo et al., 2025). The authors explained that while ASCII art is based on textual characters, it is considered a visual medium because the meaning of the art is solely determined by the shape and layout of the characters.

Across these formats, maze structure itself can vary depending on the wall encodings. The two ways to encode walls are shown in Figure 1. Line-wall representations encode walls as infinitely thin borders between cells, resulting in compact structures where all information pertains directly to navigable space. These walls are expressed differently depending on the maze representation, for example by mentioning each cell's existing walls in grid-based representations, or mentioning traversable edges between nodes in adjacency lists. In contrast, occupancy grid representations use blocked cells instead of walls, and thus encode both free and blocked cells explicitly, producing larger structures that more closely resemble real-world environments but include redundant information. Converting an $n \times n$ line-wall maze to an occupancy grid results in a $(2n + 1) \times (2n + 1)$ grid, substantially increasing representational size.

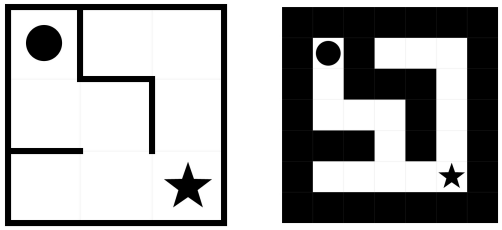
This methodological diversity complicates direct comparison between studies, as observed performance differences may stem from representational choices rather than inherent model capabilities, leaving it unclear whether reported limitations in spatial reasoning are artifacts of suboptimal spatial representations or reflect properties of the LLMs.

The choice of input representation shapes the type of reasoning the LLM must perform. Grid-based representations preserve spatial structure, but leave it to the LLM to infer connectivity information. In contrast, graph-based encodings emphasize connectivity by presenting mazes as graphs of connected nodes, thereby reducing the task to reasoning over established links while omitting an explicit spatial layout, requiring the LLM to interpret positions and relative configurations. Visual formats introduce an additional layer, where the LLM must first parse the depiction itself before any spatial reasoning can occur. These differences impose distinct cognitive demands on the LLM. As a result, the spatial representation determines which aspects of the navigation problem are made explicit and which must be inferred, motivating the treatment of spatial representation as a central experimental variable.

A similar consideration applies to the output FoR, which determines the reasoning depth required to express a solution. Assuming the input FoR is an allocentric view expressed in absolute coordinates, using the same output FoR would require the LLM to perform only pathfinding, as the solution can be specified without relational analysis or instruction generation. An allocentric output FoR expressed in grid-aligned absolute directions (up, down, left, right) increases the reasoning depth by requiring the LLM to derive spatial relations between steps and translate them into directions, while maintaining the same global perspective as the input. An egocentric (relative) output FoR imposes the highest reasoning demands, as it additionally requires orientation tracking and action generation from a dynamic, first-person perspective.

Representation and output FoR are thus separate but complementary factors: the former determines the cognitive load of reasoning about the input, while the latter governs the required reasoning depth of the task. Together, they reveal if and how LLMs

can solve spatial reasoning tasks.



(a) Example of a 3×3 line-wall maze

(b) Example of a 7×7 occupancy grid maze

Figure 1: Examples of a line-wall maze and its corresponding occupancy grid maze

1.1 Research objectives

To address the research gaps regarding LLM performance across the navigational pipeline, this thesis investigates optimal spatial representations and output FoRs. It specifically evaluates whether certain combinations of these factors enhance the ability of LLMs to solve navigational tasks. Accordingly, we address the following research question: *How do spatial representations and frames of reference influence Large Language Models' spatial reasoning capabilities, and which combinations are conducive to it?* To answer this question, we consider cross-model similarity, performance-efficiency trade-offs, and observable reasoning behavior. Subsequently, we divide the main research question into the following sub-questions:

1. Are there frames of reference or spatial representations that yield performance similarities in both reasoning and non-reasoning Large Language Models? Which variables are these and how do they achieve this?

2. Which output frame of reference and spatial representation yield high performance and a low number of output tokens?

3. Which reasoning behaviors are associated with high performance and are they influenced by the spatial representation of the input?

1.2 Hypotheses

The hypotheses are specific to the variables and skills mentioned in the research questions. The *Spatial representations* hypothesis is related to sub-questions 1, 2 and 3. The *Frame of reference* hypothesis is related to sub-questions 1 and 2. Finally, the *Spatial reasoning failures* hypothesis is related to the main research question and addresses the research gap concerning LLMs' capabilities across the complete spatial reasoning pipeline. No hypothesis is offered for sub-question 3. Due to the unmapped interactions between spatial representations and internal reasoning traces, this component of the study is exploratory, aimed at identifying behavioral markers that correlate with navigational success.

1.2.1 Spatial representations

We formulate two separate hypotheses regarding which spatial representation characteristics will yield better performance: structured, textual encodings will outperform visually based inputs, and compact representations will outperform verbose ones.

Textual, structured maze representations (both grid-based and graph-based) will yield better outcomes than visually grounded maze representations. Purely visual maze representations are not expected to be effective, even for multimodal LLMs, because the visual encoder of a multimodal LLM has lower fidelity for structured spatial information than tokenized text processing. Previous studies noted that multimodal LLMs' ability to perform structured, multi-step logical reasoning based on visual embeddings remains inconsistent across reasoning steps (Thawakar et al., 2025; R. Zhang et al., 2024). Maze representations that rely on implicit spatial layout, such as character-based visual arrangements (e.g. ASCII), require the LLM to infer spatial relations from the layout and shapes of the characters. Previous studies have shown that LLMs do not consistently demonstrate robust capabilities for interpreting such spatial layouts (Jia et al., 2025; Jiang et al., 2024; W. Li et al., 2024).

Compact encodings are expected to yield better performance than verbose encodings, because compact input data is thought to improve reasoning ability (Tan et al., 2024), as essential information is presented concisely and attention to irrelevant information is reduced, resulting in shorter reasoning paths, which may improve success rates (Struppek et al., 2025). Because occupancy grid mazes contain more cells than line-wall mazes, their descriptions are larger for most encodings. So, we expect occupancy grid mazes to yield lower performance.
Highly verbose textual encodings may overload the

LLM's context window making it hard to capture long-range dependencies (Y. Li et al., 2024; Zhou et al., 2025), introduce ambiguity, and increase the likelihood of misinterpretation (Jiang et al., 2024), thereby degrading performance. Maze representations that include redundant per-element details are therefore expected to dilute critical structural information and hinder effective reasoning.

1.2.2 Frame of reference

We hypothesize that consistency between input and output FoRs will lead to higher performance than conditions requiring FoR transformations. While Premsri and Kordjamshidi (2025) have shown this empirically, we expect this outcome due to the increased required reasoning depth that FoR transformations and continuous orientation tracking demand. Furthermore, Z. Zhang et al. (2025) explained that the inherent ambiguity of language describing spatial relations additionally contributes to task difficulty as a result of FoR transformations, which is also supported by Einarsson (2025), who highlighted the gap between "linguistic and spatial intelligence" in LLMs.

1.2.3 Spatial reasoning failures

We hypothesize that spatial reasoning failure can occur in two distinct stages of the navigational process: during pathfinding, or during relational analysis and FoR transformation. These stages were identified and discussed separately by Agrawal et al. (2025), who underscored that while LLMs are capable of finding valid paths, they have limited ability to reason about those paths. As stated in the introduction, this decomposition is observational; it does not imply that the LLM internally proceeds through these stages as separate steps. Furthermore, the failure modes are distinguished based on performance differences between FoRs. While the same ordering of FoR performance is expected as outlined in the previous hypothesis, it is interpreted here from a diagnostic perspective, where deviations and relative gaps between FoRs provide insight into underlying failure mechanisms.

Pathfinding failure The first phase where failure can occur is during pathfinding. This failure concerns the LLM's inability to maintain a coherent internal spatial representation. If pathfinding is the bottleneck, the failure occurs prior to any linguistic translation into navigation steps; consequently, we would observe consistently low performance across all output FoRs. Furthermore, testing across multiple maze sizes can help determine the nature of this limitation: a fundamental inability to model space would result in failure across all maze sizes, whereas a performance decay correlated with increasing maze size would indicate a capacity limit in maintaining larger mental maps. Notably, if certain input spatial representations yield significantly better results, it would suggest that the LLM's spatial mapping is not fundamentally absent but is instead non-robust and highly dependent on input spatial representations.

Relational analysis and FoR transformation failure The second phase where failure can occur is during the relational analysis and FoR transformation, during which a valid LLM-generated path is translated into navigational instructions. This failure mechanism is uniquely identifiable by a performance gap between output FoRs, as described and explained in the previous hypothesis about FoRs.

1.3 Approach

This thesis implements a controlled experimental setup that systematically varies input spatial representation, output FoR, and maze size. We evaluate two LLMs from the Google DeepMind Gemini family, using Gemini 2.5 Flash-Lite as a non-reasoning model, and Gemini 2.5 Pro as its reasoning counterpart. A factorial design is employed across three maze sizes, with 50 mazes per size, each represented in 11 different spatial representations. For each instance, both LLMs are prompted to produce solutions in three distinct output FoRs, enabling systematic comparison across conditions.

The task is realized using *perfect mazes*, which are a class of mazes where all cells are reachable via a single unique path without loops (Bellot et al., 2021). Prior research found that LLMs struggle to solve such mazes because cyclic graphs are more prevalent in LLM training data, and that their structural redundancy can simplify navigation by providing multiple routes to the same location (Fatemi et al., 2023). Furthermore, the acyclic nature of these mazes increases task difficulty because any incorrect turn may lead down a long, incorrect path, that will inevitably lead to a dead end, necessitating backtracking (Fatemi et al., 2023). Thus, perfect mazes minimize ambiguity in solution validity and reduce the likelihood that correct paths result from random guessing. Consequently, successful maze solving more accurately reflects an LLM's deliberate navigational reasoning. The complexity of this environment is further supported by the rapid growth in the number of unique perfect mazes as grid size increases. For grids of size $n \times n$, with $2 \leq n \leq 5$, the number of possible span-

ning trees is 4, 192, 100,352, and 557,568,000, respectively (Heinz, 2011).
Performance is evaluated using a custom "completion score" metric, the percentage of the path navigated correctly before the first error; alongside the mean number of output tokens as a proxy for efficiency. To our knowledge, this setup constitutes the first integrated evaluation that jointly varies spatial representation, FoR, and maze size within a single framework, directly addressing the previously identified gaps of fragmented benchmarks and spatial representations.

2 Methods

This section details the task setup, followed by model configurations, dataset specifications and prompt design. It ends with the setup of a keyword analysis and an explanation of the evaluation metrics.

2.1 Task description

The experiment was conducted by creating a dataset, querying the LLMs via their API, and storing both the final answers and the internal reasoning traces of the reasoning LLM. The core task for the LLMs was to determine a valid path from a specified start location to a goal location while respecting a fixed set of navigation rules. We employed three maze sizes (3×3, 6×6, and 15×15 for line-wall; mapped as explained in Section 1 to 7×7, 13×13, and 31×31 for occupancy grids), and used 50 mazes per maze size. Each of the 50 mazes per size was tested on one reasoning and one non-reasoning LLM across a combinatorial grid of 11 spatial representations and 3 output FoRs ($11 \times 3 = 33$ conditions per LLM). So, the total number of runs across 50 repetitions, 3 sizes, 33 conditions and 2 LLMs is $50 \times 3 \times 33 \times 2 = 9,900$. A complete overview of all variables is listed in Table 1. For all non-visual spatial representations, the input FoR was absolute coordinates; for JPG and ASCII the spatial layout is implicit.
The three FoRs that the LLMs were asked to use in their final answers are:
1. **Absolute coordinates:** an allocentric frame using absolute coordinates in *(row, column)* convention. Hereinafter referred to only as "coordinates" for brevity.
2. **Absolute directions:** an allocentric frame using absolute directions such as up, down, left, and right, to describe movements.
3. **Egocentric:** expresses actions relative to an agent's location and orientation, using directions such as forward, backward, left and right.

The first two frames are both allocentric frames, but differ in descriptor type.

Table 1: Overview of variables and all possible conditions. Three maze input representations will hereinafter only be referred to by their abbreviations, which are shown in brackets.

Variable name	Conditions
LLM	Gemini 2.5 Flash-Lite, Gemini 2.5 Pro
Maze representation	Adjacency List JSON (AL-JSON), Adjacency List Text (AL-Text), Tagged per-cell (Tagged), JSON, JPG, ASCII
FoR	coordinates, absolute directions, egocentric
Wall encoding	line-wall, occupancy grid
Maze size (line-wall/occ. grid)	3×3/7×7, 6×6/13×13, 15×15/31×31

2.2 Model selection and configuration

To ensure that the models could process the image inputs, we employed multimodal LLMs. We included one LLM with reasoning capabilities and one without. Additionally, the reasoning-enabled model was required to allow extraction of its internal reasoning trace for further analysis. The chosen models are Gemini 2.5 Pro, a reasoning multimodal LLM, and Gemini 2.5 Flash-Lite, a non-reasoning multimodal LLM.
The API settings are shown in Table 2. The default settings for Temperature, Top-P, Top-K, and candidate count were used (default in Nov. 2025, the starting time of testing). Gemini 2.5 Pro's default token output limit was used, so as not to limit the LLM's reasoning abilities. Gemini 2.5 Flash-Lite had a custom output token limit of 650 tokens. This limit was calculated to be approximately twice the number of tokens used in the longest solution in the dataset and therefore would not constrain the LLM's ability to provide a complete answer, but instead served as a safeguard to limit the monetary cost of the experiment.

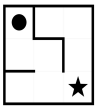
<table>
<tr><th></th><th>JSON</th><th>Adjacency List JSON (AL-JSON)</th><th>Adjacency List Text (AL-Text)</th><th>Tagged per-cell (Tagged)</th><th>ASCII</th><th>JPG</th></tr>
<tr><td>Line-Wall Maze</td><td><pre>{
  "size": {
    "columns": 3,
    "rows": 3
  },
  "start": [
    0,
    0
  ],
  "end": [
    2,
    2
  ],
  "grid": [
    [
      {
        "walls": {
          "N": true,
          "E": true,
          "S": false,
          "W": true
        }
      },</pre></td><td><pre>{
  "size": {
    "columns": 3,
    "rows": 3
  },
  "start": [
    0,
    0
  ],
  "end": [
    2,
    2
  ],
  "grid": [
    {
      "node": [
        0,
        0
      ],
      "neighbors": [
        [
          1,
          0
        ]
      ]
    },</pre></td><td><ADJLIST_START> (1,1) <–> (2,1) ;
... (5,4) <–> (5,5) <ADJLIST_END>
<ORIGIN_START>(1,1)<ORIGIN_END>
<TARGET_START>(5,5)<TARGET_END></td><td><|0-0|> <|uprightleft_wall|> <|origin|>
<|0-1|> <|updownleft_wall|> ...
<|2-2|> <|downright_wall|> <|target|></td><td>None</td><td>  </td></tr>
<tr><td>Occupancy Grid Maze</td><td><pre>{
  "size": {
    "columns": 7,
    "rows": 7
  },
  "start": [
    1,
    1
  ],
  "end": [
    5,
    5
  ],
  "grid": [
    [
      1,
      1,
      1,
      1,
      1,
      1,
      1
    ],</pre></td><td><pre>{
  "size": {
    "columns": 3,
    "rows": 3
  },
  "start": [
    0,
    0
  ],
  "end": [
    2,
    2
  ],
  "grid": [
    {
      "node": [
        0,
        0
      ],
      "neighbors": [
        [
          1,
          0
        ]
      ]
    },</pre></td><td><ADJLIST_START> (1,1) <–> (2,1) ;
... (5,4) <–> (5,5) <ADJLIST_END>
<ORIGIN_START>(1,1)<ORIGIN_END>
<TARGET_START>(5,5)<TARGET_END></td><td><|0-0|> <|occupied_wall|> ... <|1-1|> <|origin|>
<|1-2|> <|occupied_wall|> <|1-3|> <|blank|> ...
<|5-5|> <|target|> ... <|6-6|> <|occupied_wall|></td><td><pre>#######
#S#   #
#
#   # #
#
E#
#######</pre></td><td>  </td></tr>
</table>

Figure 2: Examples of each spatial representation. The top row lists the name of each maze representation, and three abbreviations to be used hereinafter. The left column differentiates between wall encodings.

Table 2: Used settings for each LLM. All values are default (Google Cloud, 2025a, 2025b), except for Flash-Lite's maximum output token budget.

Parameter	Gemini 2.5 Pro	Gemini 2.5 Flash-Lite
Temperature	1.0	1.0
Top-P	0.95	0.95
Top-K	64	64
Candidate count	1	1
Maximum output tokens	65,536 Incl. thinking	650

2.3 Dataset

This subsection details how maze generation was performed, and lists additional information about the 11 input spatial representations. The final dataset, results and all code are publicly available at https://github.com/vvannoesel/LLM_maze_solver (Van Noesel, 2025).

Maze generation Maze complexity can be quantified using various metrics, including solution path length, dead-end density, diversity, number of turns, and number of junctions (Buck, 2015; Gabrovšek, 2019; Kozlova et al., 2015; Vijaya et al., 2024). Depth-First Search (DFS) generates mazes with long solution paths and high diversity (Kozlova et al., 2015; Vijaya et al., 2024). Our objective was to evaluate the complete pipeline of spatial reasoning for navigation. The difficulty of individual choices that an LLM must make during the reasoning process is less critical than assessing whether the LLM can make and consistently track multiple consecutive decisions. Furthermore, it is important that the maze generation algorithm is capable of creating *perfect*

mazes. Therefore, we selected randomized DFS, an algorithm that generates acyclic mazes of moderate complexity, reflecting ordinary navigational decision-making tasks, while being simple to implement. This is also the maze generation algorithm that was used by Dao and Vu (2025), Einarsson (2025), Ivanitskiy et al. (2023), and Spies et al. (2025), who use mazes to query LLMs.

Spatial representations Dataset creation was performed by generating the mazes and converting them into the 11 spatial representations. The spatial representations and three corresponding abbreviations are defined in Figure 2, and more detailed examples are shown in Appendix B.1. For brevity, these abbreviations will be used exclusively when referring to specific input spatial representations.

All maze representations, except one, were derived from literature. To ensure uniformity across modalities, we introduce a JSON-formatted adjacency list as an additional maze representation. This format serves as a JSON counterpart to the textual graph encoding (AL-Text), analogous to how the JSON visual description complements the Tagged textual description of the maze layout. The proposed maze representation therefore provides consistency across both content type (grid-based vs. graph-based encoding), and format (text vs. JSON). A line-wall ASCII spatial representation is excluded because ASCII drawings inherently correspond to the dimensions of an occupancy grid.

For the maze images, JPG files were used. This image compression was not essential for obtaining the required image properties, so formats such as PNG may also be used. The images were 800x800 pixels, and each maze uses the color #939393 and a line thickness of 1 pixel for the gray grid lines.

Maze sizes The experiment was conducted on maze sizes 3×3, 6×6, and 15×15 for line-wall mazes, corresponding to 7×7, 13×13, and 31×31 for occupancy grid mazes. By evaluating performance across a range of maze sizes rather than a single size, we aim to distinguish between fundamental limitations and capacity-driven decay.

To observe performance similarities between LLMs across increasing maze sizes, a 3×3 maze was included to document the performance development of Gemini 2.5 Flash-Lite as maze size increased. The 6×6 maze was selected based on preliminary experiments indicating that this size marks the point at which the reasoning LLM's performance begins to degrade, while the non-reasoning LLM's performance approaches zero. Finally, the 15×15 maze was chosen as the largest maze size, as preliminary experiments indicated that performance at this size is declining but not yet zero for many spatial representations. We conducted 50 repetitions per maze size. For maze sizes 6×6/13×13 and 15×15/31×31 each of the 50 mazes is unique; for maze size 3×3/7×7 the dataset is comprised of 14 unique mazes.

Sample size The decision to run 50 repetitions per maze size was based on the expected 95% confidence interval half-widths. Assuming a normal distribution for prediction simplicity, for N = 50 and a t-distribution with 49 degrees of freedom, the 95% confidence interval half-width equals $t_{0.025}(49) \cdot \frac{SD}{\sqrt{N}} \approx 0.284 \cdot SD$. In the worst possible case ($SD = 50$ percentage points, occurring when 25 runs yield 0% and 25 runs yield 100%), the half-width is 14.2 percentage points. In a relatively low-variance case (e.g., 40 of 50 runs scoring 100% and 10 runs scoring 75%, giving a mean of 95% and $SD = 10.1$ percentage points), the half-width is 2.9 percentage points. In a high-variance case (e.g., scores uniformly distributed across [0%, 100%], giving a mean of 50% and $SD = 29.2$ percentage points), the half-width is 8.3 percentage points.

2.4 Prompt engineering and design

To prioritize the core experimental objectives, we excluded system-level instructions but included a brief role specification in each prompt. Given the constrained research duration, it was not feasible to validate a universal system prompt that could interact with all 33 task-specific prompts without introducing unintended side-effects such as bias in decision-making (Neumann et al., 2025). Consequently, the decision to omit system-level instructions was a precautionary measure.

For comparability across spatial representations, each prompt had the same structure. Each contained definitions of task, rules, required answer format, required output FoR, and an explanation of the spatial representation. In cases where an egocentric output FoR was requested, the prompt specified a south-facing starting orientation. An example is shown below, and all prompts are included in Appendix B.2.

"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *cols** maze is shown as an image, where thick black lines are impassable walls, thin light gray lines are passable cell borders, the circle is the start and the star is the end; the top-left

`corner is (0,0) in (row, col).`
`Instructions:`
`1. You can only move up, down, left, or right.`
`2. You cannot move diagonally or through walls, only from one cell to an adjacent cell.`
`3. Your output must be a single, comma-separated sequence of steps. For example: up, down, right, right, down.`
`4. Provide only the final list of moves in your response."`
* *`rows`* `x` *`cols`* is replaced by the maze's size during testing.

2.5 Keyword search

To investigate how LLMs approach maze solving, which reasoning behaviors are associated with high performance, and whether spatial representation influences these behaviors (sub-question 3), a keyword search was performed on the internal reasoning traces of Gemini 2.5 Pro. This required identifying relevant reasoning behavior categories and associated keywords. These were derived from a subset of the internal reasoning traces (line-wall AL-Text, 15×15 runs 1–10, coordinates output FoR), which was selected because it contains variation in performance and thus reasoning behaviors. A complete list of identified categories, explanations, and keywords can be found in Appendix D.3, Table 7.

The categories and keywords were identified in collaboration with Gemini 3 Pro (Google Cloud, 2026) as an assistive tool (all settings left to the default values, listed in Table 6, Appendix D.3). First, Gemini 3 Pro proposed initial reasoning behavior categories, after which we jointly identified candidate keywords per category from the dataset subset. Each keyword occurrence was then manually reviewed in context, and keywords appearing in unrelated contexts were removed.

For validation, a second subset was used (15×15 run 1, all spatial representations, coordinates output FoR). Each reasoning trace was manually categorized, and the automated keyword search was applied to the same subset. The automated results were compared to the manual classification, and the keyword list was amended accordingly, resulting in the final set of keywords and categories.

During keyword selection, the focus was on words used during reasoning behaviors, rather than words used to describe them. For example, the term "Breadth-First Search" was omitted because the LLM often mentions it without actually using the algorithm. The keyword list also included explicit exclusion phrases. For example, the keyword "confident" was paired with the exclusion phrase "not confident".

The keyword search was restricted to four of the six maze sizes in the Gemini 2.5 Pro dataset, excluding the 3×3 and 7×7 conditions, as their short reasoning traces and high completion scores introduced noise and obscured correlations. The search was automated using a custom Python script. The number of mentions within each reasoning trace was not considered; instead, a binary indicator of presence or absence was used.

2.6 Evaluation metrics

We evaluated LLM performance through two distinct primary metrics. First, we defined completion score as a measure of navigational accuracy; this is calculated as the number of consecutive correct steps generated from the start of the response until the first error (or the target cell, whichever comes first), normalized by the length of the ground-truth solution path. The formula can be found in Appendix C, Equation 1. This yields a percentage that reflects how far along the solution path the LLM reached before making the first mistake (i.e. 50% indicates that the LLM successfully reached 50% of the solution path before making the first mistake). In the coordinates output FoR, each "step" corresponds to a coordinate. The solution sequence includes the coordinates of both the start and end cells. As a result, if the start cell is correctly identified but the first movement away from this cell is incorrect, the response may still receive a score above 0%. In other FoRs, this case would yield a score of 0%.

Second, the mean number of output tokens was utilized as a proxy for efficiency, encompassing the total count of output tokens, including internal reasoning tokens in the case of Gemini 2.5 Pro.

Even though the occupancy grid mazes are larger than the line-wall mazes, they encode the same underlying maze structure and thus represent the same task in a different manner. The completion score metric is a reflection of how far into the maze the LLM has successfully navigated. So, because both wall encodings represent the same maze, and the metric is proportional to the solution path length, their completion scores and number of output tokens can be compared and ranked against each other.

Lastly, the metric for the keyword search was the percentage of runs in which each reasoning behavior occurs.

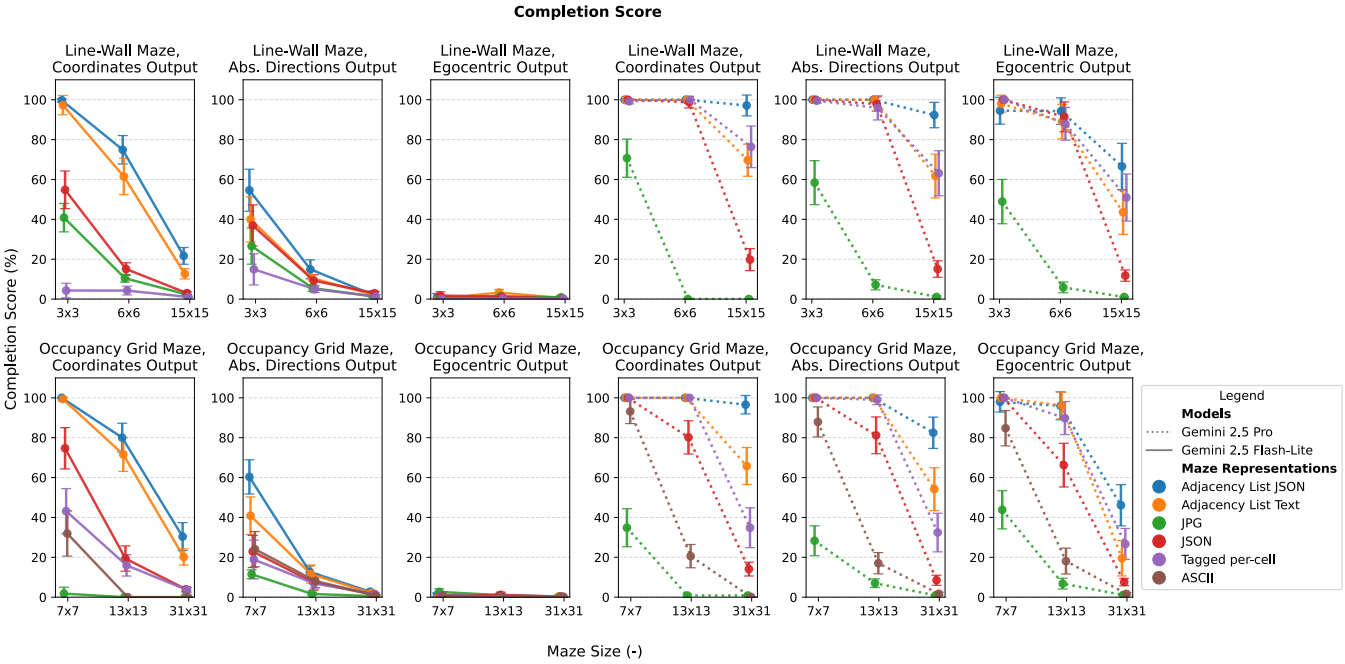


Figure 3: Average completion score achieved by each spatial representation, for each maze size and output FoR. The error bars show the bounds of the 95% confidence interval of that datapoint in percentage points.

3 Results

Each subsection presents results corresponding to a research sub-question, following the order in which the questions were posed.

3.1 Completion score

Figure 3 shows the completion score achieved by each spatial representation, output FoR, and LLM, addressing sub-question 1. This figure includes the size of the 95% confidence interval for each datapoint (see Appendix D.1 for the full overview of 95% confidence interval half-widths). 67% of conditions have a 95% confidence interval half-width below 5.0 percentage points (pp), and the maximum observed half-width is 11.8 pp. The half-widths differ somewhat by maze size: the 6×6/13×13 mazes are mostly well-behaved (only 1 condition above 10.0 pp), while the remaining maze sizes have 9 (15×15/31×31) and 8 (3×3/7×7) conditions with half-widths of 10.0 pp or higher. This occurs in 3×3/7×7 mazes because small mazes produce more bimodal all-or-nothing score distributions. In 15×15/31×31 mazes, this is largely the result of inconsistent performance with the line-wall Tagged and line-wall AL-Text spatial representations.

Although Gemini 2.5 Flash-Lite inherently cannot execute a formal reasoning phase and Gemini 2.5 Pro can, the top-performing spatial representations are similar for both LLMs. Gemini 2.5 Flash-Lite's highest performance is yielded by the AL-JSON maze representation, followed by AL-Text, across maze sizes and FoRs. For the line-wall mazes, JSON comes in third place. For the occupancy-grid mazes, coordinates output, JSON is also in third place.
Similarly, in Gemini 2.5 Pro, AL-JSON yields the highest performance across FoRs, sizes and wall encodings. For second and third place, AL-Text and Tagged alternate with each other, depending on FoR and wall encoding.
Another similarity between the LLMs, is each FoR's relative performance. The coordinates FoR yields the highest overall completion scores and the smallest confidence intervals. The absolute directions FoR results in noticeably lower performance for Gemini 2.5 Flash-Lite. For Gemini 2.5 Pro, performance in the absolute directions FoR remains similar to the coordinates FoR, although completion scores decline slightly for the 15×15 and 31×31 mazes. Finally, the egocentric output FoR shows the worst outcomes overall, with completion scores approaching 0% for the non-reasoning LLM, and completion scores below 100% from maze size 3×3/7×7 onward for the

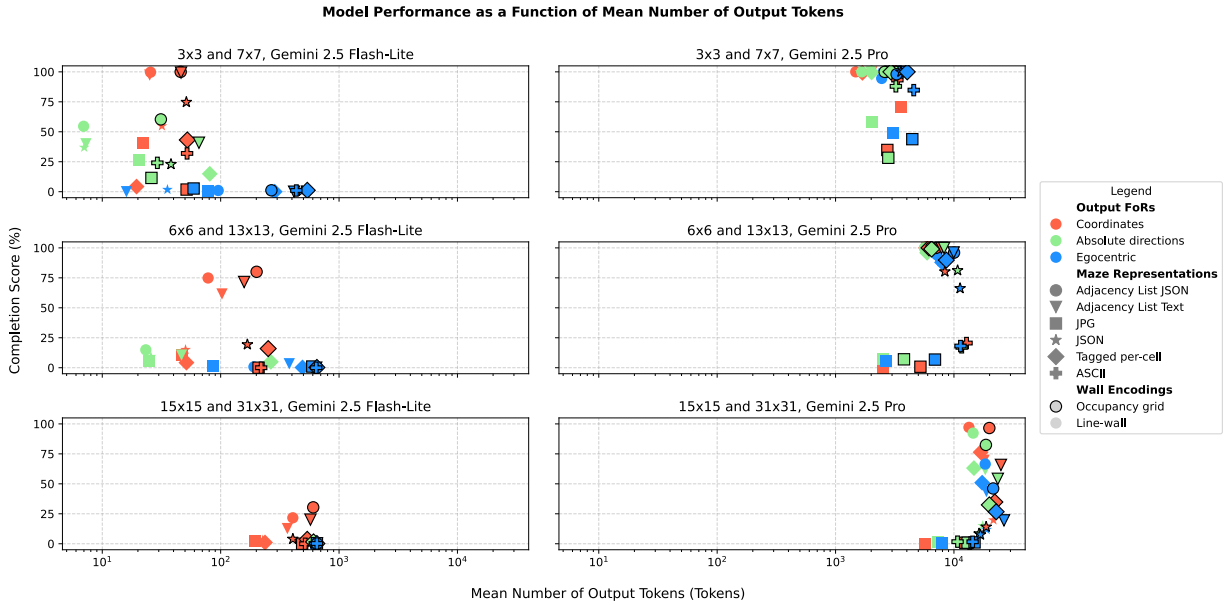


Figure 4: Completion score as a function of mean number of output tokens per task, displayed for each LLM (left: Gemini 2.5 Flash-Lite, right: Gemini 2.5 Pro), and maze size (top: 3×3/7×7, middle: 6×6/13×13, bottom: 15×15/31×31). For Gemini 2.5 Pro, the output token-count includes the internal reasoning trace.

reasoning LLM. Gemini 2.5 Pro also shows larger confidence intervals in the egocentric FoR, compared to the others.

3.2 Output tokens

To answer sub-question 2, Figure 4 shows the cost of each LLM's answer in terms of the mean number of output tokens, alongside the average completion score achieved. The count includes all output tokens; for Gemini 2.5 Flash-Lite this is only the final answer, and for Gemini 2.5 Pro this is internal reasoning trace and final answer combined. Overall, the mean number of output tokens varies strongly with output FoR for Gemini 2.5 Flash-Lite, while it scales primarily with maze size for Gemini 2.5 Pro.

The left column shows Gemini 2.5 Flash-Lite's results. For this LLM, the coordinates FoR exhibits the lowest variation in mean number of output tokens (horizontal spread) and the highest outcomes, which is consistently achieved by the two graph-based spatial representations. The absolute directions output FoR shows a larger spread in mean number of output tokens and lower performance than the coordinates FoR. The egocentric output FoR yields the highest mean number of output tokens and the lowest performance. The convergence towards low completion scores and high mean number of output tokens that is shown in the bottom-left subplot, is a result of syntax failure, which will be discussed further in section 4.3.

The right column of Figure 4 shows the reasoning LLM's performance. Here, the mean number of output tokens is primarily a function of maze size. The top-three spatial representations that are both high-performing and low in token output (top-left cluster) are listed in Tables 3 and 4. These are distinguished based on the ratio between completion score and mean number of output tokens of all datapoints in the top-left clusters. Table 3 (Gemini 2.5 Flash-Lite) excludes maze size 15×15/31×31 due to the high occurrence of syntax failure, which causes low completion scores and excessively high token output.

Notably, both line-wall AL-JSON and occupancy grid AL-JSON are among the most high performing and output token-efficient for both LLMs. Additionally, these spatial representations are the most expensive in terms of input tokens. Line-wall JSON is ranked 2^{nd} and 3^{rd} in Table 4, and has the second-highest input token cost. A complete overview of input tokens per spatial representation is shown in Appendix D.2, Figure 8.

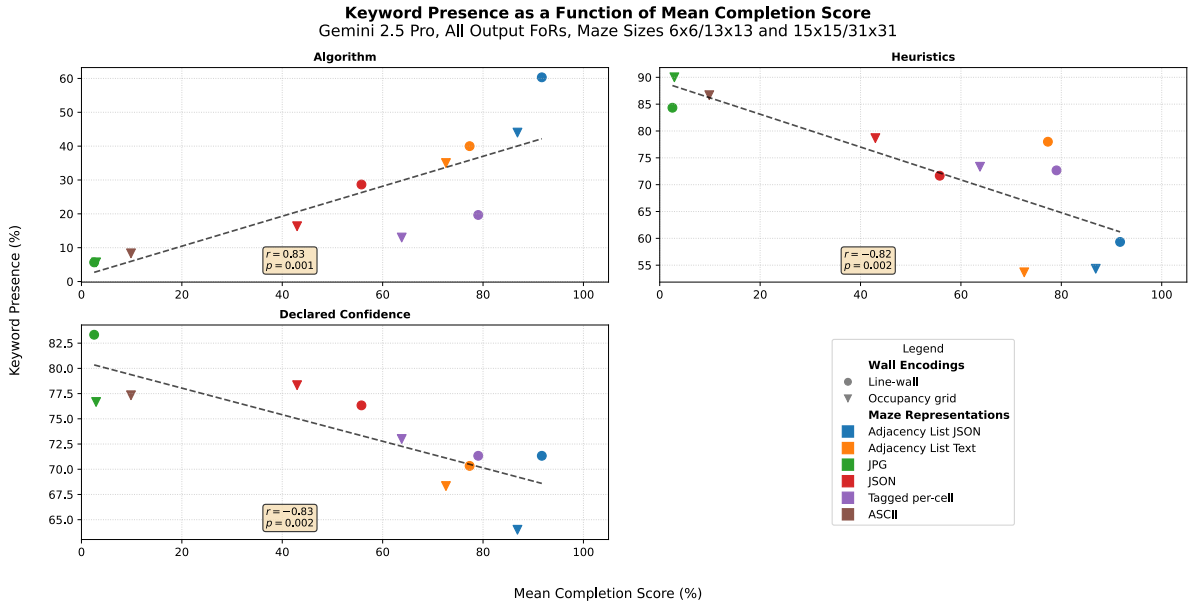


Figure 5: Keyword presence as a function of mean completion score. The data is averaged over all Gemini 2.5 Pro's reasoning traces for maze sizes 6×6/13×13 and 15×15/31×31. "r" is the correlation coefficient, with "p" as its associated statistical significance. **Note that the scale on the y-axes is different for each subplot.**

Table 3: Gemini 2.5 Flash-Lite's top three input spatial representations that achieve the highest completion score at the lowest mean number of output tokens per maze size, ranked by completion-score-to-mean-output-token ratio. Maze size 15×15/31×31 is excluded due to high occurrence of syntax failure.

Rank	3×3/ 7×7	6×6/ 13×13
1	Line-wall AL-JSON coordinates	Line-wall AL-JSON coordinates
2	Line-wall AL-Text coordinates	Line-wall AL-Text coordinates
3	Occ. grid AL-JSON coordinates	Occ. grid AL-Text coordinates

Table 4: Gemini 2.5 Pro's top three input spatial representations that achieve the highest completion score at the lowest mean number of output tokens (thinking + final answer), per maze size, ranked by completion-score-to-mean-output-token ratio.
*"abs. dir." is shorthand for "absolute directions".

Rank	3×3/ 7×7	6×6/ 13×13	15×15/ 31×31
1	Line-wall AL-JSON coordinates	Line-wall AL-JSON abs. dir.	Line-wall AL-JSON coordinates
2	Line-wall JSON coordinates	Line-wall AL-JSON coordinates	Line-wall AL-JSON abs. dir.
3	Line-wall AL-JSON abs. dir.	Line-wall JSON abs. dir.	Occ. grid AL-JSON coordinates

3.3 Keyword search

A keyword search is used to answer sub-question 3. This analysis reveals that there are multiple behaviors associated with performance inside the reasoning traces of Gemini 2.5 Pro. The behavior categories that have a statistically significant correlation ($p \leq 0.05$) to completion score are shown in Table 5, accompanied by their explanations and keywords. Figure 5 shows the correlations between the keywords occurring for each spatial representation throughout the dataset (excluding 3×3 and 7×7) and the achieved completion scores. This figure shows that there are two solving strategies that the LLM uses: heuristics and algorithmic solving. Use of graph-solving algorithms is positively correlated

Table 5: Keyword search categories that have statistically significant correlations with performance, their definitions, and keywords. Some keywords are excluded from this table for brevity, see Table 7 in Appendix D.3 for the complete list of words.
*"down-right" has many variations such as "down and right", which are left out of this table.

Category and Explanation	Keywords
Algorithm naming: Explicitly describes using Breadth-First or Depth-First Search, or other formal algorithm	visited, queue, queuing, dequeuing, enqueuing, parent
Heuristics: Uses heuristics such as right-hand rule, going down and right, or "visual" search	down-right*, southeast, eyes, right-hand, wall-following, heuristic, visually tracing, visual
Declared confidence: Claims success with vague or unsupported language	I made it, I've made it, confident, it works, paid off, solved, correct, complete, this is the solution, "easy, right?", no problem, **Exclusion**: not confident

with completion score. Predominant use of heuristics to solve the maze, and unfounded declarations of confidence are negatively correlated with completion score.

The "Declared Confidence" subplot shows that unsupported declarations of confidence are universally present. Similarly, the "Heuristics" subplot shows that use of heuristics to solve the mazes is present in all spatial representations, though in varying amounts. Finally, the use of algorithms to solve the mazes is not as prevalent; JPG, ASCII, occupancy grid JSON, and occupancy grid Tagged use algorithms in less than 20% of all instances.

The figure also reveals a pattern in the order of placement of each maze representation. The ordering along the trend-lines from left to right is: JPG, ASCII, JSON, Tagged and AL-Text, and finally AL-JSON. The order of the pattern in the "Algorithms" subplot is mirrored, compared to the pattern in the "Heuristics" subplot. This means that maze representations where heuristics are used most, algorithms are used least, and vice versa.

4 Discussion

The discussion systematically addresses the research questions by evaluating how the results support or reject each hypothesis. Section 4.1 answers sub-question 1, Section 4.2 addresses sub-question 3, Section 4.3 discusses sub-question 2, and finally Section 4.4 discusses the third hypothesis (from Section 1.2.3).

4.1 Performance similarities

Performance similarities across LLMs illuminate the variables' strengths and weaknesses, and indicate whether their performance could be generalizable, answering sub-question 1. Each variable is discussed based on its associated hypothesis.

4.1.1 Spatial representation hypothesis

This hypothesis set two separate expectations: structured, textual encodings would outperform visually based ones, and concise inputs would yield better outcomes than verbose ones. Our results prove the first expectation and reject the second one.

The data clearly supports the expectation that visually grounded maze representations would yield the lowest performance, as ASCII and JPG yield the worst completion scores. Data show that the LLMs cannot infer the basic maze layout from these maze representations, which is the leading cause for their low performance. Completion scores in the coordinates output FoR for the 6×6/13×13 and 15×15/31×31 maze sizes show that Gemini 2.5 Pro fails to correctly identify the starting coordinates in 78% and 55% of JPG and ASCII cases, respectively. Gemini 2.5 Flash-Lite fails in 50% and 100% of these cases, respectively. This indicates that poor performance on visual inputs is often caused by a fundamental misunderstanding of the maze representation, rendering subsequent search ineffective.
The failure of the ASCII spatial representation aligns with documented difficulties of LLMs in interpreting spatial semantic layouts (Jia et al., 2025; Jiang et al., 2024; W. Li et al., 2024). Failures in the JPG spatial representation appear to stem

from multimodal parsing limitations. Gemini 2.5 Pro's reasoning traces suggest that the model relies on coarse global descriptions, such as approximate start and goal locations or overall wall structure, rather than detailed per-cell information required for maze solving. Additionally, our image design may contribute to ambiguity. Grid lines in the JPG images are one pixel wide, which may be interpreted as artifacts during multimodal parsing. JPG compression may also reduce visual clarity. Future work should therefore compare performance using uncompressed images and wider grid lines to evaluate the impact of these design choices.
When considering the completion scores from Gemini 2.5 Flash-Lite, graph-based spatial representations clearly outperform grid-based spatial representations, which was also observed in the analysis of other non-reasoning LLMs by Agrawal et al. (2025). This distinction cannot be made in Gemini 2.5 Pro's completion scores. Although AL-JSON consistently outperforms any other, AL-Text does not consistently outperform Tagged, while JSON consistently performs the same as, or worse than these three, for the reasoning model.

The expectation that verbosity yields lower completion scores is rejected by the similar completion scores between wall encodings and the high number of input tokens for AL-JSON. The results show high completion scores for both line-wall and occupancy grid mazes, even though there are more reachable cells in the occupancy grid mazes than there are in the line-wall maze, meaning occupancy grid mazes are inherently more verbose than line-wall mazes. In Gemini 2.5 Flash-Lite, AL-Text and AL-JSON show comparable completion scores between wall encodings. In Gemini 2.5 Pro's results, differences only start showing at maze size 15×15/31×31 for the AL-JSON and Tagged spatial representations, with AL-JSON even yielding similar completion scores at 15×15/31×31 in the two allocentric FoRs. Although the occupancy grid AL-JSON spatial representation is concise in the sense that it only lists reachable neighboring cells and thus does not contain information about redundant cells, it is also verbose: AL-JSON has the highest number of input tokens out of all maze representations, for both wall encodings (see Figure 8 in Appendix D.2). The verbosity stems from describing the neighbor-relationships in detail, and mentioning each connection between neighbors twice (once from each cell).

4.1.2 Frame of reference hypothesis

Across models we see that outputs in the coordinates FoR yield the highest performance, while egocentric outputs consistently yield the lowest performance. This similarity aligns with our hypothesis, based on Premsri and Kordjamshidi (2025), as LLMs achieve their best performance when input and output FoRs are the same. Additionally, the task demands associated with changing FoRs require greater reasoning depth and may increase the likelihood of errors.

In summary, structured inputs outperform visually grounded ones, while compact inputs do not necessarily outperform verbose ones. To answer sub-question 1 and address research gap 2 concerning fragmented spatial representations, only AL-JSON works well for both LLMs, FoRs, maze sizes, and wall encodings. Furthermore, the coordinates output FoR yields the highest completion scores for both LLMs. However, what these results do not yet make clear, is what happens within the LLMs that causes such differences when spatial representations and FoRs are changed. The next section will explore how the reasoning LLM leverages the input spatial representations, through keyword search analysis of Gemini 2.5 Pro's reasoning traces.

4.2 Keyword search

The keyword search provides insight into the relationship between spatial representations and reasoning behavior, which helps us theorize why some yield better results. Our findings are twofold (sub-question 3): spatial representations appear to influence which solving strategy is selected, and the best performing spatial representation can accommodate use of multiple strategies.

The emergent ordering of maze representations in Figure 5 shows that those with higher heuristic usage exhibit lower algorithm usage. While correlational, this pattern is consistent with the maze representation influencing the LLM's solving strategy. The highest completion scores are yielded by maze representations that rely on both algorithms and heuristics as a solving strategy. Thus, neither strategy is inherently good or bad, but exclusive dependence on heuristics yields the worst outcomes. The graph-based maze representations AL-JSON and AL-Text more frequently lead to algorithmic reasoning than others. These formats are unambigu-

ous and explicitly define neighboring cells, which can be directly used as inputs in graph-search algorithms such as DFS and BFS.
The remaining four maze representations more frequently lead to heuristic reasoning. These maze representations lack explicit relational structure, requiring the LLM to infer spatial relationships before applying formal algorithms. Heuristic approaches such as visual search, movement toward the target, or wall-following require less preprocessing, and we theorize that they therefore occur more frequently.
The Tagged and JSON maze representations yield much higher completion scores than JPG and ASCII, even though they use the same solving strategy. This is due to the LLMs' misunderstanding of the inputs, rather than the strategy.

These findings suggest that the maze representation determines the solving strategy, and thus the final performance. So LLMs' spatial reasoning abilities are highly dependent on the input spatial representation. More broadly, this indicates that LLM performance on spatial problems may generally benefit from spatial representations that explicitly encode relationships between entities, rather than requiring LLMs to infer connectivity from implicit or visual layouts.

4.3 Output tokens

The results for the mean number of output tokens directly answer sub-question 2, and provide supporting information for sub-question 1. Furthermore, this section explains Gemini 2.5 Flash-Lite's convergence towards low completion scores and high mean number of output tokens.

The most output token-efficient and high performing FoRs and spatial representations vary per LLM (sub-question 2). In the case of Gemini 2.5 Flash-Lite, the coordinates output FoR and both the AL-JSON and AL-Text representations are the clear winners. In the case of Gemini 2.5 Pro, both the coordinates and absolute directions FoRs, as well as the AL-JSON and JSON representations accomplish high performance at a low mean number of output tokens. What stands out from these results, is that the AL-JSON maze representation is identified as output token-efficient for both LLMs (sub-question 2), even though it uses the highest number of input tokens. This shows that compact inputs do not always lead to efficient reasoning, contrary to our hypothesis in Section 1.2.1. Possibly the explicit and structured input decreases the need for LLMs to perform additional parsing or preprocessing before task solving, as explained in the keyword search analysis.

Furthermore, the bottom-left subplot in Figure 4 shows an overall convergence towards low completion score and a high mean number of output tokens. This results from an increased frequency of syntax failure, referred to as the "repeat curse" (Yao et al., 2025), who argue that it is a self-reinforced repetition of tokens or phrases as a consequence of feature activation, mentioning that more challenging questions will yield a more pronounced repeat curse. This failure manifests as an output consisting of random repetitions of possible actions (e.g. "right, right, right") until the token limit is reached, often resulting in poor performance, as shown in Figure 6.

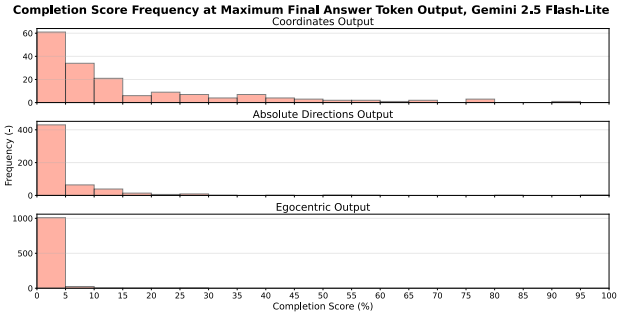


Figure 6: The frequency of the completion scores that are achieved at each occurrence of maximum token output in Gemini 2.5 Flash-Lite, counted over all maze sizes. **Note that the scale on the y-axes is different for each subplot**.

4.4 Spatial reasoning failures

This section evaluates the third hypothesis, which consists of two sub-claims which will each be discussed separately. First, Section 4.4.1 will address the remaining question of why performance differences occur across FoRs. Then, Section 4.4.2 will discuss how the effect of increasing maze sizes is used to distinguish between different failure types within the pathfinding phase. The distinction between spatial reasoning phases serves in this thesis as an analytical lens rather than a claim about how the LLM internally solves the task. Together, these analyses contribute to answering the main research question and address the research gap about fragmented evaluation methods for LLMs on navigational tasks.

4.4.1 Frame of reference

This section uses the observed differences in FoR performance to show that failures primarily occur during relational analysis and FoR transformation, and explains why FoR transformations are challenging for LLMs.

Hypotheses 2 (Section 1.2.2) and 3 (Section 1.2.3) predict the same overall FoR performance pattern: the coordinates FoR will outperform all others, and the egocentric FoR will yield the lowest results. However, hypothesis 2 does so observationally, while hypothesis 3 uses the pattern diagnostically to identify in which phase of the spatial reasoning pipeline failure occurs, providing insight into the LLMs' spatial reasoning abilities and addressing the first research gap. Section 4.1.2 mentions that the data supports the prediction made in both hypotheses. Hypothesis 3 explains that this pattern points to failures occurring during relational analysis and frame transformations rather than during pathfinding.
Z. Zhang et al. (2025) explain why relational analysis and transformations between FoRs are challenging for LLMs. The authors attribute difficulties with FoR transformations to the inherent ambiguity of spatial language. Spatial terms are context-dependent, vary across individuals, and require transformations that depend on situational context (Z. Zhang et al., 2025). Consequently, switching between frames of reference introduces ambiguity and increases reasoning complexity. Z. Zhang et al. (2025) further reported that LLMs tend to prefer absolute directions over egocentric outputs, while showing limited robustness with both. Our results also show a decline in consistency in the egocentric FoR, as indicated by the larger 95% confidence intervals.
In coordinate-based outputs, the path and final output are the same, requiring no transformations, and no added ambiguity. In contrast, egocentric outputs require iterative reasoning, where the LLM must track both position and orientation while generating stepwise navigational instructions, which introduces two potential sources of failure. First, ambiguity in spatial language may increase the difficulty of reasoning about relative directions. Second, egocentric outputs require iterative state updates, which increase reasoning depth.

The performance differences between FoRs show that performance degrades specifically when reasoning over paths is required, rather than during pathfinding itself, demonstrating that FoR transformations introduce additional reasoning complexity that exposes LLMs' main limitations in spatial reasoning.

4.4.2 Maze size

The hypothesis in Section 1.2.3 states that decreasing performance at larger maze sizes is indicative of reaching a limit in the LLMs' capacity to maintain a mental map, whereas consistently poor performance is indicative of inherent model limitations. Upon this distinction we can decide whether each model is capable of spatial reasoning.
Gemini 2.5 Flash-Lite does not have a separate reasoning phase in which pathfinding can be completed before perspective transformation. Rather, these actions happen interleaved within a single generation pass. This LLM's completion scores are low for most spatial representations at mazes of size 3×3/7×7, and with increasing maze size, completion scores continue to decay while the mean number of output tokens increases. As stated in the hypothesis, low overall completion scores are indicative of an inherent model limitation to perform the task. Furthermore, we observe the highest completion scores in the coordinates FoR, and considerably lower completion scores as FoR transformation demands increase. This suggests that the model is capable of pathfinding in small environments when no FoR transformations are required, given that a structured spatial representation is provided that explicitly encodes relational data. Thus, to partially answer our main research question: the model is not capable of spatial reasoning because its ability to maintain a mental map is limited, and it is not capable of translating between FoRs. Given a small maze and graph-based input, it can do pathfinding.
Gemini 2.5 Pro's completion scores show that each spatial representation experiences performance degradation with increasing maze size, regardless of output FoR. The reasoning trace length increases with maze size (Figure 4), which suggests that performance degradation is linked to increased reasoning complexity and context requirements. The observed decline in performance with increasing maze size cannot be conclusively explained based on the current results and metrics. Several plausible factors may contribute to this trend, including training on lower-order graphs (Agrawal et al., 2025), degradation in reasoning performance as input length grows while remaining within the model's context window, or limitations imposed by exceeding that window altogether. As these explanations cannot be discerned

with the present experimental design, further research is required to determine the underlying cause. Performance decay with increased maze size, together with the findings in the previous section, suggest that this LLM has the ability to model space, up to a certain maze size and depending on the spatial representation. Beyond this limit, the LLM is constrained in maintaining and translating spatial relationships over multiple reasoning steps. To expand our answer to the main research question: the reasoning LLM is capable of spatial reasoning depending on the spatial representation and type of FoR transformation, up to a certain maze size. The model shows robust spatial reasoning when using the Line-wall AL-JSON spatial representation, as it yields relatively high completion scores even after tasks of simple FoR transformations (coordinates to absolute directions).

4.5 Limitations

Completion score The completion score penalizes the whole answer after the first mistake, even if everything after the mistake is correct. This amplifies small errors (for example, one rogue step inserted early into an otherwise correct path yields a low score), and makes it impossible for LLMs to include backtracking behavior in their final answer, even though finding the shortest/most efficient path was not the task objective. Furthermore, this obscures error types as small detours and illegal moves are indistinguishable, and when used on various maze sizes it may give the impression that performance is decaying even when the absolute number of correct consecutive steps until the first mistake remains constant. However, this metric also has advantages such as measuring step-by-step fidelity, detecting breakdowns, offering more insight than binary pass/fail metrics for the task, and allowing for comparison.
Furthermore, the completion score does not reflect the 53 instances where the LLMs' answers continue after successfully solving the maze. These cases achieve a completion score of 100%, although they show a lack of ability to track when the task is completed. These cases are indicated in the results posted on our GitHub page.
Output token limit Gemini 2.5 Flash-Lite's token limit of 650 was implemented after the third run, when it became apparent that in some cases, the LLM generates a nonsensical output that continues up until the maximum output token limit is reached. The manual token limit of 650 was calculated based on the tokenization of the ground truth answer of a 31×31 maze, using Gemini's documentation (Google AI, 2025). It has become apparent that a miscalculation occurred due to inaccurate reported values in this documentation, as some ground truth answers exceed 650 tokens. However, in every case, the first mistake occurred well before 650 tokens. So, while the token limit did not impact the completion scores, it did impact the mean number of output tokens (Figure 4). This limits our ability to analyze the LLM's reasoning effort, and identify output token-efficient representations in the mazes of size 15×15/31×31. The extent of this limitation can be uncovered by rerunning the tests without token limits, and observing if completion scores remain the same or increase. If they increase, our token limit impacted all facets of this research.
JPG generation The JPG input images have grid lines with a width of 1 pixel. Considering that the 800×800 pixels image will likely be resized by the multimodal LLM, a line thinner than 1 pixel after preprocessing could be interpreted by the LLM as an artifact, potentially affecting the LLM's ability to correctly perceive spatial structure. Future work could mitigate this issue by increasing grid line thickness.
Keyword search Keyword searches are inherently limited, as they do not always capture the intended meaning or context of a word in a dataset. For example, the terms "eyes" and "visited" may appear in general descriptions of exploration, but can also be used in either heuristic ("eyes") or algorithmic ("visited") solution strategies, making their interpretation ambiguous. These words remained in the list of keywords because they did not appear within a descriptive context in either the identification or validation subsets. Furthermore, the keywords and categories were identified and validated based on two limited subsets of the whole dataset. Keyword identification is based on 10 runs of a single spatial representation, and validation was performed on 1 run of all 11 spatial representations, both within a single FoR. Therefore, it is possible that some keywords or phrases are missing, leading to missed correlations between LLM reasoning behavior and completion scores. We tried to mitigate these effects by averaging over the data, and manual validation, although this can introduce new biases.
Lastly, the keyword set is tied to the specific start and end coordinates of the evaluated maze sizes. As a result, extending the benchmark to additional maze sizes or different start and end positions would require adapting the keywords accordingly.
Dataset We intended to generate 50 unique mazes for each size. However, for the 3×3/7×7 mazes, our code produced only 14 of the 192 possible unique mazes, due to non-uniform sampling of the DFS al-

gorithm. This effect is mitigated by size in the larger mazes. Importantly, the generated set of 14 unique mazes covers all possible solution paths (shown in Appendix E, Figure 10), and two of the solution paths each appear twice in the dataset, which are indicated in green in the figure. The difference between the generated mazes and the remaining 178 mazes, is the placement of a single wall in the dead-end paths, thus not introducing new or more challenging solution paths.

For Gemini 2.5 Flash-Lite, the impact appears minimal, as repeated runs of the same maze still produced varied completion scores and numbers of output tokens. The impact on Gemini 2.5 Pro also appears minimal: spatial representations that perform well on 3×3/7×7 mazes continue to perform well on larger mazes, suggesting that performance on smaller mazes is not inflated by recurring instances.

5 Conclusion

This study addressed two gaps in LLM spatial reasoning research: fragmented benchmarking of navigational competencies, and fragmented spatial input representations. To close these gaps, this study performed an integrated evaluation containing all facets of spatial reasoning for navigation, and systematically compared 11 input spatial representations.

The AL-JSON maze representation in both wall encodings consistently achieved the highest completion scores across models, FoRs, and maze sizes. The performance ordering of FoRs exposes a limitation for spatial reasoning, as decreased performance aligns with the increasing reasoning depth required by these FoRs.

Across both models, line-wall AL-JSON with coordinate outputs was the most consistently performant and output token-efficient configuration.

Analysis of Gemini 2.5 Pro's reasoning behavior showed that higher completion scores were associated with solving strategies combining algorithms and heuristics, while unsupported declarations of confidence correlated with lower performance. Spatial representations influenced the applied solving strategies, although the exact mechanism remains unclear. AL-JSON was the only representation that showed a near-equal presence of heuristic and algorithmic solving, suggesting versatility across approaches.

Overall, spatial reasoning capability differed substantially between models. Gemini 2.5 Flash-Lite did not demonstrate spatial reasoning capabilities, but did exhibit pathfinding ability restricted to small maze sizes, and graph-based input maze representations. Notably, wall encoding did not impact this performance. In contrast, Gemini 2.5 Pro demonstrated spatial reasoning capability across multiple spatial representations and FoRs, although maze size still constrained performance. Remarkably, AL-JSON achieved high performance by all measures and across models, despite having the largest input token count, showing that compact inputs are not necessarily optimal.

Overall, both FoR and spatial representation greatly influence LLM spatial reasoning ability, affecting performance, output efficiency, and reasoning behavior. The findings suggest that the structured, graph-based line-wall AL-JSON spatial representation combined with coordinate-based outputs is generally most conducive to spatial reasoning, although effectiveness remains dependent on model capability and maze size.

5.1 Future work

The discussion mentions three possible directions for future work. First, the underlying cause for the observed decay in Gemini 2.5 Pro's completion scores cannot be determined with the current experimental design, requiring further research.

Second, future work should focus on re-investigating LLMs' understanding of the JPG representation, due to the limitations that have been mentioned in the Discussion and Limitations.

Third, the consistent performance of the AL-JSON maze representation raises questions about the mechanisms underlying its effectiveness, and generalization to other LLM families. Although AL-Text encodes the same information, the graph-based JSON format systematically outperformed its textual counterpart. Future work should investigate whether this advantage stems from structural regularity, tokenization effects, or differences in how models parse various formats. Understanding these mechanisms would clarify whether the observed performance gains are specific to spatial reasoning tasks or reflect a broader preference for structured graph representations, and if these advantages exist outside the Google DeepMind Gemini family.

A related direction is to evaluate whether AL-JSON spatial representations generalize beyond spatial reasoning. Many domains such as social networks, human behavior modeling, and organizational structures can be represented as graphs. If AL-JSON representations consistently improve reasoning performance across these domains, this would suggest that graph-structured inputs are broadly conducive to increased reasoning depth. Systematic cross-

domain experiments would be required to assess the generality of this spatial representation's advantage.

Another avenue for future work concerns spatial representation transformation. If LLMs can reliably convert arbitrary inputs into AL-JSON representations, the importance of the original input format may be reduced. In this setting, LLMs could first translate inputs into a structured spatial representation before performing reasoning. This approach resembles Chain-of-Thought prompting (Wei et al., 2022), where intermediate reasoning steps improve performance. Future work should examine whether representation transformation functions similarly, including whether it improves performance, increases reasoning stability, or introduces additional sources of error.
Together, these directions aim to better understand the role of structured spatial representations in LLM reasoning and to determine whether representation transformation can serve as a general strategy for improving spatial and graph-based reasoning tasks.

References

Agrawal, P., Vasania, S., & Tan, C. (2025). Can LLMs Perform Structured Graph Reasoning Tasks? *Pattern Recognition*, 287–308. https://doi.org/10.1007/978-3-031-78172-8_19

Bellot, V., Cautrès, M., Favreau, J.-M., Gonzalez-Thauvin, M., Lafourcade, P., Le Cornec, K., Mosnier, B., & Rivière-Wekstein, S. (2021). How to generate perfect mazes? *Information Sciences*, *572*, 444–459. https://doi.org/10.1016/j.ins.2021.03.022

Buck, J. (2015). *Mazes for Programmers : Code Your Own Twisty Little Passages*. The Pragmatic Bookshelf.

Dao, A., & Vu, D. B. (2025). AlphaMaze: Enhancing Large Language Models' Spatial Intelligence via GRPO. https://arxiv.org/abs/2502.14669

Deng, H., Zhang, H., Ou, J., & Feng, C. (2025). Can LLM Be a Good Path Planner Based on Prompt Engineering? Mitigating the Hallucination for Path Planning. https://arxiv.org/abs/2408.13184

Doula, A., Mühlhäuser, M., & Guinea, A. S. (2025). SafePath: Conformal Prediction for Safe LLM-Based Autonomous Navigation. https://arxiv.org/abs/2505.09427

Einarsson, H. (2025). MazeEval: A Benchmark for Testing Sequential Decision-Making in Language Models. https://arxiv.org/abs/2507.20395

Fatemi, B., Halcrow, J., & Perozzi, B. (2023). Talk like a Graph: Encoding Graphs for Large Language Models. https://arxiv.org/abs/2310.04560

Fung, P., Bachrach, Y., Celikyilmaz, A., Chaudhuri, K., Chen, D., Chung, W., Dupoux, E., Gong, H., Jégou, H., Lazaric, A., Majumdar, A., Madotto, A., Meier, F., Metze, F., Morency, L.-P., Moutakanni, T., Pino, J., Terver, B., Tighe, J., … Malik, J. (2025). Embodied AI Agents: Modeling the World. https://arxiv.org/abs/2506.22355

Gabrovšek, P. (2019). Analysis of Maze Generating Algorithms. *IPSI Transactions on Internet Research*, *15*(1), 23–30. https://ipsitransactions.org/journals/papers/tir/2019jan/p5.pdf

Giadikiaroglou, P., Lymperaiou, M., Filandrianos, G., & Stamou, G. (2024). Puzzle Solving using Reasoning of Large Language Models: A Survey. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 11574–11591. https://doi.org/10.18653/v1/2024.emnlp-main.646

Google AI. (2025). https://ai.google.dev/gemini-api/docs/tokens?lang=python

Google Cloud. (2025a, June). https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro

Google Cloud. (2025b, July). https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/gemini/2-5-flash-lite

Google Cloud. (2026, January). https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/3-pro

Heinz, A. P. (2011). The On-Line Encyclopedia of Integer Sequences [Contribution to sequence A007341]. https://oeis.org/A007341

Hsu, J., Mao, J., Tenenbaum, J. B., Goodman, N. D., & Wu, J. (2025). What Makes a Maze Look Like a Maze? https://arxiv.org/abs/2409.08202

Ivanitskiy, M. I., Spies, A. F., Räuker, T., Corlouer, G., Mathwin, C., Quirke, L., Rager, C., Shah, R., Valentine, D., Behn, C. D., Inoue, K., & Fung, S. W. (2023). Structured World Representations in Maze-Solving Transformers. https://arxiv.org/abs/2312.02566

Jia, Q., Yue, X., Huang, S., Qin, Z., Liu, Y., Lin, B. Y., You, Y., & Zhai, G. (2025). ASCIIEval: Benchmarking Models' Visual Percep-

tion in Text Strings via ASCII Art. https://arxiv.org/abs/2410.01733

Jiang, F., Xu, Z., Niu, L., Xiang, Z., Ramasubramanian, B., Li, B., & Poovendran, R. (2024). ArtPrompt: ASCII Art-based Jailbreak Attacks against Aligned LLMSs. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15157–15173. https://doi.org/10.18653/v1/2024.acl-long.809

Kozlova, A., Brown, J. A., & Reading, E. (2015). Examination of representational expression in maze generation algorithms. *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, 532–533. https://doi.org/10.1109/CIG.2015.7317902

Latif, E. (2024). 3P-LLM: Probabilistic Path Planning using Large Language Model for Autonomous Robot Navigation. https://arxiv.org/abs/2403.18778

Li, F., Hogg, D. C., & Cohn, A. G. (2024). Advancing Spatial Reasoning in Large Language Models: An In-Depth Evaluation and Enhancement Using the StepGame Benchmark. https://arxiv.org/abs/2401.03991

Li, W., Duan, M., An, D., & Shao, Y. (2024). Large Language Models Understand Layout. https://arxiv.org/abs/2407.05750

Li, Y., Li, Z., Wang, P., Li, J., Sun, X., Cheng, H., & Yu, J. X. (2024). A Survey of Graph Meets Large Language Model: Progress and Future Directions. https://arxiv.org/abs/2311.12399

Lin, J., Gao, H., Feng, X., Xu, R., Wang, C., Zhang, M., Guo, L., & Xu, S. (2025). Advances in Embodied Navigation Using Large Language Models: A Survey. https://arxiv.org/abs/2311.00530

Luo, K., Fu, M., Peguero, J., Malik, H., Patil, A., Lin, J., Overborg, M. V., Sarmiento, R., & Zhu, K. (2025). ASCIIBench: Evaluating Language-Model-Based Understanding of Visually-Oriented Text. https://arxiv.org/abs/2512.04125

Mane, D., Harne, R., Pol, T., Asthagi, R., Shine, S., & Zope, B. (2023). An Extensive Comparative Analysis on Different Maze Generation Algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(2s), 37–47. https://ijisae.org/index.php/IJISAE/article/view/3557

Neumann, A., Kirsten, E., Zafar, M. B., & Singh, J. (2025). Position is Power: System Prompts as a Mechanism of Bias in Large Language Models (LLMs). *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, 573–598. https://doi.org/10.1145/3715275.3732038

Premsri, T., & Kordjamshidi, P. (2025). FoREST: Frame of Reference Evaluation in Spatial Reasoning Tasks. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 34977–35003. https://doi.org/10.18653/v1/2025.emnlp-main.1772

Shi, Z., Zhang, Q., & Lipani, A. (2022). StepGame: A New Benchmark for Robust Multi-Hop Spatial Reasoning in Texts. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(10), 11321–11329. https://doi.org/10.1609/aaai.v36i10.21383

Shojaee, P., Mirzadeh, I., Alizadeh, K., Horton, M., Bengio, S., & Farajtabar, M. (2025). The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity. https://arxiv.org/abs/2506.06941

Spies, A. F., Edwards, W., Ivanitskiy, M. I., Skapars, A., Räuker, T., Inoue, K., Russo, A., & Shanahan, M. (2025). Transformers Use Causal World Models in Maze-Solving Tasks. https://arxiv.org/abs/2412.11867

Struppek, L., Hintersdorf, D., Struppek, H., Neider, D., & Kersting, K. (2025). Focused Chain-of-Thought: Efficient LLM Reasoning via Structured Input Information. https://arxiv.org/abs/2511.22176

Tan, X., Wang, H., Qiu, X., Cheng, Y., Xu, Y., Chu, W., & Qi, Y. (2024). Struct-X: Enhancing Large Language Models Reasoning with Structured Data. https://arxiv.org/abs/2407.12522

Tariq, M. T., Hussain, Y., & Wang, C. (2025). Robust mobile robot path planning via llm-based dynamic waypoint generation. *Expert Systems with Applications*, *282*, 127600. https://doi.org/10.1016/j.eswa.2025.127600

Thawakar, O., Dissanayake, D., More, K., Thawkar, R., Heakl, A., Ahsan, N., Li, Y., Zumri, M., Lahoud, J., Anwer, R. M., Cholakkal, H., Laptev, I., Shah, M., Khan, F. S., & Khan, S. (2025). LlamaV-o1: Rethinking Step-by-step Visual Reasoning in LLMs. https://arxiv.org/abs/2501.06186

Van Noesel, V. (2025). *LLM maze solver* [Accessed: 31-03-2026]. https://github.com/vvannoesel/LLM_maze_solver

Vijaya, J., Gopu, A., Vinayak, K. V. S. G., Singh, T. J., & Malhotra, K. (2024). Analysis of

Maze Generation Algorithms. *2024 5th International Conference on Innovative Trends in Information Technology (ICITIIT)*, 1–6. https://doi.org/10.1109/ICITIIT61487.2024.10580178

Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter brian, b., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, *35*, 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

Yao, J., Yang, S., Xu, J., Hu, L., Li, M., & Wang, D. (2025). Understanding the Repeat Curse in Large Language Models from a Feature Perspective. *Findings of the Association for Computational Linguistics: ACL 2025*, 7787–7815. https://doi.org/10.18653/v1/2025.findings-acl.406

Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.-W., Gao, P., & Li, H. (2024). MathVerse: Does Your Multi-modal LLM Truly See the Diagrams in Visual Math Problems? https://arxiv.org/abs/2403.14624

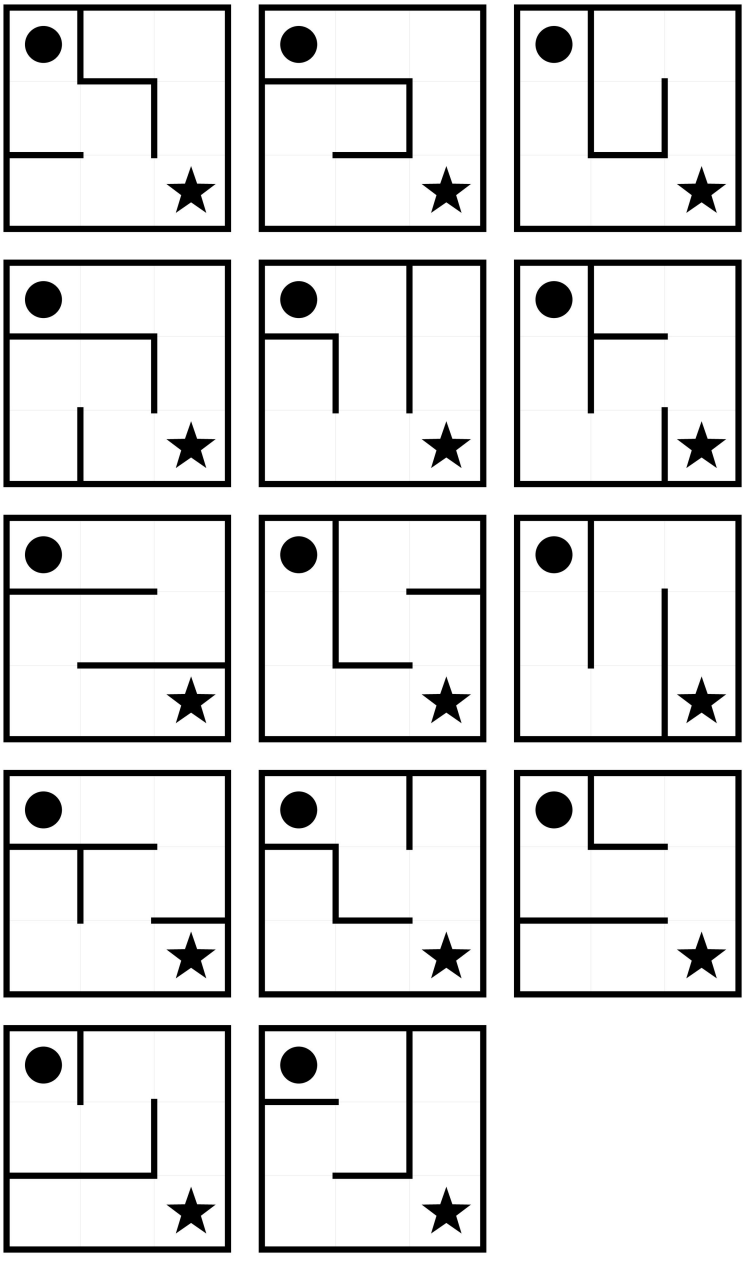
Zhang, Z., Hu, F., Lee, J., Shi, F., Kordjamshidi, P., Chai, J., & Ma, Z. (2025). Do Vision-Language Models Represent Space and How? Evaluating Spatial Frame of Reference Under Ambiguities. https://arxiv.org/abs/2410.17385

Zhou, H., Du, J., Zhou, C., Yang, C., Xiao, Y., Xie, Y., & Huang, X. (2025). Each Graph is a New Language: Graph Learning with LLMs. *Findings of the Association for Computational Linguistics: ACL 2025*, 17548–17559. https://doi.org/10.18653/v1/2025.findings-acl.902

A Maze Dataset

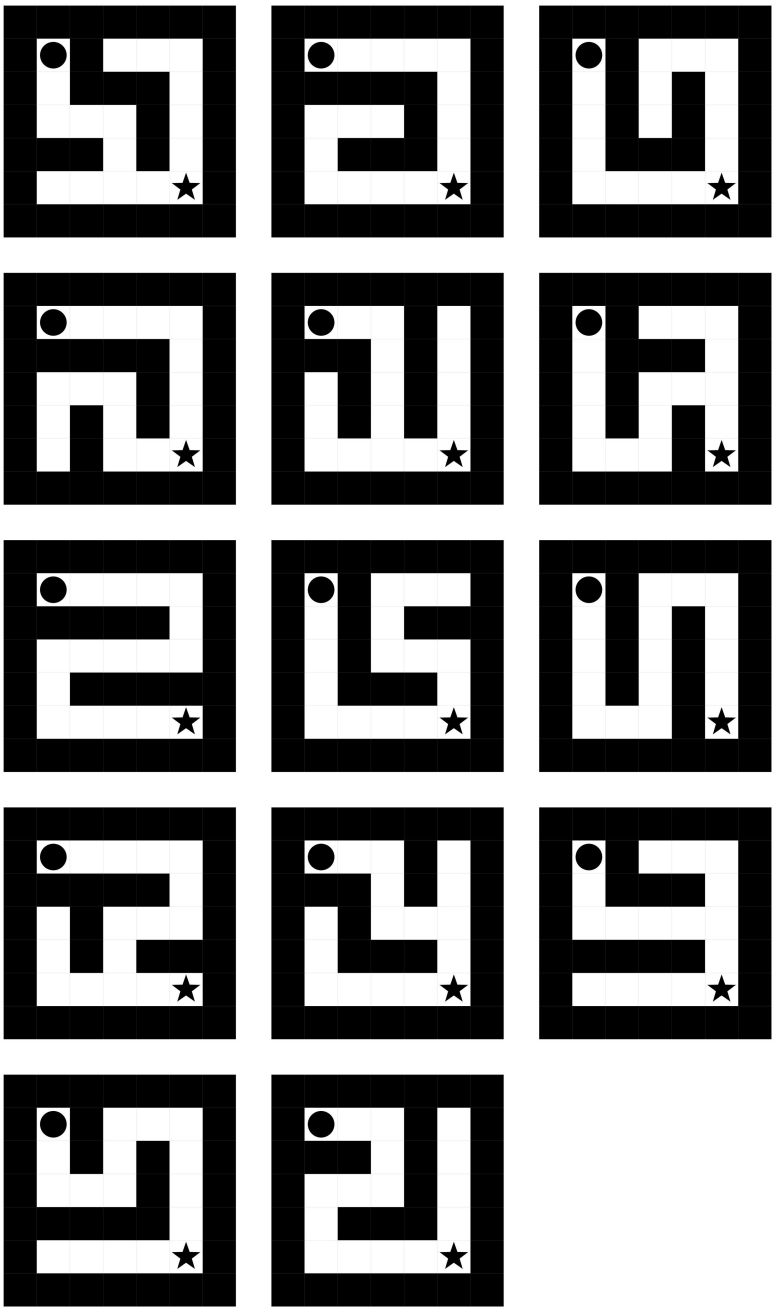
A.1 3×3 Mazes

All 14 unique 3×3 line-wall mazes that appear in the dataset.



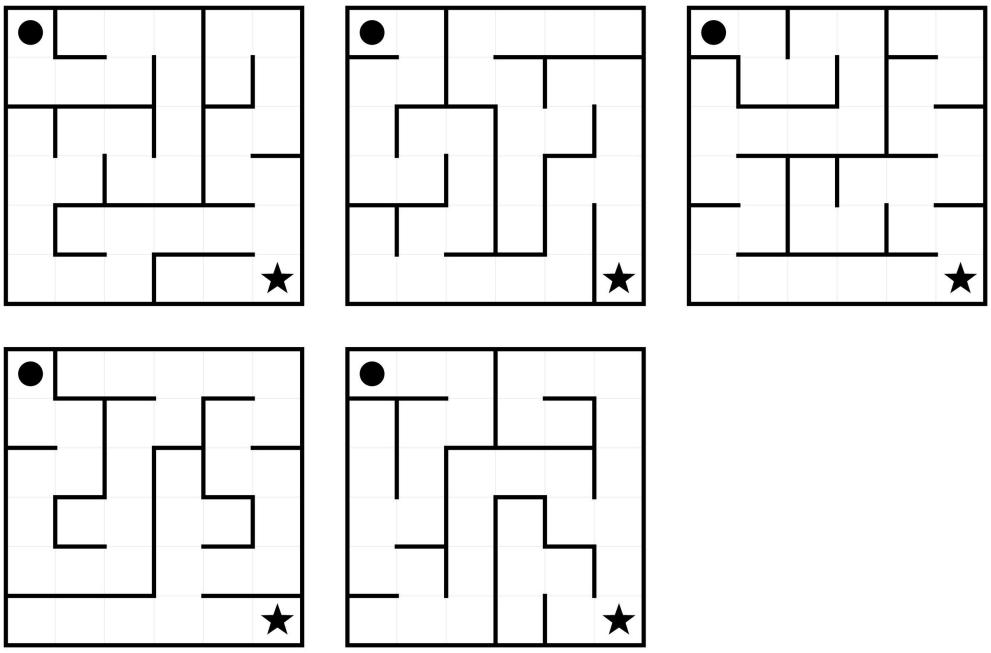
A.2 7×7 Mazes

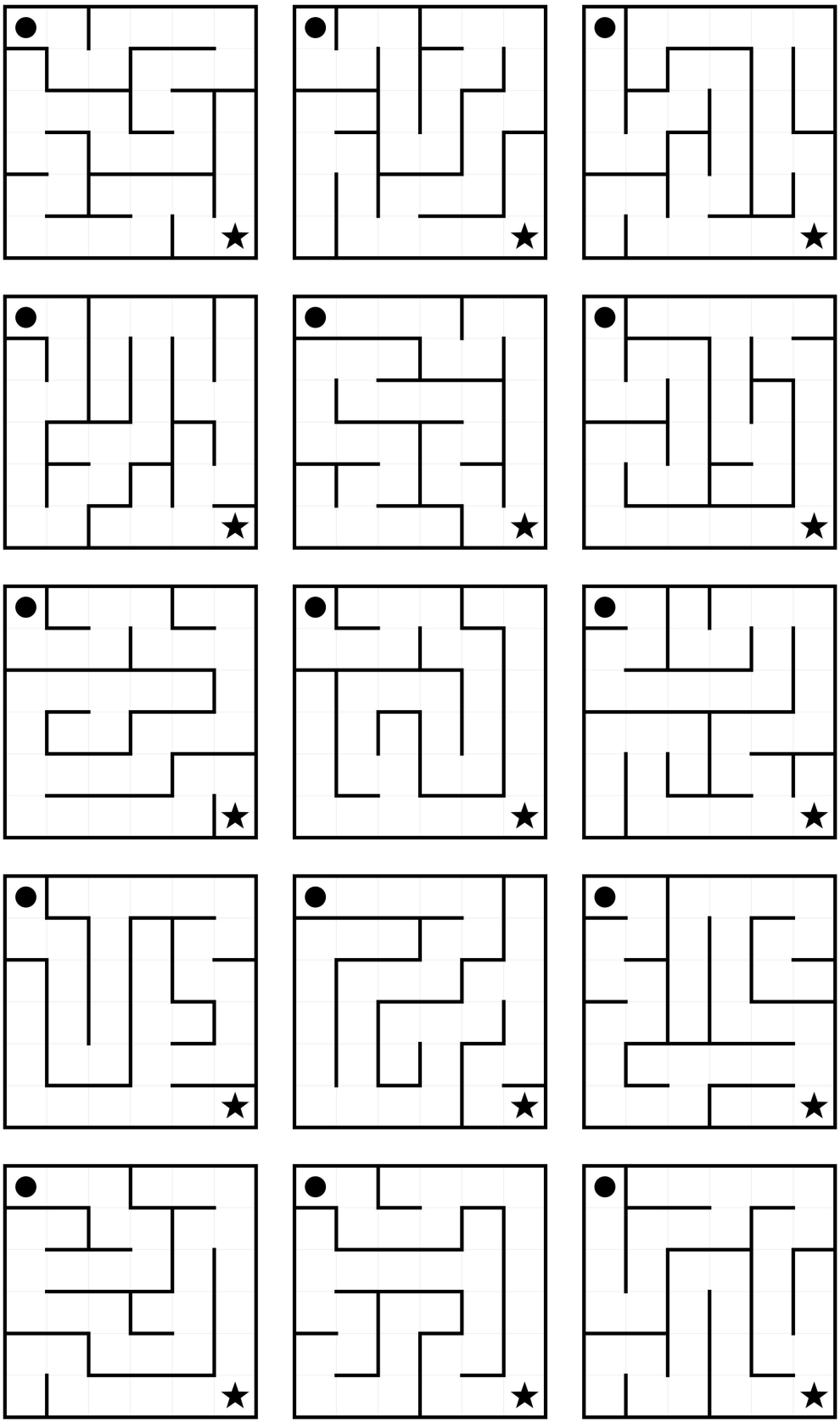
All 14 unique 7×7 occupancy grid mazes that appear in the dataset.

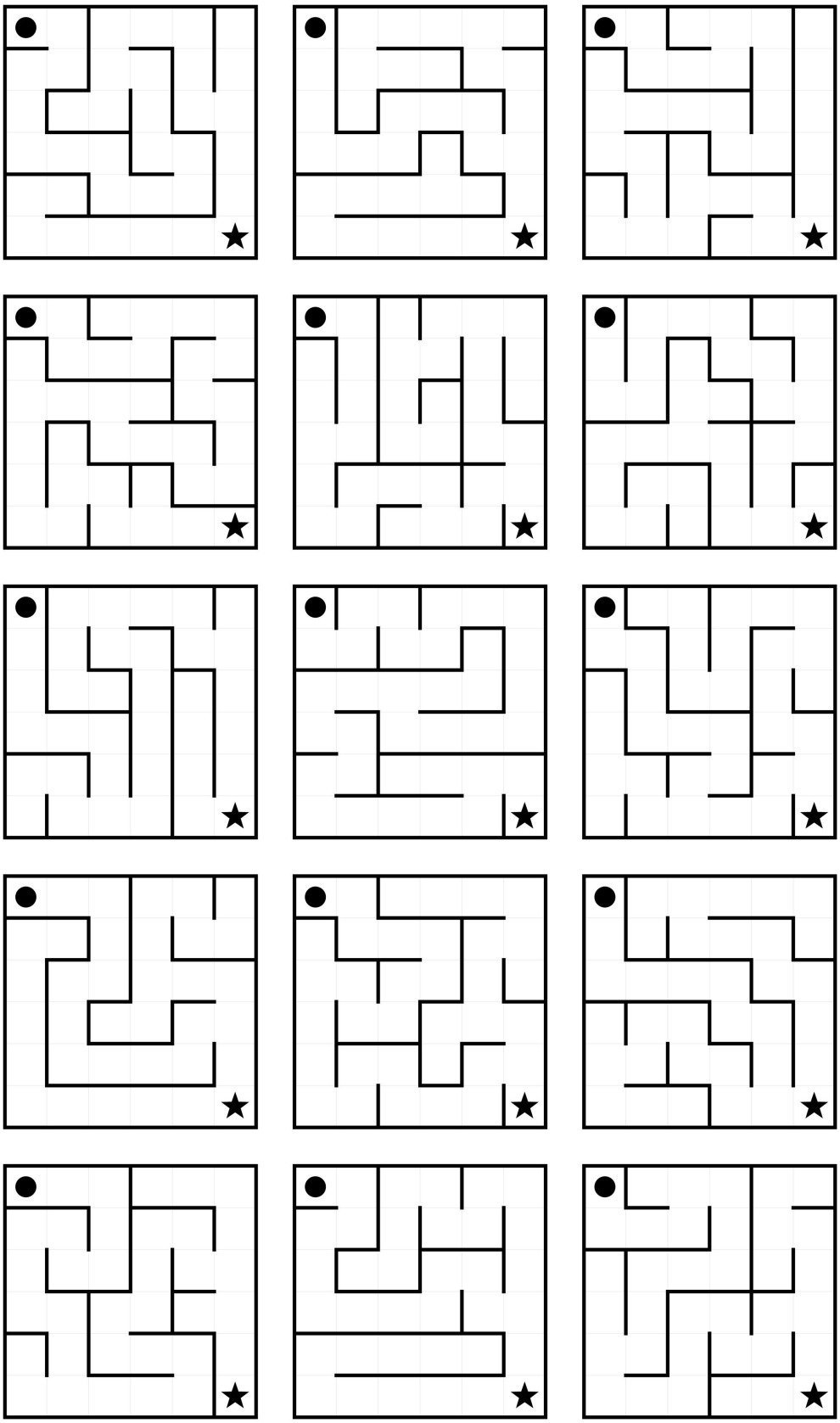


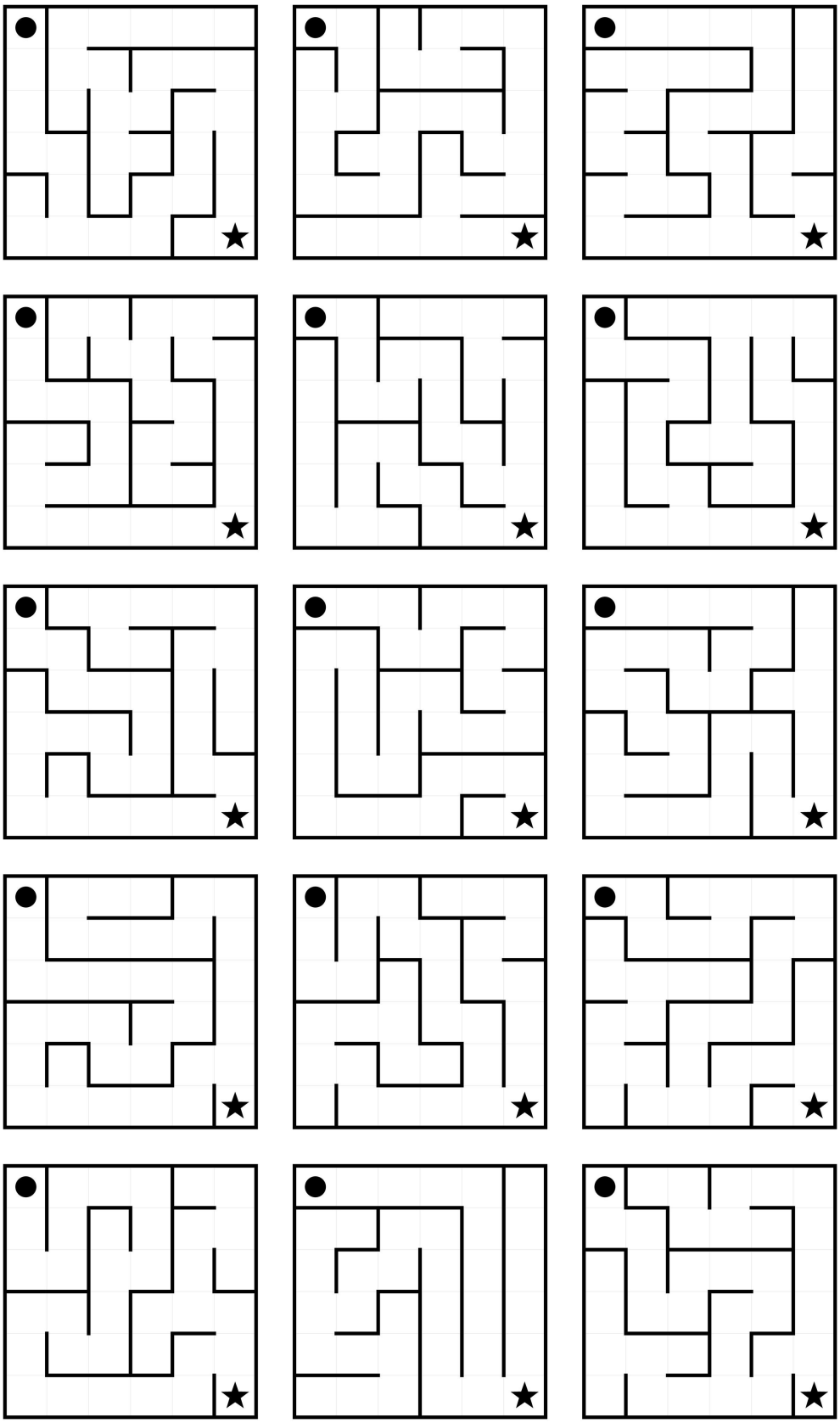
A.3 6×6 Mazes

All 50 unique 6×6 line-wall mazes that appear in the dataset.



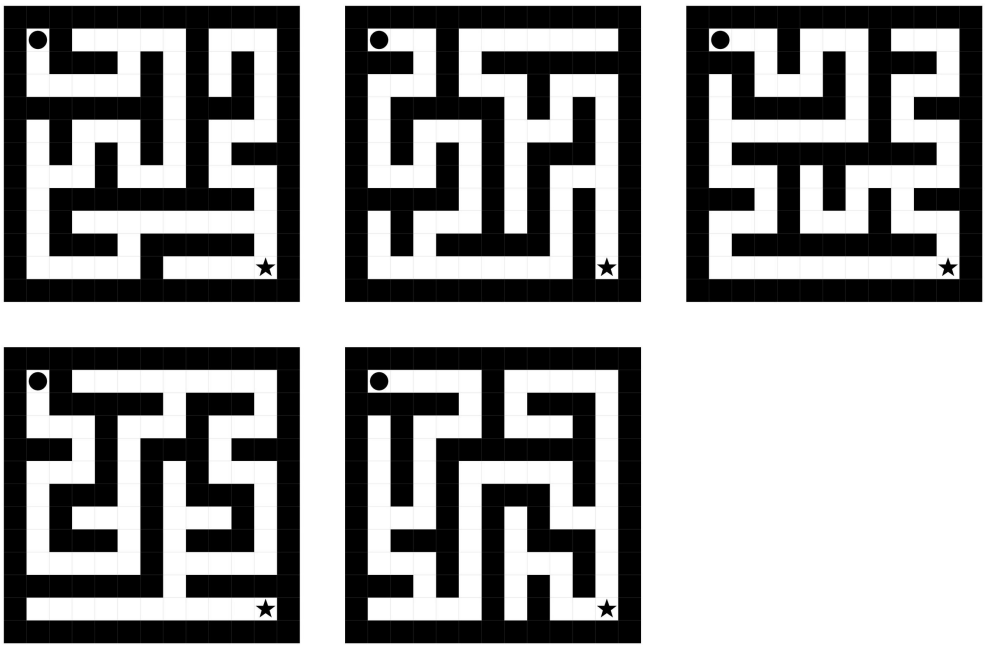


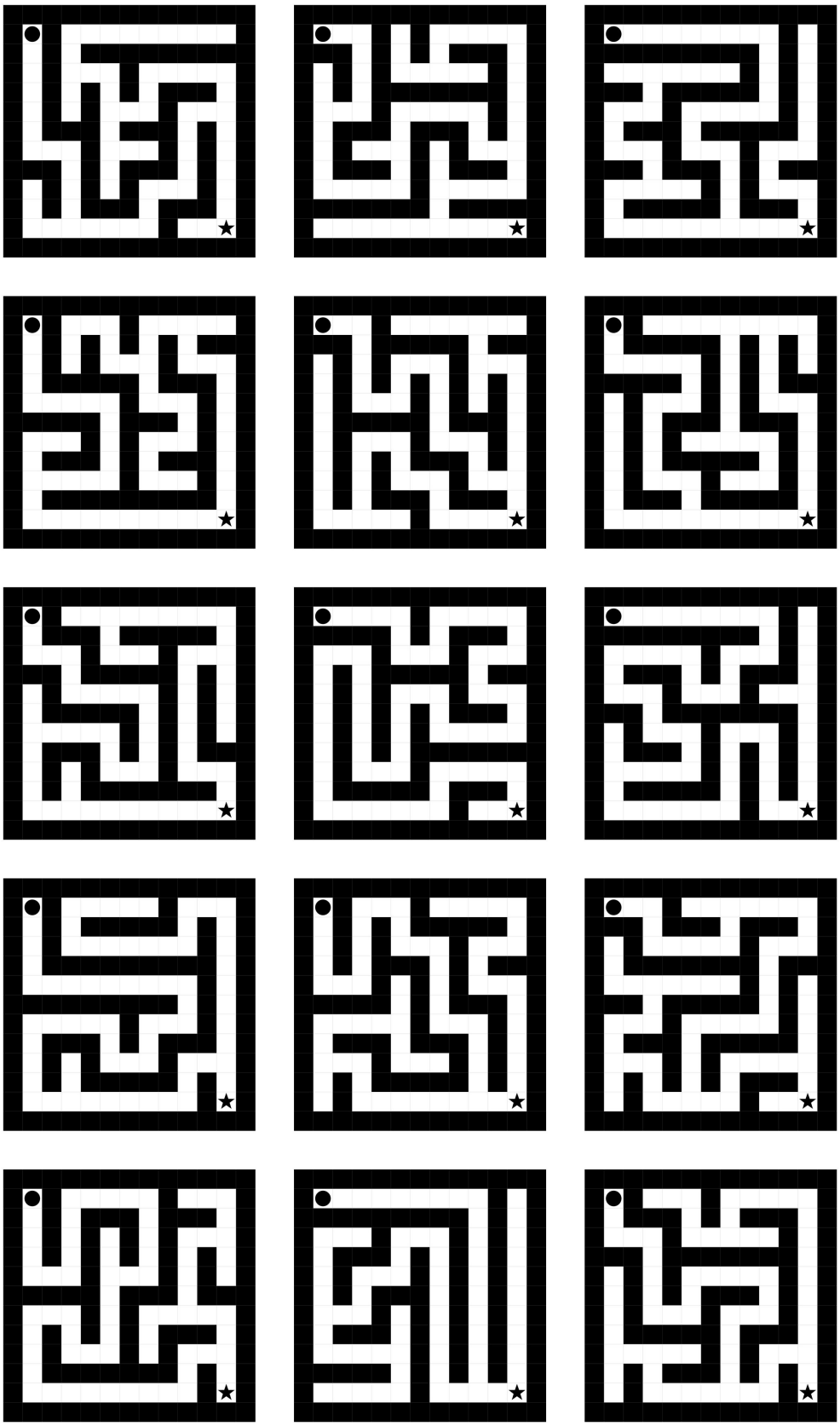


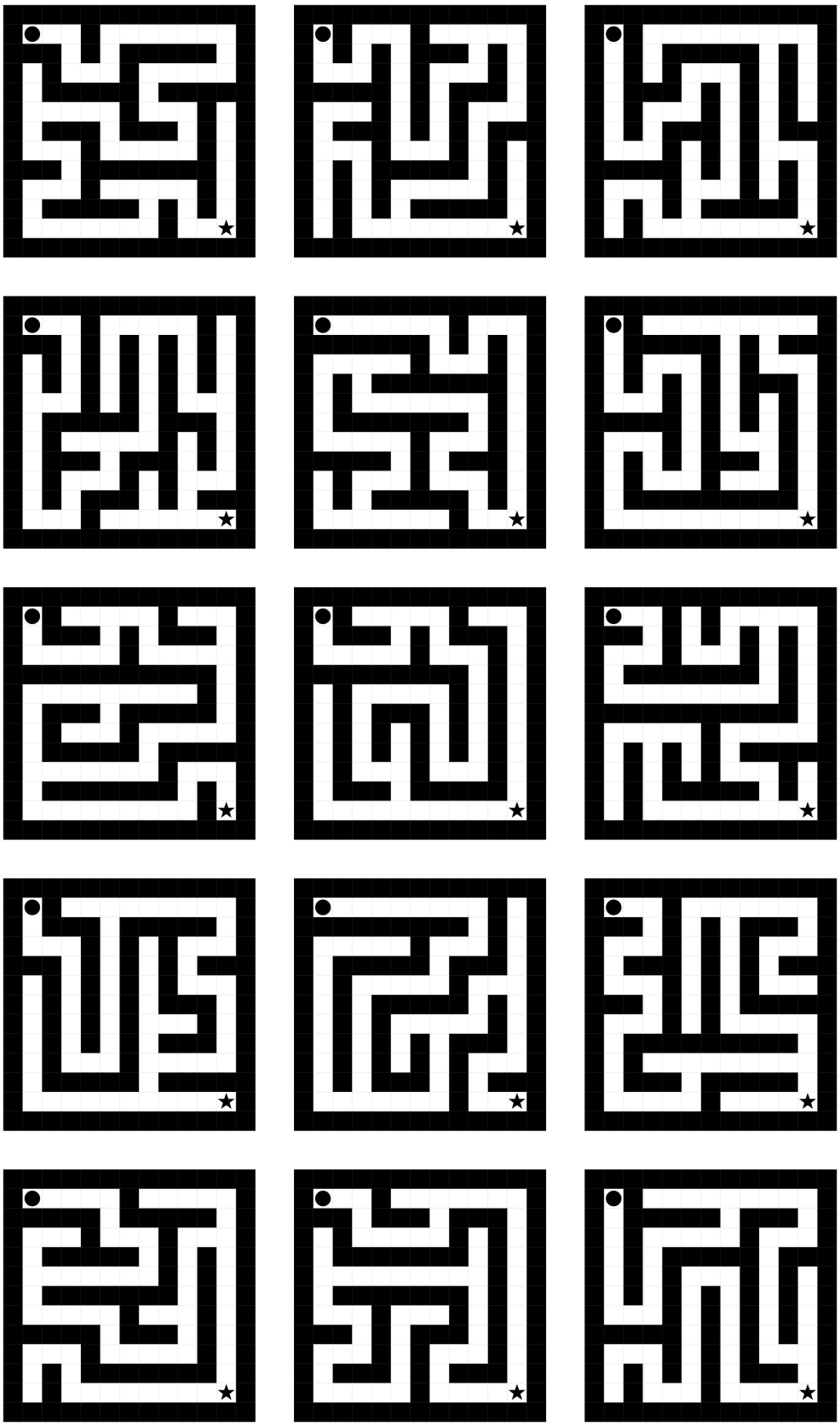


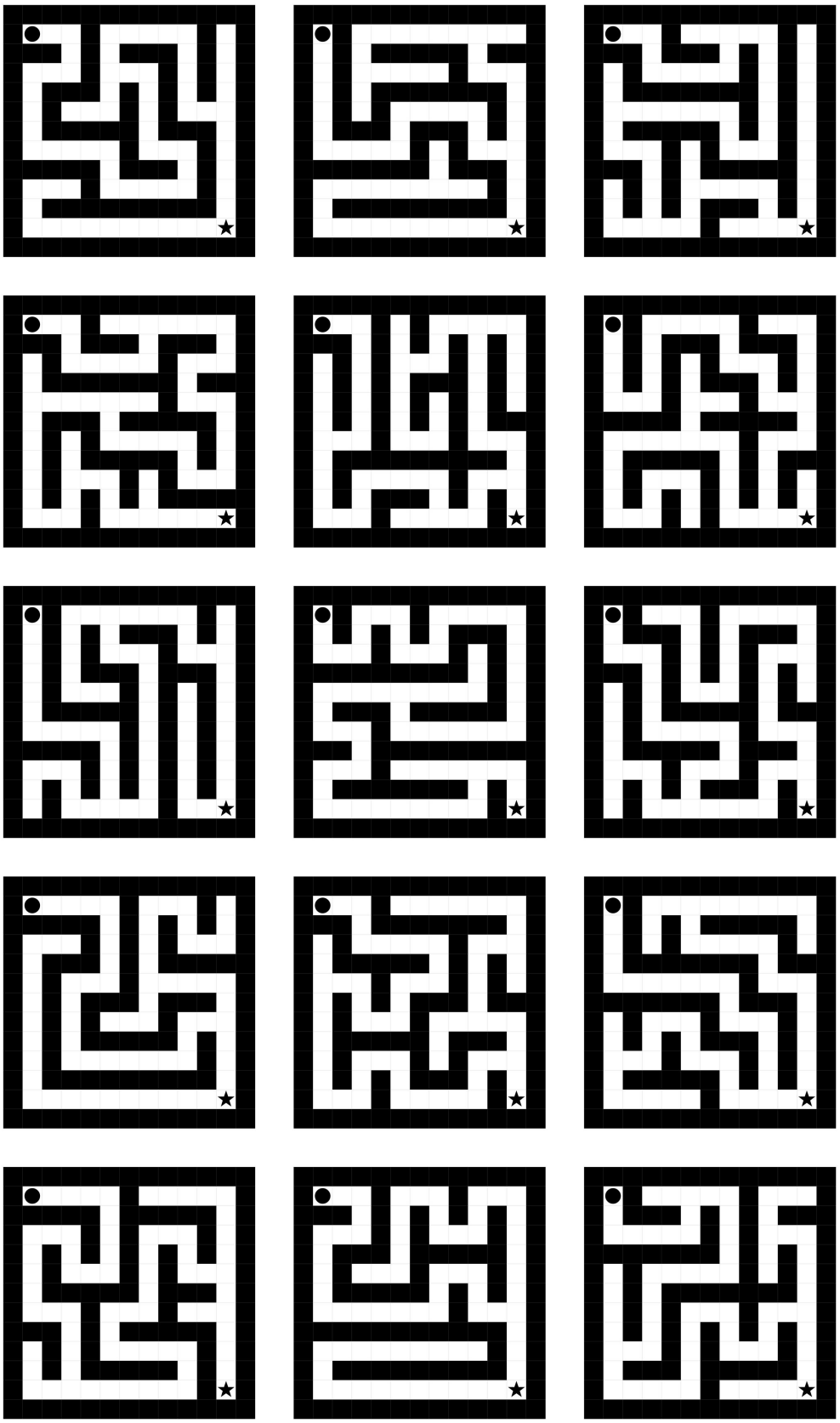
A.4 13×13 Mazes

All 50 unique 13×13 occupancy grid mazes that appear in the dataset.



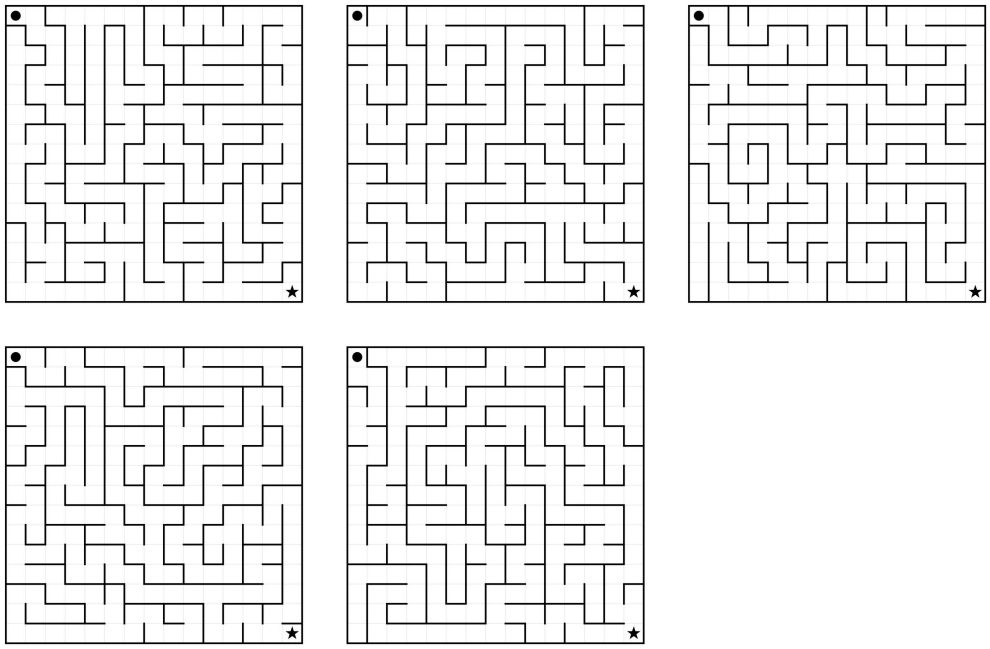


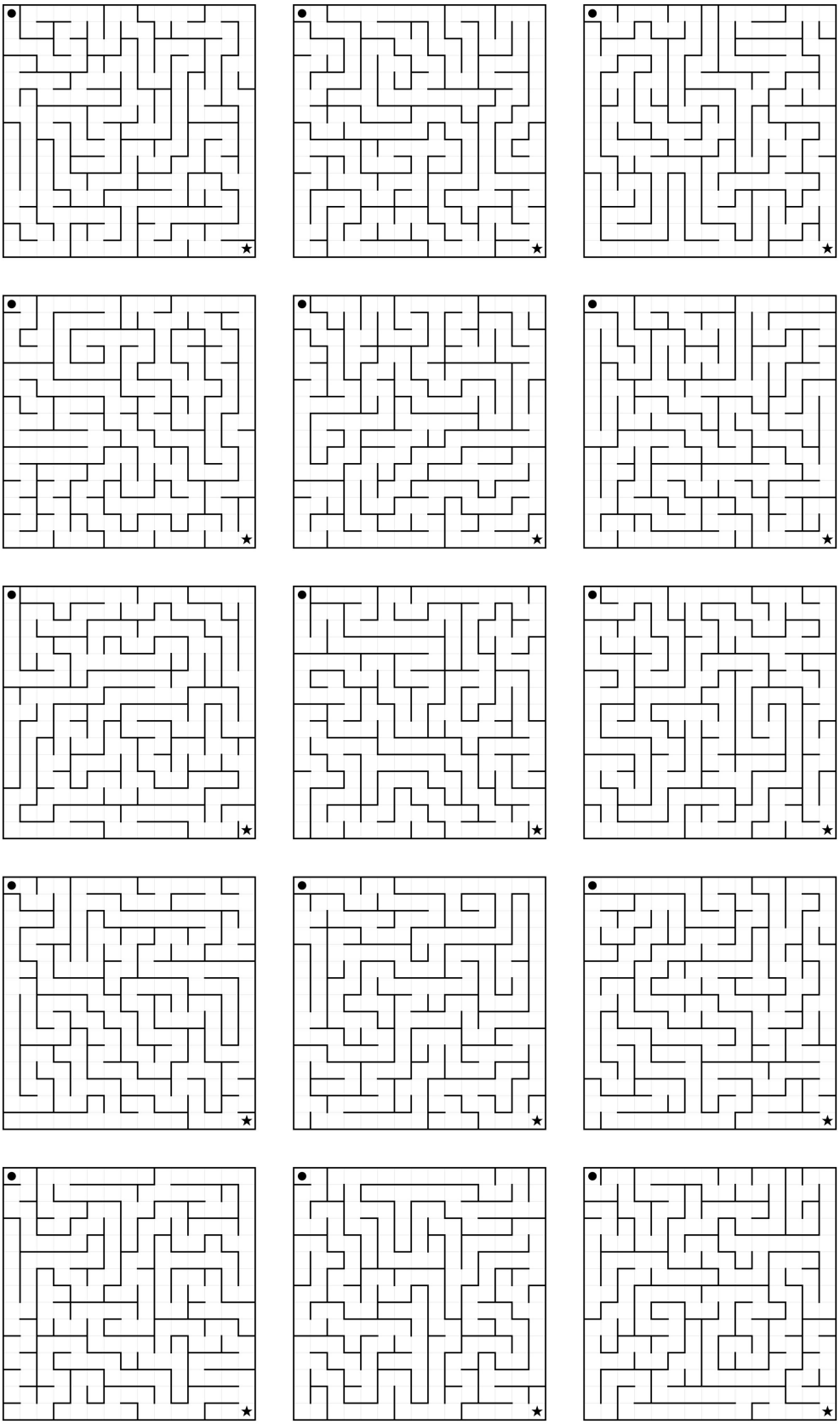


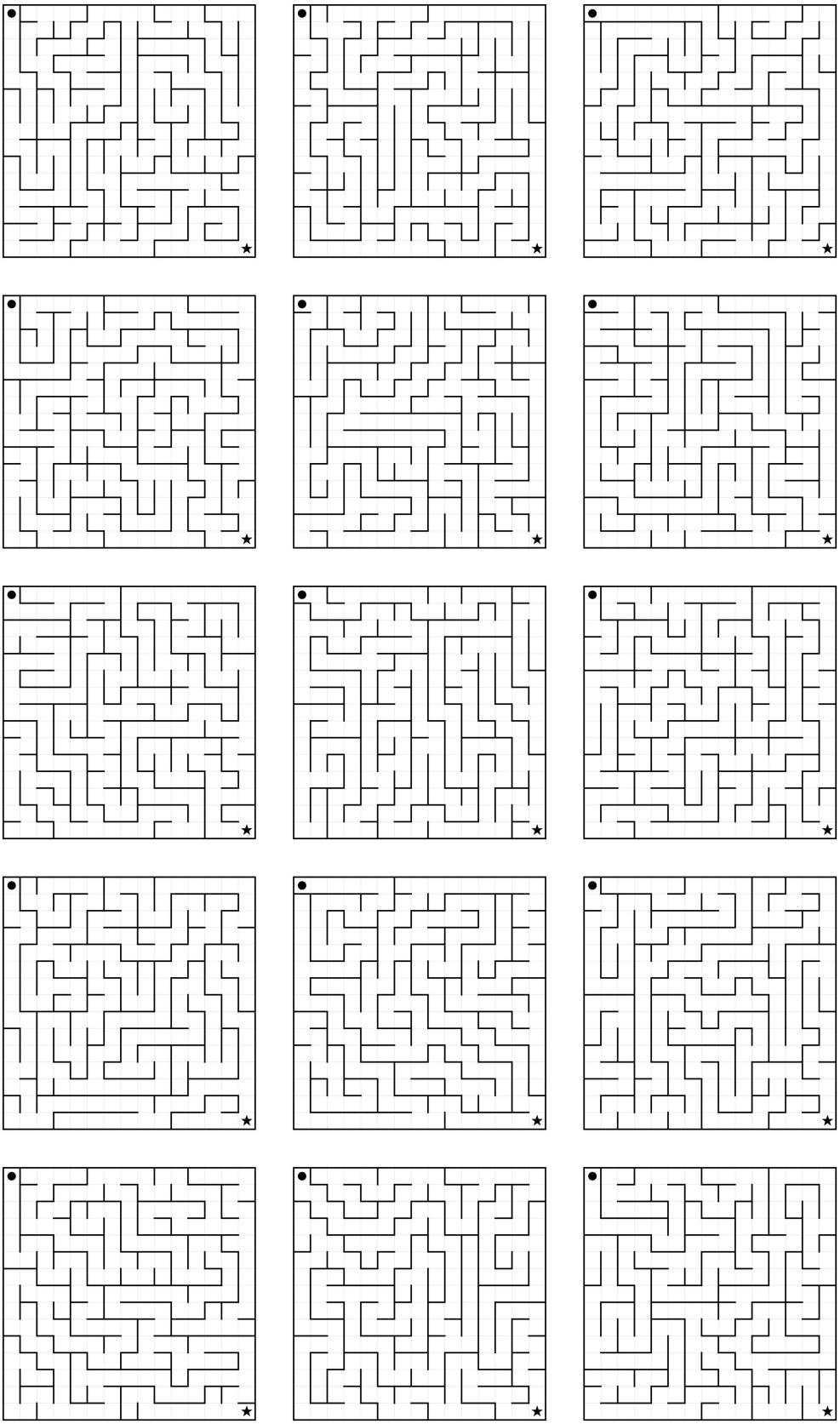


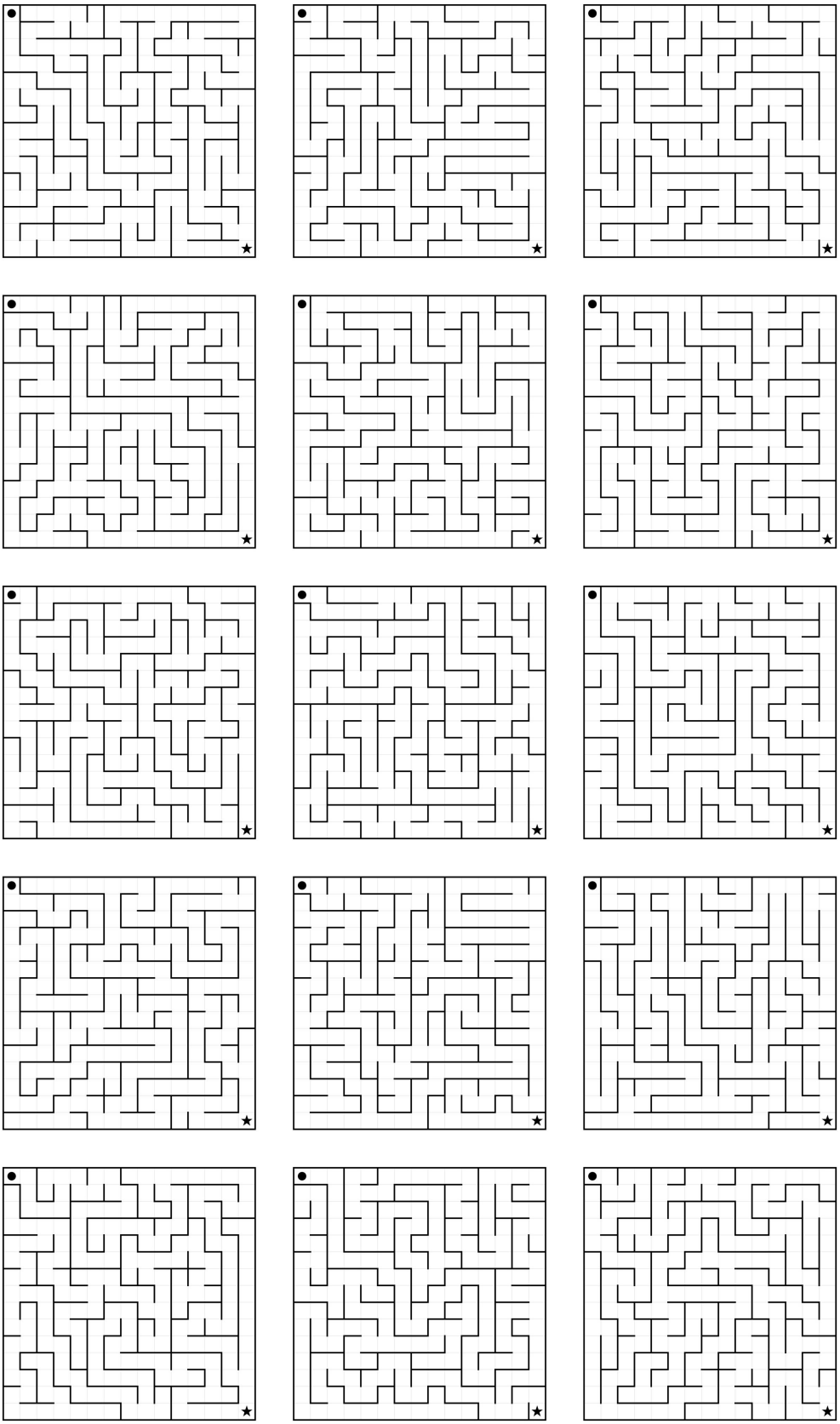
A.5 15×15 Mazes

All 50 unique 15×15 line-wall mazes that appear in the dataset.



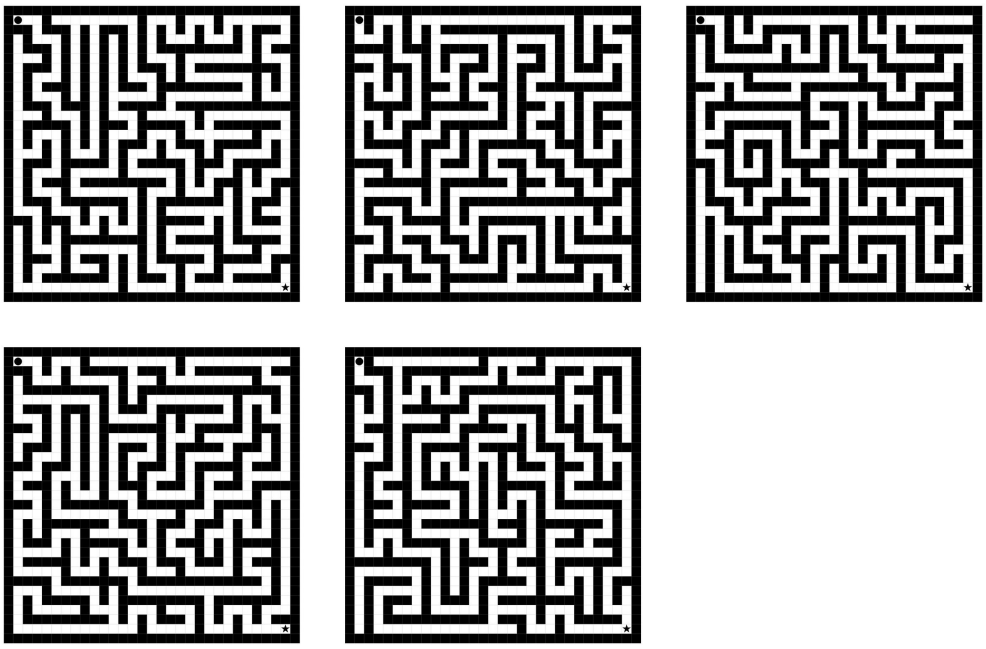


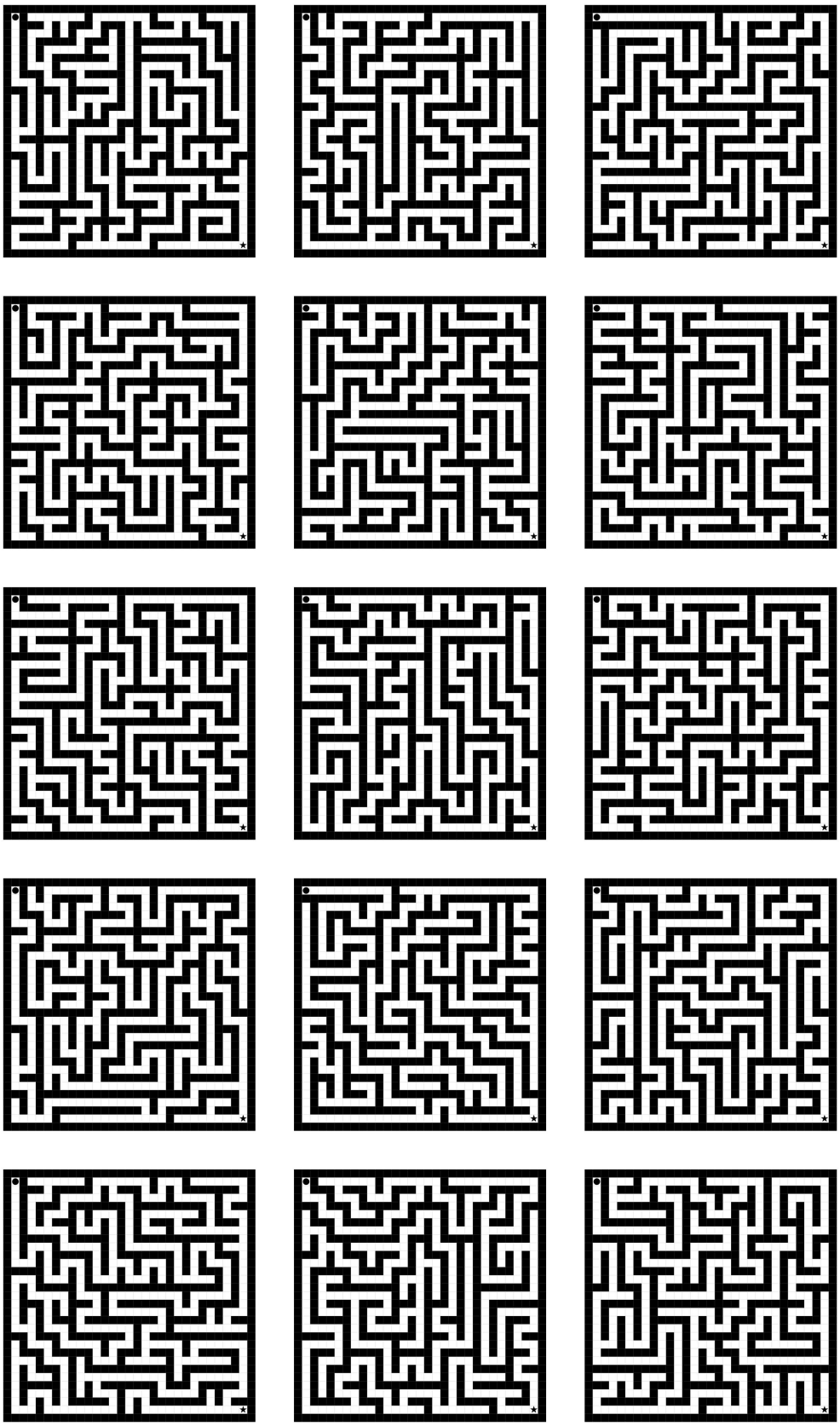


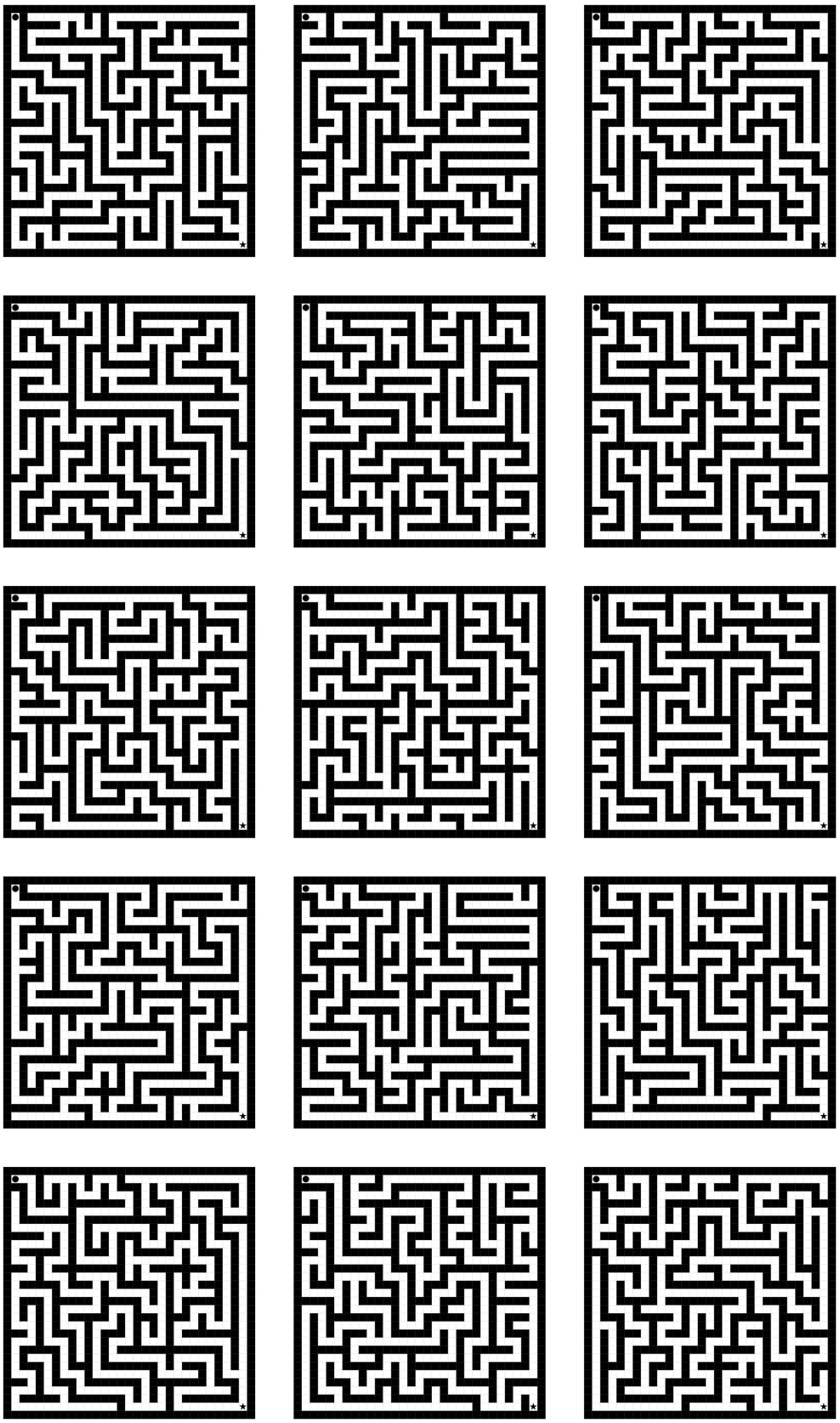


A.6 31×31 Mazes

All 50 unique 31×31 occupancy grid mazes that appear in the dataset.







B Methods

B.1 Spatial representations

Listing 1: The top two rows of a maze in line-wall JSON representation. Each cell is contained within { }, and each row is contained within square brackets.

```
{
    "size": {
        "columns": 3,
        "rows": 3
    },
    "start": [
        0,
        0
    ],
    "end": [
        2,
        2
    ],
    "grid": [
        [
            {
                "walls": {
                    "N": true,
                    "E": true,
                    "S": false,
                    "W": true
                }
            },
            {
                "walls": {
                    "N": true,
                    "E": false,
                    "S": true,
                    "W": true
                }
            },
            {
                "walls": {
                    "N": true,
                    "E": true,
                    "S": false,
                    "W": false
                }
            }
        ],
        [
            {
                "walls": {
                    "N": false,
                    "E": false,
                    "S": true,
                    "W": true
                }
            },
            {
                "walls": {
                    "N": true,
                    "E": true,
                    "S": false,
                    "W": false
                }
            },
            {
                "walls": {
                    "N": false,
                    "E": true,
                    "S": false,
                    "W": true
                }
            }
        ],
```

Listing 2: The top row of a maze in line-wall Al-JSON representation. Each cell is called "node" followed by its coordinates.

```
{
    "size": {
        "columns": 3,
        "rows": 3
    },
    "start": [
        0,
        0
    ],
    "end": [
        2,
        2
    ],
    "grid": [
        {
            "node": [
                0,
                0
            ],
            "neighbors": [
                [
                    1,
                    0
                ]
            ]
        },
        {
            "node": [
                0,
                1
            ],
            "neighbors": [
                [
                    0,
                    2
                ]
            ]
        },
        {
            "node": [
                0,
                2
            ],
            "neighbors": [
                [
                    1,
                    2
                ],
                [
                    0,
                    1
                ]
            ]
        },
```

Listing 3: A whole maze in occupancy grid JSON format. Each cell is denoted by one number, and each row is contained within square brackets. "1" signifies an occupied cell, and "0" signifies an open cell.

```
{
    "size": {
        "columns": 7,
        "rows": 7
    },
    "start": [
        1,
        1
    ],
    "end": [
        5,
        5
    ],
    "grid": [
        [
            1,
            1,
            1,
            1,
            1,
            1,
            1
        ],
        [
            1,
            0,
            1,
            0,
            0,
            0,
            1
        ],
        [
            1,
            0,
            1,
            1,
            1,
            0,
            1
        ],
        [
            1,
            0,
            0,
            0,
            1,
            0,
            1
        ],
        [
            1,
            1,
            1,
            0,
            1,
            0,
            1
        ],
        [
            1,
            0,
            0,
            0,
            0,
             0,
            1
        ],
        [
            1,
            1,
            1,
            1,
            1,
            1,
            1
        ]
    ]
}
```

Listing 4: The top row of a maze in occupancy grid AL-JSON representation. Each cell is called "node" followed by its coordinates.

```
{
    "size": {
        "columns": 7,
        "rows": 7
    },
    "start": [
        1,
        1
    ],
    "end": [
        5,
        5
    ],
    "grid": [
        {
            "node": [
                1,
                1
            ],
            "neighbors": [
                [
                    2,
                    1
                ]
            ]
        },
        {
            "node": [
                1,
                3
            ],
            "neighbors": [
                [
                    1,
                    4
                ]
            ]
        },
        {
            "node": [
                1,
                4
            ],
            "neighbors": [
                [
                    1,
                    5
                ],
                [
                    1,
                    3
                ]
            ]
        },
        {
            "node": [
                1,
                5
            ],
            "neighbors": [
                [
                    1,
                    4
                ],
                [
                    2,
                    5
                ]
            ]
        },
```

B.2 Prompts

A full prompt consists of the spatial representation-specific instruction + the output-specific instruction + the maze.

B.2.1 Spatial representation-specific instructions

During testing, "*rows* x *columns*" is replaced by the true size of the maze.

Line-wall JPG:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is shown as an image, where thick black lines are impassable walls, thin light gray lines are passable cell borders, the circle is the start and the star is the end; the top-left corner is (0,0) in (row, col)."

Line-wall JSON:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is represented as a JSON grid, which describes each cell as a set of four (N/E/S/W) walls with boolean 'True' (impassable) or 'False' (passable) values and includes start/end coordinates; the top-left corner is (0,0) in (row, col)."

Line-wall Adjacency List JSON:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is represented as a JSON adjacency list, which lists all available neighbors for each cell and provides start/end coordinates; the top-left corner is (0,0) in (row, col)."

Line-wall Adjacency List Text:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is represented as a graph via an adjacency list that lists which cells are connected (e.g., (0,0) <-> (0,1)) and marks the start and end with <ORIGIN> and <TARGET> tags; when converted to a grid, the top-left corner is (0,0) in (row, col)."

Line-wall Tagged per-cell:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is represented as a grid in a tokenized manner, where each cell is described by its walls (e.g., <|updownleft_wall|>) and the start and end cells are marked with <|origin|> and <|target|>, respectively; the top-left corner is (0,0) in (row, col)."

Occupancy grid JPG:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is shown as an image, where white cells are passable, black cells are impassable walls, thin light gray lines are traversable cell borders, the circle is the start and the star is the end; the top-left corner is (0,0) in (row, col)."

Occupancy grid JSON:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is represented as a JSON grid, where cells are '1' (inaccessible) or '0' (accessible), and start/end coordinates are provided; the top-left corner is (0,0) in (row, col)."

Occupancy grid Adjacency List JSON:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is represented as a JSON adjacency list, which lists all available neighbors for each cell and provides start/end coordinates; the

top-left corner is (0,0) in (row, col)."

Occupancy grid Adjacency List Text:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is represented as a graph via an adjacency list that lists which cells are connected (e.g., (0,0) <-> (0,1)) and marks the start and end with <ORIGIN> and <TARGET> tags; when converted to a grid, the top-left corner is (0,0) in (row, col)."

Occupancy grid ASCII:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The maze is represented as a *rows* x *columns* ASCII grid using '#' for walls, ' ' for corridors, 'S' for the start, and 'E' for the end; the top-left corner is (0,0) in (row, col)."

Occupancy grid Tagged per-cell:
"You are a maze-solving expert. Your goal is to find the path from start to end. Do not use external tools or write code to solve the maze.
The *rows* x *columns* maze is represented as a grid in a tokenized manner, where inaccessible cells are tagged as <|occupied_wall|>, accessible cells are tagged as <|blank|>, and the start and end cells are tagged with <|origin|> and <|target|>, respectively; the top-left corner is (0,0) in (row, col)."

B.2.2 Output-specific instructions

Absolute coordinates ('coordinates') output prompt:
"Instructions:
1. You cannot move diagonally or through walls, only from one cell to an adjacent cell.
2. Create a comma-separated sequence of all coordinates on the path from start to end, including the start and end points. For example: (0,0),(1,0),(1,1),(2,1),(3,1).
3. Provide only the final list of coordinates from start to end in your response."

Absolute directions output prompt:
"Instructions:
1. You can only move up, down, left, or right.
2. You cannot move diagonally or through walls, only from one cell to an adjacent cell.
3. Your output must be a single, comma-separated sequence of steps. For example: up, down, right, right, down.
4. Provide only the final list of moves in your response."

Egocentric output prompt:
"Instructions:
1. Give instructions to an agent in the maze. The agent begins by facing south. You can only use the following four actions:
Forward: this moves the agent 1 step in the direction it is facing.
Left: this turns the agent 90° to the left and then moves the agent 1 step in the new direction it is facing.
Right: this turns the agent 90° to the right and then moves the agent 1 step in the new direction it is facing.
Backward: this turns the agent 180° and then moves the agent 1 step in the new direction it is facing.
2. You cannot move diagonally or through walls, only from one cell to an adjacent cell.
3. Your output must be a single, comma-separated sequence of steps. For example: forward, left, right, forward, right.
4. Provide only the final list of instructions in your response."

C Metrics

The formula for completion score is shown in Equation 1. The numerator takes the lowest value between the number of correct consecutive steps, or the ground truth solution length. In practice, this means that the maximum score is capped at 100%, and any additional steps that continue beyond the target cell are disregarded and the answer yields a completion score of 100%.

$$\text{Completion score} = \frac{x}{L_{gt}} \times 100 \tag{1}$$

$$x = \begin{cases} CCS & \text{if CCS} \leq L_{gt} \\ L_{gt} & \text{otherwise} \end{cases}$$

- CCS: Correct Consecutive Steps, counted from the start of the LLM's answer until the first error. Unit: steps.
- L_{gt}: Ground truth solution length. The number of steps in the ground truth solution of the maze in the requested FoR. Unit: steps.
- Completion score: reflects how far along the solution path the LLM reached. Unit: percentages.

D Results

D.1 Confidence intervals

Half-widths of the 95% confidence intervals of the mean completion scores, expressed in percentage points. They are computed separately for each combination of spatial representation, output FoR, maze size, and LLM. Because the completion scores are not normally distributed, bounded, and made up of a small sample size, the confidence interval half-widths are calculated using Bias-Corrected and Accelerated (BCa) bootstrapping with 10.000 resamples and a random seed of 42.

Half-Widths of the 95% Confidence Interval for the Mean Completion Scores, Given in Percentage Points, Computed Separately for Each Combination of Spatial Representation, Output FoR, Maze Size, and LLM

3x3/7x7 Mazes

Representation	CI half-width Coordinates Output, Gemini 2.5 Pro	CI half-width Abs. Directions Output, Gemini 2.5 Pro	CI half-width Egocentric Output, Gemini 2.5 Pro	CI half-width Coordinates Output, Gemini 2.5 Flash-Lite	CI half-width Abs. Directions Output, Gemini 2.5 Flash-Lite	CI half-width Egocentric Output, Gemini 2.5 Flash-Lite
Line-wall AL-JSON	0.0	0.0	6.8	0.9	10.5	1.8
Line-wall AL-TXT	0.0	0.0	4.2	4.8	11.3	0.0
Line-wall JPG	9.6	11.0	11.1	7.1	9.1	1.5
Line-wall JSON	0.0	0.0	0.0	9.4	10.3	1.9
Line-wall Tagged	1.3	1.2	0.0	3.7	7.8	0.0
Occupancy grid AL-JSON	0.0	0.0	5.1	0.0	8.6	0.9
Occupancy grid AL-TXT	0.0	0.0	0.0	1.1	9.4	0.0
Occupancy grid ASCII	6.1	7.5	8.9	11.4	8.8	1.8
Occupancy grid JPG	9.5	7.5	9.6	3.2	2.2	1.5
Occupancy grid JSON	0.0	0.0	0.0	10.4	8.1	1.0
Occupancy grid Tagged	0.0	0.0	0.0	11.3	9.7	1.4

Half-Widths of the 95% Confidence Interval for the Mean Completion Scores, Given in Percentage Points, Computed Separately for Each Combination of Spatial Representation, Output FoR, Maze Size, and LLM

6x6/13x13 Mazes

Representation	CI half-width Coordinates Output, Gemini 2.5 Pro	CI half-width Abs. Directions Output, Gemini 2.5 Pro	CI half-width Egocentric Output, Gemini 2.5 Pro	CI half-width Coordinates Output, Gemini 2.5 Flash-Lite	CI half-width Abs. Directions Output, Gemini 2.5 Flash-Lite	CI half-width Egocentric Output, Gemini 2.5 Flash-Lite
Line-wall AL-JSON	0.0	0.0	6.7	7.1	4.7	1.1
Line-wall AL-TXT	0.0	0.0	8.7	9.1	3.8	1.5
Line-wall JPG	0.0	2.5	2.7	2.0	1.6	0.7
Line-wall JSON	2.9	3.5	7.5	3.1	2.9	1.0
Line-wall Tagged	0.0	6.1	8.2	2.2	1.7	0.4
Occupancy grid AL-JSON	0.0	0.0	7.0	7.2	3.1	0.0
Occupancy grid AL-TXT	0.0	0.0	6.6	8.5	4.6	0.1
Occupancy grid ASCII	5.8	5.3	6.5	0.0	3.4	0.0
Occupancy grid JPG	0.7	2.2	2.7	0.0	0.6	0.5
Occupancy grid JSON	8.3	9.2	11.0	6.4	4.2	1.3
Occupancy grid Tagged	0.0	2.4	8.3	5.4	2.9	0.2

Half-Widths of the 95% Confidence Interval for the Mean Completion Scores, Given in Percentage Points, Computed Separately for Each Combination of Spatial Representation, Output FoR, Maze Size, and LLM

15x15/31x31 Mazes

Representation	CI half-width Coordinates Output, Gemini 2.5 Pro	CI half-width Abs. Directions Output, Gemini 2.5 Pro	CI half-width Egocentric Output, Gemini 2.5 Pro	CI half-width Coordinates Output, Gemini 2.5 Flash-Lite	CI half-width Abs. Directions Output, Gemini 2.5 Flash-Lite	CI half-width Egocentric Output, Gemini 2.5 Flash-Lite
Line-wall AL-JSON	5.2	6.4	11.6	4.2	0.9	0.3
Line-wall AL-TXT	8.1	11.0	11.1	2.6	0.9	0.4
Line-wall JPG	0.1	0.5	0.4	0.4	0.7	0.5
Line-wall JSON	5.5	4.1	2.8	0.8	0.9	0.1
Line-wall Tagged	10.4	11.3	11.8	0.4	0.9	0.3
Occupancy grid AL-JSON	4.6	7.9	10.3	7.0	0.8	0.0
Occupancy grid AL-TXT	9.3	10.8	8.9	4.1	0.9	0.4
Occupancy grid ASCII	0.2	0.7	0.6	0.0	0.4	0.2
Occupancy grid JPG	0.3	0.3	0.5	0.0	0.4	0.3
Occupancy grid JSON	3.5	2.5	1.8	1.3	0.6	0.3
Occupancy grid Tagged	10.0	9.7	7.8	1.2	0.6	0.2

D.2 Supporting figures

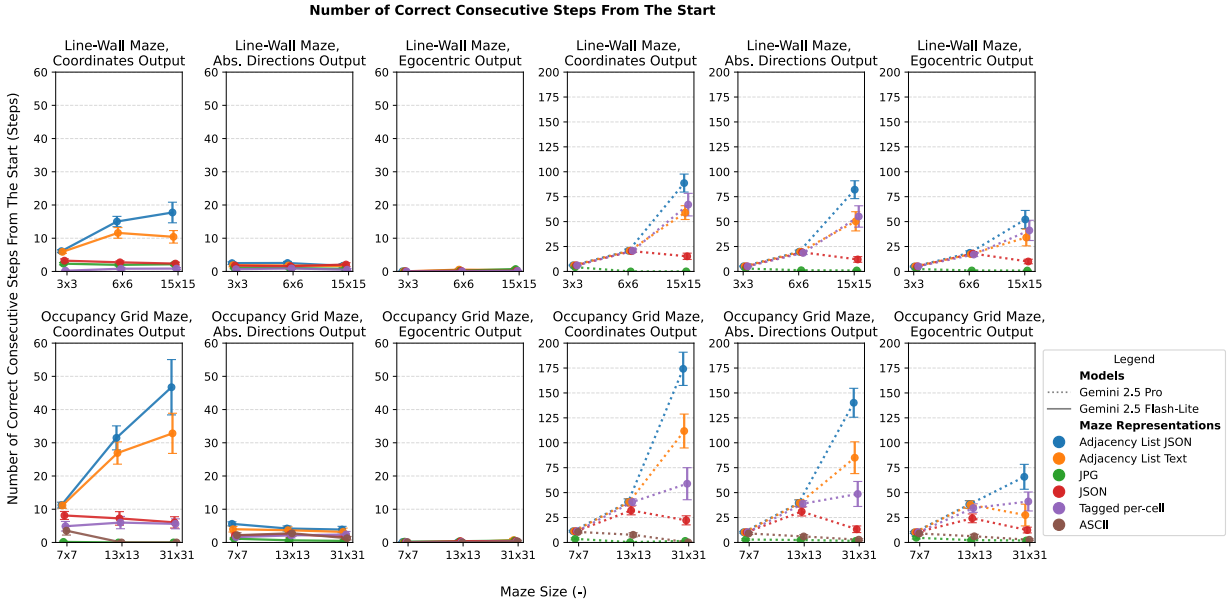


Figure 7: The average number of correct consecutive steps counted from the start of the LLM's answer until the first mistake or reaching the target cell, whichever occurs first, at each maze size. The six left-hand figures reflect Gemini 2.5 Flash-Lite's performance, the six right-hand figures reflect Gemini 2.5 Pro's performance. **Note that the scale on the y-axes is different for the two models**. The error bars show the bounds of the 95% confidence interval of that datapoint in steps.

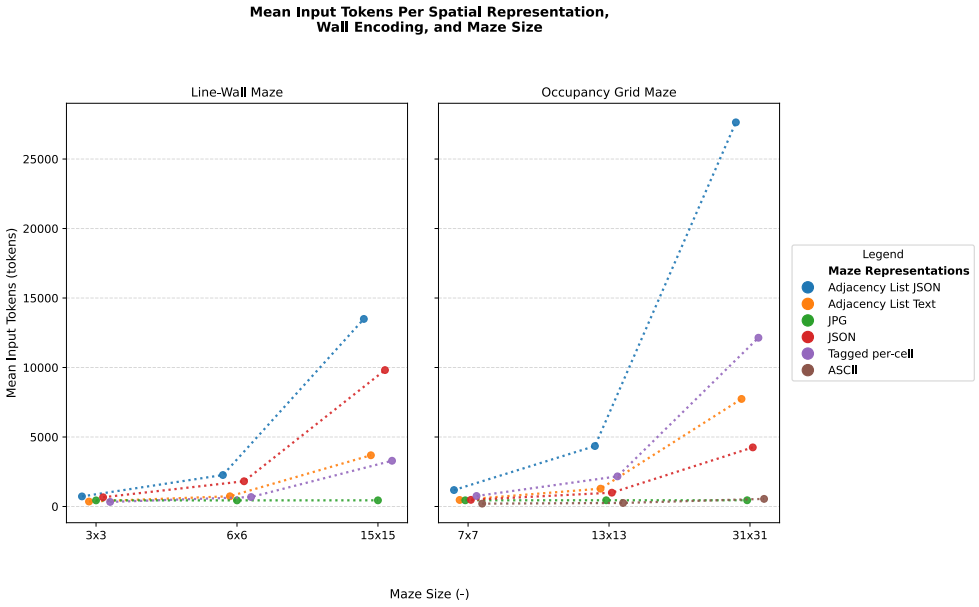


Figure 8: The mean number of input tokens for each spatial representation and maze size. The number of input tokens is the same for both LLMs, and the difference between FoRs is negligible.

D.3 Keyword search

Table 6: Used settings for Gemini 3 Pro, when used as an assistive tool for identifying keywords and categories. This model has been discontinued as of March 26 2026. All values are default (Google Cloud, 2026).

Parameter	Gemini 3 Pro
Temperature	1.0
Top-P	0.95
Top-K	64
Candidate count	1
Maximum output tokens	65,536 Incl. thinking

Table 7: All categories explored during keyword search, their definitions, and exhaustive list of keywords. *"examin" works for "examination", "examine", and "examining".

Category and Explanation	Keywords
Algorithm naming: Explicitly describes using Breadth-First Search, Depth-First Search, or other formal algorithm	visited, queue, queuing, dequeuing, enqueuing, parent

Heuristics: Uses heuristics such as right-hand rule, going down and right, or "visual" search	down-right, right-down, down and right, downwards and to the right, downward and to the right, down then right, right then down, to the right and down, right then down, to the right and down, right-hand, wall-following, heuristic, visually tracing, visual, eyes, "down, right", "right, down", southeast
Full Restart: Restarts the search from scratch, reflects on strategy choice, or changes approach	I'll need to visualize the maze in a different way, I'll begin from the starting node again, I'll begin from the start again, I did it again, restart, several attempts, trial and error, I had to go back, rethink, more systematic, new method, try again, re-routing, re-evaluate, switch, I start again, start fresh, change my approach, another approach, change my tactic, refining the approach, strategy shift, second trace
Reverse search: Tries solving from the target cell backward	working backward, from the target, from the end, starting at (14,14), starting at (29,29), starting at (2,2), starting at (11,11), starting at (5,5), working backward, meeting in the middle, trace a path backward, from (14,14) to (0,0), from (29,29) to (1,1), from (5,5) to (0,0), from (2,2) to (0,0), from (11,11) to (1,1), from (5,5) to (1,1), path from the end point, both ends tracing

Frustration/confusion: Expresses being stuck, frustrated, confused, or surprised	complicated, challenge, confusing, confused, stuck, struggling, struggle, difficult, difficult to parse, suspect, suspicion, no solution, typo, twists, proving to be a challenge, proves to be a challenge, twisty, complex, more elaborate than I expected, more intricate than I anticipated, trap, is the problem flawed?, oversight
Declared confidence: Claims success with vague or unsupported language	I made it, I've made it, confident, it works, paid off, solved, correct, complete, this is the solution, "easy, right?", no problem **Exclusions**: not confident
Backtracking: Mentions backtracking or hitting dead ends	backtracking, backtrack when, backtracked when, recalculate, track back, backtrack, backtracking and checking, I tried backtracking, retracing, backtracked, after backtracking, I have to backtrack **Exclusions**: backtrack when I hit, backtracking when I hit, backtrack whenever I hit, backtracking whenever I hit, backtrack when necessary, backtracking and re-checking, backtrack and re-check, prepared to backtrack, backtrack if I need to
Step-by-step: Shows explicit node-by-node tracing with coordinates	') to (', '), (', ') and (', ') or (', ') has (', '), then (', ') then (', '), and then (', ') and then (', 'Path: '(', 'options are (', **Exclusions**: (0,0) to (14,14), (1,1) to (29,29), (0,0) to (5,5), (1,1) to (11,11),(0,0) to (2,2), (1,1) to (5,5)

Verification: Claims to verify and/or double-check the final path	check the path step by step, review, carefully ensure, examin*, comparing, confirmed, thorough check, check the path, double-check, sanity check, check to verify, checking that the answer, checking the answer, make sure to check, go back over each step, verify, valid, verified **Exclusions**: any valid connection

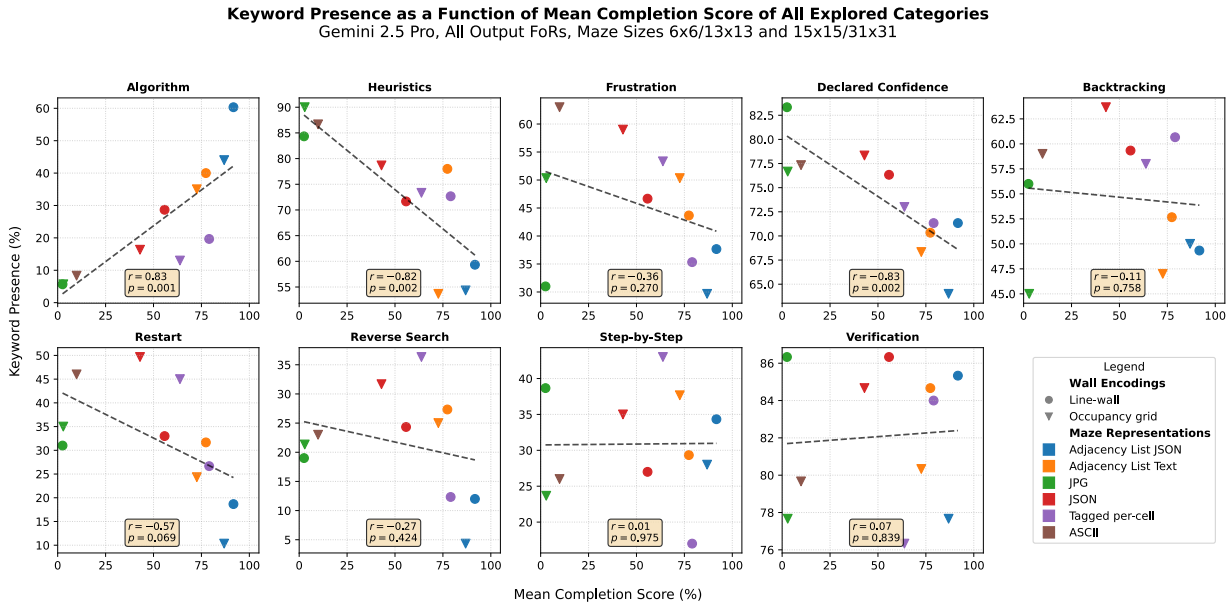


Figure 9: Keyword presence as a function of mean completion score, shown for all explored categories, including those without statistical significance and low correlation. The data is averaged over all Gemini 2.5 Pro's reasoning traces for maze sizes 6×6/13×13 and 15×15/31×31. "r" Is the correlation coefficient, with "p" as its associated statistical significance. **Note that the scale on the y-axes is different for each subplot.**

E Limitations

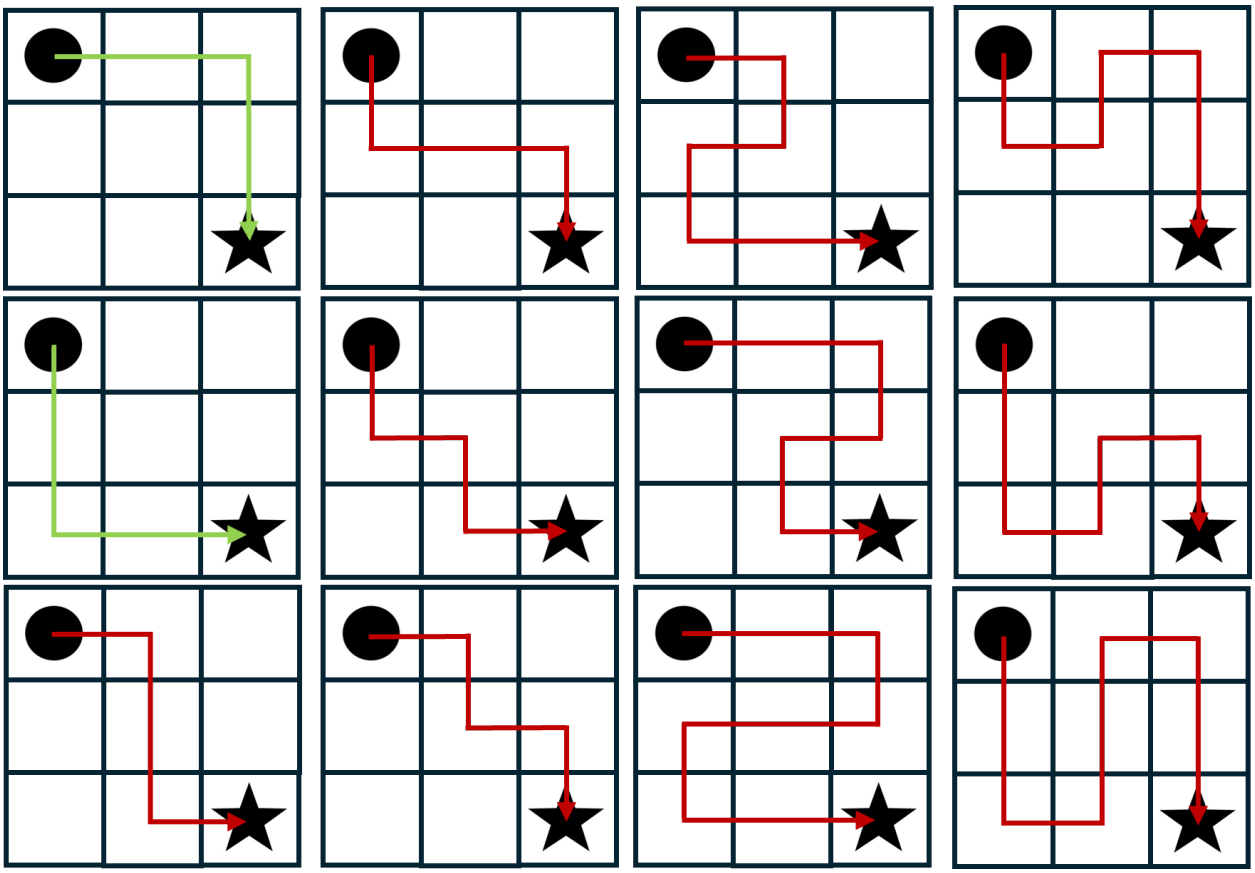


Figure 10: The full set of possible solution paths in a 3×3 maze. Our dataset contains a single unique maze for each possible path indicated in red, and two unique mazes for each path in green. The same is true for 7×7 occupancy grid mazes.