In- and Out-of-Distribution Generalization of Reasoning in Multimodal LLMs for Simple Visual Planning Tasks

Anonymous Authors[1]

Abstract

Integrating reasoning in (multimodal) large language models has recently led to significant improvement of their capabilities. However, generalization in reasoning models is still vaguely defined and poorly understood. In this work, we present an evaluation framework to rigorously examine how well chain-of-thought (CoT) approaches generalize in simple visual planning, specifically on a grid-based navigation task. The versatility of the task and its data allows us to fine-tune model variants using different input representations (visual and textual) and CoT reasoning strategies, and systematically evaluate them under both in-distribution (ID) and out-of-distribution (OOD) test conditions. Our experiments show that the out-of-distribution generalization (e.g., to larger maps) is largely impacted by the format used for input maps and CoT chains. Surprisingly, we find that reasoning traces which combine multiple text formats yield the best OOD generalization. Moreover, CoT reproducing the steps of the A* algorithm yields the state-of-the-art ID accuracy, and simple augmentation of the map solutions seen during training greatly boosts OOD results. Finally, purely text-based models consistently outperform those utilizing image-based inputs, including a recently proposed approach relying on latent space reasoning.

1. Introduction

Chain-of-thought (CoT) reasoning has been shown to be a powerful tool to improve the ability of large language models (LLMs) to solve complex tasks (Wei et al., 2022). In practice, both test-time prompting (*"Let's think step-by-step"*) (Kojima et al., 2022) or specifically designed fine-tuning schemes (Ho et al., 2023) can make LLMs generate intermediate reasoning steps which aid arriving to the correct solution. This approach may also benefit interpretability, as the user can inspect the reasoning process behind the response and potential failure points. Recently, the chain-of-thought paradigm has been extended to multimodal models to enhance their capabilities to reason about visual inputs (Huang et al., 2025; Yang et al., 2025). Despite its widespread success and adoption, we still lack a fundamental understanding of why chain-of-thought is effective and what kinds of reasoning capabilities LLMs actually acquire. Recent work further suggests that the apparent reasoning abilities of multimodal models may largely reflect statistical regularities in the training data rather than genuine algorithmic learning, as evidenced by performance degradation when inputs diverge from the training distribution (Stechly et al., 2024; Zhao et al., 2025).

In this work, we want to systematically study both in-distribution (ID) and, especially, out-of-distribution (OOD) generalization behavior of LLMs in a controlled setup. Such a comprehensive evaluation is not straightforward with common benchmarks (Lu et al., 2022; Yue et al., 2024), as there is no clear distinction between ID and OOD tasks, or an evident ground-truth algorithm to be learned. Therefore, we first build a rich evaluation environment upon the FROZENLAKE dataset (Wu et al., 2024), which requires visual planning to find the correct path to reach a treasure without falling into lakes (see Fig. 2). The fact that even small maps are challenging for state-of-the-art LLMs (Wu et al., 2024) makes it a relevant testbed to study generalization of reasoning models. Moreover, both maps and reasoning steps can be represented as images or in text-only formats (Fig. 2). Crucially, task complexity can be easily controlled through factors such as map size, start–target distance, and the number of lakes, allowing us to distinguish between in- and out-of-distribution data, and to create multiple distribution shifts for evaluating generalization to unseen data types.

We train both models without CoT supervision and reasoning models, using the same training maps but encoding them in different formats. We consider four main input formats: *image*, *description*, *grid*, and *coordinates* (the latter three being text-based). Reasoning traces are expressed either

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

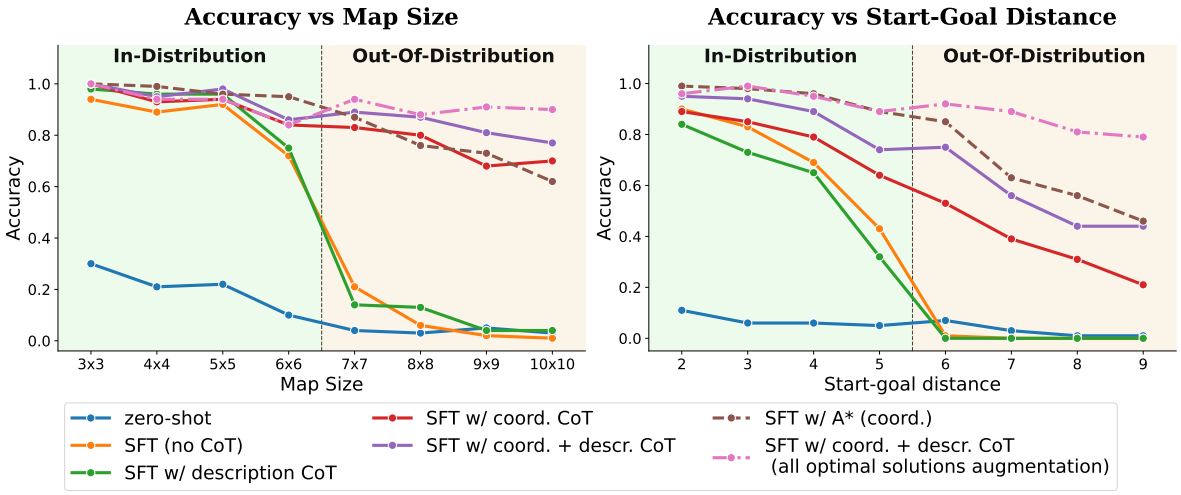

Figure 1. **Chain-of-thought (CoT) format and augmentations impact in- (ID) and out-of-distribution (OOD) generalization.** We fine-tune Qwen2.5-VL-7B-Instruct on FROZENLAKE with maps in *coordinates* format. Supervised fine-tuning (SFT) without CoT or with CoT in *description* format lead to no OOD generalization to larger map sizes or start-goal distances. Conversely, combining *coordinates* and *description* reasoning steps yields good accuracy under severe shifts. CoT reproducing the steps of the A* algorithm achieves overall the best ID and competitive OOD results. This shows that generalization of reasoning is influenced by the format of the CoT traces. Finally, augmenting the training data with multiple solutions for each map further boosts OOD accuracy.

in one of these text-based formats or in a combination of them (Fig. 3). Moreover, the *coordinates* format naturally supports A*-like reasoning traces, which provide an algorithmic strategy guaranteed to find a correct path. As most maps admit multiple valid paths, our framework also allows us to vary the number of solutions (and reasoning traces) per map while keeping the underlying training maps fixed. This lets us study solution-set augmentation as a controlled complement to input and CoT format. Across all models, performance drops on OOD data, especially when the start and goal positions are farther apart than in the training data. Surprisingly, however, we find that the choice of input and CoT formats strongly influences how sharply accuracy degrades. As shown in Fig. 1, reasoning traces based on *coordinates* + *description* and A*-like formats yield substantially better OOD performance than no CoT, *description*-only CoT, or even *coordinates*-only CoT on two challenging distribution shifts. Even on ID test maps, these models achieve state-of-the-art results, outperforming recent methods based on continuous-space reasoning (Yang et al., 2025) and specialized vision-only models (Xu et al., 2025), with our A*-like CoT model reaching 98% accuracy compared to 92% for prior methods (Table 5). Finally, we show that data augmentation is crucial for the strongest OOD results: augmenting the training maps with multiple solutions further improves OOD accuracy, reaching 85% on the most challenging fixed-size distance shift and 91% on larger maps with $d_\infty \geq 6$ (Fig. 1).

Overall, our study shows that both ID and OOD generalization in visual planning can be largely improved through appropriate multimodal data formats and simple augmentations. Our results therefore represent progress toward true algorithmic generalization in reasoning, moving beyond surface-level pattern matching.

2. Related Work

Reasoning LLMs achieve notable results on a variety of tasks (Chen et al., 2025), and understanding their functioning and limitations is an active area of research (Shojaee et al., 2025; Mirzadeh et al., 2025; Wang et al., 2025). In particular, Stechly et al. (2024) study the performance of state-of-the-art LLMs in text-based planning domain and simple synthetic tasks. In their case, CoT demonstrations are added in context rather than used for fine-tuning. Across tasks, they observe limited improvements due to CoT prompts, restricted to when these are very similar to the target test examples. Therefore, Stechly et al. (2024) argue that the model does not learn general algorithmic procedures via such demonstrations. Zhao et al. (2025) introduce DATAALCHEMY, a controllable environment which abstracts language tasks with simple symbolic inputs and transformations. In this way, they can control the OOD generalization along different distribution shifts, of both transformers trained from scratch and fine-tuned LLMs. While CoT reasoning is effective on in-distribution data, it quickly degrades under moderate shifts, again suggesting that the models learn to recognize patters from the training set rather than logical structures. In our work, we expand this line of research of studying OOD generalization of CoT reasoning: our tasks are trivial for humans but challenging

Image

Description

```
This is a 3x3 map.
The player is at:  row 3, column 2;
The hole(s) are at:  Row 2, Column 2;
Row 3, Column 3;
The goal is at:  Row 1, Column 2.
```

Grid

```
Grid:
. G .
. X .
. P X
```

Coordinates

```
Coordinates are zero-indexed as (row, col),
with (0,0) at the top-left corner.
The map is a 3 by 3 grid.
The player starts at (2,1) and the goal is
at (0,1).
Holes are located at {(1,1),(2,2)}.
Every coordinate not listed as a hole is a
safe cell.
```

Figure 2. **Maze representations.** Besides the *image* and *description* formats used by Wu et al. (2024), we introduce *grid* and *coordinates* as text-based map representations.

for SOTA LLMs, suitable for both training from scratch and fine-tuning, and can be extended to arbitrarily difficult levels. Unlike previous benchmarks such as DATAALCHEMY, our dataset let us study the interplay between input and CoT format, spanning both text and image representations, which has recently drawn attention in the context of multimodal reasoning models (Zhou et al., 2025). This reveals that in some cases significant OOD generalization can be achieved.

3. A Controlled Environment to Benchmark Generalization of Reasoning

3.1. Input and chain-of-thought format

Task. We consider the spatial planning task introduced in Wu et al. (2024), named FROZENLAKE. In this task, the model has to navigate a player through a fully observable maze. The input consists of a textual description of the rules and goal of the game, and a grid-based map of the maze, where one cell hosts the player, another the treasure, and the remaining cells are either empty or contain lakes. The model's output is a sequence of moves ("UP", "DOWN", "LEFT", "RIGHT") that should guide the player to the treasure without falling into a lake. This setting offers several advantages for our analysis: the input map can be easily represented as either an image or text, and the solutions have a clear visual and descriptive structure. Moreover, the difficulty of the task can be adjusted by, e.g., the size of the map or the distance between the start and the treasure: these controllable difficulty factors enable us to systematically examine out-of-distribution generalization of the models. Finally, despite its simplicity for humans, even powerful LLMs struggle to solve the task without fine-tuning (Wu et al., 2024), e.g. 46% accuracy for GPT-4o (Hurst et al., 2024) or 29% for Gemini (Gemini Team, 2023) (on average on small 3×3, 4×4, 5×5, and 6×6 maps), see Table 13.

Maze representations. In Wu et al. (2024), the planning task is introduced with several representations of the maze: as an *image*, an unstructured text *description* and as an ASCII *table* in Markdown-like format. We additionally introduce *(i)* an ASCII-based *grid* representation, which has a compact table-like structure, which can be considered a text-version of the image grid and *(ii)* a numerical *coordinates* representation that specifies the location of player, goal and lakes on the 2D grid but does not have any explicit visual structure (we further test a variant with alphanumerical coordinates in App. B.1). We use the four representations illustrated in Fig. 2 in the main part, and report results for additional formats in appendix. This setup allows us to use the same data both with text-only and multimodal LLMs.

Reasoning traces representations. The simplicity of the task makes the reasoning process toward a correct solution sequence straightforward, since each step corresponds to a step along a correct path to the treasure (more details on the construction of reasoning traces in App. A). This, together with the versatility of the data, enables generating multiple variants of chain-of-thought reasoning traces with different formats (see Fig. 3), whose effect on ID and OOD performance is discussed in Sec. 4.2.

- *Image*: Xu et al. (2025) use vision-only models, and their reasoning traces consist of images that show the current maze state after each move. While we do not directly employ this format in our experiments, as standard (multimodal) LLMs do not handle images in the reasoning process, it may be useful for future work on integrating visual elements in chain-of-thought.
- *Description*: This is a narration of the reasoning behind each step, discussing which moves would bring the player closer to the treasure, whether the path toward it is blocked by a lake, which are the feasible steps, and finally deciding which the next move should be.
- *Grid* and *coordinates*: We represent the current map, after the next move, with our two text-based representations discussed above, for each step of a path to the treasure.
- *Grid + description*, *coordinates + description*: We combine either *grid* or *coordinates* traces with *description*. Thus, the chain-of-thought contains first a narrated description of the reasoning process about what the next step should be, and then the *grid* or *coordinates* representation of the map after applying the selected move. These

	Image	Descriptive	Grid	Coordinates	A* CoT (coords)
Step 1		This is a 3x3 map. The direct path to the goal is moving UP. However, there is a hole UP, so I cannot move UP. As the direct path is blocked, I will move LEFT instead.	Grid: . G . . X . P . X	Coordinates: Player: (2,0) Goal: (0,1) Holes: {(1,1),(2,2)}	Grid: S=(2,1), G=(0,1), H={(1,1),(2,2)} Initial: OPEN=[(2,1)], CLOSED={} Step 1 OPEN before=[(2,1)] Pop first OPEN node: (2,1) L -> (2,0): add to OPEN; set parent[(2,0)]=(2,1) R -> (2,2): hole, skip U -> (1,1): hole, skip D -> (3,1): out of bounds, skip OPEN after=[(2,0)] CLOSED={(2,1)}
Step 2		The direct path to the goal is moving UP or RIGHT. Thus, I will move UP.	Grid: . G . P X . . . X	Coordinates: Player: (1,0) Goal: (0,1) Holes: {(1,1),(2,2)}	Step 2 OPEN before=[(2,0)] Pop first OPEN node: (2,0) L -> (2,-1): out of bounds, skip R -> (2,1): in CLOSED, skip U -> (1,0): add to OPEN; set parent[(1,0)]=(2,0) D -> (3,0): out of bounds, skip OPEN after=[(1,0)] CLOSED={(2,1),(2,0)}

Figure 3. **Reasoning traces in different formats.** We illustrate an example of the first two steps of reasoning traces in the various formats. While Xu et al. (2025) use the sequence of maze representations as *images*, we generate the corresponding steps as text-only *descriptions*, *grids*, *coordinates* and A* (with *coordinates*). In the *description* format at each step we formulate a discussion of which the next step should be, while for the *grid* and *coordinates* formats we have the visual representation of the map after the next move. The A* trace explicitly tracks the open and closed sets, parent pointers, and the skipped invalid moves at each step.

formats contain more information at the cost of being longer (and thus more expensive to generate at inference time, see Table 14).

- *A* with coordinates*: We also consider an algorithmic reasoning trace based on the exact trajectory of A* search in coordinate space. At each step, the trace writes the current search state as coordinate lists: the states waiting to be expanded, the states already expanded, and the parent links used to recover the final path. After the goal is reached, the trace follows the recorded parent links to reconstruct the path to the treasure. Compared to the traces above, this format directly exposes the search procedure rather than example solutions, but it is also longer (Table 14).

3.2. Solution multiplicity and training-set augmentation

Beyond varying the representation of a map and the corresponding reasoning trace, our framework also allows us to control the amount of solution supervision per map. While the standard training set pairs each map with a single solution trace (except A*-CoT models which take a different approach), FROZENLAKE maps often admit multiple correct paths to the goal. We therefore construct augmented training data variants that keep the same (1000) training maps fixed, but expose the model to multiple valid solutions per map. We consider three variants: up to three valid solutions per map, up to five valid solutions per map, and all optimal shortest solutions (total number of resulting training solutions in App. B.2). This factorizes the effect of solution diversity from the effect of seeing more maps: the environments are unchanged, while only the number and diversity of correct reasoning traces varies. Unless stated otherwise, experiments use the default one-solution-per-map setting. We study the effectiveness of these augmentation approaches in Sec. 4.3.

3.3. Measuring OOD generalization via challenging distribution shifts

We measure the out-of-distribution (OOD) generalization capabilities of the models along multiple axes, which control the difficulty of the task in different ways.

- **Map size:** First, by training on a range of different map sizes and testing on sizes not seen during training, we can evaluate whether the model generalizes to larger maps. Note that a larger map can contain the same maze as a smaller one, simply padded with empty cells or holes.
- **Start-goal distance:** Next, we consider the L_∞ distance between the start and the goal, i.e. for player coordinates (s_1, s_2) and treasure coordinates (g_1, g_2)

$$d_\infty = \max\{|s_1 - g_1|, |s_2 - g_2|\}.$$

  For instance, in the 3×3 example in Fig. 2, this distance equals 2. In general, a map of size $n \times n$ has $d_\infty \leq n - 1$. Therefore, we can test OOD generalization to maps with start-goal distance larger than what seen at training time.
- **Optimal solution length:** Finally, we can compute the length of the optimal path (this can be done via the A* algorithm too), and analyze how well models perform on maps where the optimal solution path is longer than those of the training set.

These factors, while connected (e.g., larger maps tend to induce larger start-goal distances and longer solutions), cap-

Table 1. **Influence of data format on ID and OOD generalization.** We report accuracy on both ID and OOD map sizes of zero-shot and fine-tuned LLMs using different combination of input and CoT traces (if any) formats. For OOD maps, we distinguish test sets where the start-goal distance is $d_\infty \geq 6$, i.e. not seen during training. *Coordinates*-based representation substantially outperforms other formats on OOD maps, especially when using *coordinates* + *description* CoT. A*-CoT over *coordinates* input achieves the highest ID accuracy.

Input format	CoT format	**ID test maps** ($d_\infty \leq 5$) 3x3	4x4	5x5	6x6	Avg	**OOD maps (random** d_∞**)** 7x7	8x8	9x9	10x10	Avg	**OOD maps** ($d_\infty \geq 6$) 7x7	8x8	9x9	10x10	Avg
Image	zero-shot	0.14	0.05	0.03	0.02	0.06	0.02	0.02	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00
	no CoT	0.89	0.84	0.72	0.52	0.74	0.51	0.37	0.23	0.12	0.31	0.02	0.00	0.00	0.00	0.01
	Descr.	0.95	0.87	0.76	0.62	0.80	0.47	0.35	0.21	0.14	0.29	0.07	0.03	0.00	0.01	0.03
Descr.	zero-shot	0.41	0.25	0.23	0.13	0.26	0.12	0.14	0.10	0.10	0.12	0.01	0.06	0.04	0.03	0.03
	no CoT	0.92	0.91	0.86	0.64	0.83	0.65	0.55	0.49	0.41	0.52	0.19	0.08	0.04	0.02	0.08
	Descr.	0.94	0.92	0.90	0.67	0.86	0.67	0.63	0.55	0.46	0.58	0.07	0.02	0.01	0.01	0.02
	Grid	0.96	0.89	0.90	0.70	0.86	0.59	0.47	0.28	0.12	0.36	0.35	0.17	0.06	0.01	0.15
	Grid + Descr.	0.98	0.93	0.93	0.69	0.88	0.70	0.51	0.41	0.26	0.47	0.44	0.28	0.11	0.07	0.23
Grid	zero-shot	0.14	0.13	0.06	0.03	0.09	0.02	0.01	0.00	0.02	0.01	0.00	0.01	0.01	0.01	0.01
	no CoT	0.95	0.89	0.83	0.62	0.82	0.55	0.39	0.34	0.26	0.38	0.04	0.02	0.00	0.00	0.01
	Descr.	0.94	0.91	0.86	0.62	0.83	0.53	0.41	0.35	0.25	0.39	0.10	0.07	0.01	0.01	0.05
	Grid	0.98	0.92	0.97	0.71	0.90	0.74	0.51	0.39	0.24	0.47	0.43	0.22	0.07	0.04	0.19
	Grid + Descr.	0.99	0.93	0.92	0.79	0.91	0.83	0.68	0.49	0.35	0.59	0.65	0.55	0.23	0.20	0.41
Coord.	zero-shot	0.30	0.21	0.22	0.10	0.21	0.14	0.09	0.08	0.08	0.09	0.04	0.03	0.05	0.03	0.03
	no CoT	0.94	0.89	0.92	0.72	0.87	0.78	0.60	0.55	0.47	0.60	0.21	0.06	0.02	0.01	0.07
	Descr.	0.98	0.96	0.96	0.75	0.91	0.72	0.63	0.55	0.49	0.59	0.14	0.13	0.04	0.04	0.09
	Coord.	**1.00**	0.93	0.94	0.84	0.93	0.93	0.88	0.88	0.83	0.88	0.83	0.80	0.68	0.70	0.75
	Coord. + Descr.	**1.00**	0.95	**0.98**	0.86	0.95	**0.97**	**0.93**	**0.90**	**0.88**	**0.92**	**0.89**	**0.87**	**0.81**	**0.77**	**0.83**
Coord.	A* w/ Coord.	**1.00**	**0.99**	0.96	**0.95**	**0.98**	0.96	0.91	0.88	0.82	0.89	0.87	0.76	0.73	0.62	0.74

ture different distribution shifts. This allows us to broadly study how reasoning LLMs for visual planning behave in face of truly unseen data, a key yet often overlooked factor for understanding the functioning of these models.

4. Experiments

4.1. Setup

Data. For training, we use the dataset introduced by Yang et al. (2025), which consists of 1,000 data points (100 3×3, 200 4×4, 300 5×5, and 400 6×6 maps). Thus, the training set only contains distances $d_\infty \leq 5$. For each training example we generate the input and reasoning traces in all formats described in Sec. 3. The in-distribution test sets (3×3, 4×4, 5×5, and 6×6 maps) are also identical to those in Yang et al. (2025), while we generated the out-of-distribution test sets via the Gym library (Brockman et al., 2016) as described in App. A.We generate 200 maps for each map size (or distance in the embedded maps case) in the corresponding OOD test sets. During evaluation, we extract the move sequence from the model output and simulate it in the environment. The sequence is correct if *(i)* the player does not enter a cell with a hole and *(ii)* the end position is on the treasure.

Models. As base model, we use Qwen2.5-VL-7B-Instruct (Bai et al., 2025), following the setup of Yang et al. (2025), including their prompt and rule specification. All models are adapted via supervised fine-tuning for 10 epochs (details in App. A). For each input format, we test the zero-shot performance of Qwen2.5-VL-7B-Instruct, and its versions fine-tuned without reasoning traces (no CoT). Then, we train models on various combinations of input and CoT formats.

4.2. Effect of input and CoT format

In-distribution performance. First, to test ID generalization, we measure the performance of the various models on maps of the same size as in the training set (3×3 to 6×6). Table 1 shows that zero-shot performance remains low across input formats, confirming that even small maps represent a challenging task for general-purpose LLMs, and FROZENLAKE was not part of the training set of the base model, making it well-suited for testing OOD behavior. *Image* inputs (with and without CoT) perform consistently worse than text-only ones, highlighting the limitations of current multimodal LLMs in fully leveraging visual inputs. CoT reasoning gives better accuracy than fine-tuning on the answers only, with the best results achieved by our *coordinates* input with *coordinates* + *description* (95% accuracy) and A*-like CoT (98%), which solves the task almost perfectly.

OOD generalization: map size. Our first OOD test set includes larger map sizes, not seen during training, i.e. 7×7, 8×8, 9×9, and 10×10. Similar to ID data, these maps are generated with random sampling of the initial position of

Table 2. **Generalization w.r.t. start-goal distance (d_∞) with fixed 10×10 map size.** We compare different input and CoT format when the training maps have fixed size 10x10 and $d_\infty < 6$, test maps size 10×10 and $d_\infty \in \{2, \dots, 5\}$ (ID) or $d_\infty \in \{6, \dots, 9\}$ (OOD). Coordinate-based CoTs improve distance extrapolation substantially over grid-based traces, and coordinate + description traces remain accurate well beyond the training distance range. A* traces over coordinates perform best in this fixed-map-size setting, achieving the strongest ID and OOD accuracy across distances.

Input format	CoT format	ID test maps ($d_\infty \leq 5$) 2	3	4	5	Avg	OOD maps ($d_\infty \geq 6$) 6	7	8	9	Avg
Image	no CoT	0.70	0.46	0.21	0.03	0.35	0.03	0.01	0.00	0.00	0.01
	Descr.	0.69	0.41	0.16	0.06	0.33	0.01	0.00	0.00	0.00	0.00
Descr.	no CoT	0.82	0.75	0.57	0.34	0.62	0.01	0.01	0.00	0.00	0.01
	Descr.	0.89	0.80	0.68	0.41	0.69	0.00	0.00	0.00	0.00	0.00
	Grid	0.82	0.76	0.67	0.49	0.69	0.34	0.20	0.10	0.04	0.17
	Grid + Descr.	0.82	0.73	0.61	0.47	0.66	0.36	0.25	0.17	0.14	0.23
Grid	no CoT	0.86	0.81	0.55	0.12	0.58	0.00	0.01	0.00	0.00	0.00
	Descr.	0.84	0.78	0.45	0.10	0.54	0.00	0.01	0.00	0.00	0.00
	Grid	0.89	0.84	0.77	0.62	0.78	0.46	0.34	0.17	0.08	0.26
	Grid + Descr.	0.86	0.84	0.74	0.61	0.76	0.50	0.33	0.21	0.10	0.29
Coord.	no CoT	0.90	0.83	0.69	0.43	0.71	0.01	0.00	0.00	0.00	0.00
	Descr.	0.84	0.73	0.65	0.32	0.63	0.00	0.00	0.00	0.00	0.00
	Coord.	0.89	0.85	0.79	0.64	0.79	0.53	0.39	0.31	0.21	0.36
	Coord. + Descr.	0.95	0.94	0.89	0.74	0.88	0.75	0.56	0.44	0.44	0.55
Coord.	A* w/ Coord.	**0.99**	**0.98**	**0.96**	**0.89**	**0.95**	**0.85**	**0.63**	**0.56**	**0.46**	**0.62**

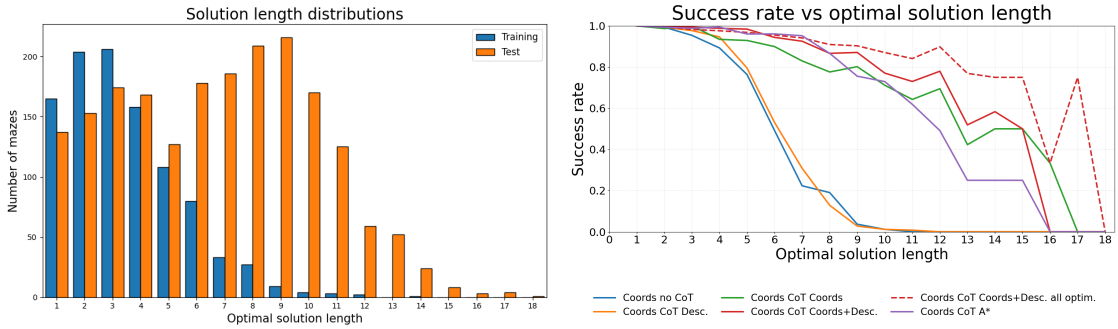

Figure 4. **OOD generalization w.r.t. optimal solution length. Left:** distribution of the length of the optimal (shortest) solution paths for both training and test maps (aggregated across map sizes). **Right:** success rate of models using *coordinates* input vs solution length.

player and treasure, without controlling for the start-goal distance (*random* d_∞ column in Table 1). Therefore, the start, goal and solution can fit into a map of size smaller than or equal to 6×6: these examples are OOD in terms of map size but not start-goal distance, as can be seen in Table 7 (in appendix). Similar to the ID case, models with text-based input significantly outperform the image-based ones. Again, models encoding maps in *coordinates* format with *coord.* (88% accuracy), *coord. + descr.* (92%) and A* CoT (89%) far outperform all others.

OOD generalization: start-goal distance. To ensure that both map size and start-goal distance are out-of-distribution, we generate new OOD maps enforcing $d_\infty \geq 6$, i.e., the distance between player and treasure is too large to fit in a 6×6 map. As shown in Table 1 (right column), in this more challenging distribution shift, the accuracy of most models quickly drops near 0% for larger maps. While the best results are still obtained, with a large margin, by the *coordinates* format (83% average accuracy), all models combining a map encoding and the text description in the reasoning process (*grid/coord. + descr.*) achieve the best performance among those with same input format. We conjecture that generating a (more or less "visual") representation of the current map (as *grid* or *coordinates*) at each step helps tracking the progress of the map navigation, while the reasoning in the natural language (the *description* CoT component) is better suited for the model to elaborate on the next move. Interestingly, the accuracy of the model using *description* format for both input and CoT sharply drops from 58% OOD maps with random d_∞ to 7% on 7×7 maps with $d_\infty \geq 6$: this demonstrates how our controlled setup can reveal severe limitations in generalization.

Table 3. **Effect of solution-set augmentation on generalization w.r.t. map size.** We report ID and OOD accuracy of LLMs fine-tuned on training sets with the same 1000 maps but different number of solutions per map. All models use *coord.* input and *coord. + descr.* CoT.

Training data variant	**ID test maps** ($d_\infty \leq 5$)					**OOD maps (random** d_∞**)**					**OOD maps** ($d_\infty \geq 6$)				
	3x3	4x4	5x5	6x6	Avg	7x7	8x8	9x9	10x10	Avg	7x7	8x8	9x9	10x10	Avg
1 sol/map (default)	**1.00**	0.95	0.98	0.86	0.95	0.97	0.93	0.90	0.88	0.92	0.89	0.87	0.81	0.77	0.83
up to 3 sol/map	**1.00**	**0.98**	0.98	0.87	**0.96**	0.98	0.96	**0.95**	**0.92**	**0.95**	0.92	**0.90**	0.87	0.85	0.89
up to 5 sol/map	**1.00**	0.96	**0.99**	**0.90**	**0.96**	**0.99**	0.95	0.94	**0.92**	**0.95**	0.92	0.89	0.82	0.83	0.86
all optimal solutions	**1.00**	0.94	0.94	0.84	0.93	0.97	**0.97**	**0.95**	**0.92**	**0.95**	**0.94**	0.88	**0.91**	**0.90**	**0.91**

Table 4. **Effect of solution-set augmentation on generalization w.r.t. start-goal distance (10x10 maps).** We report ID and OOD accuracy of LLMs fine-tuned on training sets on the same 1000 maps but with different number of solutions per map. All models use *coord.* input and *coord. + descr.* CoT).

Training data variant	**ID test maps** ($d_\infty \leq 5$)					**OOD maps** ($d_\infty \geq 6$)				
	2	3	4	5	Avg	6	7	8	9	Avg
1 sol/map (default)	0.95	0.94	0.89	0.74	0.88	0.75	0.56	0.44	0.44	0.55
up to 3 sol/map	0.94	0.95	0.92	0.87	0.92	0.80	0.69	0.52	0.48	0.62
up to 5 sol/map	0.95	0.97	**0.95**	**0.93**	**0.95**	**0.94**	0.88	**0.82**	0.77	**0.85**
all optimal solutions	**0.96**	**0.99**	**0.95**	0.89	**0.95**	0.92	**0.89**	0.81	**0.79**	**0.85**

OOD generalization: start-goal distance with fixed map size. To fully disentangle the effects of map size and d_∞, we embedded all training images (of size up to 6×6) randomly into 10×10 maps, which does not change the correct path but fixes the map size. Moreover, we generate a new test set of 10×10 maps with 200 samples for each value of $d_\infty \in \{2, \ldots, 9\}$. This setup allows us to analyze generalization w.r.t. start-goal distance in isolation, without requiring the model to learn or generalize to maps of different size. Table 2 shows the in-distribution ($d_\infty \leq 5$) and out-of-distribution ($d_\infty \geq 6$) accuracy for different data format. Similarly to the previous setups, *image* inputs yield worse results than text-based ones, and the combined CoT versions the best results. Moreover, for both no CoT and *description* CoT the accuracy drops to near 0% already at $d_\infty = 6$, i.e. the smallest distribution shift, indicating that the models do not typically learn an algorithmic solution to the task. Instead, the A*-like reasoning significantly outperforms other models in both ID (95%) and OOD (62%).

OOD generalization: optimal solution length. Next, we study generalization with regard to the length of optimal solution l. Fig. 4 (left plot) shows the distribution of the length of the shortest solution of each map for training and test set, aggregating the statistics over map sizes. As expected, including the larger OOD maps in the test set introduces a distribution shift towards longer solutions, with $l \leq 14$ for the training set, up to 18 for the test set. Fig. 4 (right) reports the accuracy of the models using *coordinates* input representations vs the solution length of the test maps. The accuracy of the LLMs with no CoT or *description* CoT drops near zero for $l \geq 9$, where very few training examples are available. Conversely, with *coordinates*, *coord. + descr.* and A* CoT the accuracy remains non-trivial until $l = 15$, yet failing afterwards. Therefore, generalization to maps with longer solutions is challenging, but even in this case our CoT representations greatly help achieving better accuracy. The results for all input formats can be found in Fig. 6.

4.3. Effect of solution-set augmentation

We then study the effect of the training data variants with multiple solutions per map discussed in Sec. 3.2. Since *coordinate* inputs with *coordinates + description* CoT is the most effective format, we then fine-tune LLMs with such formats of these three training sets. We report the results of these models, together with the default one-solution-per-map setting, in Table 3 (varying map size) and Table 4 (fixed map size). We see that the augmented training datasets yield in almost all cases notable improvements in both ID and, especially, OOD performance. In particular, using all optimal solutions increases OOD accuracy on larger maps with $d_\infty \geq 6$ from 83% to 91% (Table 3), and from 55% to 85% on OOD maps with fixed 10x10 size (Table 4). This setting gives improvements even in generalization w.r.t. optimal solution length, see Fig. 4 (right). This demonstrates that augmenting training maps with additional correct solutions is a complementary approach to optimizing data format for improving generalization to distribution shifts.

4.4. Additional analyses

Comparison to existing methods. Table 5 reports the result of existing approaches on ID maps. Li et al. (2025) fine-tune Anole (Chern et al., 2024) on a larger training set than ours for 40 epochs. Their approach, MVoT, which also includes images of the map in the reasoning traces, results in 86% accuracy. Moreover, VPFT (Xu et al., 2025) achieves 75.4% accuracy (exact match metric) with a specialized model,

Table 5. **Other baselines.** We report the performance of other baselines models and visual reasoning methods on the ID tasks (results taken from the previous works). Our best models with *Coord.* input format achieve the best accuracy on ID maps.

Model	ID test maps ($d_\infty \leq 5$)				
	3x3	4x4	5x5	6x6	Avg
GPT-4o (Xu et al., 2025)	0.68	0.58	0.35	0.24	0.46
Anole Direct (Li et al., 2025)	0.83	0.80	0.75	0.75	0.78
Anole CoT (Li et al., 2025)	0.94	0.72	0.50	0.39	0.64
MVoT (Li et al., 2025)	0.86	0.84	0.84	0.89	0.86
VPFT (Xu et al., 2025)	0.92	0.83	0.67	0.58	0.75
VPRL (Xu et al., 2025)	0.98	0.96	0.91	0.82	0.92
Coord., Coord. + Descr. CoT	**1.00**	0.95	**0.98**	0.86	0.95
Coord., A* CoT	**1.00**	**0.99**	0.96	**0.95**	**0.98**

Table 6. **Comparison to Mirage.** We compare the ID performance of models from Yang et al. (2025) to ours (20 epochs training, † trained for 10 epochs only). The continuous reasoning of Mirage does not provide benefit over our model trained without CoT data.

Model	ID test maps ($d_\infty \leq 5$)				
	3x3	4x4	5x5	6x6	Avg
Direct SFT (Yang et al., 2025)	0.88	0.81	0.73	0.47	0.72
CoT SFT (Yang et al., 2025)	0.68	0.53	0.35	0.31	0.47
Mirage Direct (Yang et al., 2025)	0.93	0.83	0.76	0.51	0.76
Mirage Direct (retrained)	0.91	0.82	**0.79**	0.51	0.76
Mirage Direct shuffled	0.92	0.85	0.82	0.50	0.77
Image, no CoT	0.91	0.85	0.76	0.59	0.78
Image, Descr. CoT †	**0.95**	**0.87**	0.76	**0.62**	**0.80**

LVM-7B (Bai et al., 2024), which only handles images as both input and output. VPRL (Xu et al., 2025) improves this to 91.6% by further fine-tuning with reinforcement learning (RL). Our models with *coordinates* input outperform all these previous approaches, achieving SOTA results (up to 98%). Since the models from Xu et al. (2025) are not available, we could not test their OOD generalization.

Comparison to Mirage. Mirage (Yang et al., 2025) integrates images into CoT by latent reasoning, i.e. the LLM can output tokens in continuous space. Mirage generates an encoded version of helper images, depicting the correct path to the goal, before giving the text-based solution. Yang et al. (2025) report that Mirage Direct, which relies on such continuous reasoning and supervised fine-tuning (SFT), outperforms SFT on the solution only (Direct SFT) and with text-based CoT (CoT SFT), all using FROZENLAKE maps as images. Table 6 compares the results reported in Yang et al. (2025) to our models also trained on *image* input format and no or *descr.* CoTs. Surprisingly, our model without CoT already outperforms Mirage Direct and Direct SFT, which use the same data: we hypothesize this is due to our better choice of training hyperparameters. The model with our *descrip.* CoT attains the best results (80% average accuracy), while Yang et al. (2025) report that this version, with their reasoning traces, largely fails (47%): we conjecture our more concise CoT formulation yields this improvement. Finally, we retrain Mirage Direct in the original setup and randomly shuffling the helper images of the training set: both get similar performance to the original Mirage Direct, suggesting that their continuous reasoning approach does not provide benefits on this task. Table 12 in appendix shows that the Mirage framework does not benefit OOD accuracy.

Other analyses. In App. B.2 we report how the average response length of correct solutions changes depending on the input and CoT formats (Table 14). Moreover, we show that increasing the number of training epochs does not affect the model performance (Fig. 5). Examples of reasoning traces generated by our models are provided in App. B.3.

5. Conclusion

Discussion. Our results confirm the observation of prior works (Stechly et al., 2024; Zhao et al., 2025) that chain-of-thought reasoning, while helping LLMs to generalize to in-distribution test samples, often fails under, even small, distribution shifts. This in turn indicates the models do not learn the algorithmic solution to the tasks but rather some form of pattern recognition and memorization. However, unlike existing works, we demonstrate that significant improvements can be achieved by carefully choosing format for input and reasoning, and augmenting the training maps with additional solutions. In this way, we obtain SOTA results on both in- and out-of-distribution data for the FROZENLAKE visual planning task. Finally, the effectiveness of A*-like CoT (which cover complex dynamics such as backtracking) and solution-set augmentation underlines the importance of rich reasoning data, with multiple and diverse behaviors, in supervised fine-tuning of LLMs.

Outlook. While the exact approaches we used to obtain these results may be not directly applicable to every planning or reasoning task, the rationale behind them as well as the takeaways from our experiments have the potential to benefit a large class of problems. Further, the flexibility of the environment we have introduced, in terms of data formats and distribution shifts, offers the possibility to construct even more complex OOD tasks and study a variety of approaches. Finally, we could study how techniques like latent reasoning and reinforcement learning affect generalization, and interact with different data formats.

Limitations. Following Yang et al. (2025) we focus on Qwen2.5-VL-7B-Instruct as our base model, since it is well-suited for extensive experiments on an academic budget, and allows us to study OOD performance in isolation. Other models may interact slightly differently with the various data formats, an interesting object for future work.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Anthropic. Introducing the next generation of claude, 2024. URL https://www.anthropic.com/news/claude-3-family.

Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Bai, Y., Geng, X., Mangalam, K., Bar, A., Yuille, A. L., Darrell, T., Malik, J., and Efros, A. A. Sequential modeling enables scalable learning for large vision models. In *CVPR*, 2024.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.

Chen, Q., Qin, L., Liu, J., Peng, D., Guan, J., Wang, P., Hu, M., Zhou, Y., Gao, T., and Che, W. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025.

Chern, E., Su, J., Ma, Y., and Liu, P. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024.

Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Ho, N., Schmid, L., and Yun, S.-Y. Large language models are reasoning teachers. In *ACL*, 2023.

Huang, W., Jia, B., Zhai, Z., Cao, S., Ye, Z., Zhao, F., Xu, Z., Hu, Y., and Lin, S. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

Li, C., Wu, W., Zhang, H., Xia, Y., Mao, S., Dong, L., Vulić, I., and Wei, F. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025.

Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., Clark, P., and Kalyan, A. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.

Mirzadeh, S. I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., and Farajtabar, M. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *ICLR*, 2025.

Shojaee, P., Mirzadeh, I., Alizadeh, K., Horton, M., Bengio, S., and Farajtabar, M. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. In *NeurIPS*, 2025.

Stechly, K., Valmeekam, K., and Kambhampati, S. Chain of thoughtlessness? an analysis of cot in planning. In *NeurIPS*, 2024.

von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

Wang, Y., Chang, F.-C., and Wu, P.-Y. A theoretical framework for ood robustness in transformers using gevrey classes. *arXiv preprint arXiv:2504.12991*, 2025.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022.

Wu, Q., Zhao, H., Saxon, M., Bui, T., Wang, W. Y., Zhang, Y., and Chang, S. Vsp: Assessing the dual challenges of perception and reasoning in spatial planning tasks for vlms. *arXiv preprint arXiv:2407.01863*, 2024.

Xu, Y., Li, C., Zhou, H., Wan, X., Zhang, C., Korhonen, A., and Vulić, I. Visual planning: Let's think only with images. *arXiv preprint arXiv:2505.11409*, 2025.

Yang, Z., Yu, X., Chen, D., Shen, M., and Gan, C. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025.

Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, 2024.

Zhao, C., Tan, Z., Ma, P., Li, D., Jiang, B., Wang, Y., Yang, Y., and Liu, H. Is chain-of-thought reasoning of llms a mirage? a data distribution lens. *arXiv preprint arXiv:2508.01191*, 2025.

Zhou, Y., Tu, H., Wang, Z., Wang, Z., Muennighoff, N., Nie, F., Choi, Y., Zou, J., Deng, C., Yan, S., et al. When visualizing is the first step to reasoning: Mira, a benchmark for visual chain-of-thought. *arXiv preprint arXiv:2511.02779*, 2025.

A. Data and Experimental Details

Data generation. For the OOD maps, we generate the maps randomly using a hole probability of 0.1. We generated 200 maps per map size (statistics about start-goal distance are reported in Table 7). Player and goal are placed at random, using rejection sampling in cases where we have a constraint on the distance between start and goal. Similarly, to embed the small maps into 10×10 size, we generate a random 10×10 map, remove player and treasure, and then place the small map randomly within the large one.

Construction of reasoning traces. To generate descriptive reasoning traces for each step of the solution sequence, we first identify the general direction from the player toward the treasure, which can include one or two of the four possible directions. In the 3×3 example, this direction is "UP." Next, we check whether this direction is blocked by a hole. If it is not, this move is selected; otherwise, we iterate over the remaining directions: first the other direction pointing toward the treasure (if it exists), and then the one not pointing towards the treasure. We enforce that the chosen direction in the reasoning trace always corresponds to the next move in the target action sequence of the training point to ensure consistency between reasoning and solution. The A* format always begins its chain-of-thought by stating its intention to solve the problem with the A* algorithm and writes down its choice of notation before beginning its simulation of the algorithm, as seen in Fig. 11.

Training setup. The statistics of the training data are reported in Table 8. For fine-tuning the models, we use the SFTTrainer from the trl (von Werra et al., 2020) package with the hyperparameters stated in Table 9. All models were trained on two A100 GPUs.

B. Additional experiments

B.1. Additional data formats

Results of additional formats. Table 10 and Table 11 report the results of all data formats on generalization to larger maps and larger start-goal distance with fixed map size respectively. The *table* format yields LLMs with behavior consistent with those using *grid*, yet typically worse results and longer, thus more computationally expensive, answers. The alphanumerical *coordinates (sheet)*, inspired by spreadsheets convention, perform similarly to or slightly worse than the numerical. Same observation holds for the A* reasoning in *coordinates (sheet)* format. This confirms that the coordinate format, with the different CoT variants, is relatively robust to small variations in the exact formulation used.

Generalization w.r.t. solution length for other formats. In Fig. 6 we report the success rate w.r.t. optimal solution

Table 7. **Average start-goal distance.** We report the average distance d_∞ between the start and goal positions for each map size. For the OOD maps with $d_\infty \geq 6$ we enforce that both map size and start-goal distance are out-of-distribution compared to the training data.

	ID test maps ($d_\infty \leq 5$)				**OOD maps (random** d_∞**)**				**OOD maps** ($d_\infty \geq 6$)			
	3x3	4x4	5x5	6x6	7x7	8x8	9x9	10x10	7x7	8x8	9x9	10x10
Mean d_∞	1.50	1.79	2.16	2.75	3.27	3.66	4.01	4.47	6.00	6.35	6.72	7.00

Table 8. **Statistics of training data.** We report the mean d_∞ as well as the mean length of the ground truth solution sequence on the training set.

	3x3	4x4	5x5	6x6	Total
Number of samples	100	200	300	400	1000
Mean d_∞	1.40	1.80	2.23	2.72	2.26
Mean solution length	2.03	2.65	3.58	4.18	3.48

Table 9. **SFT hyperparameters.** These values were used for all text-based models. The image models use the same hyperparameters as Yang et al. (2025)

Hyperparameter	**Value**
`epochs`	10
`per_device_train_batch_size`	1
`gradient_accumulation_steps`	1
`warmup_steps`	10
`learning rate`	1e-5
`weight_decay`	0.01
`optim`	AdamW
`bf16`	True

length for different input formats, similar to what shown in Fig. 4 for *coordinates*, with similar conclusions. As a further ablation, we re-train the model with *grid* input and *grid* + *description* CoT removing the examples with $l > 10$ from the training set (10 maps in total): this model (the dashed curve in the Fig. 6) performs similarly to the original one.

B.2. Other analyses

Extended comparison to Mirage on OOD maps. To complement the results in Sec. 4.4, Table 12 further reports the performance on OOD of the Mirage models (Yang et al., 2025) we retrained (the original ones are not available). Similar to ID performance, the continuous reasoning approach of Mirage does not provide benefit compared to our simple supervised fine-tuning with *image* inputs, with or without *description* CoT.

Length of reasoning. Table 14 reports the average token length of responses that lead to correct solutions. Reasoning length alone does not explain performance. In the main OOD map-size setting, the best-performing method is numerical *coordinates* + *description*, even though its responses are much shorter than A* traces and many *grid*/*table* multi-format CoTs. A* traces are substantially longer and achieve the strongest performance in the fixed 10×10 map size start-goal distance generalization setting, but they do not dominate in the OOD evaluation with varying map sizes. Likewise, *sheet-style coordinates* reduce output length compared to numerical coordinate formats, yet this token saving measure does not predictably affect accuracy: *coordinates* CoT perform similarly regardless of whether they are written numerically or in sheet-style, while numerical *coordinates* + *description* is clearly stronger than its sheet-style counterpart. The output lengths for solution-set augmentation variants are reported in Table 15. Unsurprisingly, models fine-tuned on variants that include suboptimal solutions (*up to 3 sol/map* and *up to 5 sol/map*) produce longer outputs than variants trained only on optimal solutions.

Effect of number of training epochs. To test the effect of longer training, we fine-tune LLMs with various data formats and with or without CoT for 20 and 30 epochs (as default we use 10 epochs). The detailed results are shown in Fig. 5 (in appendix). While there are minimal improvements after 20 epochs on the ID test set and after 30 epochs on the OOD test set, the overall performance and model rankings remain largely unchanged.

Detailed results of other methods. Table 13 reports the result of additional baselines models. We see that frontier LLMs such as Gemini-1.0-Pro-Vision (Gemini Team, 2023), Claude-3 (Anthropic, 2024) and GPT-4o (Hurst et al., 2024), have low zero-shot accuracy (the best is GPT-4o with 46%) on the ID test maps (results are taken from (Wu et al., 2024)). Further, we report the three fine-tuning variants from Li et al. (2025): *Direct*, *CoT*, and *MVoT*. They use Anole-7B (Chern et al., 2024) as base model which can generate interleaved text and images autoregressively. *Direct* corresponds to SFT without reasoning traces and achieves an accuracy of 78%. For *CoT*, the model is fine-tuned on reasoning traces incorporating coordinates and environment layout described in the text. In contrast to our experiments, they are not able to improve using such traces (accuracy 64%). *MVoT* also includes images of the grid in the reasoning traces and results in 86%. Note that these methods were fine-tuned on a larger training set for 40 epochs each. Finally, VPFT (Xu et al., 2025) achieves 75.4% accuracy (exact match metric) with a specialized model, LVM-7B (Bai et al., 2024), which only handles images as both input and output. VPRL (Xu et al., 2025) improves this to 91.6% by further fine-tuning with reinforcement learning (RL). This is similar to what we achieve via supervised fine-tuning alone of the

Table 10. **Influence of data format on ID and OOD generalization.** We report accuracy on both ID and OOD map sizes of zero-shot and fine-tuned LLMs using different combination of input and CoT traces (if any) formats. For OOD maps, we distinguish test sets where the distance between start and goal position is $d_\infty \geq 6$, i.e. not seen during training. Models provided with coordinate input and using coordinate CoT formats (*coordinates*/*coordinates + description*) lead to the best out-of-distribution performance, while A* formats attain the highest in-distribution performance, with both categories achieving substantial accuracy until 10×10 maps while being trained on up to 6×6 maps.

Input format	CoT format	**ID test maps** ($d_\infty \leq 5$)					**OOD maps (random** d_∞**)**					**OOD maps** ($d_\infty \geq 6$)				
		3x3	4x4	5x5	6x6	Avg	7x7	8x8	9x9	10x10	Avg	7x7	8x8	9x9	10x10	Avg
Image	zero-shot	0.14	0.05	0.03	0.02	0.06	0.02	0.02	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00
	no CoT	0.89	0.84	0.72	0.52	0.74	0.51	0.37	0.23	0.12	0.31	0.02	0.00	0.00	0.00	0.01
	Descr.	0.95	0.87	0.76	0.62	0.80	0.47	0.35	0.21	0.14	0.29	0.07	0.03	0.00	0.01	0.03
Descr.	zero-shot	0.41	0.25	0.23	0.13	0.26	0.12	0.14	0.10	0.10	0.12	0.01	0.06	0.04	0.03	0.03
	no CoT	0.92	0.91	0.86	0.64	0.83	0.65	0.55	0.49	0.41	0.52	0.19	0.08	0.04	0.02	0.08
	Descr.	0.94	0.92	0.90	0.67	0.86	0.67	0.63	0.55	0.46	0.58	0.07	0.02	0.01	0.01	0.02
	Grid	0.96	0.89	0.90	0.70	0.86	0.59	0.47	0.28	0.12	0.36	0.35	0.17	0.06	0.01	0.15
	Table	0.99	0.90	0.93	0.69	0.88	0.64	0.45	0.33	0.16	0.40	0.28	0.16	0.07	0.02	0.13
	Grid + Descr.	0.98	0.93	0.93	0.69	0.88	0.70	0.51	0.41	0.26	0.47	0.44	0.28	0.11	0.07	0.23
	Table + Descr.	0.93	0.93	0.93	0.71	0.88	0.66	0.52	0.40	0.32	0.47	0.31	0.21	0.12	0.09	0.18
Table	zero-shot	0.43	0.24	0.20	0.08	0.24	0.09	0.08	0.04	0.04	0.06	0.04	0.03	0.01	0.01	0.02
	no CoT	0.94	0.89	0.83	0.59	0.81	0.56	0.41	0.35	0.27	0.40	0.02	0.01	0.00	0.00	0.01
	Descr.	0.96	0.94	0.83	0.69	0.85	0.57	0.39	0.34	0.23	0.39	0.04	0.03	0.00	0.00	0.02
	Table	0.99	0.90	0.93	0.72	0.89	0.73	0.51	0.35	0.19	0.45	0.38	0.24	0.07	0.03	0.18
	Table + Descr.	0.97	0.93	0.95	0.78	0.91	0.81	0.57	0.41	0.27	0.51	0.48	0.32	0.12	0.06	0.25
Grid	zero-shot	0.14	0.13	0.06	0.03	0.09	0.02	0.01	0.00	0.02	0.01	0.00	0.01	0.01	0.01	0.01
	no CoT	0.95	0.89	0.83	0.62	0.82	0.55	0.39	0.34	0.26	0.38	0.04	0.02	0.00	0.00	0.01
	Descr.	0.94	0.91	0.86	0.62	0.83	0.53	0.41	0.35	0.25	0.39	0.10	0.07	0.01	0.01	0.05
	Grid	0.98	0.92	0.97	0.71	0.90	0.74	0.51	0.39	0.24	0.47	0.43	0.22	0.07	0.04	0.19
	Grid + Descr.	0.99	0.93	0.92	0.79	0.91	0.83	0.68	0.49	0.35	0.59	0.65	0.55	0.23	0.20	0.41
Coord.	zero-shot	0.30	0.20	0.18	0.10	0.20	0.13	0.08	0.08	0.11	0.10	0.04	0.05	0.04	0.02	0.03
	no CoT	0.94	0.89	0.92	0.72	0.87	0.78	0.60	0.55	0.47	0.60	0.21	0.06	0.02	0.01	0.07
	Descr.	0.98	0.96	0.96	0.75	0.91	0.72	0.63	0.55	0.49	0.59	0.14	0.13	0.04	0.04	0.09
	Coord.	**1.00**	0.93	0.94	0.84	0.93	0.93	0.88	0.88	0.83	0.88	0.83	0.80	0.68	0.70	0.75
	Coord. + Descr.	**1.00**	0.95	**0.98**	0.86	0.95	**0.97**	0.93	**0.90**	**0.88**	**0.92**	**0.89**	**0.87**	**0.81**	**0.77**	**0.83**
Coord. (sheet)	zero-shot	0.18	0.14	0.13	0.06	0.13	0.11	0.10	0.06	0.11	0.09	0.07	0.05	0.05	0.05	0.05
	no CoT	0.91	0.89	0.89	0.68	0.84	0.69	0.59	0.50	0.41	0.55	0.03	0.05	0.03	0.02	0.03
	Descr.	0.95	0.92	0.90	0.73	0.88	0.71	0.59	0.49	0.43	0.55	0.09	0.05	0.01	0.01	0.04
	Coord.	**1.00**	0.94	0.97	0.85	0.94	0.94	0.93	0.87	0.80	0.88	0.82	0.81	0.68	0.62	0.73
	Coord. + Descr.	0.97	0.95	0.95	0.79	0.92	0.90	0.87	0.79	0.81	0.84	0.78	0.75	0.65	0.63	0.70
Coord.	A* w/ coord.	**1.00**	**0.99**	0.96	**0.95**	**0.98**	0.96	0.91	0.88	0.82	0.89	0.87	0.76	0.73	0.62	0.74
Coord. (sheet)	A* w/ coord.	**1.00**	0.98	0.96	0.94	0.97	0.95	**0.94**	0.84	0.78	0.87	0.86	0.77	0.63	0.59	0.71

Table 11. **Generalization w.r.t. start-end distance (d_∞) with fixed 10×10 map size.** We compare different input and CoT format when the training maps have fixed size 10x10 and $d_\infty < 6$, test maps size 10×10 and $d_\infty \in \{2, \dots, 5\}$ (ID) or $d_\infty \in \{6, \dots, 9\}$ (OOD). Using *coordinates* input with *coordinates*, *coordinates + description*, or *A** CoT format yields the best results, with numerical coordinates outperforming their sheet-style counterparts; *grid* input combined *grid* or *grid + description* CoT format yield lower yet nontrivial results, while the accuracy of models with no CoT or CoTs in other formats drop close to 0% already at $d_\infty = 6$.

Input format	CoT format	ID test maps ($d_\infty \leq 5$) 2	3	4	5	Avg	OOD maps ($d_\infty \geq 6$) 6	7	8	9	Avg
Image	no CoT	0.70	0.46	0.21	0.03	0.35	0.03	0.01	0.00	0.00	0.01
	Descr.	0.69	0.41	0.16	0.06	0.33	0.01	0.00	0.00	0.00	0.00
Descr.	no CoT	0.82	0.75	0.57	0.34	0.62	0.01	0.01	0.00	0.00	0.01
	Descr.	0.89	0.80	0.68	0.41	0.69	0.00	0.00	0.00	0.00	0.00
	Grid	0.82	0.76	0.67	0.49	0.69	0.34	0.20	0.10	0.04	0.17
	Grid + Descr.	0.82	0.73	0.61	0.47	0.66	0.36	0.25	0.17	0.14	0.23
Table	no CoT	0.88	0.77	0.54	0.16	0.59	0.01	0.00	0.00	0.00	0.00
	Descr.	0.89	0.76	0.47	0.09	0.55	0.00	0.01	0.01	0.00	0.00
Grid	no CoT	0.86	0.81	0.55	0.12	0.58	0.00	0.01	0.00	0.00	0.00
	Descr.	0.84	0.78	0.45	0.10	0.54	0.00	0.01	0.00	0.00	0.00
	Grid	0.89	0.84	0.77	0.62	0.78	0.46	0.34	0.17	0.08	0.26
	Grid + Descr.	0.86	0.84	0.74	0.61	0.76	0.50	0.33	0.21	0.10	0.29
Coord.	zero-shot	0.11	0.06	0.06	0.05	0.07	0.07	0.03	0.01	0.01	0.03
	no CoT	0.90	0.83	0.69	0.43	0.71	0.01	0.00	0.00	0.00	0.00
	Descr.	0.84	0.73	0.65	0.32	0.63	0.00	0.00	0.00	0.00	0.00
	Coord.	0.89	0.85	0.79	0.64	0.79	0.53	0.39	0.31	0.21	0.36
	Coord. + Descr.	0.95	0.94	0.89	0.74	0.88	0.75	0.56	0.44	0.44	0.55
Coord. (sheet)	zero-shot	0.15	0.11	0.05	0.06	0.09	0.04	0.04	0.03	0.04	0.04
	no CoT	0.83	0.71	0.64	0.35	0.63	0.00	0.00	0.00	0.00	0.00
	Descr.	0.82	0.66	0.59	0.30	0.59	0.02	0.00	0.00	0.00	0.00
	Coord.	0.97	0.94	0.92	0.85	0.92	0.77	0.61	0.41	0.40	0.54
	Coord. + Descr.	0.84	0.74	0.65	0.50	0.68	0.47	0.36	0.26	0.27	0.34
Coord.	A* w/ coord.	**0.99**	0.98	**0.96**	**0.89**	**0.95**	**0.85**	0.63	**0.56**	**0.46**	**0.62**
Coord. (sheet)	A* w/ coord.	**0.99**	**0.99**	0.95	0.88	**0.95**	0.77	**0.67**	0.50	0.40	0.58

general-purpose Qwen2.5-VL-7B-Instruct with *grid/table + description* CoT (91%), see Table 1. Since the models from Xu et al. (2025) are not available, we could not test their OOD generalization.

Details about training data variants. In the previous experiments we have used a fixed training set of 1000 examples (see App. A), where for each map a single solution is shown to the model (with the exception of the A* CoT models which take a different approach). However, the solution path is not generally unique. Then, we test the effect of augmenting the training set with additional solutions for the same set of 1k maps. In particular we consider three configurations: *(i)* we sample up to three solutions for each map, if possible, which are up to four steps longer than the optimal one (total of training samples increases to 2761 when training on 3x3 to 6x6 maps, 2914 for the embedded 10x10 maps), *(ii)* we sample up to five solutions for each map as above (4173 and 4656 training samples), *(iii)* we use all optimal solutions, i.e. with shortest path, for each map (2756 and 2750 training samples). Since *coordinate* inputs with *coordinates + description* CoT is the most effective format, we then fine-tune LLMs with such formats of the three training sets described above. We keep 10 epochs regardless of the number of training samples, therefore the number of training steps effectively increases.

B.3. Additional examples

Examples of generated CoT. We show examples of CoT in *grid + description*, *coordinates + description* and A* formats our models generated for the same OOD map in Figs. 7, 8 and 9 respectively.

Examples of other input and trace formats. Examples of the *table* and *sheet-style coordinate* input formats are shown in Fig. 10, and examples of their respective trace formats, along with the sheet-coordinate variant of the A* CoT, are shown in Fig. 11.

Table 12. **Comparison to Mirage.** We additionally report the OOD performance of the models of Table 6. The latent reasoning approach of Mirage does not provide better OOD generalization than simple supervised fine-tuning without CoT.

Model	**ID test maps** ($d_\infty \leq 5$)					**OOD maps (random** d_∞**)**					**OOD maps** ($d_\infty \geq 6$)				
	3x3	4x4	5x5	6x6	Avg	7x7	8x8	9x9	10x10	Avg	7x7	8x8	9x9	10x10	Avg
Direct SFT (Yang et al., 2025)	0.88	0.81	0.73	0.47	0.72	-	-	-	-	-	-	-	-	-	-
CoT SFT (Yang et al., 2025)	0.68	0.53	0.35	0.31	0.47	-	-	-	-	-	-	-	-	-	-
Mirage Direct (Yang et al., 2025)	0.93	0.83	0.76	0.51	0.76	-	-	-	-	-	-	-	-	-	-
Mirage Direct (retrained)	0.91	0.82	0.79	0.51	0.76	0.45	0.35	0.22	0.15	0.29	0.01	0.01	0.01	0.00	0.00
Mirage Direct shuffled	0.92	0.85	0.82	0.5	0.77	0.46	0.41	0.26	0.14	0.31	0.05	0.06	0.02	0.00	0.03
Image, no CoT (ours)	0.91	0.85	0.76	0.59	0.78	0.46	0.41	0.23	0.16	0.31	0.04	0.01	0.00	0.00	0.01
Image, Descr. CoT (ours)	0.95	0.87	0.76	0.62	0.80	0.47	0.35	0.21	0.14	0.29	0.07	0.03	0.00	0.01	0.03

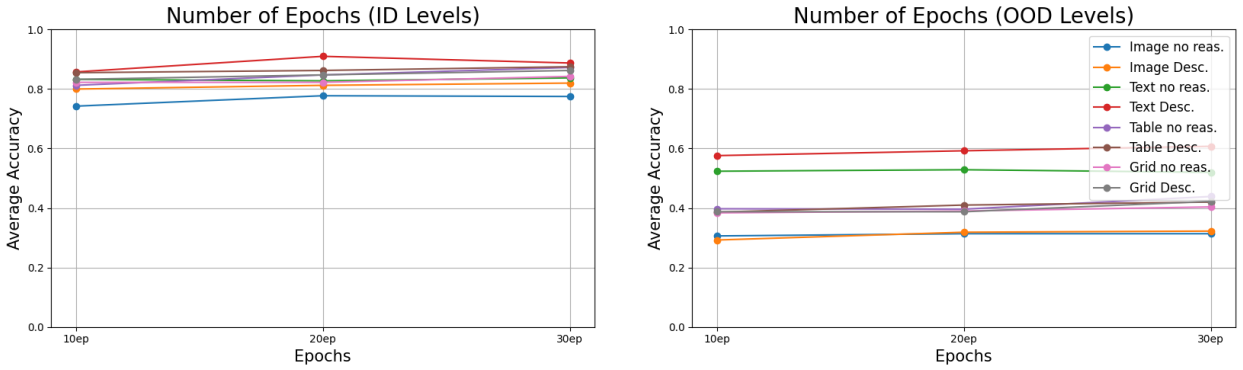

Figure 5. **Number of epochs.** We extended the fine-tuning to 20 and 30 epochs for some of the models and evaluated them on the ID and the OOD (random d_∞) test maps. However, the overall performance and model ranking remain largely unchanged.

Table 13. **Other baselines.** We report the performance of other baselines models and visual reasoning methods on the ID tasks. The results are taken from the previous works reporting them.

Model	**ID test maps** ($d_\infty \leq 5$)				
	3x3	4x4	5x5	6x6	Avg
Gemini-1.0 (Xu et al., 2025)	0.31	0.26	0.15	0.06	0.20
Claude-3 (Xu et al., 2025)	0.52	0.33	0.16	0.15	0.29
GPT-4o (Xu et al., 2025)	0.68	0.58	0.35	0.24	0.46
Anole Direct (Li et al., 2025)	0.83	0.80	0.75	0.75	0.78
Anole CoT (Li et al., 2025)	0.94	0.72	0.50	0.39	0.64
MVoT (Li et al., 2025)	0.86	0.84	0.84	0.89	0.86
VPFT (Xu et al., 2025)	0.92	0.83	0.67	0.58	0.75
VPRL (Xu et al., 2025)	0.98	0.96	0.91	0.82	0.92

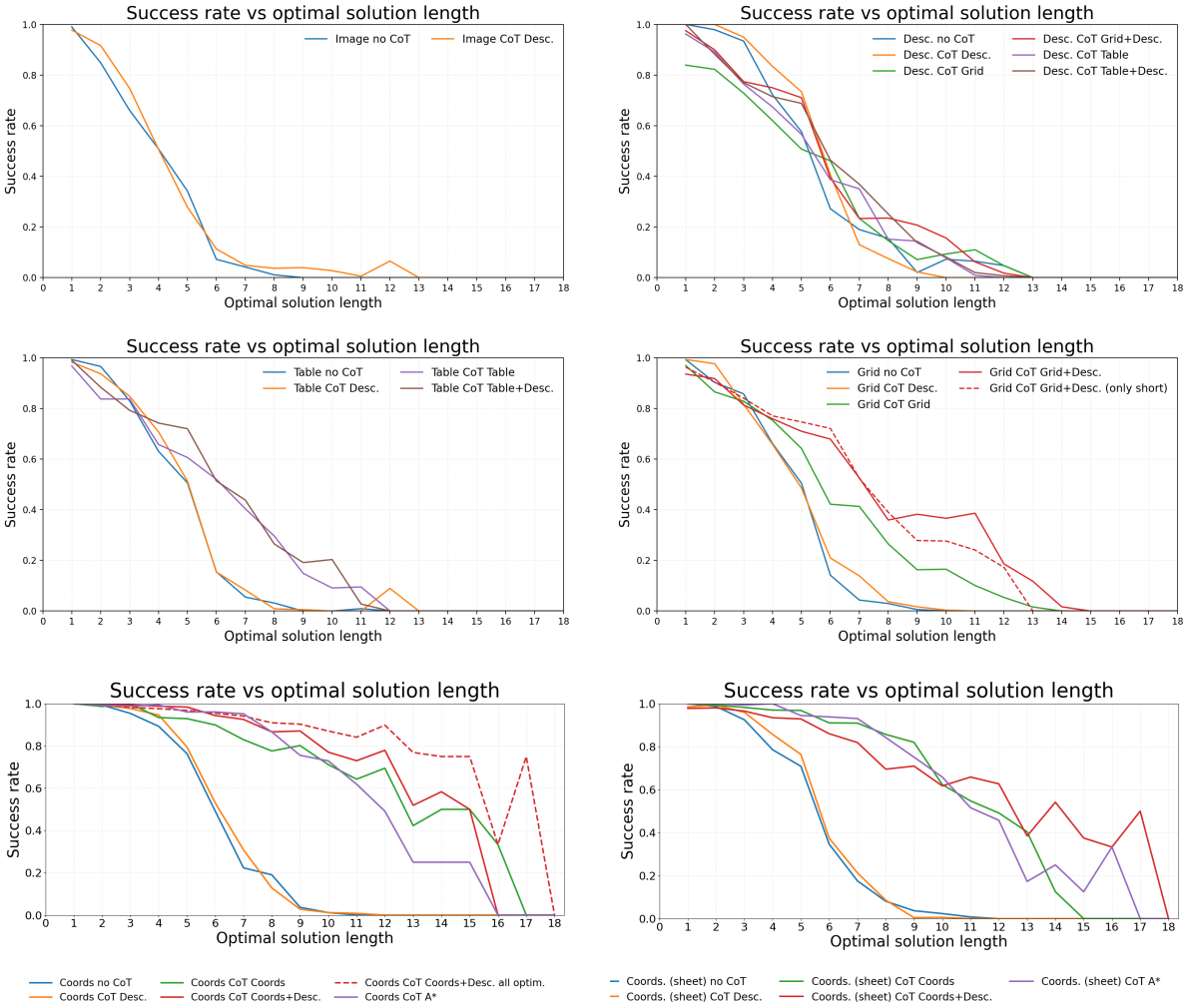

Figure 6. **OOD generalization w.r.t. optimal solution length.** We show success rate of models using different input and CoT representation over solution length, complementing the results of Fig. 4 for the *grid* format.

Table 14. **Output length.** We report the average length (in tokens) of the correct solutions, including reasoning, our the various fine-tuned models for each map size.

Input format	CoT format	**ID test maps** ($d_\infty \leq 5$)					**OOD maps (random** d_∞**)**					**OOD maps** ($d_\infty \geq 6$)				
		3x3	4x4	5x5	6x6	Avg	7x7	8x8	9x9	10x10	Avg	7x7	8x8	9x9	10x10	Avg
Image	zero-shot	210	429	309	363	328	450	590	662	547	562	524	457	774	502	564
	no CoT	9	10	10	11	10	12	11	10	9	11	19	-	-	-	19
	Descr.	52	59	64	86	65	81	74	65	54	69	196	217	-	162	192
Descr.	zero-shot	222	264	505	541	383	588	836	832	675	732	848	791	1010	833	870
	no CoT	9	10	11	12	10	13	13	13	13	13	21	19	18	22	20
	Descr.	52	62	73	87	69	86	90	91	90	89	156	127	122	157	141
	Grid	54	98	176	310	160	472	554	666	652	586	850	1023	1281	2116	1317
	Table	155	272	497	824	437	1136	1307	1534	1268	1311	2071	2534	2967	4387	2990
	Grid + Descr.	97	148	236	369	213	623	621	772	1069	771	1000	1371	1676	2413	1615
	Table + Descr.	188	320	527	848	471	1213	1451	1760	2205	1657	2238	2613	3289	4353	3123
Table	zero-shot	227	384	701	540	463	910	684	951	1027	893	848	918	886	1097	937
	no CoT	9	10	11	12	10	12	12	11	11	11	20	19	-	-	19
	Descr.	52	64	68	91	69	82	80	76	64	75	171	189	-	-	180
	Table	152	266	482	810	428	1242	1436	1687	1701	1517	2166	2652	3383	3498	2925
	Table + Descr.	192	324	544	899	490	1265	1633	1982	2755	1909	2333	2979	3813	6489	3904
Grid	zero-shot	355	668	1030	1086	785	1209	1314	1459	1567	1387	1067	1421	1523	1340	1338
	no CoT	9	10	11	12	10	12	11	11	11	11	19	18	-	-	19
	Descr.	52	61	72	83	67	81	80	79	71	77	161	145	171	150	157
	Grid	53	98	179	300	158	457	614	659	744	619	871	1081	1251	1572	1194
	Grid + Descr.	98	151	239	393	220	582	751	882	1109	831	1025	1298	1632	1880	1459
Coord.	zero-shot	138	170	151	170	157	176	223	275	225	225	297	230	258	238	256
	no CoT	11	12	13	14	12	15	15	15	15	15	22	22	20	21	21
	Descr.	54	63	76	90	71	89	88	93	96	92	160	146	137	173	154
	Coord.	64	86	131	193	118	203	254	302	381	285	357	425	516	598	474
	Coord. + Descr.	101	134	184	268	172	287	348	399	506	385	512	581	707	785	646
Coord. (sheet)	zero-shot	134	116	144	137	133	143	143	175	161	156	169	171	182	191	178
	no CoT	11	12	13	14	12	15	15	15	15	15	24	21	23	23	23
	Descr.	52	62	72	88	68	90	89	88	90	89	138	152	150	122	140
	Coord.	52	67	94	136	87	143	174	195	246	190	242	284	332	384	310
	Coord. + Descr.	86	111	152	204	138	219	261	294	375	287	390	440	515	576	480
Coord.	A* w/ coord.	476	589	724	960	687	1090	1221	1296	1484	1273	1856	2002	2400	2525	2196
Coord. (sheet)	A* w/ coord. (sheet)	392	478	548	716	534	801	899	931	1055	922	1364	1418	1703	1794	1570

Table 15. **Output length for solution-set augmentation variants.** We report the average length, in tokens, of correct solutions for *coordinates + description* models fine-tuned on the same 1000 maps but with different numbers of valid solutions per map.

Training data variant	**ID test maps** ($d_\infty \leq 5$)					**OOD maps (random** d_∞**)**					**OOD maps** ($d_\infty \geq 6$)				
	3x3	4x4	5x5	6x6	Avg	7x7	8x8	9x9	10x10	Avg	7x7	8x8	9x9	10x10	Avg
1 sol/map (default)	101	134	184	268	172	287	348	399	506	385	512	581	707	785	646
up to 3 sol/map	148	183	238	308	219	316	393	442	554	426	509	604	735	817	666
up to 5 sol/map	157	181	249	320	227	336	407	475	567	446	524	614	733	822	673
all optimal solutions	101	131	180	267	170	288	355	419	525	397	504	589	712	650	614

```
The direct path to the goal is moving UP or RIGHT.
Thus, I will move UP.
Grid:
 .  .  .  .  .  X  .  .
 .  X  .  .  .  .  X  .
 .  .  .  .  .  .  .  .
 .  .  .  .  .  .  X  .
 .  .  .  .  .  .  G  .
 .  X  .  .  .  .  .  X
 .  .  .  .  .  P  .  .
 .  .  X  .  .  .  .  .

The direct path to the goal is moving UP or RIGHT.
Thus, I will move UP.
Grid:
 .  .  .  .  .  X  .  .
 .  X  .  .  .  .  X  .
 .  .  .  .  .  .  .  .
 .  .  .  .  .  .  X  .
 .  .  .  .  .  .  G  .
 .  X  .  .  .  P  .  X
 .  .  .  .  .  .  .  .
 .  .  X  .  .  .  .  .

The direct path to the goal is moving UP or RIGHT.
Thus, I will move UP.
Grid:
 .  .  .  .  .  X  .  .
 .  X  .  .  .  .  X  .
 .  .  .  .  .  .  .  .
 .  .  .  .  .  .  X  .
 .  .  .  .  .  P  G  .
 .  X  .  .  .  .  .  X
 .  .  .  .  .  .  .  .
 .  .  X  .  .  .  .  .

The direct path to the goal is moving RIGHT.
Thus, I will move RIGHT.
Grid:
 .  .  .  .  .  X  .  .
 .  X  .  .  .  .  X  .
 .  .  .  .  .  .  .  .
 .  .  .  .  .  .  X  .
 .  .  .  .  .  .  P  .
 .  X  .  .  .  .  .  X
 .  .  .  .  .  .  .  .
 .  .  X  .  .  .  .  .

Final answer: \\boxed{UP, UP, UP, RIGHT}
```

Figure 7. **Example of Grid + Description CoT reasoning on an OOD map.** We show an example of the reasoning trace produced by the model trained on *grid* input and *grid + description* CoT. Even on an OOD map, the model first reasons in natural language on the next move, then produces the map after such move.

```
The direct path to the goal is moving UP or RIGHT.
Thus, I will move UP.
Coordinates:
Player: (6,5)
Goal: (4,6)
Holes: {(0,5),(1,1),(1,6),(3,6),(5,1),(5,7),(7,2)}

The direct path to the goal is moving UP or RIGHT.
Thus, I will move UP.
Coordinates:
Player: (5,5)
Goal: (4,6)
Holes: {(0,5),(1,1),(1,6),(3,6),(5,1),(5,7),(7,2)}

The direct path to the goal is moving UP or RIGHT.
Thus, I will move UP.
Coordinates:
Player: (4,5)
Goal: (4,6)
Holes: {(0,5),(1,1),(1,6),(3,6),(5,1),(5,7),(7,2)}

The direct path to the goal is moving RIGHT.
Thus, I will move RIGHT.
Coordinates:
Player: (4,6)
Goal: (4,6)
Holes: {(0,5),(1,1),(1,6),(3,6),(5,1),(5,7),(7,2)}

Final answer: \\boxed{UP, UP, UP, RIGHT}
```

Figure 8. **Example of Coordinates + Description reasoning on an OOD map.** We show an example of the reasoning trace produced by the model trained on *coordinates* input and *coordinates + description* CoT. On an OOD map, the model correctly reasons in natural language about each successive move and correctly updates the state of the relevant positions on the map after each step.

```
This is a shortest-path grid problem, so I can solve it with the A* algorithm. Let's simulate it.
Coordinates are zero-indexed as (row,col), with (0,0) at the top-left.
Actions: L=left, R=right, U=up, D=down.
OPEN is always written in the order A* would consider nodes, so the first entry is the next node to pop.
If several nodes are equally preferred, keep the earlier inserted one first. Neighbor order: L,R,U,D.

Grid: S=(7,5), G=(4,6), H={(0,5),(1,1),(1,6),(3,6),(5,1),(5,7),(7,2)}

Initial:
OPEN=[(7,5)]
CLOSED={}

Step 1
OPEN before=[(7,5)]
Pop first OPEN node: (7,5)
L -> (7,4): add to OPEN; set parent[(7,4)]=(7,5)
R -> (7,6): add to OPEN; set parent[(7,6)]=(7,5)
U -> (6,5): add to OPEN; set parent[(6,5)]=(7,5)
D -> (8,5): out of bounds, skip
OPEN after=[(7,6), (6,5), (7,4)]
CLOSED={(7,5)}

Step 2
OPEN before=[(7,6), (6,5), (7,4)]
Pop first OPEN node: (7,6)
L -> (7,5): in CLOSED, skip
R -> (7,7): add to OPEN; set parent[(7,7)]=(7,6)
U -> (6,6): add to OPEN; set parent[(6,6)]=(7,6)
D -> (8,6): out of bounds, skip
OPEN after=[(6,6), (6,5), (7,7), (7,4)]
CLOSED={(7,5),(7,6)}

Step 3
OPEN before=[(6,6), (6,5), (7,7), (7,4)]
Pop first OPEN node: (6,6)
L -> (6,5): already in OPEN with best or equal route, keep old parent
R -> (6,7): add to OPEN; set parent[(6,7)]=(6,6)
U -> (5,6): add to OPEN; set parent[(5,6)]=(6,6)
D -> (7,6): in CLOSED, skip
OPEN after=[(5,6), (6,5), (6,7), (7,7), (7,4)]
CLOSED={(7,5),(7,6),(6,6)}

Step 4
OPEN before=[(5,6), (6,5), (6,7), (7,7), (7,4)]
Pop first OPEN node: (5,6)
L -> (5,5): add to OPEN; set parent[(5,5)]=(5,6)
R -> (5,7): hole, skip
U -> (4,6): add to OPEN; set parent[(4,6)]=(5,6)
D -> (6,6): in CLOSED, skip
OPEN after=[(4,6), (6,5), (5,5), (6,7), (7,7), (7,4)]
CLOSED={(7,5),(7,6),(6,6),(5,6)}

Step 5
OPEN before=[(4,6), (6,5), (5,5), (6,7), (7,7), (7,4)]
Pop first OPEN node: (4,6)
Goal reached.

Parent path:
(7,5) -> (7,6) -> (6,6) -> (5,6) -> (4,6)
Actions: RIGHT, UP, UP, UP

Final answer: \\boxed{RIGHT, UP, UP, UP}
```

Figure 9. **Example of A* CoT reasoning on an OOD map.** We show an example of the reasoning trace produced by the model trained on *coordinates* input and *A** CoT. On an OOD map, the model still correctly states its intention to solve the map by simulating the A* algorithm along with its choice of notation, and correctly simulates the algorithm in each step.

Image	Table	Coordinates (sheet)

	`|  | Col 1 | Col 2 | Col 3 |`
	`| Row 1 |  _  |  *  |  _  |`
	`| Row 2 |  _  |  #  |  _  |`
	`| Row 3 |  _  |  @  |  #  |`
		`Coordinates use sheet-style labels: a1 is the top-left corner, letters move right, and numbers move down.`
		`The map is a 3 by 3 grid.`
		`The player starts at b3 and the goal is at b1.`
		`Holes are located at {b2,c3}.`
		`Every coordinate not listed as a hole is a safe cell.`

Figure 10. **Maze representations for tabular and spreadsheet-style coordinates.** We show the same maze as an *image*, an ASCII *table*, and a spreadsheet-style *coordinates* representation. The sheet coordinate system names cells by column letter and row number, with a1 at the top-left corner. This makes the state of the table compact to list in text and removes the need to list empty cells.

	Image	Table	Coordinates (sheet)	A* (sheet)

Step 1		`|  | Col 1 | Col 2 | Col 3 |`
		`| Row 1 |  _  |  *  |  _  |`
		`| Row 2 |  _  |  #  |  _  |`
		`| Row 3 |  @  |  _  |  #  |`
			`Coordinates:`
			`Player: a3`
			`Goal: b1`
			`Holes: {b2,c3}`
				`Grid: S=b3, G=b1, H={b2,c3}`
				`Initial: OPEN=[b3], CLOSED={}`
				`Step 1`
				`OPEN before=[b3]`
				`Pop first OPEN node: b3`
				`L -> a3: add to OPEN; set parent[a3]=b3`
				`R -> c3: hole, skip`
				`U -> b2: hole, skip`
				`D -> b4: out of bounds, skip`
				`OPEN after=[a3]`
				`CLOSED={b3}`

Step 2		`|  | Col 1 | Col 2 | Col 3 |`
		`| Row 1 |  _  |  *  |  _  |`
		`| Row 2 |  @  |  #  |  _  |`
		`| Row 3 |  _  |  _  |  #  |`
			`Coordinates:`
			`Player: a2`
			`Goal: b1`
			`Holes: {b2,c3}`
				`Step 2`
				`OPEN before=[a3]`
				`Pop first OPEN node: a3`
				`L -> offboard: out of bounds, skip`
				`R -> b3: in CLOSED, skip`
				`U -> a2: add to OPEN; set parent[a2]=a3`
				`D -> a4: out of bounds, skip`
				`OPEN after=[a2]`
				`CLOSED={b3,a3}`

A* before Step 1

```
This is a shortest-path grid problem, so I can solve it with the A* algorithm. Let's simulate it.
Coordinates are zero-indexed as (row,col), with (0,0) at the top-left.
Actions: L=left, R=right, U=up, D=down.
OPEN is always written in the order A* would consider nodes, so the first entry is the next node to pop.
If several nodes are equally preferred, keep the earlier inserted one first. Neighbor order: L,R,U,D.
```

Figure 11. **Reasoning traces with table and sheet-style coordinates. Top**: We show the same first two steps of the solution trace of the maze as ASCII tables, sheet-style coordinate states, and sheet-style A* reasoning. Sheet-style formats simply replace numerical notation with spreadsheet notation. For example, (0,0) is rewritten as a1. **Bottom**: The beginning of the chain-of-thought for both numerical- and sheet-style coordinate variants of the A* format.