CITY NAVIGATION IN THE WILD: EXPLORING EMERGENT NAVIGATION FROM WEB-SCALE KNOWLEDGE IN MLLMS

Dwip Dalal[†]
University of Illinois Urbana-Champaign
dwip2@illinois.edu

Utkarsh Mishra[*]
Texas A&M University

Narendra Ahuja
University of Illinois Urbana-Champaign

Nebojsa Jojic
Microsoft Research, Redmond

ABSTRACT

Leveraging multimodal large language models (MLLMs) to develop embodied agents offers significant promise for addressing complex real-world tasks. However, current evaluation benchmarks remain predominantly language-centric or heavily reliant on simulated environments, rarely probing the nuanced, knowledge-intensive reasoning essential for practical, real-world scenarios. To bridge this critical gap, we introduce the task of *Sparsely Grounded Visual Navigation*, explicitly designed to evaluate the sequential decision-making abilities of MLLMs in challenging, knowledge-intensive real-world environment. We operationalize this task with CityNav, a comprehensive benchmark encompassing four diverse global cities, specifically constructed to assess raw MLLM-driven agents in city navigation. Agents are required to rely solely on visual inputs and internal multimodal reasoning to sequentially navigate $50+$ decision points without additional environmental annotations or specialized architectural modifications. Crucially, agents must autonomously achieve localization through interpreting city-specific cues and recognizing landmarks, perform spatial reasoning, and strategically plan and execute routes to their destinations. Through extensive evaluations, we demonstrate that current state-of-the-art MLLMs, reasoning techniques (e.g., GEPA, chain-of-thought, reflection) and competitive baseline PReP significantly underperform in this challenging setting. To address this, we propose *Verbalization of Path* (VoP), which explicitly grounds the agent's internal reasoning by probing city-scale cognitive maps (key landmarks and directions toward the destination) from the MLLM, substantially enhancing navigation success.

1 INTRODUCTION

Pretraining MLLMs on web-scale, interleaved image–text corpora induces broad, transferable world knowledge, enabling robust zero-shot and few-shot generalization across diverse vision–language tasks (Xie et al., 2024; Hu et al., 2025; Gao et al., 2024). This emergent behavior has enabled foundation models to exhibit robust general reasoning and instruction-following abilities (Brown et al., 2020; Achiam et al., 2023b; Rani et al., 2023; 2025; Yang et al., 2024). Furthermore, their multimodal successors (OpenAI, 2024; Reid et al., 2024; Liu et al., 2024; Wang et al., 2024; Chen et al., 2023), enhanced by recent scaling methods (Chen et al., 2025), integrate high-resolution visual processing, extended context handling, and precise OCR and grounding capabilities. Such advancements facilitate elegant perception-to-action pipelines, enabling researchers to develop embodied agents capable of complex perception, planning, and action tasks (Huang et al., 2022; 2023; Ahn et al., 2022; Song et al., 2023; Zhou et al., 2024; Lin et al., 2025; Yue et al., 2024).

[*]Equal contribution.
[†]Work performed during a research internship at Microsoft Research, Redmond.

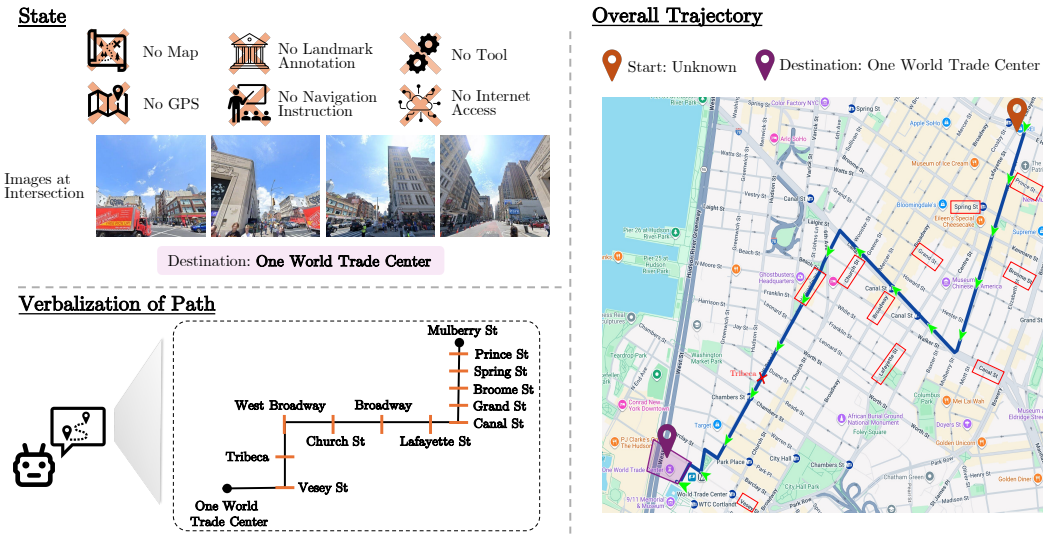


Figure 1: The figure illustrates our proposed *Verbalization of Path* (VoP) method, which elicits city-scale cognitive maps from MLLMs for city navigation in the wild. The red bounding boxes on the New York map highlight the streets and locations explicitly referenced by the MLLM during verbalization of path.

While existing works have demonstrated reasoning capabilities of embodied agents in indoor navigation (Zhou et al., 2024; Lin et al., 2025; Yue et al., 2024), code generation and planning (Song et al., 2023; Liang et al., 2022), and robotic arm manipulation (Huang et al., 2023; Ahn et al., 2022; Singh et al., 2022), these evaluations predominantly occur within simulated environments and are not knowledge intensive tasks. Therefore do not rigorously assess the agents' ability to leverage their extensive internal knowledge repositories to execute sequential decisions in dynamic, real-world scenarios.

Whereas, outdoor navigation is inherently a knowledge-intensive task demanding extensive cognitive capabilities, such as comprehensive environmental knowledge, sequential decision-making, spatial reasoning, and robust visual grounding using recognizable landmarks (Mirowski et al., 2018a). Prior studies on outdoor navigation have typically supplied agents with explicit landmark information embedded within images (Chen et al., 2019; Zeng et al., 2024; Schumann et al., 2024), significantly alleviating cognitive load by eliminating the necessity for agents to internally retrieve knowledge for self-localization and planning. In contrast, our work introduces a novel task termed *Sparsely Grounded Long-Range Navigation* where:

> Agent must navigate without any landmark annotations or explicit city navigation instructions, relying exclusively on images observed at each intersection. This task requires agents to leverage their intrinsic world knowledge to facilitate spatial understanding, accurate self-positioning, and sequential decision-making to reach the goal.

Existing datasets (Mirowski et al., 2018a; Chen et al., 2019) exhibit several limitations: 1) they contain relatively short path lengths, 2) they are restricted to one-two cities, 3) given their widespread usage over the years, there is a significant likelihood that MLLMs have been exposed to these datasets during training. Hence, we introduce a novel, diverse dataset, CityNav, comprising paths of length greater than 2 Km, and include over 50+ decision points, spanning four distinct cities, thereby testing varied capabilities of MLLMs and significantly enhancing the task complexity. Importantly, to mirror the inherently multilingual nature of real-world urban navigation (Pfeiffer et al., 2022; Chen et al., 2024), CityNav is multilingual, featuring routes with diverse language cues (e.g., street signs) across its cities. Our dataset is constructed using Google Street View panograph (Anguelov et al., 2010). Alongside the dataset, we provide a robust evaluation platform capable of deploying MLLMs directly onto the Google Street View navigation graph. The platform is explicitly designed to handle

practical navigation challenges such as dead ends, missing street connections, and abrupt transitions inherent to Google Street View.

We further introduce *Verbalization of Path (VoP)*, a mechanism designed to explicitly extract and leverage the latent world knowledge internalized by MLLMs. By prompting agents to verbalize navigation paths, as illustrated in Fig. 3, VoP substantially enhances the performance of MLLM-based agents on long-range navigation tasks. Since navigating unstructured environments is widely regarded as a fundamental hallmark of intelligence (Mirowski et al., 2018a), our results highlight the effectiveness of VoP in bridging the gap between static reasoning capabilities and dynamic, real-world sequential decision-making.

Our main contributions are 1) We introduce a new task and dataset designed to test MLLMs on long-range sequential decision-making that requires leveraging their internal world knowledge. 2) We propose a zero-shot framework *Verbalization of Path* to elicit and utilize the internal world knowledge of MLLMs for effective outdoor navigation. 3) We show that MLLMs can successfully navigate complex urban environments such as New York City, indicating that these models possess extensive structured world knowledge capable of supporting real-world spatial reasoning. 4) We demonstrate that state-of-the-art reasoning techniques (e.g., GEPA, reflection) that are effective for static reasoning fail in embodied setting.

2 Related Works

MLLMs and Sequential Decision Making. Recent advancements in prompting (Wei et al., 2022; Yao et al., 2023a; Wang et al., 2022; Yao et al., 2024; Manas et al., 2024; Wang et al., 2023) have shown LLMs can exhibit sophisticated reasoning, and significantly improve performance on tasks requiring intermediate reasoning in static environment. (Yao et al., 2023b; Shinn et al., 2023; Gao et al., 2023; Agrawal et al., 2025; Xu et al., 2023) extend this further by iteratively planning through interactive feedback and reflection. In this work, we show that while these methods perform effectively in static contexts, they degrade significantly in sequential decision-making tasks that require methods to *coax* the internal world knowledge.

Instruction-based outdoor navigation. Vision-and-Language Navigation (VLN) (Gu et al., 2022) addresses the challenge of jointly grounding linguistic instructions and visual perception in realistic environments. Prior works (Chen et al., 2019; Schumann & Riezler, 2021; 2022) introduced landmark-rich navigation dataset. (Hermann et al., 2020) aligned textual instructions with visual observations in partially observable Street View environments, while (Mirowski et al., 2018b) employed reinforcement learning to improve navigation robustness. More recent methods (Schumann et al., 2024; Zeng et al., 2024; Xu et al., 2025; Tian et al., 2024) fine-tune MLLMs with city landmark–based instruction following, whereas (Li et al., 2024) leverages driving videos to provide dense visual supervision for route following. All prior approaches rely on explicit textual or landmark-based instructions and operate over short trajectories (typically under 350 m). In contrast, our dataset CityNav focuses on long-range navigation (average path length $\approx$ 2 km) and provides no auxiliary environmental information—only images at each intersection—requiring the model to infer spatial relations, coax its internal world knowledge and plan trajectories.

3 CityNav

We model autonomous city navigation as a Partially Observable Markov Decision Process (POMDP) defined on an undirected graph $G = (V, E)$, where V denotes intersections and E represents undirected street segments, each associated with a positive length $\ell(e) > 0$ for $e \in E$.

At any given discrete time step t, the state s_t of the agent corresponds directly to its current intersection: $s_t = v_t \in V$. From an intersection v_t, the set of available actions is defined as the set of street segments incident to v_t: $\mathcal{A}(v_t) = \{e \in E \mid v_t \in e\}$. When an agent at intersection v_t selects an action $a_t = e \in \mathcal{A}(v_t)$, it deterministically transitions to the adjacent intersection v' connected by the chosen street segment $e = \{v_t, v'\}$.

The system exhibits *sparse grounding*, as visual observations are only available at intersections and entire state in form of map is not available. Specifically, upon reaching an intersection v_t, the agent

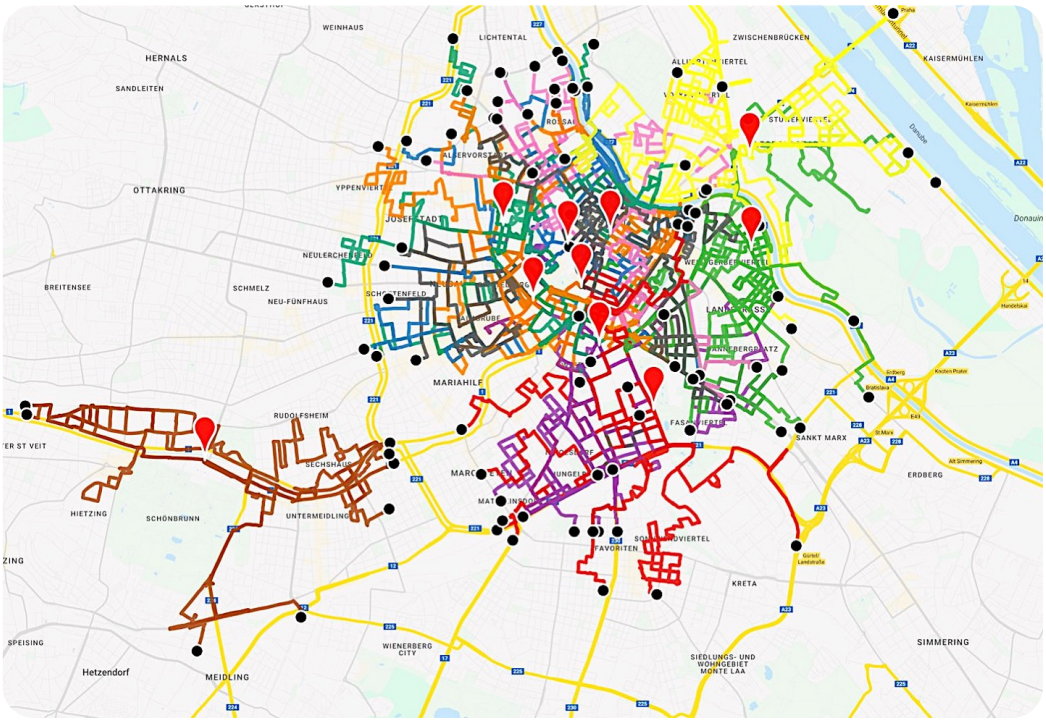


Figure 2: Dataset paths visualization of Vienna. Here the black dots mark the starting point, and the red blobs mark the destination point.

receives a set of images: $o_t = \{I_e \mid e \in \mathcal{A}(v_t)\}$, where each image I_e corresponds to visual input associated with the street segment represented by edge e. Between intersections, while navigating along a street segment, the agent receives no visual observations.

Action selection at each intersection is governed by a policy π, defined as a mapping from the current state v_t and observation set o_t to an action a_t: $\pi : V \times O \to E, \quad a_t = \pi(v_t, o_t)$, where O represents the space of possible observation sets. The policy thus dictates the agent's decision-making process, leveraging available visual information to select the next street segment to traverse.

3.1 DATASET

We curated a diverse dataset explicitly designed to evaluate multiple dimensions of navigation planning in MLLM-based embodied agents. Our city selection methodology targeted locations exhibiting considerable diversity across primary language usage, architectural style, signage characteristics, and street layout topology, including variations in grid density and road complexity. This strategic diversity ensures exposure of the model to a comprehensive array of real-world navigational scenarios.

Specifically, we selected four globally distributed cities, each presenting distinct navigational challenges designed to rigorously assess the adaptability of the model. For example, Tokyo, Japan, predominantly employs Japanese-language signage and place nomenclature, thus posing significant linguistic barriers to LLMs primarily trained on English-dominated corpora. Within each city, we systematically identified and annotated 100 distinct origin-destination pairs, forming standardized evaluation tasks.

Manual Annotation. The destinations are not always just a building, sometimes they are as big as park. So in those cases, the destination doesn't necessarily need to be just a node, it can be collection of nodes. Since it's not possible to figure this out algorithmically, we manually annotate the destinations for each of the chosen places. So we draw a polygon around the destination that acts as a boundary for the destination and when agent reaches this boundary we call it reached.

Cities	Region	Diversity	Distance	Decision Points
New York	USA	Grid-Based, Well Spaced, Rich Street Signs	1.8	44
São Paulo	Brazil	Non-Block Structure, Portuguese Language	2.0	55
Tokyo	Japan	Short Sightlines in Narrow Alleys, Japanese Language	1.9	80
Vienna	Austria	Road blocks because of rails, German Language	2.1	60

Table 1: Dataset statistics across four cities. *Diversity*: qualitative descriptors of urban form and visual/linguistic variety that affect navigation (e.g., grid regularity, sightlines, signage language). *Distance*: average path length (km) for routes in our test split. *Decision Points*: mean number of discrete navigation decisions per route (intersections).

Random Sampling of Starting Location. Starting from a seed node $v_s \in V$, our crawler aims to reach a target radial distance d_{target} from v_s. The traversal operates in two distinct phases: first, a deterministic corridor-following phase continues along nodes having an effective out-degree of 1 (excluding the backward link), until encountering the first decision junction (nodes with out-degree ≥ 2). The second phase involves a depth-first search (DFS) using an explicit junction stack with backtracking. At each junction node v_j, the crawler chooses among candidate edges $e_j^i \in E(v_j)$ according to a probability distribution computed via softmax over their angular deviation θ_i relative to the desired heading (typically directed away from the seed node). Specifically, the selection probability is: $P(e_j^i) = \frac{\exp(\cos(\theta_i)/T)}{\sum_k \exp(\cos(\theta_k)/T)}$ where the temperature parameter T anneals with increasing straight-line distance $d(v_j, v_s)$ from an initial random exploration ($T \to \infty$) toward a progressively more directional selection (lower T). To mitigate loops and encourage diverse coverage, we impose an exponential revisit penalty factor γ^{n_v} to the selection probability, where n_v is the visit count for node v, and $0 < \gamma < 1$. Once the crawler reaches the radial target distance d_{target}, it optionally continues for a small number of steps to terminate at a node with degree greater than or equal to a threshold $d_{\text{min\_final}}$.

Google Street View Pre-processing. We enhance the underlying Google Street View graph for reliable navigation tasks, we systematically identify and resolve common structural issues. One of the primary challenges in constructing our navigation graph from Google Street View data arises from structural inconsistencies such as dead ends, incomplete coverage, and asymmetric links. We define a *dead end* as a node with an outgoing edge to a neighboring node that does not lead to any further intersections (i.e., a terminal street segment with no valid successors). To ensure graph connectivity and eliminate such artifacts, we algorithmically identify and prune dead ends during preprocessing. Additionally, the Street View panograph often exhibits asymmetric connectivity, where a link from node a to node b exists, but the reverse link from b to a is absent. This breaks the undirected graph assumption required for consistent navigation. To resolve this, we crawl the underlying graph and explicitly add the missing reverse edges whenever such inconsistencies are detected, thereby restoring bidirectional connectivity and ensuring that the resulting graph is well-formed for navigation tasks.

4 AGENTNAV

4.1 GROUNDING WITH VERBALIZED PATHS

Successful outdoor navigation fundamentally requires accurate self-localization and comprehensive world knowledge. Here, we demonstrate a targeted approach to probe such knowledge explicitly from MLLMs. We augment the agent's prompt-as-policy framework with three distinct phrases that consistently elicit robust navigation performance by explicitly grounding the agent's internal state and reasoning in the external world. Specifically, we incorporate the following structured prompts: 1) *Write the exact location of the destination*: This explicitly defines the navigation goal, anchoring the agent's decision-making process to a clear terminal state. 2) *Write the current estimated exact location*: This compels the agent to continuously estimate and update its current position, serving as a precise initial condition for subsequent decisions. 3) *Write the walking directions from the current position to the destination*: Crucially, this leverages the agent's generalist knowledge, prompting it to generate actionable instructions grounded in real-world spatial relationships and pathfinding logic.

Figure 3: Illustration of state transition from S (marked by the yellow dot) to $S+1$ using the agent's internal reasoning through *Verbalization of Path*. At each state, the agent perceives visual cues and references its memory to update decisions and navigation strategy. The purple marker denotes the destination (One World Trade Center), while the green marker indicates the starting point.

4.2 Memory of AgentNav

For extensive runs averaging over 50+ decisions per trajectory, maintaining an efficient memory system is critically important. Traditional episodic memory architectures which store comprehensive information (images, decisions, analyses) for each step across multiple past episodes quickly become computationally intractable, scaling exponentially with episode length. To mitigate this issue, we strategically decompose memory management into three core components that significantly enhance efficiency (approximately a 100-fold reduction in memory overhead) within a Partially Observable Markov Decision Process (POMDP) framework: *Markovian Memory*, *Decision History*, and *Previous Visit Tracking*.

Markovian Memory. We implement *Markovian Memory* by explicitly prompting the agent to produce a memory state at each decision step. Formally, at time step t, the agent's input includes the previous memory state m_{t-1}, and along with next action a_t it outputs updated memory state m_t. This process can be expressed as: $(a_t, m_t) = \pi(v_t, o_t, m_{t-1})$ where m_t represents a sufficient statistic summarizing past observations, effectively transforming the partially observable process into a Markovian one within an augmented state space $\tilde{s}_t = (v_t, m_t)$. This mechanism eliminates the need for full episodic memory, thereby significantly reducing computational and storage costs. Furthermore, as the model becomes increasingly capable, it learns to selectively preserve only the most relevant information for effective decision-making, resulting in a more compact and adaptive memory representation.

Decision History. It maintains a structured record of the sequence of actions chosen by the agent at each intersection during the trajectory. Formally, this can be represented as: $\mathcal{H}_t = \{a_1, a_2, \ldots, a_t\}$. Maintaining this ordered sequence enables the agent to leverage its own behavioral trajectory for reasoning about prior choices, route corrections, and avoid repeated loops. By focusing on compact action traces instead of complete episodic histories, this mechanism provides a balance between computational efficiency, and long-horizon reasoning.

Previous Visit. It provides the agent with awareness of past interactions at specific intersections. Each time the agent revisits an intersection v_t, it retrieves the record of its previous decisions taken at that node, enabling it to reason about prior outcomes. Repeatedly encountering the same intersection typically indicates that the agent is caught in a local loop or has failed to make progress toward the destination. To mitigate this, the system encodes a visit count n_v for each node v, which influences the policy's exploration behavior. As n_v increases, the agent is progressively discouraged from repeating the same action-promoting exploration and preventing cyclic behavior. For example,

		New York			Tokyo			Vienna			Sao Paulo		
MLLM	**Agent Config**	**Success**	**SPL**	**D.A.**	**Success**	**SPL**	**D.A.**	**Success**	**SPL**	**D.A.**	**Success**	**SPL**	**D.A.**
GPT 4o	Base	13	0.064	39.04	4	0.046	36.79	4	0.031	35.67	3	0.040	34.69
	AgentNav	88	0.539	72.91	14	0.099	40.90	26	0.170	46.32	20	0.06	43.45
GPT 5	Base	54	0.375	55.97	10	0.088	41.24	11	0.092	40.72	7	0.051	36.98
	AgentNav	94	0.711	82.98	30	0.163	54.97	56	0.226	54.82	29	0.126	48.96
GPT 4.1	Base	15	0.097	42.27	5	0.044	38.83	2	0.037	34.66	5	0.049	35.46
	AgentNav	92	0.557	75.27	17	0.101	43.67	32	0.182	49.95	22	0.080	44.05
O3	Base	48	0.490	64.36	7	0.049	40.33	9	0.083	39.82	6	0.075	35.68
	AgentNav	95	0.759	84.93	27	0.142	52.56	38	0.190	50.73	24	0.117	50.75
Gemini 2.5 Flash	Base	12	0.060	41.57	8	0.049	39.97	1	0.010	29.31	5	0.049	35.80
	AgentNav	73	0.471	74.75	17	0.066	46.87	17	0.137	46.35	12	0.085	43.65
Qwen 2.5 VL 32b	Base	7	0.089	35.11	2	0.023	30.01	0	0.0	26.1	2	0.011	29.87
	AgentNav	32	0.153	56.39	12	0.094	40.03	12	0.119	44.94	9	0.059	37.80

Table 2: Base model vs. AgentNav across four cities. We report Success, SPL(Success weighted by Path Length), and D.A. (Decision Accuracy); higher is better. AgentNav consistently and substantially improves performance over the base MLLM across all model families, indicating importance of VoP method.

if the agent has chosen to go west multiple times from an intersection without improvement, the memory mechanism biases future actions toward unexplored directions such as east. This structured representation of visit history thus endows the agent with self-awareness of its traversal patterns, improving navigational robustness in complex city graphs.

5 EXPERIMENTATION

In this section, we present a comprehensive empirical evaluation of AgentNav. We benchmark six strong MLLMs and multiple state-of-the-art reasoning and navigation baselines on CityNav. We then quantify the contribution of each component through ablation studies (Sec. 5.5).

5.1 IMPLEMENTATION DETAILS

For each run within every city, we set a maximum limit of 150 decision points for the agent to reach its destination before automatic termination. Additionally, we restrict the maximum number of graph node transitions to 2000. Rather than executing self-positioning at every decision point, we perform self-positioning every third decision point (a separate call to same MLLM). This deliberate choice introduces minor positional uncertainty, effectively testing the robustness and accuracy of the verbalized path by challenging the agent to reason with slightly imprecise localization.

5.2 MLLMS AND REASONING BASELINES

We evaluate our method using a selection of strong closed-source and open-source multimodal foundation models to effectively probe their internal world knowledge and reasoning capabilities. Specifically, our evaluation includes: GPT-4o (Achiam et al., 2023a), a widely-used baseline model in multimodal research; GPT-4.1, which is expected to demonstrate enhanced geographical reasoning capabilities (Grainge et al., 2025); Gemini-2.5 Flash (Team et al., 2024), serving as an additional closed-source comparison; GPT-5 (thinking) & O3, known for its advanced reasoning abilities; and Qwen-2.5VL-32B (Bai et al., 2025), a powerful open-source counterpart. Our focus on these sophisticated multimodal models is motivated by our goal of *coaxing* out the latent world knowledge embedded within large-scale, web-trained models.

To systematically evaluate reasoning effectiveness, we benchmark our approach against state-of-the-art reasoning baselines, including GEPA (Agrawal et al., 2025), Chain-of-Thought (CoT) (Wei et al., 2022), Self-Reflection (GPT-4.1) (Shinn et al., 2023), and Self-Reflection (GPT-5). Here, the labels GPT-4.1 and GPT-5 denote the specific models employed during the reflective reasoning step, wherein the initial reasoning output is revisited and refined. Additionally, we compare our method with the state-of-the-art outdoor navigation baseline, PReP (Zeng et al., 2024).

	New York			Tokyo			Vienna			Sao Paulo		
Method	**Success(%)**	**SPL**	**D.A.(%)**	**Success(%)**	**SPL**	**D.A.(%)**	**Success(%)**	**SPL**	**D.A.(%)**	**Success(%)**	**SPL**	**D.A.(%)**
GPT-4.1	15	0.097	42.27	5	0.044	38.83	2	0.037	34.66	5	0.049	35.46
CoT	21	0.173	44.59	9	0.077	41.09	4	0.039	34.88	7	0.055	37.93
Self Reflection (GPT-4.1)	16	0.112	42.90	4	0.040	36.20	3	0.042	36.33	12	0.052	41.95
Self Reflection (GPT-5)	22	0.168	48.14	8	0.079	41.48	5	0.045	37.84	13	0.050	41.64
GEPA	37	0.251	43.24	10	0.036	42.97	5	0.013	39.74	17	0.093	40.21
PReP	39	0.248	36.07	5	0.010	40.68	5	0.025	38.11	22	0.157	41.11
AgentNav	92	0.557	75.27	17	0.101	43.67	32	0.182	49.95	22	0.080	44.05

Table 3: The table shows comparison results with different baselines. All the experiments here are performed using GPT-4.1. Self Reflection (GPT-4.1) means the agent used is GPT-4.1 and reflection is done with GPT-4.1. Self Reflection (GPT-5) agent used is GPT-4.1 and reflection is done with GPT-5.

5.3 Evaluation Metrics

We employ standard reasoning and navigation metrics:

• **Success:** If the agent reaches the destination node successfully, it receives a score of 1; otherwise, the score is 0.

• **SPL (Success weighted by Path Length):** SPL evaluates the agent's navigation efficiency by comparing the optimal (shortest possible) path distance d_{opt} to the actual distance traveled by the agent d_{agent}, scaled by the binary success indicator $S \in \{0, 1\}$. SPL $= S \times \frac{d_{\text{opt}}}{\max(d_{\text{agent}}, d_{\text{opt}})}$ where $S = 1$ if the agent successfully reaches the destination, and $S = 0$ otherwise.

• **Decision Accuracy (D.A.):** The percentage of correct navigation decisions (e.g., correct turns at junctions) made by the agent. A decision is classified as correct if the remaining walking distance to the destination (calculated using the Google Street View API) decreases after executing that decision.

5.4 Quantitative Results

Significant Performance Across 5 MLLMs. Table 2 demonstrates that VoP consistently delivers strong improvements across all five evaluated MLLMs. In high-density urban settings such as New York, our approach achieves significantly higher success rates and SPL scores compared to the base configurations of each model. This indicates that the *Verbalization of Path* mechanism effectively coaxes latent world knowledge from MLLMs, enabling more reliable decision-making in complex navigation tasks. Importantly, similar performance gains are observed across all four cities, highlighting the generality of VoP.

Limitations of Existing Reasoning Techniques and Navigation baseline. Table 3 highlights the limitations of state-of-the-art reasoning strategies. While these techniques have proven effective for static reasoning tasks, they fail to achieve competitive performance on our long-range, embodied navigation benchmark. Their inability to elicit sufficiently structured internal knowledge from MLLMs leads to lower success rates.

Reflection baselines degrade in long-horizon, real-world navigation. Reflection-based baselines (e.g., GEPA and reflection-style prompting) work extremely well in static or short-horizon settings but substantially underperform on our real-world, long-horizon benchmark: GEPA achieves only 37% success, and Self-Reflection achieves only 22% success even when the reflection step uses a stronger model (e.g., GPT-5).

Challenges Arising from the Diversity of CityNav. Despite the substantial gains delivered by AgentNav, Table 2 also reveals the inherent difficulty of long-horizon sequential decision-making tasks. Even with *Verbalization of Path*, absolute success rates and SPL scores remain modest for Tokyo, Vienna and Sao Paulo, reflecting the persistent gap in the reasoning abilities of current MLLMs and reasoning methods (see Table 3).

Methods	Success(%)	SPL	D.A.(%)
GPT-4.1	15	0.097	42.27
GPT-4.1 + Markovian Memory	23	0.162	47.19
GPT-4.1 + Decision History	29	0.228	55.63
GPT-4.1 + Previous Visit	35	0.298	56.67
GPT-4.1 + Partial VoP	66	0.469	63.48
AgentNav	**92**	**0.557**	**75.27**

Table 4: Performance breakdown showing contributions of different components of the AgentNav.

5.5 Ablations and Analysis

We systematically analyze the contributions of individual components of our method, as detailed in Table 4. Starting with the GPT-4.1 baseline, we observe a success rate of 15%. Incrementally integrating memory components highlights their individual impacts: incorporating *Markovian Memory* increases success to 23%, and further addition of *Decision History* boosts performance to 29%. Finally, integrating *Previous Visit* yields an additional enhancement, reaching a success rate of 35%.

Next, we further dissect the VoP method by breaking it down into two stages (Partial VoP and VoP) to clearly identify its effect. In the *Partial VoP* scenario, we instruct the agent to explicitly ground its reasoning solely based on the final destination, prompting it to write the destination's address at the beginning of its reasoning. This targeted grounding notably enhances the success rate to 66%. Subsequently, employing the complete verbalization, where the agent explicitly generates detailed walking directions from its current location to the target destination, our *AgentNav* model reaches the highest performance with a 92% success rate. This stepwise analysis underscores how each introduced stage of verbalization progressively grounds and elicits richer world knowledge from MLLMs, significantly improving the agent's capability to reason and navigate reliably in complex, real-world scenarios.

6 Limitations

While CityNav and the proposed AgentNav framework constitute a substantial advance in the systematic evaluation of MLLMs on real-world, long-range navigation tasks, several limitations persist. 1) Despite the incorporation of explicit verbalization and memory mechanisms, the absolute success rates and SPL scores remain modest for certain model–city combinations, underscoring a persistent gap between current MLLM capabilities and the requirements of large-scale, embodied spatial reasoning. 2) The benchmark currently focuses on four urban environments but does not exhaust the full spectrum of urban layouts, linguistic settings, or signage complexities. Models trained predominantly on English-centric corpora may display limited robustness and generalization to regions characterized by complex writing systems, multilingual signage, or low-visibility conditions.

7 Conclusion

In this work, we introduced CityNav, a comprehensive benchmark designed to rigorously assess MLLMs on real-world, long-range urban navigation tasks. Through extensive experiments across diverse global cities, we demonstrated that our proposed *Verbalization of Path* mechanism, complemented by strategic memory components, effectively coaxes the intrinsic world knowledge from MLLMs, resulting in significant performance improvements over existing reasoning methods. These findings underline current limitations of MLLMs for embodied sequential decision-making tasks, emphasizing the need for continued research in robust, adaptive reasoning frameworks.

REFERENCES

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

Joshua Achiam, Sarina Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, F. L. Aleman, Daniel Almeida, Jan Altenschmidt, Sam Altman, Shantanu Anadkat, et al. GPT-4 Technical Report, 2023b.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2020.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12538–12547, 2019.

Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. On scaling up a multilingual vision and language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14432–14444, 2024.

Zhe Chen, Jiapeng Wu, Weijian Wang, Wensheng Su, Gongwei Chen, Sheng Xing, Mulin Zhong, Qihang Zhang, Xizhou Zhu, Liang Lu, Bo Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks, 2023.

Zhe Chen, Weijian Wang, Yue Cao, Yufei Liu, Zizheng Gao, Enming Cui, Junyang Zhu, Shilong Ye, Hao Tian, Zheng Liu, Lianlong Gu, Xinlong Wang, Qirong Li, Yi Ren, Zhiyu Chen, Jian Luo, Jinghao Wang, Tianhe Jiang, Bin Wang, Changxin He, Bo Shi, Xiaoqing Zhang, Hanyu Lv, Yu Wang, Wenqi Shao, Peng Chu, Zhenguo Tu, Tao He, Zheng Wu, Haohua Deng, Jinrui Ge, Kehan Chen, Kai Zhang, Liang Wang, Mingsong Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799. PMLR, 2023.

Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaojian Ma, Tao Yuan, Yue Fan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. Multi-modal agent tuning: Building a vlm-driven agent for efficient tool usage. *arXiv preprint arXiv:2412.15606*, 2024.

Oliver Grainge, Sania Waheed, Jack Stilgoe, Michael Milford, and Shoaib Ehsan. Assessing the geolocation capabilities, limitations and societal risks of generative vision-language models. *arXiv preprint arXiv:2508.19967*, 2025.

Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Eric Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. *arXiv preprint arXiv:2203.12667*, 2022.

Karl Moritz Hermann, Mateusz Malinowski, Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, and Raia Hadsell. Learning to follow directions in street view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11773–11781, 2020.

Chuxuan Hu, Dwip Dalal, and Xiaona Zhou. A Dataset-Centric Survey of LLM-Agents for Data Science, 2025.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pp. 9118–9147. PMLR, 2022.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

J. Li, A. Padmakumar, G. Sukhatme, and M. Bansal. VLN-Video: Utilizing driving videos for outdoor vision-and-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18517–18526, 2024.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Haotian Liu, Chenyang Li, Yuhang Li, Bo Li, Yutong Zhang, Shiyu Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge. `https://llava-vl.github.io/blog/2024-01-30-llava-next/`, 2024.

Kumar Manas, Stefan Zwicklbauer, and Adrian Paschke. Cot-tl: Low-resource temporal knowledge representation of planning instructions using chain-of-thought reasoning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9636–9643. IEEE, 2024.

P. Mirowski, M. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, A. Zisserman, R. Hadsell, et al. Learning to navigate in cities without a map. In *Advances in Neural Information Processing Systems*, 2018a.

Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Andrew Zisserman, Raia Hadsell, et al. Learning to navigate in cities without a map. *Advances in neural information processing systems*, 31, 2018b.

OpenAI. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`, 2024.

Jonas Pfeiffer, Gregor Geigle, Aishwarya Kamath, Jan-Martin O. Steitz, Stefan Roth, Ivan Vulić, and Iryna Gurevych. xgqa: Cross-lingual visual question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2497–2511. Association for Computational Linguistics, 2022.

Anku Rani, S.M Towhidul Islam Tonmoy, Dwip Dalal, Shreya Gautam, Megha Chakraborty, Aman Chadha, Amit Sheth, and Amitava Das. FACTIFY-5WQA: 5W aspect-based fact verification through question answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), pp. 10421–10440, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.581. URL https://aclanthology.org/2023.acl-long.581/.

Anku Rani, Dwip Dalal, Shreya Gautam, Pankaj Gupta, Vinija Jain, Aman Chadha, Amit Sheth, and Amitava Das. SEPSIS: I can catch your lies – a new paradigm for deception detection. In Jin Zhao, Mingyang Wang, and Zhu Liu (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 97–128, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-254-1. doi: 10.18653/v1/2025.acl-srw.7. URL https://aclanthology.org/2025.acl-srw.7/.

Machel Reid, Nikolay Savinov, Dmitry Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024.

R. Schumann and S. Riezler. Generating landmark navigation instructions from maps as a graph-to-text problem. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 489–502, 2021.

R. Schumann and S. Riezler. Analyzing generalization of vision and language navigation to unseen outdoor areas. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 7519–7532, 2022.

R. Schumann, W. Zhu, W. Feng, T.-J. Fu, S. Riezler, and W. Y. Wang. VELMA: Verbalization embodiment of LLM agents for vision and language navigation in Street View. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18924–18933, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2998–3009, 2023.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

H. Tian, J. Meng, W. S. Zheng, Y. M. Li, J. Yan, and Y. Zhang. Loc4plan: Locating before planning for outdoor vision and language navigation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 4073–4081, 2024.

Peiyi Wang, Shusheng Bai, Shuo Tan, Shiji Wang, Zhe Fan, Jun Bai, Kexin Chen, Xiang Liu, Jinghao Wang, Wenhai Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Junlin Xie, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. Large multimodal agents: A survey. *arXiv preprint arXiv:2402.15116*, 2024.

Weijia Xu, Andrzej Banburski-Fahey, and Nebojsa Jojic. Reprompting: Automated chain-of-thought prompt inference through gibbs sampling. *arXiv preprint arXiv:2305.09993*, 2023.

Yunzhe Xu, Yiyuan Pan, Zhe Liu, and Hesheng Wang. Flame: Learning to navigate with multimodal llm in urban environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9005–9013, 2025.

Ao Yang, Botao Yang, Bo Zhang, Bo Hui, Bing Zheng, Bowen Yu, Chaoqun Li, Dong Liu, Fei Huang, Hong Wei, et al. Qwen2.5 technical report, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

Yao Yao, Zuchao Li, and Hai Zhao. Got: Effective graph-of-thought reasoning in language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2901–2921, 2024.

Junpeng Yue, Xinrun Xu, Börje F Karlsson, and Zongqing Lu. Mllm as retriever: Interactively learning multimodal retrieval for embodied agents. *arXiv preprint arXiv:2410.03450*, 2024.

Qingbin Zeng, Qinglong Yang, Shunan Dong, Heming Du, Liang Zheng, Fengli Xu, and Yong Li. Perceive, reflect, and plan: Designing llm agent for goal-directed city navigation without instructions. *arXiv preprint arXiv:2408.04168*, 2024.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024.

APPENDIX

A GOOGLE STREET VIEW POLICY COMPLIANCE STATEMENT

Our study uses Google Street View imagery only at evaluation time and only via official Google-provided interfaces. The evaluation platform executes the agent directly on the Street View navigation graph and fetches the panorama views transiently for decision making; we do not scrape, bulk-download, mirror, or redistribute Street View imagery. The public release accompanying this paper consists of annotations (origin–destination pairs, destination polygons), code, and evaluation scripts. No images will be released. Human annotation was performed by the authors and is limited to drawing destination boundaries; no personally identifying information was collected or added. As documented in the paper (dataset description in Section 3.1; summary statistics in Table 1), imagery appears only as on-the-fly observations at intersections, multilingual scene text may be visible within those images, and all faces and license plates remain blurred as provided by Google, since we do not alter or post-process Street View content.

To align fully with Google's Street View and Maps Platform Terms of Service, we (i) access imagery solely through authorized APIs/viewers; (ii) preserve all provider attribution, blurring, and watermarks returned by the service; (iii) avoid storing or caching raw imagery beyond ephemeral runtime needs; (iv) release no derivative image dataset (only text/graph metadata and author-created annotations under our license); (v) require downstream users to supply their own API keys and to accept and comply with Google's Terms when reproducing our results; and (vi) forbid any use of our code or annotations to scrape, de-watermark, reverse-engineer, or otherwise circumvent Google's technical and policy safeguards. Our figures, where illustrative thumbnails are necessary for scholarly reporting, are minimal and strictly for explanation of results; they do not constitute redistributable imagery or a dataset. These measures, together with our annotations-only release and authors-only human labeling protocol, ensure that the work adheres to Google's Street View policy while enabling reproducible research on long-range, real-world navigation.

B ADDITIONAL DETAILS

LLM parameters. OpenAI (GPT-4o, GPT-4.1; non-reasoning): `temperature=1.0`, `top_p=1.0`, `presence_penalty=0`, `frequency_penalty=0`, `n=1`, `stream=false`, `max_tokens=8000`.
Gemini 2.5 Flash: `temperature=1.0` (0–2), `top_p=0.95` (0–1), `top_k=64`, `candidate_count=1`, `max_tokens=8000`.
Qwen 2.5-VL (FP16; Ollama): `temperature=0.8`, `top_p=0.9`, `top_k=40`, `repeat_penalty=1.1`, `repeat_last_n=64`, `num_predict=-1`.

Streetview API Parameters. The Street View Static API parameters were as follows:

- `size=512x512` (pixels): output resolution of each crop.
- `fov=90` (degrees, horizontal field of view): the angular width of the panorama that is projected into the image (i.e., how wide the virtual camera "sees").
- `pitch=+30` (degrees): camera tilt relative to the horizon; positive values tilt upward.

The API returns a rectilinear view extracted from the underlying panorama. We set `fov=90`$^\circ$ to balance scene coverage with per-pixel detail, and `pitch=+30`$^\circ$ to de-emphasize the ground plane

(road surface) and emphasize facades/skyline features that are more informative. The `heading` parameter is derived from the link direction using the 3-hop link calculation method described in the paper.

Agent Run Parameters. We run the agent for `max_steps=2000` and `max_decision_points=150`.

C IMPROVEMENTS TO GOOGLE STREETVIEW BASE GRAPH

The base panoramas and connectivity sourced from Google Street View are inherently noisy and contain multiple flaws, which required careful mitigation to ensure dataset reliability. We explicitly apply the following improvements during dataset construction to refine the base graph and increase the resulting dataset's value:

1. **Robust crawler seeding under isolated sub-networks and indoor panoramas.** Often, the initial panorama we choose for the crawler may be part of an isolated sub-network or an indoor panorama. In such a case, we manually prune the isolated sub-network, and reiterate.
2. **Increased exploration range beyond random-walk initialization.** Initially, data collection relied on a random-walk strategy for selecting starting points. However, this approach restricted the crawler's exploration radius, preventing sufficient geographical coverage. To address this, we introduced a dynamic temperature (T) parameter that strategically decreases randomness as a function of distance, effectively guiding the crawler toward diverse and more distant locations.
3. **Eliminating dead ends in Street View panographs.** Street View panoramas frequently contain dead ends, nodes that lead exclusively to terminal segments without further intersections, causing agents to become trapped in infinite loops. To resolve this, we algorithmically detect and prune such dead-end nodes during preprocessing, ensuring robust graph connectivity and eliminating navigational artifacts.
4. **Manual annotation via destination polygon construction.** Destinations in our dataset are not limited to individual buildings; they often span extensive areas, such as parks or complexes. Consequently, representing a destination as a single node is often inadequate, necessitating a collection of nodes instead. To address this and precisely define termination criteria, we manually annotate polygons around each destination, clearly delineating their spatial boundaries.
5. **Fixing asymmetric connectivity and unexpected node jumps.** Street View panoramas frequently exhibit asymmetric connectivity, where a node links to another without a reciprocal connection. To resolve this, we explicitly identify and add missing reverse edges, restoring graph symmetry. Additionally, we mitigate unexpected node jumps caused by Street View errors through careful calibration and enforcement of a distance threshold.
6. **Precise orientation alignment for image capture.** To obtain meaningful images at an intersection, we recalibrate the panorama's default heading, which frequently points inaccurately toward buildings. Specifically, we move three nodes ahead from the initial position, compute the optimal heading aligned with the street's actual direction, and then capture the image. This careful orientation ensures consistency between the imagery and real-world street alignments.

D STRONG PROMPT-BASED BASELINES

We use prompt ablations to illustrate the difficulty of our setting and to contextualize the novelty of VoP. Specifically, we construct *strong prompt-based baselines* that combine common prompting strategies (structured reasoning, confidence scoring, elimination, self-critique, and multi-agent role decomposition) but intentionally exclude our Verbalization of Path (VoP) mechanism. This evaluation tests whether sensible alternative prompts can substitute for VoP in long-range, sparsely grounded navigation.

Prompt	C1	C2	C3	C4	C5	C6	C7	Score (NY)
P1	✓		✓					19
P2	✓	✓	✓					27
P3	✓		✓	✓	✓			23
P4	✓		✓		✓			20
P5			✓			✓		29
P6	✓		✓				✓	26
P7	✓	✓	✓	✓	✓		✓	30

Table 5: Prompt-component ablation on the New York split. Each prompt variant P1–P7 enables a subset of components C1–C7.

We design a *modular* prompting template composed of interpretable components (C1–C7). Each component targets a specific capability (e.g., extracting visual cues, structured reasoning, or self-verification). We instantiate multiple prompt variants (P1–P7) by enabling different subsets of components and evaluate them on the New York split. Table 5 reports which components are included in each variant and the resulting score. The example of the prompts formed from these components are in Appendix Section F.

Component definitions. Our prompt template is built from the following modules:

- **C1 (Visual inference instructions).** The agent infers geographic and semantic cues from each image (e.g., signboards, shop names, traffic flow) and uses them for navigation.
- **C2 (Self-confidence estimation).** The agent assigns calibrated confidence scores to candidate actions and selects the highest-confidence direction.
- **C3 (Multi-step reasoning protocol).** The agent follows a 4-step observe–infer–plan–question process before deciding.
- **C4 (Cardinal direction estimation).** The agent estimates the target's approximate cardinal direction (N/S/E/W) from the current location and uses it to guide decision making.
- **C5 (Elimination-based decision making).** The agent systematically eliminates unlikely directions to select the best remaining candidate.
- **C6 (Multi-agent decomposition).** A multi-agent system with specialized roles (image analyzer, planner, decision maker) that communicate and aggregate their outputs.
- **C7 (Self-critique and consistency checking).** The agent generates a detailed self-critique explaining why the chosen action is correct and why alternatives are incorrect.

Table 5 shows that while these prompt-only baselines are carefully engineered and combine multiple best practices, they still achieve limited performance on New York (best score: P7 with 30). This indicates that the challenge is not resolved by stronger phrasing or additional reasoning scaffolds alone: even "high-quality" prompts fail to reliably support long-horizon localization and planning in this split. In contrast, VoP explicitly probes city-scale cognitive map, which is precisely the capability these prompt-only variants lack, supporting the novelty and necessity of our VoP design.

E AGENTNAV PROMPT

```
You are at an intersection with 4 possible directions (options). Below are images
for each option. Analyze the images to determine the best direction towards Grand
Central Terminal. Use the following information to guide your decision:

The images correspond to the following options/directions:
Option step0_option0: facing East (118°)
Option step0_option1: facing South (212°)
Option step0_option2: facing North (29°)
Option step0_option3: facing West (301°)

Estimated position: Intersection of 9th Avenue and West 57th Street, Hell's Kitchen
(Midtown West), Manhattan, New York City. (evidence: )

Write exact location of the destination as precise as you can as first sentence in
analysis.
Write current estimated exact location as next sentence.
One paragraph on reaching plan i.e. Write the walking directions from the current
estimated location to the exact location of the destination.
Put plan in memory with current progress.
Use images for analysis of current position and where to go.
If you see the destination in the image go in that direction.
For each decision mention concrete reason as to why this decision was chosen. Exact
perfect real reason.
If no such reason, it is a random exploration.
You are navigating streetview panoramas where linking may be unexpected, so it is
possible
that direct route may not be possible. If stuck go around.

Return a JSON object strictly matching this schema. The 'decision' MUST be the unique
string ID of your chosen option (e.g., 'stepX_optionY'):
{
  "type": "object",
  "required": ["analysis", "decision", "memory"],
  "properties": {
    "analysis": {"type": "string"},
    "decision": {"type": "string"},
    "memory": {"type": "string"}
  }
}

VALID OPTION IDS (choose exactly one and place it in the 'decision' field):
step0_option0 | step0_option1 | step0_option2 | step0_option3

EXAMPLE OF THE EXPECTED JSON FORMAT (fill with your own analysis, decision, and
memory):
{
  "analysis": "Your reasoning here",
  "decision": "step0_option0",
  "memory": "Any memory to retain for future steps"
}
```

F BASELINE PROMPTS

F.1 GEPA

```
You are at an intersection with 3 possible directions (options). Below are images
for each option. Analyze the images to determine the best direction towards Tokyo
Station. Use the following information to guide your decision:

The images correspond to the following options/directions:
Option step0\_option0: facing North (32°)
Option step0\_option1: facing South (214°)
Option step0\_option2: facing North (39°)

You are a navigation agent tasked with guiding a user to a specified landmark
destination using only visual observation, basic spatial reasoning, and a compass
direction at each intersection. You are immersed in a first-person, street-level
panoramic environment (like Google Street View), moving step by step through
possible navigation options. You do not have access to any map, GPS, or external
city-specific data. There are no locals to ask for help.

At each decision point, you receive a set of panoramic images, each corresponding to
a possible movement option, and the compass bearing for each. You also know which
direction you just arrived from (to avoid immediate backtracking). Your goal is to
select the option most likely to move you closer to the named destination.

You must rely solely on:
- Visual cues in the images—look for features commonly associated with city
infrastructure (e.g., wide avenues, density of buildings, parks, open spaces, rivers,
bridges, architectural styles, landmark silhouettes, signage, traffic density, etc).
- Orientation and direction—reason about the destination's likely location using
general world and city knowledge (e.g., major train stations are typically central,
art museums often near parks or cultural districts, government buildings might be
near water or grand avenues).
- Past movement pattern—avoid unnecessary backtracking and detect when you might be
circling, making lateral progress, or moving away from urban cores or likely
landmark locations.
- Street grid logic—urban environments often have repeating structures; use building
density, street widths, and landmark visual profiles to estimate center vs perimeter
or determine proximity to likely landmark locations.
- Cumulative context / memory—summarize your progress toward the destination,
recalling key landmarks or changes in urban patterns.

Correct navigation judgments involve:
- Progressing toward dense, central, or recognizable landmark environments based on
destination type.
- Preferring direct, unbroken routes unless detours are visually or logically
warranted.
- Adjusting movement as clues appear (e.g., spotting the landmark or matching
landmark silhouettes, more signage, dramatic increases in foot traffic or iconic
city fixtures).
- Recognizing when to switch from major arteries to more local access as the
landmark nears, based on cues from the environment.

Your output at each intersection should briefly explain the reasoning, select the
optimal movement option that best advances toward the goal, and update your
memory/trajectory summary to reflect the decision and any new visual or spatial
clues.

Assume the task must generalize to any city or landmark—even those you have never
seen described explicitly before. You must not depend on any hardcoded city,
neighborhood, or path knowledge.
```

```
Your entire approach should be grounded in visual/spatial inference,
destination-type reasoning, compass use, avoidance of immediate repetition, and
adaptive updating as more information is visually or contextually revealed.

[JSON schema and example omitted for brevity]
```

F.2 TEST PROMPT P7

```
You are at an intersection with 3 possible directions (options). Below are images
for each option. Analyze the images to determine the best direction towards Empire
State Building. Use the following information to guide your decision:

The images correspond to the following options/directions:
Option step0\_option0: facing North (22°)
Option step0\_option1: facing West (299°)
Option step0\_option2: facing East (119°)

Estimated position: This location is on Columbia Street near the intersection with
Cherry Street, in the Two Bridges neighborhood of Lower Manhattan, New York City
(evidence: The buildings visible are part of the large residential complexes in this
area, just east of the Manhattan Bridge and near the FDR Drive.)

First and foremost, your are independent. NO one is reading your responses. You are
writing them for yourself and yourself alone. So do not worry about keeping them
proper for someone else. Respond in a way people talk, not in the way they write.
These responses are your Thoughts, not answers to someone else.
For the analysis of each image produce an at least 300 word analysis of the image.
followed by 300 words explaining why this image could be the right decision, also
why this could be the wrong decision. Then for the reasoning text, provide at least
1000 words of reasoning, where you explain why you are correct, and also approach it
from the opposite perspective and write 500 words on why you are wrong. Finally, use
both perspectives to understand which one it is, then , and only then make a final
decision.
For the reasoning process, follow a process of elimination, and choose the least
wrong option rather than the most correct once since you have no way of knowing which
is correct.

For the self positioning, along with positioning, explain in 500 words why you think
you are correct, and in 500 words why you think you are wrong, and only then weigh
both options to produce the confidence score.

If you decide you are somewhere, and that your destination is somewhere else,  then
explain why and from where in your knowledge base you derived the information.

Maintain all information in your memory as this is the only thing persistent in your
mind. Maintain global information here, like what is where in which direction, what
you have explore etc. be creative. This needs to be at minimum 500 words

You need to pay attention to the previous visits text. if you have been to an
intersection before and went one way then don't go back unless there is an extremely
strong reason. if there is one explain it.
When you don't know where you are, it is best to move in a direction, check it out,
then come back and then explore the other directions. This way, you can explore all
options corresponding to the choices for one decision. then when you have finally
explored all directions, you can make a better decision. in order to remember the
exploration knowledge be sure to include the relevant information in your notes.
IF you see the destination, drop all reasoning, drop all doubts, drop all process,
put "dont care, see destination" in all of the reasoning texts, forget everything
and choose the direction that takes you towards the visible destination. FORGET
EVERYTHING AND GO TO THE DESTINATION.

That being said, if you do go towards the destination, you'll have noted it in your
memory right? And then if immediately after you return to the same exact
intersection, it would suggest that what you thought was the destination was
incorrect, so well, consider that as well.
```

```
Another thing is, since you are exploring in a streetview environment, you move
through linked panoramas, sometimes you will notice that even when you go in a
certain direction you keep ending up in the same spot. it would mean that even if the
direction is correct, the panoramas are linked in a way that you keep coming back to
the same spot. Perhaps that direction is a dead end, hence you will have to make a
plan to go around it or something else.

Keep a good track of the intersections you have been to so that if you return to an
intersection you have been to, you don't make a stupid choice.
In your memory keep track of your movements, keep sense of your moves and use the
rough net movement to understand where you are and if you are back tracking. going X
direction 4 times then opposite of X 4 times likely means you are back. you can treat
the cardinal directions as roughly making a graph paper grid and then you can use the
net movement to understand where you are.
Remember the memory you create right now would be given to you verbatim at the next
intersection. So be careful how you phrase things. Please write the memory strictly
in past tense.
Remember, at every intersection the images are named image1, image2, image3 etc. so
don't put comments about the image lables in your memory as you will get confused. if
you need to do so, remember image x as intersection y. Since you do not have a visual
memory, it will be hard to identify what you have seen before., hence remember this.
The memory is of supreme importance so if you need it to be , make it 2000 or more
words.
Think step by step.

The json format is paramount. Do not deviate from it. no matter what since your
output wont be parsed otherwise.

[JSON schema and example omitted for brevity]
```

F.3 TEST PROMPT P4

```
You are at an intersection with 3 possible directions (options). Below are images
for each option. Analyze the images to determine the best direction towards Empire
State Building. Use the following information to guide your decision:

The images correspond to the following options/directions:
Option step0\_option0: facing North (22°)
Option step0\_option1: facing West (299°)
Option step0\_option2: facing East (119°)

Estimated position: This location is at the intersection of Cherry Street and
Rutgers Street, in the Lower East Side near the Two Bridges neighborhood of
Manhattan, New York City (evidence: The images show the residential towers of the
Rutgers Houses and the surrounding cityscape characteristic of this area.)

At every intersection, follow this rigorous sequence—do not skip, merge, or reorder
steps. You must enforce each rule as stated.

1. Absolute Immediate Exclusion: Under no circumstances may you select a direction
that (a) is marked as a dead end, (b) is the direction you just arrived from, or (c)
is flagged as looping/cycling (i.e., a path already revisited from this intersection
with no progress or returning here in a cycle)—unless and ONLY unless every other
remaining possibility is also categorically excluded. Exclude all taboo options
first, before any further reasoning.

2. Mandatory Novelty/Least-Traversed Prioritization: From the directions remaining
after exclusion, strictly prioritize the untried or least-recently-tried
direction(s) at this intersection. If several are tied as least-explored, you must
evaluate their images with the next step to break ties. If all are genuinely
equivalent, select randomly and document this tiebreak in your analysis and memory.
```

```
3. Compulsory Comparative Image Analysis: For each remaining candidate,
systematically analyze the corresponding images for direct evidence of advancement,
entrance to new territory, ongoing streets, signage, visible landmarks, or blockers.
Explicitly note any cues for or against progress, and only let clear, unambiguous
visual evidence override novelty prioritization. Never allow vague hope or
regional/directional bias to overrule exclusion or novelty unless the current image
provides categorical new information (e.g., unmistakable landmark, impassable
barrier, or prominent destination feature).

4. Explicit Tie-Breaking: In the rare event that two or more candidate directions
remain equally untried (or equally least-visited) and no image cues break the tie,
select randomly among them. State in your analysis/memory if random selection was
necessary.

5. Fallback for Deadlocks: If, after all exclusion and above prioritization, every
direction is either failed, cycled, or dead-ended, and no image offers new hope, you
may select the path least recently attempted as a last resort. Clearly specify this
as a 'deadlock fallback' move in both your analysis and your memory update. Update
your memory to avoid endless repetition: mark this intersection as in a deadlock
state and the selected fallback as attempted.

6. Explicit Ruling Out of All Non-Chosen Directions: For every available direction,
justify its exclusion or lower ranking—label whether it was omitted due to exclusion
rule, prior confirmed cycle, visual dead end, lack of promising cues, or previous no
progress. Write these reasons individually, not as generic groups.

7. Precise, Structured Memory Update Per Intersection: After each decision, update
your memory to explicitly record for this intersection:
    - The direction(s) now confirmed as dead ends or persistent cycles (list these as
    taboo/blocked);
    - The direction(s) explored but not confirmed dead—potentially still viable, to
    be remembered if fallback needed;
    - The untried or least-tried directions, prioritized for future steps;
    - Any new observations, visual cues, or notable changes from the current images,
    each attached to the relevant direction;
    - Whether random tiebreaking or fallback logic was applied for this decision.
Do NOT merely copy memory from previous steps—always recompose specifically for the
current intersection.

8. Exploration Rule for Recurring Cycles: If you have visited this intersection two
or more times without making forward progress, you MUST now prioritize any
still-untried or least-recently-tried direction, unless the images now conclusively
show it as unviable. In such case, state both the recurrence and your forced
prioritization (or the contraindicating cue) in your analysis and memory.

Always apply this entire sequence at each intersection. Do not allow any global,
habitual, or abstract destination direction to substitute for exclusion or local
intersection evidence. Your analysis and memory update must reflect this logic
point-by-point, per intersection, to maximize navigational accuracy and coverage,
rigorously avoid cycles and repetition, and systematically drive towards the
destination.

[JSON schema and example omitted for brevity]
```

F.4 TEST PROMPT P3

```
You are at an intersection with 3 possible directions (options). Below are images
for each option. Analyze the images to determine the best direction towards Empire
State Building. Use the following information to guide your decision:
```

```
The images correspond to the following options/directions:
Option step0\_option0: facing North (22°)
Option step0\_option1: facing West (299°)
Option step0\_option2: facing East (119°)

Estimated position: These images are taken from the area around the intersection of
Grand Street and Columbia Street, in the Lower East Side, Manhattan, New York City
(evidence: The views show the nearby residential high-rises which are part of the
Baruch Houses complex, along with Grand Street and Columbia Street street signs and
crossings visible.)

Before you answer, run through this checklist—do not skip any step:

1. VISUAL SCAN
   • Examine every image closely. Note landmark silhouettes, skyline cues,
   street/avenue signs or numbers, arrows, and road width.
   • If the destination itself or a sign pointing to it is visible in a photo, choose
   that option immediately.

2. ELIMINATE NON-STARTERS
   • Remove the option that matches the direction you just came FROM, unless all
   other paths are confirmed dead-ends.
   • Ignore options the system already flagged as dead-ends.

3. RE-ASSESS ORIENTATION
   • Using recent heading history and any street-number clues, re-estimate where the
   destination lies (N, S, E, W) relative to you *at this moment*—do NOT assume
   yesterday's best heading is still optimal.

4. ROAD PROMISE
   • Prefer routes that look longer, wider, busier, or keep a downtown skyline ahead.
   A major road that turns toward the destination is usually better than a minor
   side street that continues your old heading.

5. TIE-BREAKER ORDER (apply only if still uncertain)
   a) Photo with destination/sign
   b) Street numbers getting closer to the goal (e.g., in a numbered grid)
   c) Greater building density matching expected city centre
   d) Unexplored path over a previously visited one to avoid loops

6. OUTPUT FORMAT
   • analysis: Briefly cite the key visual cues, orientation reasoning, and why
   competing options were rejected.
   • decision: ONLY the option id (e.g., "step42\_option3").
   • memory: ONE concise sentence (<20 words) updating your high-level navigation
   belief (e.g., "Turning east toward landmark after north stint").

[JSON schema and example omitted for brevity]
```

G EXAMPLES

Figures 4–6 illustrate AgentNav's Verbalization of Path (VoP) mechanism in Tokyo, Vienna and Sao Paulo.

Figure 4: AgentNav navigating to Karlskirche in Vienna, Austria.

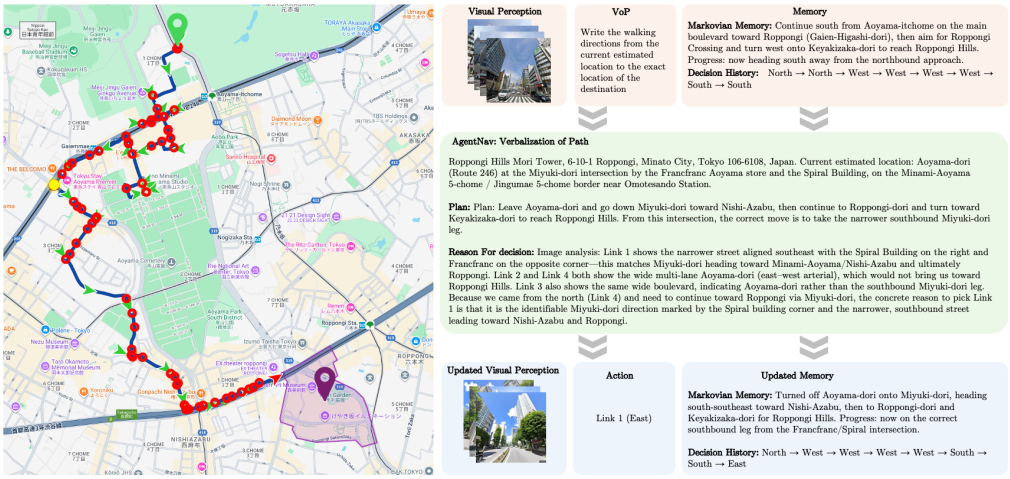
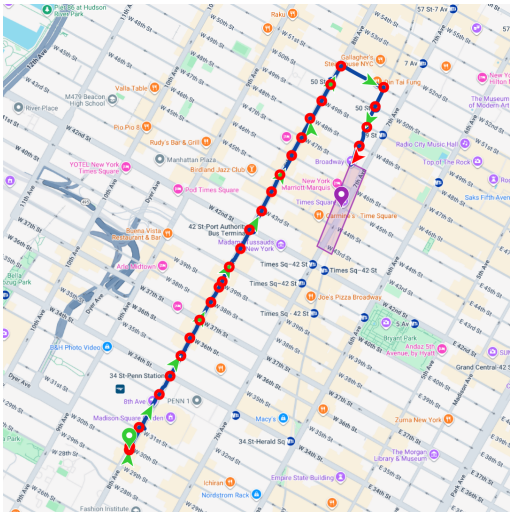
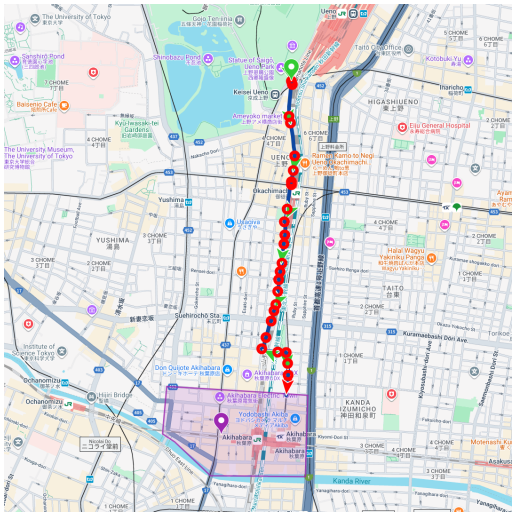


Figure 5: AgentNav navigating to Roppongi Hills in Tokyo, Japan.

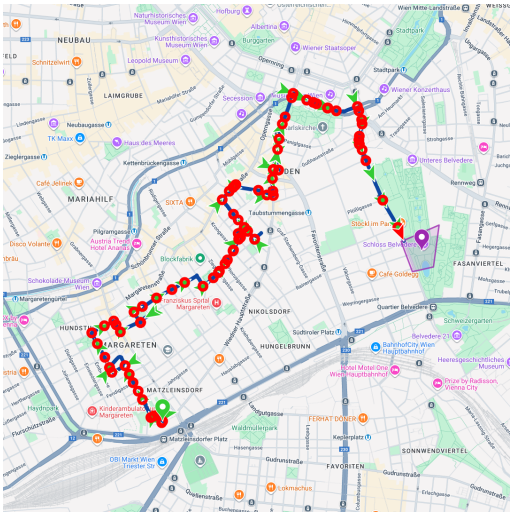
Figure 6: AgentNav navigating to Beco do Batman (Batman Alley) in Sao Paulo, Brazil.



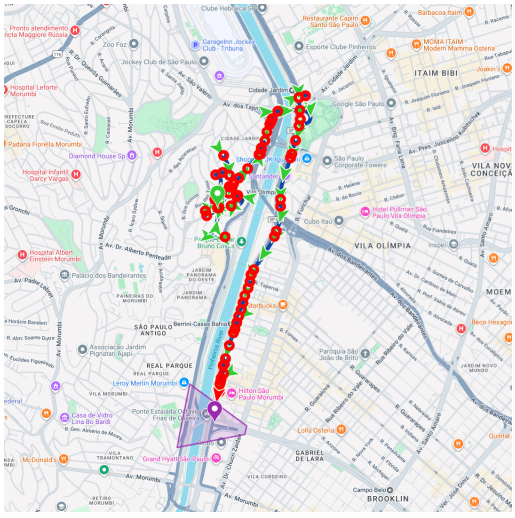
(a) New York, Destination: Times Square



(b) Tokyo, Destination: Akihabara

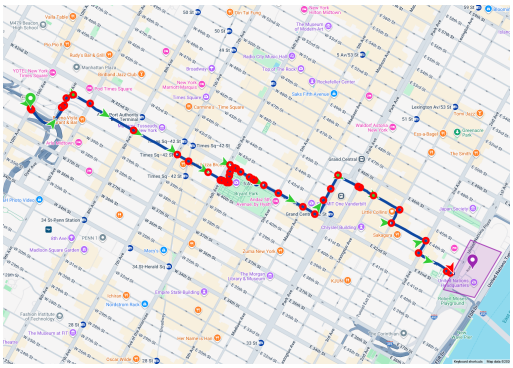


(c) Vienna, Destination: Belvedere Palace



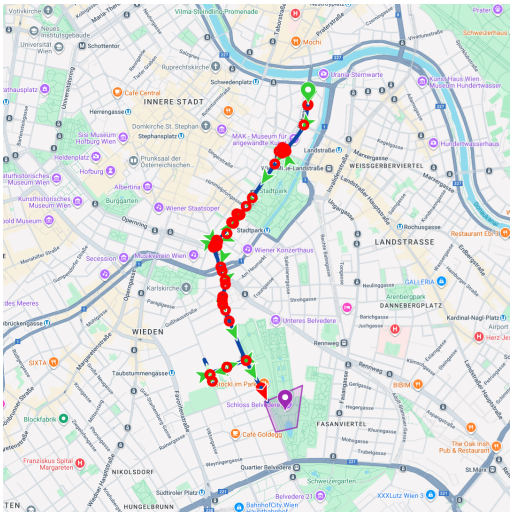
(d) Sao Paulo , Destination: Ponte Estaiada

Figure 7: Sample navigation paths (Set 1). Green markers indicate starting locations and purple polygons mark destination areas.

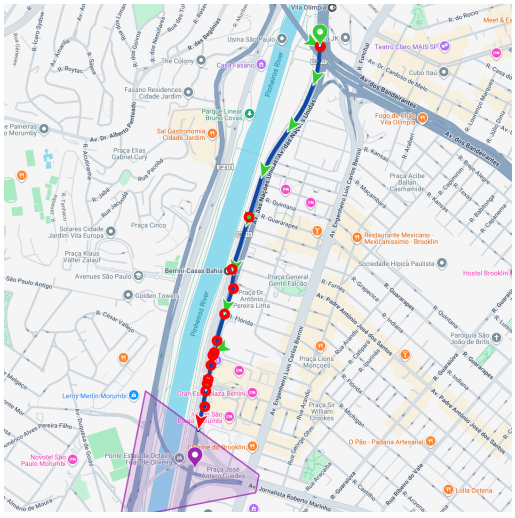


(a) New York, Destination: United Nations Headquarters

(b) Tokyo, Destination: Roppongi Hills



(c) Vienna, Destination: Belvedere Palace



(d) Sao Paulo, Destination:Ponte Estaiada

Figure 8: Sample navigation paths (Set 2). Green markers indicate starting locations and purple polygons mark destination areas.