Can LLMs Learn to Map the World from Local Descriptions?

Anonymous EMNLP submission

Abstract

Recent advances in Large Language Models (LLMs) have demonstrated strong capabilities in tasks such as code generation and mathematical reasoning. However, their potential to internalize structured spatial knowledge remains underexplored. This study investigates whether LLMs, grounded in locally relative human observations, can construct coherent global spatial cognition by integrating fragmented relational descriptions. We focus on two core aspects of spatial cognition: spatial perception, where models infer consistent global layouts from local positional relationships, and spatial navigation, where models learn road connectivity from trajectory data and plan optimal paths between unconnected locations. Experiments conducted in a simulated urban environment demonstrate that LLMs not only generalize to unseen spatial relationships between points of interest (POIs) but also exhibit latent representations aligned with real-world spatial distributions. Furthermore, LLMs can learn road connectivity from trajectory descriptions, enabling accurate path planning and dynamic spatial awareness during navigation.

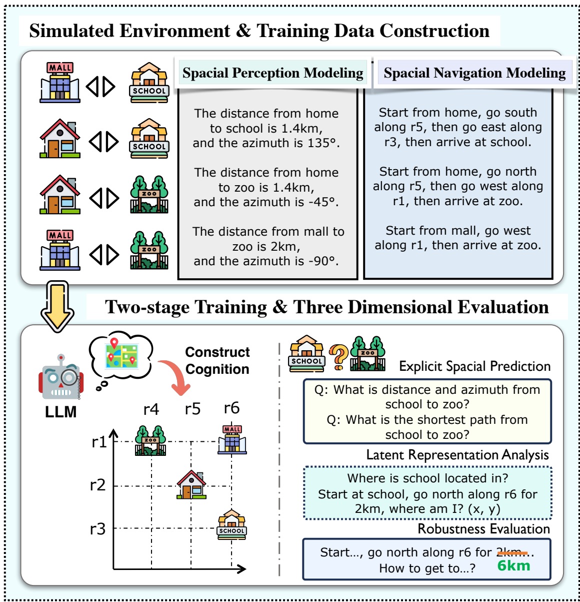

Figure 1: Summary of our research framework: First, construct a simulated environment and generate training data with relative spatial relations and shortest paths. Then, train the LLM and evaluate its spatial cognition via explicit prediction tasks, latent representation analysis, and route interference experiments.

1 Introduction

Recent advances in large language models (LLMs) have demonstrated impressive performance across diverse tasks, including code generation, mathematical reasoning, and natural language generation (Chen et al., 2021; Shao et al., 2024; Kojima et al., 2022). LLMs are trained on vast amounts of human-generated text (Achiam et al., 2023; Bai et al., 2023), including structured resources such as Wikipedia and informal unstructured dialogues. Since human language inherently relies on local semantic relationships, this enables LLMs to excel at capturing these context-dependent associations. However, it remains unexplored whether they can implicitly acquire a deep, structured understanding of global information from large amounts of fragmented, localized data—and apply it to reasoning and planning tasks such as spatial reasoning, route optimization, or multi-step inference.

A prime example of a domain requiring structured understanding is ***spatial cognition***—the ability to construct coherent mental representations of physical environments. In human communication, spatial relationships are often conveyed through relational language (*e.g.*, "The library is 100 meters southeast of the park"), which, though concise, encodes rich geometric information such as direction, distance, and topology. Humans seamlessly integrate such fragmented descriptions into unified mental maps, demonstrating a remarkable capacity to *derive global spatial understanding from localized cues*. This global cognition supports higher-level spatial reasoning, navigation, and planning.

This reliance on human perspectives raises a fundamental yet underexplored question: ***To what extent can LLMs, grounded in locally relative human observations, develop a coherent understanding of global space?*** In essence, we are asking if LLMs can learn to ***"connect the dots"*** on a vast, unseen map. This challenge goes beyond processing coordinate data—it requires understanding of spatial language, integrating fragmented descriptions, and building consistent mental maps. The model must comprehend geometric relationships (*e.g.*, direction, distance), synthesize incomplete information, and maintain logical coherence across global spatial contexts—all without relying on visual input or explicit coordinates.

While prior studies have demonstrated that LLMs can encode geospatial information (Liétard et al., 2021; Treutlein et al., 2024) and model world knowledge from sequential data (Nanda et al., 2023; Hazineh et al.; Vafa et al., 2024), they have not systematically evaluated the ability of LLMs to construct a global spatial understanding solely from local, relative relationships between POIs.

To explore this, we conduct a comprehensive analysis focusing on two core aspects of spatial cognition: **1) Spatial Perception**: The ability of LLMs to integrate local descriptions of distances and azimuth angles between POIs into a global understanding of spatial layouts without explicit coordinate information. **2) Spatial Navigation**: The ability of LLMs to extract topological knowledge from local shortest paths and perform shortest path planning between previously unseen POIs in the absence of explicit road network information.

To enable a controlled investigation, we construct a simulated urban environment and introduce a two-stage training and analysis framework guided by two core research questions, which leverages two complementary data modalities: (1) relational spatial descriptions capturing pairwise distances and directions between points of interest (POIs); and (2) trajectory descriptions representing shortest paths across the environment. We analyze whether spatial cognition is formed and how it is expressed through three experimental paradigms: (1) *Explicit spatial prediction*, assessing task-level prediction; (2) *Latent representation analysis*, probing geometry in hidden states; and (3) *Robustness evaluation*, measuring stability under navigational perturbations. The key findings are as follows:

- **LLMs can construct global spatial cognition from local observations**: LLMs demonstrate spatial perception by inferring unseen POI relationships, and spatial navigation by planning optimal paths between unconnected locations—revealing coherent global understanding emerging from fragmented linguistic input.
- **LLMs can develop implicit spatial representations**: LLMs encode absolute coordinates within their latent space, aligned with real-world geometry, and dynamically track their position during navigation—indicating the emergence of implicit spatial abstraction without explicit coordinates.
- **LLM's spatial navigation remains fragile under perturbation**: LLMs exhibit limited robustness to path perturbations, with their recovery ability dependent on the distribution of training data, suggesting that their understanding of road spatial information is limited, lacking a continuous and precise representation.

2 Global Setup

Simulation Environment. To facilitate controlled investigation and data collection, we construct a synthetic 100×100 grid map representing a simplified urban layout. Roads run along horizontal and vertical lines ($x = i$ or $y = j$, for $0 \leq i, j \leq 100$), with traversal weights w randomly sampled from $[0.8, 1.2]$ to simulate varying traffic conditions—higher weights indicate faster travel. We randomly place $N_{POI} = 1024$ points of interest (POIs) on the grid, each assigned a unique identifier p_k ($k \in 1, 2, \ldots, 1024$). Each grid unit represents 1 km, with the x-axis pointing east and the y-axis north. In addition, we explore the real-world data and synthetic data in the Appendix C.3.

Task Formulation. To explore whether LLMs can develop spatial cognition from natural language descriptions, we define two research tasks that capture key aspects of spatial cognition: **(1) *Global Spatial Perception*** — Can the model build a globally consistent understanding of spatial layouts based on local, relational language descriptions? **(2) *Dynamic Spatial Planning and Navigation*** — Can the model infer the structure of an underlying road network from local shortest-path descriptions, and use this knowledge to dynamically plan routes between previously unseen pairs of POIs?

Data. **(1) The *Relational Spatial Dataset*** is used in the first stage to train the model to infer global spatial structure from local pairwise relations. Each sample computes the Euclidean distance $d(p_i, p_j)$

and azimuth $\alpha(p_i, p_j) \in [-180°, 180°]$ between POIs (p_i, p_j), expressed through templated natural language (*e.g.*, "The distance from p_i to p_j is 2.5 km, and the azimuth is 135 degrees."). To enhance linguistic diversity, we vary the surface realizations of each template. **(2) The *Trajectory Dataset*** is used in the second stage to train dynamic spatial navigation. The road network is modeled as a weighted graph, and shortest paths between POIs are computed using Dijkstra's algorithm. Each path is translated into multi-step natural language instructions (Dijkstra, 1959) (*e.g.*, "Start at p_i, go east on r_3 for 3 km, then north on r_8 for 2 km to reach p_j"), capturing both directional and topological structure. These datasets are introduced through continuous pre-training in two stages: first, to build coherent spatial representations from relational cues; and second, to acquire navigation capabilities based on learned connectivity.

Model and Two-Stage CPT Training. We adopt a two-stage continual pre-training (CPT). Continual pre-training enables the model to gradually learn general linguistic knowledge and world knowledge from training data, without being constrained by task-specific objectives. Our research focus is on whether LLMs can construct a globally consistent spatial map from localized relational inputs, thereby demonstrating how spatial understanding can be internalized as the model's cognitive ability through CPT. The two training stages correspond to our datasets: the first uses pairwise relational data to foster global spatial perception; the second uses path-based training to develop spatial navigation abilities. We use QWEN2.5-0.5B (Yang et al., 2024b) as our base model. We also examine the impact of model size and architecture, with results in Appendix E.2.

Analysis Approach Overview. To systematically investigate the emergence of spatial cognition in LLMs, we design experiments along three complementary dimensions: *functional ability*, *internal representation*, and *behavioral robustness*. This framework moves beyond surface-level performance to probe the cognitive structures formed during training. Specifically, we assess whether the model can generate accurate spatial predictions, internalize geometry-consistent representations, and maintain stable behavior under perturbations.

- ***Explicit spatial prediction*** Evaluate the model's ability to perform spatial perception and navigation by predicting distances, azimuths, or shortest paths between unseen POI pairs.
- ***Latent representation analysis*** Analyze the spatial structure encoded in the model's latent space. We apply probing methods to assess whether these representations exhibit geometry-consistent properties, such as encoding absolute coordinates or tracking positions during navigation.
- ***Robustness evaluation*** tests whether the model can navigate accurately under perturbations, focusing on its ability to recover from trajectory deviations and plan effectively under uncertainty.

Together, these experiments progress from functional assessment to structural interpretation and robustness evaluation, offering a comprehensive view of how spatial cognition is encoded, composed, and utilized within LLMs.

3 Modeling Global Spatial Perception from Pairwise Relational Observations

In this section, we investigate the capacity of LLMs to develop a holistic understanding of spatial layout from local spatial relationships, without access to absolute coordinates.

3.1 Results of Explicit Spatial Prediction

Setting. We first evaluate whether the LLM can predict spatial relationships between unseen POI pairs. We adopt the *relational spatial dataset* in Section 2, and evaluate the model's performance under different train-test split ratios. We primarily use an 8:2 split, while also testing 6:4 and 4:6. To avoid data leakage, reciprocal POI pairs (e.g., $p_i \to p_j$, $p_j \to p_i$) are always assigned to the same subset. We denote the trained model as MODEL$_{per}$.

Split Ratio	Distance		Azimuth	
	MRPE (%) ↓	R^2 ↑	MRPE (%) ↓	Spearman ↑
8:2	0.11	1.00	0.79	1.00
6:4	0.85	1.00	3.67	0.98
4:6	2.63	0.99	5.36	0.98

Table 1: Prediction performance on distance and azimuth for unseen POI pairs across different train/test splits. MRPE is the Mean Relative Percentage Error; R^2 and Spearman reflect consistency in distance and azimuth predictions, respectively.

LLMs exhibit generalized spatial perception across unseen POI pairs. As shown in Table 1, MODEL$_{per}$ achieves low mean relative percentage errors—0.11% for distance and 0.79% for azimuth—demonstrating strong consistency with the

Split Ratio	X			Y			Euclidean Distance	
	MSE ↓	MAE ↓	R² ↑	MSE ↓	MAE ↓	R² ↑	Mean ↓	Std. ↓
Base	887.76	25.99	-0.01	878.72	25.10	-0.10	39.19	15.18
8:2	1.16	0.78	1.00	0.91	0.71	1.00	1.18	0.82
6:4	1.30	0.76	1.00	1.55	0.82	1.00	1.26	1.12
4:6	2.60	1.24	1.00	3.86	1.45	1.00	2.13	1.39

Table 2: Performance of the MLP probe in predicting the absolute coordinates of POIs from the LLM's last hidden states. Base refers to the untrained LLM. **X/Y Coordinate Accuracy**: the accuracy of the predicted x and y coordinates using MSE (Mean Squared Error), MAE (Mean Absolute Error) and R^2 (Coefficient of Determination). **Euclidean Distance**: the Euclidean distance between the predicted and true coordinates.

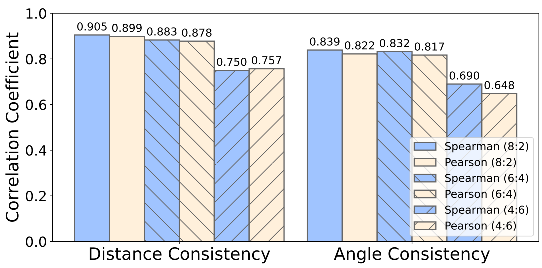

Figure 2: Consistency between POI latent representations and actual spatial locations. Spearman and Pearson coefficients measure monotonic and linear relationships, respectively.

ground truth. This highlights the model's ability to infer spatial relationships between unseen POI pairs, confirming its success in generalizing spatial perception from local relative relationships.

The strength of generalization is affected by training data scale. As the proportion of training data increases, the model's accuracy in predicting the relative spatial positions of unseen POI pairs improves, with errors decreasing from 2.63% to 0.11% across different train/test splits. This trend underscores the critical role of training data scale in enhancing the model's ability to develop a robust and generalizable global spatial perception.

3.2 Do LLMs Construct Structured Latent Spatial Representations?

Setting. To investigate whether the model develops spatial perception beyond explicit prediction, we conduct a series of experiments on its latent space. These experiments aim to evaluate whether the model encodes spatial coordinate information, how it aligns with physical geometry, and whether spatial relationships can be compositionally inferred.

Latent representations encode absolute coordinates. First, we use an MLP probe ($Probe_{loc}$) to examine whether the model implicitly encodes absolute POI coordinates in its last hidden state. Specifically, we encode POI names p_i using MODEL$_{per}$ and extract their last hidden states as latent representations. These vectors are then fed into $Probe_{loc}$, a non-linear regressor that maps them to 2D spatial coordinates (x, y). We randomly assign 90% of the POIs for training and use the remaining 10% for evaluation. The specific MLP configuration is provided in Appendix B.

As shown in Table 2, predictions from $Probe_{loc}$ yield low Mean Absolute Error, high R^2, and small Euclidean deviations, indicating that the last hidden states of MODEL$_{per}$ effectively capture absolute coordinate information. This suggests that the model not only learns local spatial relations between POIs, but also internalizes a coherent global spatial structure with precise absolute positioning.

Latent spatial layout aligns with physical geometry. We further examine the consistency between the last hidden states of POIs and their actual geographic locations. For any three distinct POIs (p_i, p_j, and p_k), we explore two types of spatial consistency: **1)** ***Distance Consistency***: the correlation between hidden space vector distances (p_i-p_j, p_j-p_k, p_i-p_k) and corresponding Euclidean distances on the map. **2)** ***Angle Consistency***: the alignment between angles formed by hidden state vectors and those formed by the physical locations.

The results in Figure 2 show a strong alignment between the POIs' spatial layout in latent space and their real-world geography, with consistently high Spearman and Pearson correlations for both distance and angle consistency. This suggests that the model's spatial understanding is internalized in its latent representations, beyond mere prediction accuracy. Unlike probing methods, which train an external model to extract absolute coordinates, this experiment directly examines the latent representations, providing more direct evidence of the

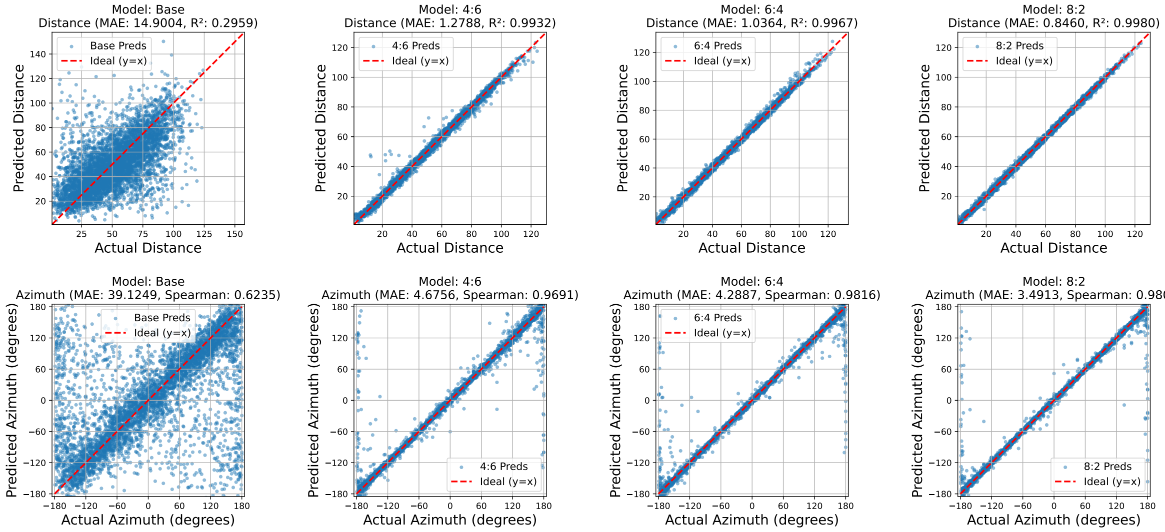

Figure 3: Latent spatial composition evaluation. An MLP predicts distance and azimuth between POI pairs using their concatenated hidden states. We use MAE to measure the deviation between the predicted and true values, and use R² and Spearman correlation to assess the consistency.

model's structured spatial understanding.

Spatial relations are recoverable via compositional inference over latent representations. Building upon these consistent findings, we further investigate whether the latent representations of individual POIs can be compositionally manipulated to infer relative spatial relationships. For any two POIs, p_i and p_j, we extract their last hidden states, concatenate them, and feed the result into an MLP probe (denoted as $Probe_{geo}$), which is trained to regress a 2D output representing the distance and azimuth between p_i and p_j. We randomly select 100 POIs for evaluation and use the remaining POIs for training.

Figure 3 shows that $Probe_{geo}$ accurately predicts the distance and azimuth between POIs with low MAE (0.85 & 3.49) and high R^2 (1.00 & 0.98), indicating that relative spatial relationships between POI pairs can be directly derived by composing their individual POI latent representations. This further validates the correctness of the spatial structure captured in the model's latent space and demonstrates the compositionality of its representations, enabling spatial reasoning tasks to be performed directly through the combination of latent vectors.

4 Modeling Spatial Navigation from Local Trajectories

We investigate the ability of LLMs to learn road connectivity and spatial navigation capabilities from local trajectory data. The custom evaluation metrics defined in this section are shown in Table 3.

4.1 Results of Explicit Spacial Prediction

Setting. To facilitate generalization analysis, we hold out a subset of 200 POIs (denoted as P_{heldout}), which selectively participate in shortest-path training. For the remaining POIs (denoted as P_{main}), we generate shortest-path trajectories for all valid point pairs, and use 80% of these pairs for training. We denote the trained model as MODEL$_{nav}$.

Models generalize shortest-path planning to unseen POI pairs by learning from localized trajectories. To evaluate model performance under the partially observable condition where all POIs appear as either origins or destinations (but not both) in the training data, we incorporate P_{heldout} by adding trajectories between P_{heldout} and P_{main} POIs, while paths between P_{heldout} POIs remain unseen (denoted as **Bridged Exposure** setting).

Table 4 shows that MODEL$_{nav}$ excels in shortest-path prediction, with an exact match accuracy of 83.63%. This suggests that the model effectively generalizes road connectivity patterns, not just memorizing seen trajectories, but also performing well on unseen POI pairs.

Models exhibit an emerging ability to compose spatial layout understanding and road network topology for navigation in unseen regions. To further investigate whether the model can leverage the spatial layout understanding established in Section 3 to perform shortest-path navigation in

Metric	Full Name	Description
SED	*Start-End Deviation*	Euclidean distance between the predicted start/end points $(\hat{p}_i, \hat{p}_j)$ and actual POIs (p_i, p_j); composed of Start Point Deviation (SPD) and End Point Deviation (EPD).
VRP	*Valid Road Proportion*	Proportion of legal roads selected at each step based on the current position.
SPA	*Shortest Path Accuracy*	Fraction of predicted trajectories that exactly match the true shortest path.
VMR	*Vector Magnitude Ratio*	Compares straight-line distances between (p_i, p_j) and $(\hat{p}_i, \hat{p}_j)$ to assess distance similarity.
VCS	*Vector Cosine Similarity*	Cosine similarity between displacement vectors $p_i \to p_j$ and $\hat{p}_i \to \hat{p}_j$, indicating directional consistency.
FD	*Fréchet Distance*	Measures geometric similarity between predicted and ground truth trajectories via path point sequences.
FSA	*First-Step Accuracy*	Proportion of correct first road selections after applying perturbation to the initial point.
SA	*Subsequent Accuracy*	Proportion of correct road selections in all subsequent steps after the first.
DD	*Destination Deviation*	Euclidean distance between the final predicted destination and the actual end point.

Table 3: Evaluation Metrics for Predicted Shortest Paths and Path Perturbations

Method	**Accuracy**				**Consistency**		
	SPD $\downarrow$	**EPD** $\downarrow$	**VRP** ($\uparrow$%)	**SPA** ($\uparrow$%)	**VMR** ($\uparrow$1.0)	**VCS** ($\uparrow$1.0)	**FD** ($\downarrow$0.0)
Zero-Exposure (Base)	49.26	49.81	87.97	0.00	0.97	0.10	58.39
Zero-Exposure	5.33	10.20	94.84	0.00	0.97	0.96	13.76
Bridged Exposure	0.06	0.48	96.07	83.63	1.00	1.00	0.91

Table 4: Performance of different training settings on shortest path prediction between POIs in P_{heldout}. (Base) denotes the model trained on the base model.

unseen regions, we ensure that the POI set P_{heldout} does not participate in the training data (denoted as **No-Exposure** setting, these unseen POIs represent unseen regions). We then compare the performance between: (1) Perception-MODEL$_{nav}$- the model trained on MODEL$_{per}$ (with spatial layout understanding), and (2) Base-MODEL$_{nav}$- the model trained on the base model (as baseline).

The results in Table 4 reveal that MODEL$_{nav}$, while trained on MODEL$_{per}$ without direct exposure to P_{heldout} POIs during shortest-path training, performs better than the baseline. The model shows improvements in both Start-End Deviation (SPD, 49.26 $\to$ 5.33) and significant gains in directional (VCS, 0.10 $\to$ 0.96) and geometric (FD, 58.39 $\to$ 13.76) consistency metrics compared to the baseline. This suggests that while MODEL$_{nav}$ may not yet fully excel at shortest-path navigation in unseen regions, it demonstrates the ability to combine the understanding of POI spatial layout with the understanding of road network topology.

4.2 Can LLMs Develop Spatial Perception of POI Positions Based on the Shortest Path Trajectory Data?

Setting. We next examine whether the model retains spatial perception of POI locations. To this end, we compare models trained under the **Bridged Exposure** setting on MODEL$_{per}$ and the base model (denoted as Perception-MODEL$_{nav}$ and Base-MODEL$_{nav}$). The untrained base model is also included for comparison.

The model still demonstrates an understanding of the spatial layout of POIs in its latent representations. To assess whether the model's latent space still encodes absolute coordinate information, we apply the same probing strategy as in Section 3.2. As shown in Table 5, although the spatial perception learned by Base-MODEL$_{nav}$ is less precise than that of Perception-MODEL$_{nav}$, the model trained solely on shortest-path trajectories shows significant improvements across all evaluation metrics compared to the base model (*e.g.*, X-MAE: 25.99 $\to$ 7.08, X-R^2: -0.01 $\to$ 0.89). This demonstrates that even when trained solely on shortest-path trajectories, the model's latent space can encode a certain degree of absolute coordinate information, highlighting the effectiveness of such data in fostering deeper spatial perception.

The model can dynamically recognize its current position during the navigation process. We evaluate the model's ability to encode absolute coordinates at each step of a predicted path using the same probing setup as in previous experiments. This allows us to assess whether the model can dynamically track spatial positions as the path unfolds. To do so, we segment each predicted path into discrete navigation steps (*e.g.*, "go east along r_1 for 4km"). At each step, we extract the model's last hidden state from the full input sequence up to that point. The true coordinate of the current location is used as supervision for probe training. For evaluation, we randomly select 200 POIs as held-out points and use the remaining POIs to construct

Model	X			Y			Euclidean Distance	
	MSE ↓	MAE ↓	R^2 ↑	MSE ↓	MAE ↓	R^2 ↑	Mean ↓	Std. ↓
Absolute Coordinate Probing								
Base Model	887.76	25.99	-0.01	878.72	25.10	-0.10	39.19	15.18
Perception-MODEL$_{nav}$	8.53	2.16	0.99	10.21	2.40	0.99	3.54	2.49
Base-MODEL$_{nav}$	100.75	7.08	0.89	85.52	7.13	0.89	11.29	7.67
Step-wise Coordinates Probing								
Base Model	713.44	19.76	0.05	621.05	18.39	0.17	30.39	20.30
Perception-MODEL$_{nav}$	6.51	1.84	0.99	6.96	1.94	0.99	3.01	2.10
Base-MODEL$_{nav}$	22.60	2.89	0.97	21.98	2.90	0.97	4.72	4.71

Table 5: Performance of the MLP probe in predicting the absolute coordinates of POIs and dynamic position coordinates at each step of the generated navigation path from the model's last hidden states.

the training set. We sample 20,000 training trajectories using only the training POIs as both start and end points, and 1,000 evaluation trajectories where the endpoints are drawn from the held-out POIs.

As shown in Table 5, at each step of the model's navigation, the absolute coordinate position can be clearly extracted from its hidden state (*e.g.*, X-R^2 0.05 → 0.97). This demonstrates the model's ability to encode and dynamically update its current position at each navigation step, indicating its capacity for dynamic spatial location cognition.

4.3 Are LLMs Robust to Path Perturbations When Navigating to a Destination?

Setting. To assess the robustness of the model to trajectory perturbations, we introduce controlled deviations during path prediction to simulate realistic detours, and evaluate whether the model can still reach the intended destination. These experiments are based on the Perception-MODEL$_{nav}$ defined in Section 4.2. We define p_{perturb} as the perturbation point and p_{target} as the immediate location reached after the deviation. Based on this, we design several perturbation strategies. We identify the step in the predicted trajectory corresponding to the road segment with the highest traversal speed and designate it as the critical step, denoted as s_{critical}.

Type	FSA (%)	SA (%)	DD (km)
No Pert.	100.00	100.00	0.00
Road Pert.	11.85	62.70	26.99
Distance Pert.	58.79	77.71	20.24
Direction Pert.	43.61	74.87	56.08

Table 6: Navigation performance under various perturbation strategies applied at critical path steps.

The model exhibits poor robustness against random disturbances. We apply the following types of random perturbations to s_{critical}: **1) Road Perturbation**: Replace the original road name in s_{critical} with a different road (the direction should be modified accordingly); **2) Distance Perturbation**: Randomly adjust the distance at s_{critical}, ensuring that it does not exceed the remaining distance to the destination. **3) Direction Perturbation**: Invert the heading direction in s_{critical} (*e.g.*, "east" → "west").

We select 10,000 cases with original correct predictions by the model for evaluation.

The experimental results in Table 6 show that the model performs poorly when confronted with random perturbations, and its robustness varies across different types of disturbances. Specifically, in the road perturbation scenario, the model only has an 11.85% chance of selecting the first valid passable road, indicating that it does not have a precise understanding of its current location, or it lacks clarity on the available roads at its current position. This suggests that the model's understanding of the road network is not coherent.

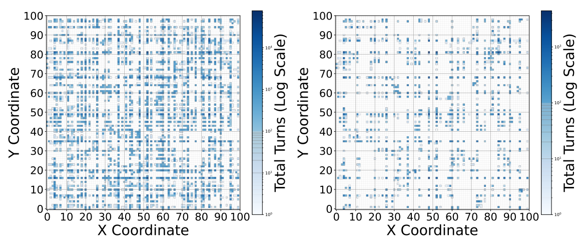

Figure 4: Heatmap of turning point frequencies in shortest paths. The left side shows the training data statistics, while the right side shows the test data statistics.

The model's robustness to disturbances largely depends on the distribution of the training data. The results reveal a distinction between distance (or direction) perturbation and road perturbation. When subjected to distance or direction perturbations, the model remains on the original high-speed road, such as those within s_{critical}. In contrast, road perturbations often randomly cause the model to

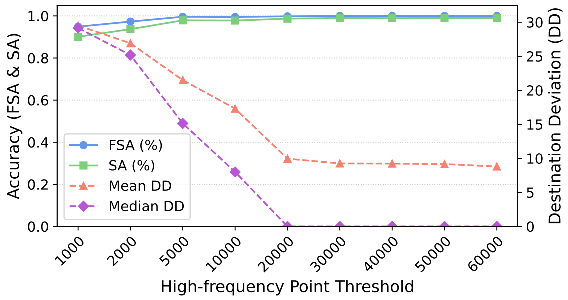

Figure 5: The model's performance under different frequency thresholds. A higher frequency threshold indicates that the new starting point after the interference appears more frequently in the paths of the training data.

deviate onto slower roads. The roads within s_{critical}, which are frequently selected in the shortest-path training data and feature higher-frequency entry and exit points along with corresponding turning patterns, are more robust to disturbances.

To analyze the impact of turning points, we count the frequency of points in the datasets where the direction of movement changes. High-frequency turning points generally correspond to transitions between high-speed and regular roads (Figure 4). We control the selection of p_{perturb} and p_{target} by ensuring the location frequency exceeds a threshold τ. We select 8,464 cases and analyze how the model's performance varies across different τ.

As shown in Figure 5, the model's performance improves with an increase in the frequency threshold (τ) for selecting p_{perturb} and p_{target}. This suggests that the model is more robust to perturbations at high-frequency turning points—enabling it to recover more effectively and reorient towards the correct destination. We provide further analysis and examples in Appendix D.3.

These results suggest that although the model exhibits a degree of robustness to perturbations, its recovery ability is highly dependent on the frequency of turning points encountered during training. This reliance implies that the model's understanding of the road network is likely fragmented and localized, rather than comprehensive and global.

5 Related Work

World Cognition Previous studies have demonstrated that LLMs can encode real-world geospatial (Liétard et al., 2021; Treutlein et al., 2024) information and temporal (Gurnee and Tegmark) information within their internal representations. However, most of these studies use pretrained LLMs in non-anonymized experiments and have not fully explored the source of these capabilities. Concurrently, many works have focused on the ability of LLMs to learn and internalize rules of the physical world or form a "world cognition" of specific tasks from sequential data, such as in board games (Nanda et al., 2023; Li et al., 2023; Hazineh et al.) or simulated navigation (Jin and Rinard, 2024; Martorell, 2025; Vafa et al., 2024). Unlike predicting the next token based on sequential data, our work focuses on whether LLMs can create a global understanding from natural language descriptions of local observations.

Urban Space Reasoning Some works focus on evaluating and enhancing the geospatial reasoning capabilities of LLMs (Feng et al., 2024a,b; Li et al., 2024). These studies enhance LLMs through knowledge training, external information, or tool use to adapt to spatial reasoning tasks in urban scenarios. In contrast, we focus on evaluating whether LLMs can reconstruct a global spatial understanding from local descriptions.

Spatial Cognition Spatial cognition capabilities are essential for LLMs to understand physical environments and perform tasks involving spatial reasoning. Many works focus on evaluating and enhancing the spatial cognition capabilities of LLMs (Mirzaee et al., 2021; Momennejad et al., 2023; Ramakrishnan et al., 2024), particularly in MLLM settings involving spatial memory and path reasoning (Yang et al., 2024a; Wu et al., 2024; Yu et al., 2025). Our work examines text-only LLMs' ability to construct global spatial cognition from localized natural language observations, without relying on global information or coordinates.

6 Conclusion

Our study shows that LLMs can develop a global spatial understanding by training on local relative positions and shortest-path data. This is evident in their ability to generalize to unseen POI-pair relationships and in the strong alignment between latent representations and real-world geographic structures. These findings suggest that the model can autonomously build structured spatial cognition from unstructured language to support spatial reasoning. However, its limited robustness to navigation disturbances reveals the constraints of its understanding of road network structures.

Limitations

Our study reveals that during the training process, the model develops an understanding of the global spatial distribution of Points of Interest (POIs) through the description of local relative relationships. However, how the model utilizes such spatial understanding when explicitly predicting positional relationships and shortest-path trajectories between unseen point pairs has not been fully analyzed. Our experiments lack an in-depth investigation into the internal mechanisms underlying the model's explicit predictions of relative positions and shortest-path trajectories, which we will explore in future work. Furthermore, our training process has impacted the model's original general language capabilities. Given that our work is primarily analytical rather than enhancement-oriented, although the issue of catastrophic forgetting does exist, it does not affect our evaluations or conclusions within the context of our assessment scenarios. Nevertheless, in future downstream application scenarios, how to balance the model's general capabilities with its internal spatial cognitive abilities remains an open research question.

Ethics Statement

We hereby acknowledge that all authors of this work are aware of the provided ACL Code of Ethics and honor the code of conduct.

Datasets Source All studies in this work are based on a simulated, synthetically constructed dataset. The generated data is solely for model analysis research and contains no other usable information. To ensure privacy and ethical compliance, the dataset has been anonymized with placeholder names and contains no real-world information. As a result, the risk of sensitive information leakage is effectively eliminated.

AI assistants AI assistants (ChatGPT) were solely used to improve the grammatical structure of the text.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

E. W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1).

Jie Feng, Yuwei Du, Tianhui Liu, Siqi Guo, Yuming Lin, and Yong Li. 2024a. Citygpt: Empowering urban spatial cognition of large language models. *arXiv preprint arXiv:2406.13948*.

Jie Feng, Jun Zhang, Junbo Yan, Xin Zhang, Tianjian Ouyang, Tianhui Liu, Yuwei Du, Siqi Guo, and Yong Li. 2024b. Citybench: Evaluating the capabilities of large language model as world model. *CoRR*.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*.

Dean Hazineh, Zechen Zhang, and Jeffrey Chiu. Linear latent world models in simple transformers: A case study on othello-gpt. In *Socially Responsible Language Modelling Research*.

Charles Jin and Martin Rinard. 2024. Emergent representations of program semantics in language models trained on programs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 22160–22184. PMLR.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.

Wenbin Li, Di Yao, Ruibo Zhao, Wenjie Chen, Zijie Xu, Chengxue Luo, Chang Gong, Quanliang Jing, Haining Tan, and Jingping Bi. 2024. Stbench: Assessing the ability of large language models in spatio-temporal analysis. *arXiv preprint arXiv:2406.19065*.

Bastien Liétard, Mostafa Abdou, and Anders Søgaard. 2021. Do language models know the way to rome? In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 510–517.

Nicolas Martorell. 2025. From text to space: Mapping abstract spatial models in llms during a grid-world navigation task.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. 2021. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598, Online. Association for Computational Linguistics.

Ida Momennejad, Hosein Hasanbeig, Felipe Vieira Frujeri, Hiteshi Sharma, Nebojsa Jojic, Hamid Palangi, Robert Ness, and Jonathan Larson. 2023. Evaluating cognitive maps and planning in large language models with cogeval. *Advances in Neural Information Processing Systems*, 36:69736–69751.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Santhosh Kumar Ramakrishnan, Erik Wijmans, Philipp Kraehenbuehl, and Vladlen Koltun. 2024. Does spatial cognition emerge in frontier models? *arXiv preprint arXiv:2410.06468*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Johannes Treutlein, Dami Choi, Jan Betley, Samuel Marks, Cem Anil, Roger B Grosse, and Owain Evans. 2024. Connecting the dots: Llms can infer and verbalize latent structure from disparate training data. *Advances in Neural Information Processing Systems*, 37:140667–140730.

Keyon Vafa, Justin Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. 2024. Evaluating the world model implicit in a generative model. *Advances in Neural Information Processing Systems*, 37:26941–26975.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024. Mind's eye of llms: Visualization-of-thought elicits spatial reasoning in large language models. In *Neural Information Processing Systems*.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2024a. Thinking in space: How multimodal large language models see, remember, and recall spaces. *arXiv preprint arXiv:2412.14171*.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. 2024b. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Dazhou Yu, Riyang Bao, Gengchen Mai, and Liang Zhao. 2025. Spatial-rag: Spatial retrieval augmented generation for real-world spatial reasoning questions. *arXiv preprint arXiv:2502.18470*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Notation Table

	Definition
	Task Formulation
$\mathcal{G}$	A graph where intersections are nodes, roads remain unchanged, and the average travel speed is used as the edge weight.
p_i	Names of Points of Interest.
r_i	Names of roads.
MODEL_{per}	LLM trained on data describing the relative positional relationships between POIs.
MODEL_{nav}	LLM trained on data describing the shortest path trajectories.
	Metrics
MSE	Mean Squared Error, a metric quantifying the average squared difference between predictions and actual values.
MRPE	Mean Relative Percentage Error, a metric quantifying the average relative percentage difference between predictions and actual values.
MAE	Mean Absolute Error, a metric quantifying the average absolute difference between predictions and actual values.
RMSE	Root Mean Squared Error, a metric quantifying the square root of the average squared difference between predictions and actual values.
R^2	R-squared, a metric quantifying the proportion of the variance in the dependent variable that is predictable from the independent variable(s).
Spearman	Spearman correlation coefficient, a metric measuring the strength and direction of the monotonic relationship between two variables.
Pearson	Pearson correlation coefficient, a metric measuring the strength and direction of the linear relationship between two variables.
FD	Fréchet Distance, a metric that measures the similarity between two curves by considering the location and ordering of points.

Table 7: The notation table.

In Table 7, we list the notations and abbreviations in this paper, together with their definitions.

B Training Parameters

LLM Training For the continual pre-training of the LLM, we use 4×A800 80G GPUs with a batch size of 128, a learning rate of 1.0e^{-4}, and a warmup ratio of 0.1, training for 10 epochs. Additionally, we designate the POI names $P = \{p_i\}_{i=1}^{1024}$ and road names $R = \{r_i\}_{i=1}^{200}$ as special tokens.

For the SFT of the LLM, we train on a single A800 80G GPU with a batch size of 256, a learning rate of 3.0e^{-5}, a warmup ratio of 0.1, and train for 10 epochs. We use Llama-Factory as our training framework (Zheng et al., 2024).

Probe We use the MLPRegression model from scikit-learn (Pedregosa et al., 2011). The MLP probe we use consists of two hidden layers, with 128 and 64 neurons, and ReLU activation functions. The model is trained using the Adam optimizer with an initial learning rate of 0.001, and L2 regularization (alpha = 0.0001) with adaptive learning rate adjustment. The maximum number of training epochs is set to 500, and early stopping is enabled based on validation set performance (patience = 100 epochs), with a validation set proportion of 10%. The batch size is adjusted automatically during training, and data is shuffled before each epoch to improve generalization. All models and tools are publicly available for research purposes.

C Experimental Details in Modeling Spatial Cognition

C.1 POI Distribution

The spatial distribution of POI points in Section 2 is shown in the Figure 6.

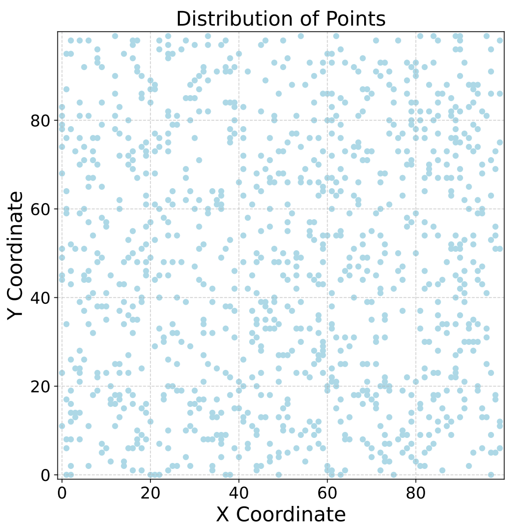

Figure 6: POIs Distribution.

C.2 Data Format

We provide examples of the data format used for training, as shown in Table 8.

C.3 Additional Experiment

POI-in-Area Prediction We design a simple spatial reasoning task that is inconsistent with the form of the training data in Section 2. We train the MODEL_{per} through supervised fine-tuning to determine whether a specific POI lies within a given

Data Format
The distance from p_i to p_j is 1000 meters, with an azimuth of 30 degrees.
The distance from p_i to p_j is 1000 meters, and the azimuth from p_i to p_j is 30 degrees.
The azimuth from p_i to p_j is 30 degrees, with a distance of 1000 meters.
Q: What is the distance from p_i to p_j? A: 1000 meters.
Q: What is the azimuth from p_i to p_j? A: 30 degrees.
Q: What is the azimuth and distance from p_i to p_j? A: 30 degrees and 1000 meters.

Table 8: Different Forms of Training and Evaluation Data for Positional Relationship Description.

Data Format
Start at p_i, then go north on r_i for 2km, then go east on r_j for 10km, and you will arrive at p_j.
To get from p_i to p_j, go along r_1 heading north for 2km, then go along r_2 heading east for 10km.
What is the shortest path from p_i to p_j? Answer: First, go north on r_1 for 2km, then go east on r_2 for 10km.
What is the shortest path from p_i to p_j? Answer: Go along r_1 heading north for 2km, then go along r_2 heading east for 10km.

Table 9: Different Forms of Training and Evaluation Data for Shortest Path Description.

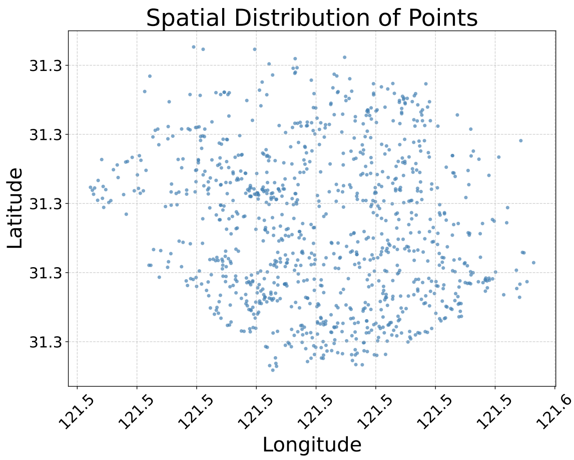

Figure 7: Real-World POIs Distribution.

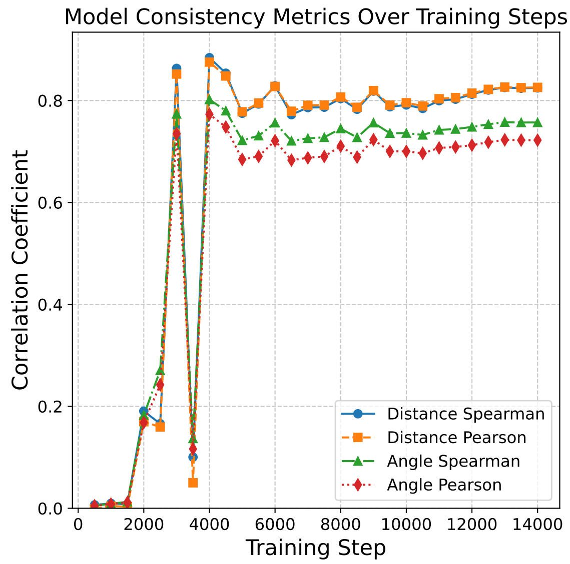

Figure 8: Consistency of POI hidden state vectors with actual spatial locations across training steps.

region. We consider two types of region descriptions: 1) a circular region defined by a central POI and a given radius; 2) a triangular region formed by three POIs. The LLM is required to provide a "yes" or "no" answer.

Additionally, we reserve a quarter of the POI points in the Map region, which are not included in the region descriptions of the SFT training data and are only used for evaluation. The remaining POIs are randomly sampled and divided into training and testing sets. We directly use prediction accuracy for evaluation.

Real World POIs The representation used in our synthetic data is universal and transferable. Real-world geographic data can be represented using our method and used for training, with no substantial differences between synthetic and real-world data.

The focus of our work is to evaluate whether LLMs can construct global cognition from discrete local descriptions; thus, using synthetic data is appropriate here. Building spatial cognition in practical scenarios and accomplishing related downstream tasks will be the focus of future work.

Also, collecting real-world data is challenging, especially the shortest path between two POIs, as the shortest path depends not only on distance but also on road conditions of each segment. In our synthetic dataset, we design a road weighting mechanism to simulate real-world road conditions. This weight represents the average driving speed of each segment. For routes with the same straight line distance, driving speeds may vary due to factors such as road roughness or curvature.

To enhance the realism and generalizability of the experiments, we sample 1,000 real-world POIs,

POI Type	Circle (%)	Triangle (%)
Included (8:2)	98.8	96.2
Excluded (8:2)	97.1	97.9
Included (6:4)	98.9	97.8
Excluded (6:4)	96.1	95.8

Table 10: Prediction accuracy for POI-in-Area experiment.

represented by their geographic coordinates (latitude and longitude). As with the synthetic data, we compute the pairwise Euclidean distances and azimuths, and split the dataset into training and testing sets (80/20).

The LLM training parameters remain consistent with those used for the synthetic data. The prediction accuracy for distances and azimuths on unseen POI pairs is shown in Table 11.

Distance		Azimuth	
MRPE (%) ↓	**R^2 ↑**	**MRPE (%) ↓**	**Spearman ↑**
0.30	1.00	0.53	1.00

Table 11: Prediction performance for distance and azimuth on unseen POI pairs in real-world scenarios.

The experimental results indicate that in more complex real-world scenarios, the model can also accurately model global positional cognition based on local relative positional relationships. We do not conduct experiments on the shortest path in real-world scenarios. This is because shortest path data is often difficult to collect in real-world settings, and our synthetic data simulates traffic conditions on real roads through weights, which is sufficient for our evaluation scenarios.

D Experimental Details in Modeling Spatial Navigation

D.1 Data Format

We provide examples of the data format used for training and evaluation of MODEL$_{nav}$, as shown in Table 9.

D.2 Metric Calculations

Start-End Deviation (SED) : evaluates the spatial accuracy of the predicted path description by computing the Euclidean distance between predicted and ground truth coordinates at both the start and end points. The predicted trajectory is reconstructed by simulating the movement along a parsed sequence of road-based navigation steps using map information. The final metric is reported as a tuple: Start Deviation (SD) and End Deviation (ED). Detailed computation logic is provided in Algorithm 1.

Valid Road Proportion (VRP) : measures the proportion of valid road choices at each step of the predicted path description. The path is parsed into a sequence of steps, and for each step, the algorithm checks if the road and direction are valid according to the map's connectivity and direction rules. The final metric, VRP, is the ratio of valid steps to the total steps in the path description. If no steps are described, the VRP is defined as 0. Detailed computation logic is provided in Algorithm 2.

Shortest Path Accuracy (SPA) : measures the proportion of cases where the model-generated trajectory exactly matches the ground truth shortest path.

D.3 Case Study

Failure Analysis In terms of distance and azimuth prediction, the model demonstrates high accuracy, with most errors occurring in the shortest path prediction task, especially in the presence of perturbations.

To better understand the failure modes of the model in shortest path prediction, we conducted an error analysis. We categorized prediction errors into three types: 1) start point errors, 2) intermediate path errors, and 3) end point errors. Since end point errors are always a consequence of one of the first two types, we do not report them separately.

The breakdown of errors on the test set (in terms of error count / total number of test cases) is as follows:

- Start point errors: 917 / 39800
- Intermediate path errors: 5656 / 39800

In intermediate path errors, we record the step at which the first error occurs. The distribution is as follows:

Step	1	2	3	4	5	6	7	8	9
Error Count	1819	2254	1049	367	113	39	13	1	1

Table 12: Distribution of the step where the first error occurred in intermediate path errors

We further categorize the causes of intermediate path errors into the following types:

- Direction errors: 894
- Road name errors: 37 (cases where the direction is correct but the road name is incorrect)
- Distance errors: 4725

These results indicate that most errors stem from a single incorrect step in the intermediate path (errors mainly occur in the early steps, primarily because the average number of steps across all cases is 5.2).

It is worth noting that most intermediate path errors are caused by incorrect distance predictions, accounting for 4725 out of 5656 cases.

Disturbance Case Figure 12 demonstrates the performance of LLM in handling intermediate disturbances under different turning point frequencies. As the frequency increases, LLM exhibits stronger robustness against disturbances and can reach the final destination after being disturbed. When the frequency is low, the model is more prone to output interruptions (*e.g.*, not knowing where to go).

D.4 Additional Experiment

Model	Distance % ↓	Azimuth % ↓
Perception-MODEL$_{nav}$	3.08	5.52
Base-MODEL$_{nav}$	12.03	13.84

Table 13: Evaluation results for distance and azimuth prediction, evaluated using MRPE.

The model remains capable of performing explicit spatial relationship prediction. To assess whether the model directly trained on path data can still understand the relative positional relationships between POIs, we fine-tune it with supervised training to predict the distance and azimuth between POI pairs. We use 200 POIs to construct the test set, while the remaining POIs are used to generate the training data (randomly sample 100,000 cases).

The results in Table 13 show that training the base model on shortest-path trajectories (Base-MODEL$_{nav}$) allows it to capture the relative spatial relationships between POI pairs, achieving reasonable performance in both distance and azimuth prediction, with MRPE values of 12.03% and 13.84%, respectively. This suggests that, even without directly relying on local distance and azimuth information between POI pairs, the model is still able to leverage shortest-path trajectories to build a certain level of global spatial perception. This also indicates that shortest-path trajectories, as a topologically structured data format, are effective in constructing an understanding of spatial layout.

E Additional Experiments and Results

E.1 Training Strategy

Training Strategy	Distance		Azimuth	
	MRPE ↓	R^2 ↑	MRPE ↓	Spearman ↑
CPT	0.11	1.00	0.79	1.00
SFT	0.003	1.00	0.025	1.00

Table 14: The performance of the model's prediction of distance and azimuth for unseen POI pairs under different training strategies.

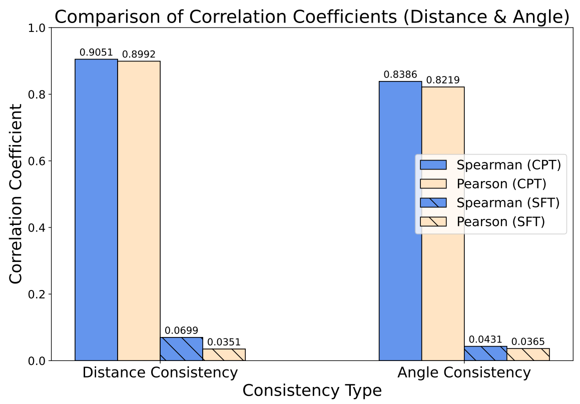

Figure 9: Consistency of POI point last hidden state vector with actual spatial location in terms of distance and angle under different training strategies.

Our primary experiments adopt a continual pre-training approach for LLM training. In addition to this, we explore the use of SFT for training MODEL$_{per}$ and MODEL$_{nav}$. For training MODEL$_{per}$, we retain the question-answer format data from the original complete dataset and adopt an 80/20 split for training and test sets. For MODEL$_{nav}$, we follow the **Bridged Exposure** strategy. We evaluate whether the LLM trained with SFT can perform explicit predictions and construct cognitive representations in the latent space.

We conduct training using 4×A800 80G GPUs, with a batch size of 512 and a learning rate set to 3.0e-5. The LLM is trained for 10 epochs.

Spatial Perception The experimental results for evaluating Spatial Cognition are shown in Table 14, Table 15, Table 16 and Figure 9.

Experimental results show that while SFT-trained LLM outperform CPT-trained LLM in dis-

Training Strategy	X			Y			Euclidean Distance	
	MSE ↓	MAE ↓	R² ↑	MSE ↓	MAE ↓	R² ↑	Mean ↓	Std. ↓
Base	887.76	25.99	-0.01	878.72	25.10	-0.10	39.19	15.18
CPT	1.16	0.78	1.00	0.91	0.71	1.00	1.18	0.82
SFT	406.66	15.41	0.46	373.35	14.23	0.53	23.10	15.69

Table 15: Performance of the MLP probe in predicting the absolute coordinates of POIs from the LLM 's last hidden states under different training strategies.

Training Strategy	Distance		Azimuth	
	MAE (km)	R²	MAE (°)	Spearman
Base	14.90	0.03	39.12	0.62
CPT	0.85	1.00	3.49	0.98
SFT	31.62	-2.92	66.48	0.38

Table 16: Latent spatial composition evaluation. An MLP predicts distance and azimuth between POI pairs using their concatenated hidden states.

tance and azimuth prediction accuracy, they exhibit weaker latent spatial cognition, as evidenced by blurred awareness of absolute coordinates in hidden states and poor alignment between latent vector distributions and actual spatial layouts.

This result is expected, as the POI name tokens in the SFT training process do not directly contribute to the loss calculation. Consequently, their embeddings are not explicitly optimized, leading to a lack of structured distribution in the latent space. This highlights the importance of continual pre-training for fostering deeper internal representations. At the same time, it suggests that a well-structured latent distribution of individual POIs is not strictly necessary for predicting relative relationships between unseen POI pairs.

Spatial Navigation The experimental results for evaluating Spatial Navigation are shown in Table 17 and Table 18.

In addition, we further train the continual pre-trained model $\text{M}\small{\text{ODEL}}_{per}$ using the sft approach for the shortest path task, and evaluate its robustness against disturbances. The experimental results are shown in Table 19 and Figure 10.

Experimental results show that Cognition-*ModelTwo* trained via SFT exhibits robustness comparable to that of the CPT-trained counterpart, with both being influenced by the training data distribution—performing better at critical points with larger thresholds. Meanwhile, when facing random disturbances, the SFT-trained model reaches destinations closer to the target on average, but demonstrates a significantly lower proportion of selecting valid roads at each step.

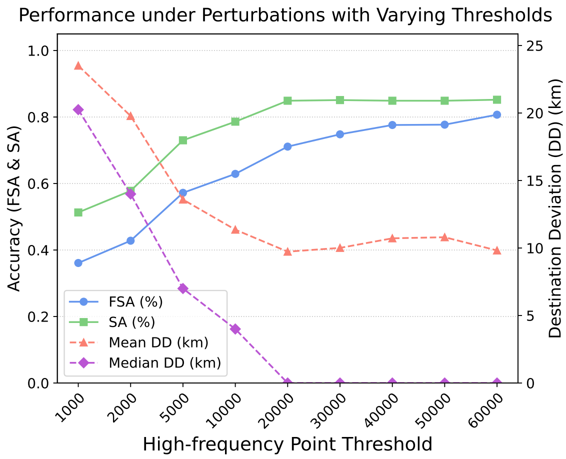

Figure 10: Performance metrics (FSA, SA, Mean/Median DD) versus high-frequency point thresholds. Left y-axis: FSA/SA; Right y-axis: DD (km).

E.2 Model Architecture and Scale

To investigate the impact of architecture and parameter scale on models' spatial cognition, in addition to Qwen2.5-0.5B used in the main experiments, we further examine the performance of Qwen2.5-1.5B, Qwen2.5-3B, and LLaMA-3.2-1B. (AI@Meta, 2024).

Spatial Perception The results are shown in Table 25, Table 24 and Table 26. Experiments show that for the Qwen2.5 series models, as the model parameter scale increases, no significant improvement is observed in the explicit prediction of distance and azimuth, nor in the probing accuracy of absolute coordinates. Even when the model parameter scale is small (0.5B), it already achieves high accuracy. In addition, models with different architectures (LLaMA) also demonstrate highly accurate modeling cognition of relative positions and absolute coordinates, presenting consistent experimental conclusions.

Training Strategy	Accuracy				Consistency		
	SPD ↓	EPD ↓	VRP (↑%)	SPA (↑%)	VMR (↑1.0)	VCS (↑1.0)	FD (↓0.0)
CPT	0.06	0.48	96.07	83.63	1.00	1.00	0.91
SFT	0.02	0.02	99.65	97.34	1.00	1.00	0.11

Table 17: Performance of different training settings on shortest path prediction between POIs in P_{heldout}.

Model	X			Y			Euclidean Distance	
	MSE ↓	MAE ↓	R^2 ↑	MSE ↓	MAE ↓	R^2 ↑	Mean ↓	Std. ↓
Absolute Coordinate Probing								
Base Model	887.76	25.99	-0.01	878.72	25.10	-0.10	39.19	15.18
Cognition-CPT	8.53	2.16	0.99	10.21	2.40	0.99	3.54	2.49
Base-CPT	100.75	7.08	0.89	85.52	7.13	0.89	11.29	7.67
Cognition-SFT	13.05	2.89	0.98	12.88	2.76	0.99	4.39	3.84
Base-SFT	630.21	20.83	0.25	659.85	21.14	0.25	32.55	15.19
Step-wise Coordinates Probing								
Base Model	713.44	19.76	0.05	621.05	18.39	0.17	30.39	20.30
Cognition-CPT	6.51	1.84	0.99	6.96	1.94	0.99	3.01	2.10
Base-CPT	22.60	2.89	0.97	21.98	2.90	0.97	4.72	4.71
Cognition-SFT	11.97	2.53	0.98	12.78	2.56	0.98	4.07	3.56
Base-SFT	39.01	3.91	0.95	80.64	5.13	0.89	7.50	5.21

Table 18: Performance of the MLP probe in predicting the absolute coordinates of POIs and dynamic position coordinates at each step of the generated navigation path from the LLM 's last hidden states.

Method	FSA (%)	SA (%)	DD (km)
No Pert.	100.00	100.00	0.00
Road Pert.	8.14	52.31	12.42
Distance Pert.	14.95	60.29	9.46
Direction Pert.	6.01	59.53	40.89

Table 19: Evaluation Results for Different Types of Perturbations Trained via SFT.

Spatial Navigation The results are shown in Table 27 and Figure 11. The experimental results show that for the Qwen2.5 series models, as the scale of the model parameter increases, the prediction accuracy of the shortest path improves (89.0% → 89.9% → 91.9%), but the robustness against path interference does not improve. Moreover, the LLaMA model exhibits poor performance in learning local path information and accomplishing shortest path navigation, with a notable bias in identifying the starting point.

E.3 Linear vs. Non-linear Probe

Setup We use the LinearRegression model from scikit-learn. It relies on a direct mathematical solution to find the best-fit line, and we used its default configuration. For the non-linear probe, we use the same MLP configuration as in the main experiment.

Results We use MODEL$_{per}$ and Base-MODEL$_{nav}$ to compare linear and non-linear probes in several experiments involving probing. The experimental results are shown in the Table 23, Table 20.

Conclusion The results in Table 23 demonstrate that a linear probe can map hidden states to actual coordinate values, indicating the presence of linearly accessible coordinate information within the hidden representations of the LLM. However, non-linear regression achieves higher prediction accuracy. Furthermore, in the LLM trained on shortest-path trajectory data, the performance of the linear probe deteriorates significantly, with the average Euclidean distance increasing from 3.01 to 18.68. This suggests that non-linear probes are better suited for capturing position information in more complex tasks.

The experimental results in Table 20 show that when performing regression to predict distance and azimuth by combining the hidden states of two POIs, the linear probe performs poorly (R^2 of only 0.20 for distance prediction). This suggests that we cannot achieve combined prediction through simple linear regression, which may also be related to how we process the two POI vectors (*e.g.*, concatenation).

E.4 Data Construction Template

Setup In addition to the data construction template adopted in the main experiment described in Section C.2, we also experiment with other forms

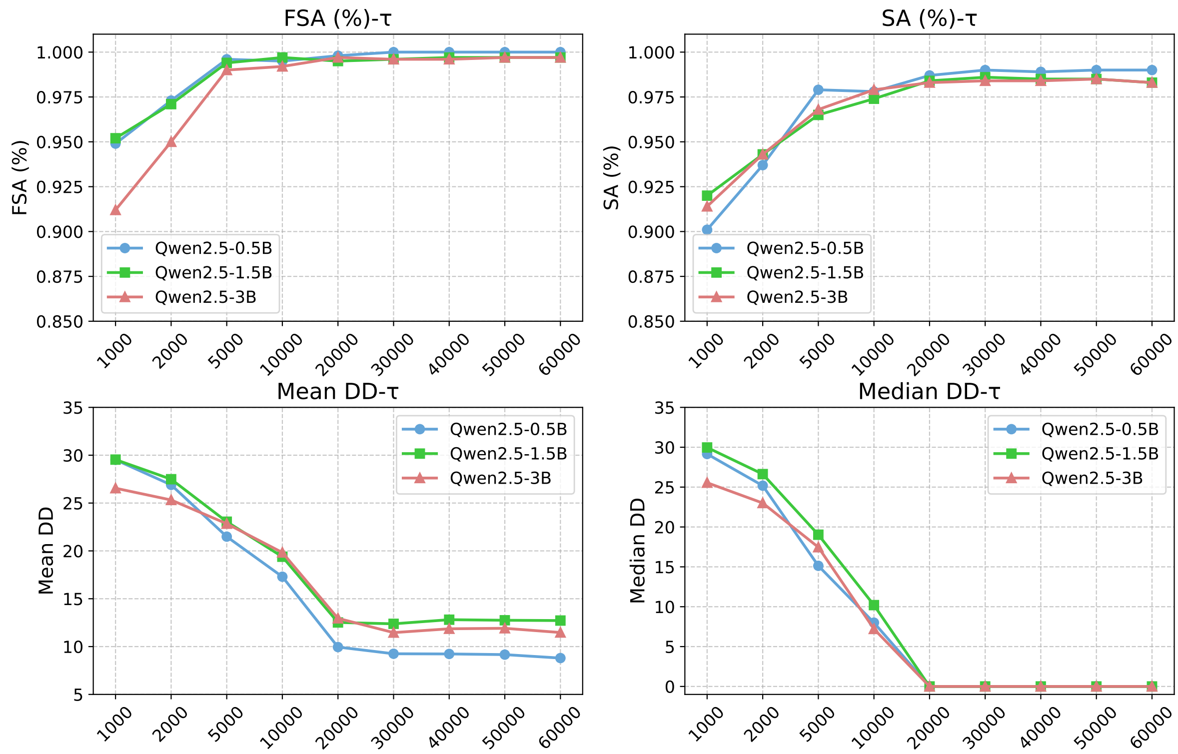

Figure 11: The robustness performance of models with different parameter scales when facing path interference.

Probe Type	Distance		Azimuth	
	MAE (km)	R²	MAE (°)	Spearman
Non-linear	0.85	1.00	3.49	0.98
Linear	17.89	0.20	51.94	0.78

Table 20: Latent spatial composition evaluation. An MLP predicts distance and azimuth between POI pairs using their concatenated hidden states.

Data Format
At an azimuth of 30 degrees from p_i, p_j is located 1000 meters away.
p_j lies 1000 meters from p_i at an azimuth of 30 degrees.
The azimuth from p_i to p_j is 30 degrees, and the separation is 1000 meters.
Q: How far is p_j from p_i? *A*: 1000 meters.
Q: In what direction does p_j lie relative to p_i? *A*: An azimuth of 30 degrees.
Q: What is the direction and separation between p_i and p_j? *A*: An azimuth of 30 degrees and a distance of 1000 meters.

Table 21: An Alternative Template for Training and Evaluation Data of Positional Relationship Description.

of templates to explore the impact of data construction templates on model performance. For the

Template	Distance		Azimuth	
	MRPE (%) ↓	**R²** ↑	**MRPE (%)** ↓	**Spearman** ↑
type 1	0.11	1.00	0.79	1.00
type 2	1.10	1.00	0.92	1.00

Table 22: The model's prediction performance under different data construction templates: Type 1 represents the original template used in the main experiment, and Type 2 represents the replaced template.

training, prediction, and evaluation of distance and azimuth, the templates we use are shown in Table 21.

Results We all adopt an 8:2 split ratio between the training set and the evaluation set. The experimental results of the two different data construction templates are shown in Table 22. In addition, after training the model using the replaced template, we still attempt to use the original template as the input for model evaluation, specifically the question "What is the distance from p_i to p_j?". The model's mean relative prediction error (MRPE) for distance prediction remains only 1.30%.

Conclusion The experimental results show that after replacing with more diverse templates, the prediction errors of the model are still controlled

Probe Type	X			Y			Euclidean Distance	
	MSE ↓	**MAE ↓**	**R² ↑**	**MSE ↓**	**MAE ↓**	**R² ↑**	**Mean ↓**	**Std. ↓**
Absolute Coordinate Probing								
Non-linear	1.16	0.78	1.00	0.91	0.71	1.00	1.18	0.82
Linear	21.18	3.61	0.97	12.70	2.75	0.99	4.99	2.99
Step-wise Coordinates Probing								
Non-linear	6.51	1.84	0.99	6.96	1.94	0.99	3.01	2.60
Linear	238.65	11.97	0.68	228.98	11.73	0.69	18.68	10.86

Table 23: Performance of the MLP probe in predicting the absolute coordinates of POIs and dynamic position coordinates at each step of the generated navigation path from the LLM 's last hidden states.

Model	X			Y			Euclidean Distance	
	MSE ↓	MAE ↓	R² ↑	MSE ↓	MAE ↓	R² ↑	Mean ↓	Std. ↓
Qwen2.5-0.5B	1.16	0.78	1.00	0.91	0.71	1.00	1.18	0.82
Qwen2.5-1.5B	6.83	1.96	0.99	3.40	1.47	1.00	2.73	1.66
Qwen2.5-3B	5.84	1.79	0.99	4.72	1.71	0.99	2.90	1.75
LlaMA-3.2-1B	5.71	1.94	0.99	6.97	1.99	0.99	3.07	1.81

Table 24: Performance of the MLP probe in predicting the absolute coordinates of POIs from the LLM 's last hidden states under different models.

Model	Distance		Azimuth	
	MRPE ↓	R² ↑	MRPE ↓	Spearman ↑
Qwen2.5-0.5B	0.11	1.00	0.79	1.00
Qwen2.5-1.5B	0.28	1.00	1.30	0.99
Qwen2.5-3B	0.11	1.00	0.89	1.00
LlaMA-3.2-1B	1.71	1.00	3.99	0.98

Table 25: The performance of the model's prediction of distance and azimuth for unseen POI pairs under different models.

Model	Distance		Azimuth	
	MAE (km)	R²	MAE (°)	Spearman
Qwen2.5-0.5B	0.85	1.00	3.49	0.98
Qwen2.5-1.5B	1.61	0.99	5.81	0.97
Qwen2.5-3B	0.84	1.00	3.81	0.98
LlaMA-3.2-1B	1.18	1.00	4.32	0.96

Table 26: Latent spatial composition evaluation. An MLP predicts distance and azimuth between POI pairs using their concatenated hidden states.

Model	Accuracy				Consistency		
	SPD ↓	**EPD** ↓	**VRP** (↑%)	**SPA** (↑%)	**VMR** (↑1.0)	**VCS** (↑1.0)	**FD** (↓0.0)
Qwen2.5-0.5B	0.07	0.47	97.5	89.0	1.00	1.00	0.81
Qwen2.5-1.5B	0.04	0.27	97.6	89.9	1.00	1.00	0.59
Qwen2.5-3B	0.03	0.24	98.0	91.9	1.00	1.00	0.49
LlaMA-3.2-1B	32.18	1.16	96.5	27.4	1.05	0.74	23.30

Table 27: Performance of different training settings on shortest path prediction between POIs in P_{heldout}.

within a very small range (1.1%), which indicates that the templates have little impact on the model's construction of such spatial cognitive ability. Moreover, when using an evaluation method different from the templates, the performance of the model is still not significantly affected (1.1% → 1.3%), which suggests that LLM has strong generalization ability and can understand texts with the same meaning but different forms.

F Other Statements

Our use of existing artifacts are consistent with their intended use, and we follow their license and terms.

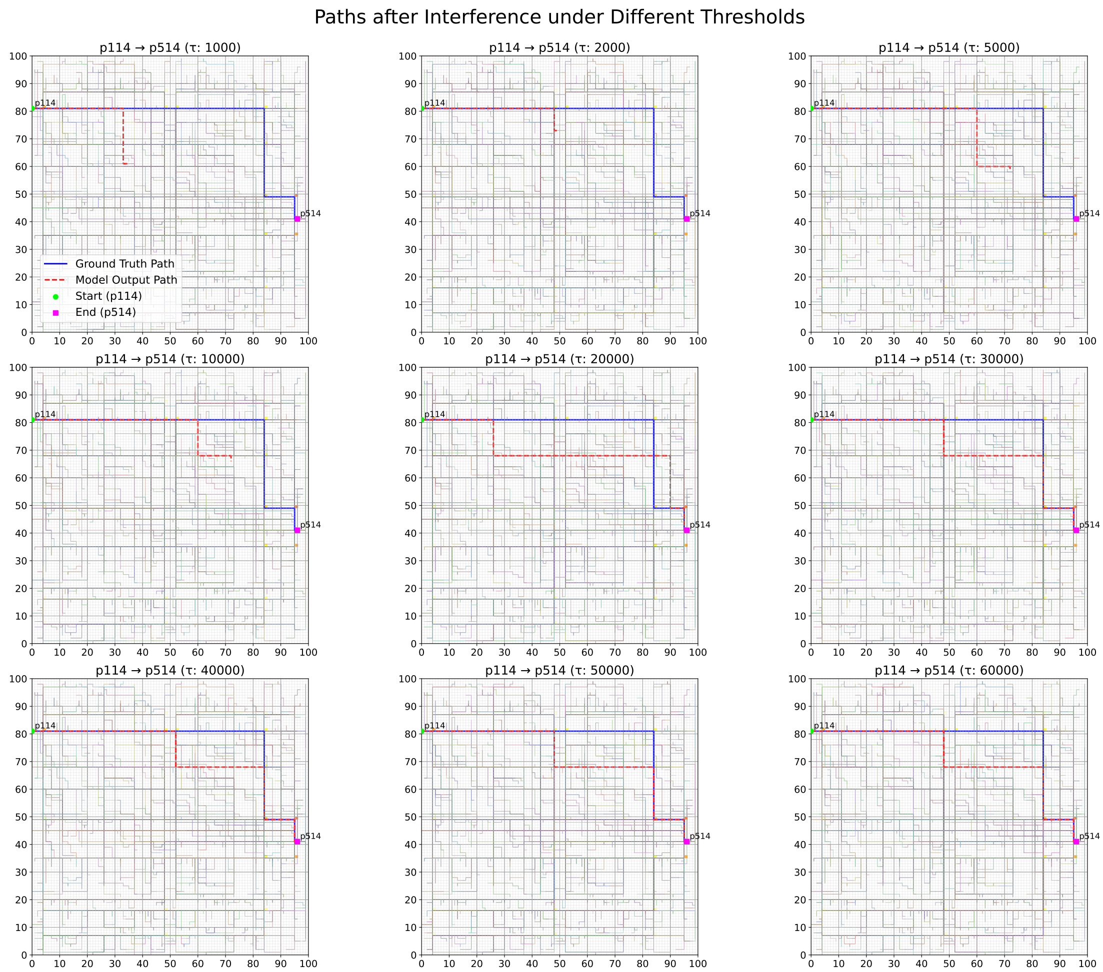

Figure 12: Case study on the model's behavior under interference during navigation at different statistical frequencies.

Algorithm 1 SED: Start-End Deviation Calculation

Input: Ground truth start coordinates P_{start_gt}, Ground truth end coordinates P_{end_gt}, LLM-generated textual path description $\mathcal{A}$, Map information $\mathcal{M}_{map}$
Output: Start-End Deviation SED ▷ Euclidean distance between predicted and ground truth points
$\mathcal{S} \leftarrow$ ParsePathDescription($\mathcal{A}$) ▷ Parse $\mathcal{A}$ into sequence of steps $\mathcal{S} = [(r_1, d_1, l_1), \ldots, (r_n, d_n, l_n)]$
if $|\mathcal{S}| < 2$ **then**
  $P_{start\_pred} \leftarrow P_{start\_gt}$ ▷ Use ground truth start if path description has fewer than 2 steps
else
  Let $(r_1, d_1, l_1) = \mathcal{S}[1]$ ▷ First step details
  Let $(r_2, d_2, l_2) = \mathcal{S}[2]$ ▷ Second step details
  $P_{intersect} \leftarrow$ FindIntersection($r_1, r_2, \mathcal{M}_{map}$) ▷ Find intersection of the first two roads (position after the first step)
  if $P_{intersect}$ is valid **then** ▷ Check if a valid intersection was found
    $P_{start\_pred} \leftarrow$ MoveAlongRoad($P_{intersect}, r_1,$ Opposite$(d_1), l_1, \mathcal{M}_{map}$) ▷ Backtrack from intersection to estimate start
  else
    $P_{start\_pred} \leftarrow P_{start\_gt}$ ▷ Fallback to ground truth start if intersection is indeterminate
  end if
end if
$P_{current} \leftarrow P_{start\_pred}$ ▷ Initialize current position
if $|\mathcal{S}| > 0$ **then** ▷ Simulate the path if steps exist
  for each step (r_i, d_i, l_i) in $\mathcal{S}$ **do**
    $P_{current} \leftarrow$ MoveAlongRoad($P_{current}, r_i, d_i, l_i, \mathcal{M}_{map}$) ▷ Update position
  end for
end if
$P_{end\_pred} \leftarrow P_{current}$ ▷ The final position is the predicted end position
$SD \leftarrow$ EuclideanDistance($P_{start\_pred}, P_{start\_gt}$) ▷ Calculate Start Deviation
$ED \leftarrow$ EuclideanDistance(P_{end_pred}, P_{end_gt}) ▷ Calculate End Deviation
return (SD, ED) ▷ Return deviations at both start and end points

▷ Helper Functions:
▷ - ParsePathDescription($\mathcal{A}$): Parses the textual path description $\mathcal{A}$ into a structured list $\mathcal{S}$ of tuples, where each tuple is $(road\_id, direction, length)$.
▷ - FindIntersection($r_a, r_b, \mathcal{M}_{map}$): Returns the geographic coordinates of the intersection between road segment r_a and road segment r_b based on $\mathcal{M}_{map}$. Returns an invalid/null state if no relevant intersection exists.
▷ - MoveAlongRoad($P_{origin}, r, d, l, \mathcal{M}_{map}$): Calculates the coordinates resulting from starting at P_{origin}, moving along road r in direction d for distance l, according to $\mathcal{M}_{map}$.
▷ - Opposite(d): Returns the direction directly opposite to d (e.g., Opposite(North) = South).
▷ - EuclideanDistance(P_1, P_2): Computes the L2 norm (straight-line distance) $||P_1 - P_2||_2$.

Algorithm 2 VRP: Valid Road Proportion Calculation

Input: Ground truth start coordinates P_{start_gt}, LLM-generated textual path description $\mathcal{A}$, Map information $\mathcal{M}_{map}$

Output: Valid Road Proportion VRP ▷ Proportion of steps choosing a valid next road

$\mathcal{S} \leftarrow$ ParsePathDescription($\mathcal{A}$) ▷ Parse $\mathcal{A}$ into sequence of steps $\mathcal{S} = [(r_1, d_1, l_1), \ldots, (r_n, d_n, l_n)]$

if $|\mathcal{S}| < 2$ **then**

  $P_{start\_pred} \leftarrow P_{start\_gt}$ ▷ Use ground truth start if path description has fewer than 2 steps

else

  Let $(r_1, d_1, l_1) = \mathcal{S}[1]$ ▷ First step details

  Let $(r_2, d_2, l_2) = \mathcal{S}[2]$ ▷ Second step details

  $P_{intersect} \leftarrow$ FindIntersection($r_1, r_2, \mathcal{M}_{map}$) ▷ Find intersection of the first two roads (position after the first step)

  if $P_{intersect}$ is valid **then** ▷ Check if a valid intersection was found

    $P_{start\_pred} \leftarrow$ MoveAlongRoad($P_{intersect}, r_1,$ Opposite$(d_1), l_1, \mathcal{M}_{map}$) ▷ Backtrack from intersection to estimate start

  else

    $P_{start\_pred} \leftarrow P_{start\_gt}$ ▷ Fallback to ground truth start if intersection is indeterminate

  end if

end if

$P_{current} \leftarrow P_{start\_pred}$ ▷ Initialize current position

$valid\_steps \leftarrow 0$ ▷ Initialize counter for valid road choices

$total\_steps \leftarrow |\mathcal{S}|$ ▷ Total number of steps in the described path

if $total\_steps > 0$ **then** ▷ Simulate the path if steps exist

  for each step (r_i, d_i, l_i) in $\mathcal{S}$ **do**

    $\mathcal{R}_{valid} \leftarrow$ GetValidNextRoads($P_{current}, \mathcal{M}_{map}$) ▷ Get set of valid (road_name, road_direct)

    if $(r_i, d_i) \in \mathcal{R}_{valid}$ **then** ▷ Check if the chosen road and direction are valid options

      $valid\_steps \leftarrow valid\_steps + 1$ ▷ Increment valid step count

    end if

    $P_{current} \leftarrow$ MoveAlongRoad($P_{current}, r_i, d_i, l_i, \mathcal{M}_{map}$) ▷ Update position

  end for

end if

if $total\_steps == 0$ **then**

  $VRP \leftarrow 0$ ▷ Define VRP as 0 for empty paths

else

  $VRP \leftarrow valid\_steps/total\_steps$ ▷ Calculate the proportion of valid steps

end if

return VRP

▷ Helper Functions:

▷ - ParsePathDescription($\mathcal{A}$): Parses the textual path description $\mathcal{A}$ into a structured list $\mathcal{S}$ of tuples $(road\_id, direction, length)$.

▷ - MoveAlongRoad($P_{origin}, r, d, l, \mathcal{M}_{map}$): Calculates coordinates after moving from P_{origin} along road r in direction d for distance l.

▷ - Opposite(d): Returns the direction opposite to d.

▷ - GetValidNextRoads($P_{pos}, \mathcal{M}_{map}$): Returns a set of valid next moves as (road_id, direction) tuples accessible from position P_{pos}. This considers connectivity and travel direction rules based on map data $\mathcal{M}_{map}$.

▷ - EuclideanDistance(P_1, P_2): Computes the L2 norm $||P_1 - P_2||_2$. (Included for consistency, though not used in VRP calculation itself).