FROM MAP TO STREET: A BENCHMARK FOR MULTI-SCALE GEOSPATIAL REASONING IN INSTRUCTION-DRIVEN GEOGRAPHIC LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction-driven geographic localization is a fundamental geospatial-reasoning task in which a model must fuse multiple sources of evidence to locate a real-world place described by natural-language instructions. Despite its practical importance, existing benchmarks capture only a narrow, static version of this task: they provide fixed inputs, allow little active exploration of geospatial space, and evaluate models primarily by final localization accuracy. To address this limitation, we present **IDOL**, a comprehensive benchmark for **I**nstruction-**D**riven **O**pen-world geographic **L**ocalization, which requires an agent to locate a place that a colloquial instruction describes by fusing multi-modal evidence across map, satellite, and street-level views. The benchmark covers eight cities and seven languages, and defines two task families: pinpointing a unique place from visual cues, and finding any place satisfying an open-ended predicate. For each family, we design a diverse suite of metrics that assess the reasoning process rather than the final answer alone. Experimental results show that while powerful models such as GPT-5.6-Sol and Gemini-3.1-Pro demonstrate great potential, they remain limited in fine-grained visual understanding and spatial grounding; moreover, their performance varies across cities and languages, exposing the imbalance in current models' reasoning capabilities. Our benchmark-generation and evaluation frameworks can be found at `https://anonymous.4open.science/r/IDOLBenchmark/`.

1 INTRODUCTION

Instruction-driven geographic localization aims to find one or more places that satisfy a user query that is vague about where the place is, yet precise about what it looks like. It is particularly valuable when users do not know a place's location but can describe it through visual cues, as commonly occurs in travel assistance, urban navigation, and local search (De Vries et al., 2018; Chen et al., 2019; Wu et al., 2026). Unlike image geographic localization, which maps a given image to coordinates (Vo et al., 2017; Pramanick et al., 2022; Xue et al., 2026), and retrieval-based localization, which confines the search to a known area and localizes a query image by retrieving from a database (Lee et al., 2022; Shao et al., 2023; Berton et al., 2025), this task requires a model to extract visual descriptions from vague instructions, fuse information across multiple modalities, and align visual features with natural language. In recent years, multi-modal large language models (MLLMs) with greatly improved cross-modal reasoning have emerged as a promising approach to this task (Jia et al., 2026; Xue et al., 2026).

Consider a typical request a traveler makes when asking for directions, "I am looking for that green steel bridge with colorful padlocks hanging all over both sides of the railings". A conventional method based on a large language model (LLM) handles this by extracting keywords from the query and retrieving a point of interest (POI) from the map as the answer (Yu et al., 2026; Wu et al., 2026; Dihan et al., 2025). While efficient, it cannot confirm whether a candidate place actually satisfies the visual constraints stated in the query, such as the green steel bridge and the colorful padlocks, nor can it compare how well multiple candidates match the query. Both limitations can reduce localization accuracy. To answer such geographic localization questions accurately, a model must integrate evidence across multiple modalities and verify visual details across multiple spatial scales.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Building upon these observations, we present **IDOL**, a comprehensive **I**nstruction-**D**riven **O**pen-world geo-**L**ocalization benchmark designed to evaluate the ability of MLLM-based agents to understand vague instructions and to fuse and align multi-modal information. IDOL covers eight cities with distinct visual appearances, including New York, Tokyo, and Johannesburg, as well as seven languages, including English, Chinese, and Zulu. Every localization question in IDOL specifies concrete visual features for the target place. Depending on whether a unique answer exists, questions fall into two categories: pinpointing a unique place from visual cues, or finding any place satisfying an open-ended predicate.

Motivated by the need for visual verification, we design a multi-scale geographic information retrieval and fusion framework driven by an MLLM-based agent. Specifically, given a colloquial instruction, the agent is equipped with three geospatial tools that query real-world geographic sources at different spatial scales, namely text-based keyword retrieval of candidate places, coarse-grained visual verification through overhead satellite imagery, and fine-grained visual verification through 360° street view. The agent autonomously decides which tool to call and when to stop, committing to one or more places that satisfy the instruction only after cross-checking the visual evidence.

For evaluation, we design distinct metrics for the two task families. For questions with a unique answer, we report comprehensive accuracy metrics together with a detailed failure analysis. For open-ended questions with multiple answers, we design a dedicated evaluation rubric and rely on an LLM-as-judge protocol (Zheng et al., 2023) to assess the answer along two dimensions: whether it satisfies the query's visual constraints and whether the reasoning process behind it is correct.

Experimental results show that while powerful MLLMs such as GPT-5.6-Sol and Gemini-3.1-Pro demonstrate great potential, reaching up to 64.6% S@100 and a median distance error down to 14 m on fixed-answer queries, they remain limited in fine-grained visual understanding and spatial grounding. Furthermore, we observe hallucination-like overconfidence in some models such as GPT-5.6-Luna, which directly output answers that appear semantically plausible yet do not match the actual scene, without invoking any visual tool. We additionally observe substantial cross-city variation and smaller but consistent cross-language variation, where models are less accurate in dense, visually complex cities such as Tokyo, and in low-resource languages such as Zulu, than in visually distinctive cities such as San Francisco and in dominant languages such as English. These findings establish IDOL as a challenging benchmark for future geographic localization systems.

In summary, our contributions are as follows:

- We introduce **IDOL**, an instruction-driven, multi-scale geographic localization benchmark that covers eight cities and seven languages. It defines two task families and every query specifies a visual constraint.
- We design a general multi-scale geographic information retrieval and fusion framework in which an MLLM-based agent autonomously composes three geospatial tools across different spatial scales to gather and cross-check visual evidence.
- Our extensive evaluation reveals that current MLLMs remain limited in fine-grained visual understanding and spatial grounding, and exhibit substantial cross-city and smaller but consistent cross-language performance variation, positioning IDOL as a rigorous benchmark for future research.

2 Related Work

2.1 Benchmarks for Geographic Localization

Geographic localization has been studied through a family of static benchmarks such as IM2GPS (Hays & Efros, 2008) and PlaNet (Weyand et al., 2016) that map a street-level image to coordinates, and cross-view benchmarks such as CVUSA (Workman et al., 2015), CVACT (Liu & Li, 2019) that match a ground-level image against an aerial view. These benchmarks share a static, outcome-centric paradigm in which a fixed input is mapped to a fixed output and evaluation reduces to coordinate distance or retrieval accuracy.

The rise of MLLMs has produced a new wave of benchmarks that examine geographic reasoning beyond raw coordinate prediction. GLOBE (Li et al., 2026) and GRE Suite (Wang et al., 2026)

Table 1: Comparison of IDOL with existing geo-localization benchmarks. **Instruction**: the query is a natural-language instruction rather than an image. **Visual constraints**: the query specifies visual features. **Multi-scale**: the evidence spans multiple spatial scales. **Open World**: the model operates in the open world with live external sources. ✓, ×, and △ denote yes, no, and partial, respectively.

Benchmark	Input	Instruction	Visual constraints	Multi-scale	Tool	Open World	Cities/Lang
GeoRC (Talreja et al., 2026)	Image	×	×	×	×	✓	Global/1
ERGeoBench (Xue et al., 2026)	Image	×	×	×	✓	×	Global/1
GeoPAVE (Jiang et al., 2026)	Image	×	×	△	✓	✓	Global/1
UrbanGround (Ju et al., 2026)	3D agent	×	×	✓	✓	×	1/1
Geovista (Wang et al., 2025)	Image	×	×	△	✓	✓	Global/1
EVGeoQA (Wu et al., 2026)	Text	✓	×	×	✓	✓	3/1
IDOL (ours)	**Text**	✓	✓	✓	✓	✓	**8/7**

reinforce image geographic localization with explicit reasoning chains, GeoRC (Talreja et al., 2026) evaluates geolocation reasoning chains, ERGeoBench (Xue et al., 2026) probes embodied reasoning and geographic localization in MLLMs. However, these benchmarks still treat the input as a single image or a fixed scene and none grounds the answer in multiple spatial scales.

A more recent line of work turns localization into an agentic, tool-driven task. GeoPAVE (Jiang et al., 2026) introduces a perceive-and-verify framework for open-world geographic localization with multi-round tool use, Geovista (Wang et al., 2025) augments geolocation with web retrieval, EVGeoQA (Wu et al., 2026) evaluates multi-objective geo-spatial exploration through a tool-augmented agent, and UrbanGround (Ju et al., 2026) places agents in a real-scale three-dimensional city. These works demonstrate the promise of agentic geo-localization, yet they are image-driven, scale-limited, and confined to a single city or language. Table 1 contrasts IDOL with these representative benchmarks.

2.2 Applications of MLLMs in Geospatial Reasoning

Driven by exceptional inductive reasoning and information-synthesis capabilities, MLLMs have been broadly applied across vision and language tasks (Alayrac et al., 2022; Li et al., 2025; Zhou et al., 2025; Wu et al., 2025). To handle the unique spatial constraints and heterogeneous data of geospatial reasoning, researchers have integrated MLLMs with established agent frameworks such as ReAct (Yao et al., 2023), Toolformer (Schick et al., 2023), and ToolLLM (Qin et al., 2024), empowering them to make decisions and interact with tools within geographic environments. Along this line, Spatial-RAG (Yu et al., 2026) resolves spatial reasoning questions through a dual-retrieval strategy, and CityGPT (Feng et al., 2025) injects street-network and urban functional-zone knowledge via instruction tuning.

While these works apply MLLMs to broad geo-spatial reasoning and urban understanding, a parallel line of work targets geographic localization specifically (Li et al., 2026; Wang et al., 2026; Jiang et al., 2026; Wang et al., 2025; Ji et al., 2026; Jin et al., 2026); these works underscore that reasoning and tool use improve geo-localization, yet they remain image-driven and do not structure the search across multi-level spatial scales. Our benchmark is built upon these advances and provides a specialized, instruction-driven, multi-scale adaptation for geographic localization.

3 IDOL Dataset Construction

3.1 Problem Formulation

Unlike image geographic localization that finds a place directly from a given image (Talreja et al., 2026; Jia et al., 2026), a query Q in IDOL is a colloquial natural-language instruction that describes a place through its visual features while omitting its name and exact location.

We divide this task into two types according to whether the query identifies a unique target: Type 1 queries require localizing a single visually distinctive place, whereas Type 2 queries require finding one or more places that satisfy an open-ended visual predicate.

The goal is to find a real-world place P that satisfies the visual constraint expressed in Q by fusing evidence across multiple geographic scales, including map, satellite image, and street view.

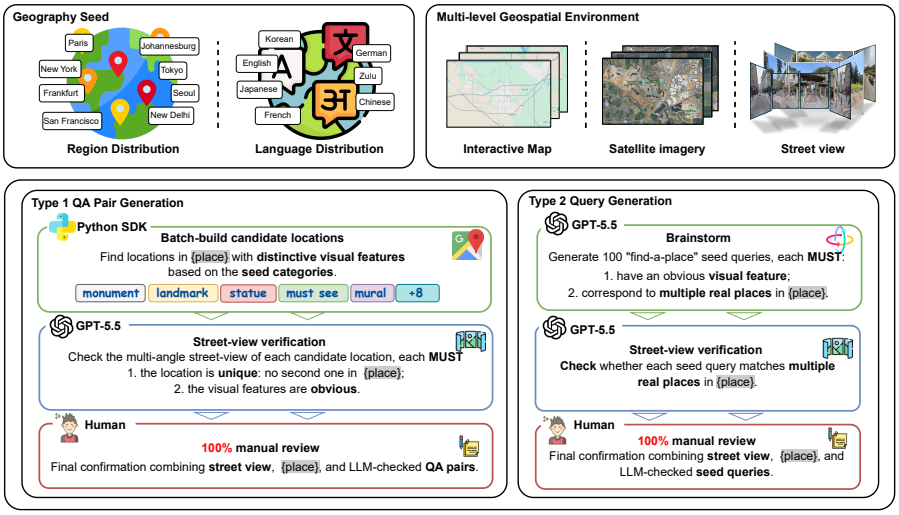

Figure 1: IDOL Dataset Construction Pipeline.

3.2 Data Acquisition and Overview

To ensure the comprehensiveness and diversity of the benchmark, IDOL covers eight cities with distinct visual appearances, namely Frankfurt, Johannesburg, New Delhi, New York, Paris, San Francisco, Seoul, and Tokyo, and seven languages, including English, Chinese, French, German, Japanese, Korean, and Zulu [1]. Table 2 summarizes the statistics of the benchmark.

All geographic data are obtained through the Google Maps Platform[2].

3.3 Query Generation

Type 1 queries: unique-place localization. As shown in Figure 1, we follow a pipeline of category keyword retrieval, MLLM-based automated verification, and human review. Since a Type 1 query must be answerable from visual appearance alone and point to a single unique place, we manually construct a fixed set of category keywords with prominent visual features and low repetition, such as "landmark", "must see" (full list in Appendix A.7), and retrieve a large set of candidate places matching these keywords from the Google Maps Platform.

For each candidate we fetch four directional street-view images at headings of 0, 90, 180, and 270 degrees, and prompt a powerful MLLM (GPT-5.5) to judge whether the scene contains distinctive and unique visual features. When the model deems a candidate to satisfy the requirement, it generates a one-sentence visual description and three colloquial queries that mention a rough location and visual cues. We automatically discard any query that leaks the place name and retain the coordinate and place name as the ground truth.

Table 2: Statistics of the IDOL dataset. Counts are per language, and every query is localized into seven languages, yielding 5,761 Type 1 queries and 1,365 Type 2 queries in total.

City	Type 1	Type 2	Total
Frankfurt	140	31	171
Johannesburg	83	16	99
New Delhi	85	17	102
New York	115	30	145
Paris	113	31	144
San Francisco	127	27	154
Seoul	80	18	98
Tokyo	80	25	105
Total	823	195	1,018

Type 2 queries: open-ended predicate search. Unlike a Type 1 query, which targets a single unique landmark, a Type 2 query asks for any place satisfying a visual requirement. Inspired by recent work on LLM-based task and idea generation (Wang et al., 2023; Lim & Perrault, 2024; Guo et al., 2025), we first prompt a language model (GPT-5.5) to iteratively brainstorm can-

[1] A local language spoken in Johannesburg, which is one of the eight cities in the IDOL benchmark.
[2] https://mapsplatform.google.com/

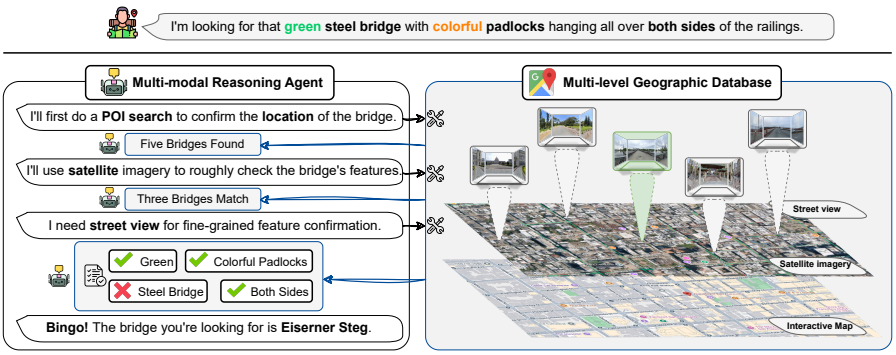

Figure 2: Overview of the IDOL evaluation framework.

didate predicates, each comprising a precise verification criterion and a colloquial query, under the constraint that the requirement is confirmable from street view and satisfied by multiple places in the city. To guarantee that every predicate admits multiple valid answers in a city, we enumerate candidates through keyword search on Google Maps Platform and verify each candidate by inspecting an overhead satellite view first, and, when that is inconclusive, the street-level views at five sampled nearby points. A predicate is retained only when at least three distinct places satisfy it. Notably, Type 2 queries in IDOL do not provide reference answer sets; instead, each query carries a description of the visual features shared by all valid places, which supports subsequent result analysis.

3.4 Quality Control and Human Review

Automatic query generation is efficient but not fully reliable. First, the generating MLLM can misread fine-grained visual details and hallucinate descriptions that do not match the scene. Second, a Type 1 query is intended to correspond to a single unique place in a city, yet the visual distinctiveness judged by the MLLM is subjective rather than an absolute guarantee, and exhaustively checking all street views in a city is impractical.

To address these issues, we subject every generated query to a full manual inspection, in which a reviewer examines each candidate together with its evidence images and checks that the visual description matches the actual scene and that the visual features are genuinely distinctive and unique. Notably, even for the rare Type 1 query that could plausibly match several similar places, the reviewer's joint confirmation reflects the inherent ambiguity of real-world queries, which is itself a property IDOL is designed to capture. In addition, our quantitative failure analysis in Section 5.5 confirms that such residual ambiguity accounts for only a small fraction of Type 1 failures, leaving the benchmark's quality unaffected. Overall, the combination of MLLM-based automatic verification and comprehensive human review filters out nearly 70% of the keyword retrieval queries, ensuring the high quality of the IDOL benchmark.

We provide more detail about query generation in Appendix A.5.

4 Multi-Scale Information Fusion and Evaluation Framework

4.1 Geospatial Toolset

We prioritize the evaluation of the agent's ability to gather and fuse evidence across different spatial scales, rather than simple information extraction. Accordingly, we design three geospatial tools:

- ***SearchPlaces*** performs keyword retrieval over the map and returns candidate places with their metadata including names, addresses, and coordinates.
- ***SatelliteView*** returns a satellite image of a given coordinate, revealing its overhead layout and surroundings.
- ***StreetView*** returns four directional street-view images of a given coordinate, revealing its fine-grained ground-level appearance.

As illustrated in Figure 2, the agent itself decides when and which tool to invoke and when to stop to commit an answer. The detailed parameter settings of each tool are provided in Appendix A.6.

4.2 Evaluation Metrics

The two task types are scored with different metrics. Type 1 queries (unique-place localization) are evaluated by spatial proximity to the ground-truth coordinate, whereas Type 2 queries (open-ended predicate search) are evaluated by visual evidence supporting the returned places.

Metrics for Type 1 queries. Let p_i^* denote the ground truth coordinate of query i, $\hat{p}_i$ the coordinate returned in the agent's final answer, and $d(\cdot,\cdot)$ the Haversine distance in meters. The primary metric is the success rate at distance threshold τ ($S@\tau$), and median distance error (MDErr):

$$S@\tau \;=\; \frac{1}{N}\sum_{i=1}^{N}[\![\, d(\hat{p}_i,\, p_i^*) \le \tau \,]\!], \qquad \tau \in \{25, 50, 100, 500\},$$

$$\text{MDErr} \;=\; \text{median}_{\,i\in\mathcal{A}}\; d(\hat{p}_i,\, p_i^*).$$

where N is the number of queries and $\mathcal{A}$ is the set of answered queries.

A success rate alone conflates retrieval failure, where the target never enters the candidate set, with verification failure, where the target is present but selected wrongly. To separate the two, we report the oracle success rate ($OSR@\tau$) and the oracle median distance error (OSR-MDErr), both computed over the candidate set $C_i \neq \varnothing$ accumulated across all *SearchPlaces* tool calls:

$$OSR@\tau \;=\; \frac{1}{N}\sum_{i=1}^{N}[\![\, \min_{c\in C_i} d(c,\, p_i^*) \le \tau \,]\!], \qquad \tau \in \{25, 50\},$$

$$\text{OSR-MDErr} \;=\; \text{median}_{\,i}\; \min_{c\in C_i} d(c,\, p_i^*).$$

Metrics for Type 2 queries. Since a Type 2 query admits multiple valid answers, we evaluate the agent's trajectory rather than a single predicted coordinate. Given a query q, the agent returns a set of places P_q, where each place $p \in P_q$ is located at $\mathrm{loc}(p)$. We deem p a valid answer if it is both *observed* and *satisfied*. A place is *observed*, denoted $o(p)$, if some street-view or satellite action in the trajectory retrieves imagery within 50 m of $\mathrm{loc}(p)$; it is *satisfied*, denoted $s(p)$, if a MLLM-based judge, given only the retrieved imagery, confirms that p meets the visual predicate of q. These two gates assess the presence and the correctness of visual evidence, respectively.

We report five metrics: the answer rate ($\mathrm{AnsRate}$) and average answer count (AvgAns) characterize answering behavior; the observation rate ($\mathrm{ObsRate}$) and success rate (Succ) measure the fraction of queries for which at least one returned place is observed and valid, respectively; and the verification accuracy (VerAcc) is the fraction of observed places confirmed as satisfied:

$$\mathrm{AnsRate} = \frac{1}{N}\sum_{q=1}^{N}[\![\, |P_q| \ge 1 \,]\!], \qquad \mathrm{AvgAns} = \frac{1}{|\mathcal{A}|}\sum_{q\in\mathcal{A}} |P_q|,$$

$$\mathrm{ObsRate} = \frac{1}{N}\sum_{q=1}^{N}[\![\, \exists\, p \in P_q : o(p) \,]\!], \qquad \mathrm{Succ} = \frac{1}{N}\sum_{q=1}^{N}[\![\, \exists\, p \in P_q : o(p) \wedge s(p) \,]\!],$$

$$\mathrm{VerAcc} = \frac{\big|\{p : o(p) \wedge s(p)\}\big|}{\big|\{p : o(p)\}\big|}.$$

where N is the number of queries and $\mathcal{A} = \{q : |P_q| \ge 1\}$ the set of answered queries.

5 Experiment

5.1 Selection of Evaluated MLLMs

Evaluated models. To comprehensively evaluate the geographic localization capabilities of MLLMs on the IDOL benchmark, we select a diverse set of MLLMs spanning different ven-

Table 3: Experimental results on Type 1 queries across different MLLMs on the IDOL dataset.

Model	S@25	S@50	S@100	S@500	MDErr (m)	OSR@25	OSR@50	OSR-MDErr (m)
Gemini-3.1-Pro	**50.9%**	**57.0%**	**64.6%**	**78.8%**	27	56.9%	63.4%	14
Gemini-3.5-Flash	46.8%	51.6%	58.0%	69.8%	**14**	**65.7%**	**71.1%**	6
MiniMax-M3	38.7%	43.5%	51.2%	63.6%	83	53.4%	58.6%	29
MiMo-V2.5	38.8%	42.4%	47.5%	60.9%	74	52.5%	57.2%	20
GLM-4.6V	31.2%	35.1%	41.2%	57.1%	264	40.9%	44.0%	109
GLM-5.3-Flash	43.2%	48.5%	56.9%	66.5%	38	52.3%	57.1%	25
Qwen3.5-27B	34.7%	38.2%	44.8%	60.0%	146	44.4%	49.8%	76
Qwen3.5-Plus	39.1%	44.0%	50.5%	63.5%	63	54.1%	57.0%	22
Qwen3.8-Max	45.5%	49.2%	55.9%	69.9%	97	52.0%	57.7%	48
GPT-5.6-Luna	35.0%	39.5%	48.1%	62.1%	57	53.8%	58.6%	9
GPT-5.6-Terra	38.9%	42.9%	50.4%	68.9%	73	54.7%	58.7%	16
GPT-5.6-Sol	47.5%	50.1%	60.2%	78.1%	44	64.6%	70.5%	**3**

Table 4: Experimental results on Type 2 queries across different MLLMs. H, G, Q, and M denote the human and the three MLLM judges, Gemini, Qwen, and MiMo, respectively.

Model	AnsRate	AvgAns	ObsRate	Succ(Q)	VerAcc(H/Q)	Succ Rank (G/Q/M)
MiniMax-M3	96.9	2.95	96.9	82.1	78.0/59.1	1/1/2
Gemini-3.5-Flash	89.2	2.47	89.2	76.9	72.2/66.0	2/2/1
MiMo-V2.5	97.4	2.33	94.9	74.4	69.8/59.1	3/3/4
Qwen3.5-Plus	99.0	2.35	91.8	71.8	68.4/57.6	4/4/3
Gemini-3.1-Pro	99.5	2.29	84.6	70.3	58.7/59.9	5/5/7
Qwen3.5-27B	100.0	2.02	97.9	68.7	67.3/54.9	6/6/5
GLM-4.6V	99.0	2.27	94.4	63.6	71.4/47.4	7/7/6
GPT-5.6-Sol	99.5	2.42	50.8	32.8	75.0/59.1	8/8/8
GPT-5.6-Terra	100.0	2.33	45.6	28.7	75.0/58.3	9/9/9
GPT-5.6-Luna	100.0	2.32	29.2	15.4	62.5/47.9	10/10/10

dors, parameter scales, and reasoning paradigms, including the Gemini series (Gemini-3.5-Flash and Gemini-3.1-Pro), the GPT series (GPT-5.6-Luna, GPT-5.6-Sol, and GPT-5.6-Terra), the Qwen series (Qwen3.5-27B, Qwen3.5-Plus, and Qwen3.8-Max), the GLM series (GLM-4.6V and GLM-5.3-Flash), MiniMax-M3, and MiMo-V2.5.

Judge model. The evaluation metrics for Type 2 queries rely on an MLLM-based judge. Rather than trust a single judge, we use three strong MLLMs from different vendors: Qwen3.8-Max, Gemini-3.1-Pro, and MiMo-V2.5. We additionally uniformly sample 600 observed places for human annotation to measure human-judge agreement.

The source of every model is listed in Appendix A.2.

5.2 Experimental Results and Analysis

We report localization results for Type 1 queries in Table 3 and process-oriented results for Type 2 queries in Table 4. All metrics are computed independently for each language and city, and then averaged. We highlight three findings below.

Fine-grained localization remains far from solved. Even the strongest model, Gemini-3.1-Pro, locates only 50.9% of Type 1 targets within 25 m and 64.6% within 100 m, and performance degrades sharply as the threshold tightens for every model. The gap between the coarse and fine thresholds indicates that current MLLMs can localize to the right neighborhood but struggle to pinpoint the exact place, which is precisely what Type 1 queries are designed to test.

Retrieval and verification are separable bottlenecks for Type 1 queries. The OSR metrics disentangle where localization fails. A large gap between $OSR@\tau$ and $S@\tau$ implies that the target was retrieved but not selected, isolating verification as the bottleneck, whereas a low $OSR@\tau$ implies a retrieval failure upstream. Gemini-3.5-Flash attains the highest oracle success rate (71.1% at 50 m) but converts only 51.6% of it into correct answers. GPT-5.6-Sol exhibits accurate retrieval, with an oracle median error of 3 m, yet a comparable success-rate gap, again attributing its errors to selection rather than search. In contrast, Gemini-3.1-Pro narrows this gap to roughly 6 points, indicating that its advantage stems from reliable verification once candidates are in hand.

Process-oriented evaluation for Type 2 queries exposes shortcut answering. For the majority of models, Type 2 success is tightly coupled with observation. Models that consistently ground their

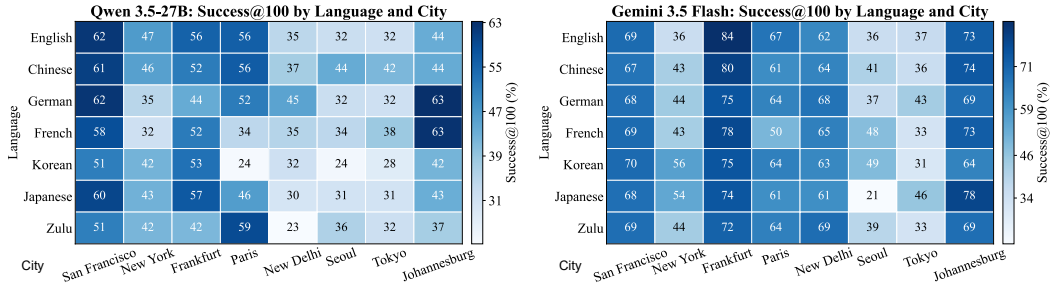

Figure 3: Regional and language differences in S@100 for Qwen3.5-27B and Gemini-3.5-Flash.

answers in retrieved imagery, such as MiniMax-M3 and MiMo-V2.5, also attain the highest success rates and occupy the top ranks under all three judges, with MiniMax-M3 leading at 82.1%. Across these models the two rates move together, confirming that inspecting a candidate before committing to it is the decisive step in solving a Type 2 query.

The GPT-5.6 family departs sharply from this pattern: although these models answer almost every query and return comparable numbers of candidates, they have substantially lower observation rates than the other evaluated models. Such low observation rates raise the concern that these models bypass the visual tools and answer directly from parametric knowledge. We further re-verify the GPT-5.6 family's answers in Section 5.6 to check whether they actually satisfy the query's requirements.

We also conduct a comprehensive ablation study and tool-call-count analysis of the three framework tools on Type 1 queries. The results show that all three tools, especially the *SearchPlaces*, contribute to higher accuracy, and that both *SearchPlaces* and *SatelliteView* improve answering efficiency by reducing the number of tool calls. We attribute this to the two tools enabling the model to coarsely screen candidates early: *SearchPlaces* narrows the candidate pool while *SatelliteView* tends to filter out clearly mismatched places. The full analysis is reported in Appendix A.1.

5.3 Reliability of the MLLM-based Judge for Type 2 Query Evaluation

To calibrate model's judgment errors on Type 2 queries, we uniformly draw 600 samples of observed places and have human annotators label each one under a blind protocol. Beyond raw agreement (Agr.), we report Cohen's κ, the **Miss** rate (judge false-rejects) and the **F**alse-**P**ass rate (judge false-accepts), both conditional on the human label and reflecting bias direction rather than overall error.

Table 5: Human vs judge agreement.

Judge	Agr.	κ	Miss	FP
Qwen3.8-Max	78.7	0.526	22.3	18.8
Gemini-3.1-Pro	74.3	0.430	25.7	25.9
MiMo-V2.5	66.1	0.338	40.8	16.1

As shown in Table 5, Qwen3.8-Max attains the highest agreement (78.7%) and κ (0.526). We therefore adopt Qwen3.8-Max as the primary judge for all Type 2 results shown in Table 4. The consistent ranking of the evaluated models across the three judges, reported in Table 4, further indicates that the metrics for Type 2 queries are robust to the choice of judge. Appendix A.3 presents representative cases and a finer-grained analysis.

5.4 Cross-City and Cross-Language Performance Variation

We evaluate Qwen3.5-27B and Gemini-3.5-Flash across the eight cities and seven languages and find that both region and language affect accuracy, yet the regional difference is more significant.

As shown in Figure 3, averaging over models and languages, performance spans a roughly 28-point range, from about 63% in Frankfurt down to 35% in Tokyo, with the dense, visually crowded metropolises Tokyo, Seoul, and New York proving hardest and the more visually distinctive Frankfurt, San Francisco, and Johannesburg easiest. The language effect is smaller, spanning only about 5 points from 52.9% in Chinese to 48.0% in Korean, strongest in dominant languages such as Chinese and English and weakest in low-resource languages such as Korean and Zulu.

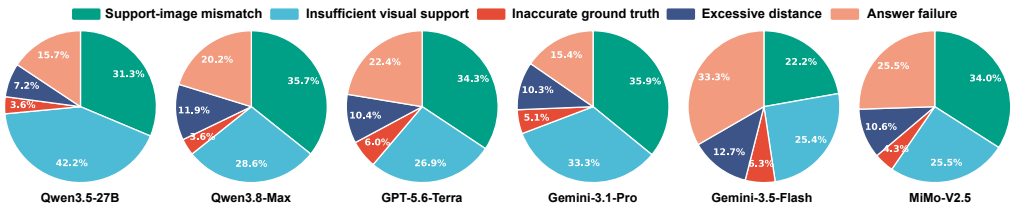

Figure 4: Distribution of Type 1 failure causes for six representative models.

5.5 Failure Analysis for Type 1 Queries

For six representative models (Qwen3.5-27B, Qwen3.8-Max, GPT-5.6-Terra, Gemini-3.1-Pro, Gemini-3.5-Flash, MiMo-V2.5), we randomly sample about 70 query-answer pairs that each model fails under $S@100$ and, through blind human review, classify each into one of five causes:

- **Support-image mismatch**: the retrieved scene contradicts the query's visual constraint.
- **Insufficient visual support**: the retrieved scene matches the visual constraint partially.
- **Inaccurate ground truth**: human review deems the answer correct, but it is scored as a failure because the ground truth is not exhaustive.
- **Excessive distance**: a matching place appears in the imagery, but it lies too far away.
- **Answer failure**: no valid answer is produced.

Fine-grained visual understanding is the dominant bottleneck. As shown in Figure 4, averaged over the six models, support-image mismatch and insufficient visual support together cause about 63% of failures, far outweighing others. Both errors trace to the same weakness: the model cannot resolve fine-grained visual differences in the imagery and therefore misjudges the vague visual constraints stated in the query, echoing the retrieval-verification gap in Section 5.2.

Query quality is high. The only data-side cause, inaccurate ground truth, is uniformly small across all six models (3.6%–6.3%). It reflects the inherent ambiguity of real-world instructions, where occasionally more than one place in a city fits a query. As noted in Section 3, such ambiguity cannot be fully eliminated but is far too rare to affect the benchmark's conclusions or quality. We provide representative examples of each failure cause in Appendix A.4.

5.6 Analysis of Ungrounded Type 2 Answers from the GPT-5.6 Family

We additionally extract every answer coordinate returned by the GPT-5.6 family without prior image observation and conduct a follow-up evaluation using Qwen3.8-Max as the judge. Unlike the evaluation protocol mentioned in Section 4.2, here we re-fetch complete evidence at each coordinate including four street-view headings plus one overhead satellite view.

Table 6: Re-verification of the GPT-5.6 family.

Model	Valid rate
GPT-5.6-Sol	62.5%
GPT-5.6-Terra	58.8%
GPT-5.6-Luna	54.7%

As shown in Table 6, even the GPT-5.6-Sol has only 62.5% of its ungrounded answers confirmed to satisfy the visual predicate. The hallucination rate reaches nearly half for GPT-5.6-Luna. These results indicate that the multimodal hallucination of even the strongest MLLMs remains pronounced, and that grounded visual verification is still indispensable.

6 Conclusion

We presented **IDOL**, a benchmark for instruction-driven, multi-scale geographic localization in which an MLLM-based agent must fuse evidence across map, satellite, and street-level views to locate a place from a colloquial description. Covering eight cities and seven languages, IDOL defines two task families and evaluates the reasoning process rather than the final answer alone.

By exposing key bottlenecks such as fine-grained localization, separable retrieval and verification failures, and regional and language gaps, IDOL serves as a rigorous testbed to guide the development of agents that can robustly fuse multi-scale evidence to localize places in the open world.

AI USE STATEMENT

We disclose the role of generative AI in three stages of this work:

Benchmark construction. The queries in IDOL are semi-automatically generated with the assistance of multimodal and language models, which retrieve and screen candidate places and draft the natural-language queries described in Section 3. To ensure data quality, every generated query and its visual evidence undergo full manual review under a documented protocol; the complete procedure is detailed in Section 3 and Appendix A.5.

Code. Part of the benchmark-generation and evaluation code is written with the assistance of generative AI; all generated code is manually checked for correctness before use.

Manuscript. We use generative AI to polish the language and check the grammar and spelling of this manuscript. All authors review the final text and take full responsibility for its content.

ETHICS STATEMENT

Privacy. All data in IDOL come from the publicly available Google Maps Platform and contain no personally identifiable information, and all platform data are used in compliance with the Google Maps Platform terms of service.

Manual review and annotation. All manual review and annotation in this work are performed by the authors themselves. No external annotators are employed and no crowd-sourcing platform is used.

REPRODUCIBILITY STATEMENT

We release the benchmark-construction and evaluation frameworks at `https://anonymous.4open.science/r/IDOLBenchmark/`. The dataset statistics are described in Section 3, the query-generation pipeline and human-review protocol in Section 3 and Appendix A.5, the geospatial tools and their parameter settings in Appendix A.6, the evaluation metrics in Section 4.2, and the sources of all evaluated models in Appendix A.2. We also provide the prompts used in the framework in Appendix A.8.

Access to Google Maps Platform data. The Google Maps Platform terms of use do not permit us to publicly redistribute the place information and imagery retrieved from it, so we do not release the raw images or place metadata directly. Instead, the code repository provides a complete specification of the tool calls together with the cached platform identifiers recorded for every API call issued during our experiments to help researchers retrieve the same return values used in our experiments and reproduce our results.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Gabriele Berton, Alex Stoken, and Carlo Masone. Astroloc: Robust space to ground image localizer. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5811–5820. IEEE, 2025.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12530–12539. IEEE, 2019.

Harm De Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367*, 2018.

Mahir Labib Dihan, Md Tanvir Hassan, Md Tanvir Parvez, Md Hasebul Hasan, Md Almash Alam, Muhammad Aamir Cheema, Mohammed Eunus Ali, and Md Rizwan Parvez. Mapeval: A map-based evaluation of geo-spatial reasoning in foundation models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13774–13813. PMLR, 13–19 Jul 2025.

Jie Feng, Tianhui Liu, Yuwei Du, Siqi Guo, Yuming Lin, and Yong Li. Citygpt: Empowering urban spatial cognition of large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 591–602, 2025.

Sikun Guo, Amir Hassan Shariatmadari, Guangzhi Xiong, Albert Huang, Myles Kim, Corey M. Williams, Stefan Bekiranov, and Aidong Zhang. Ideabench: Benchmarking large language models for research idea generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 5888–5899, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737419.

James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *2008 ieee conference on computer vision and pattern recognition*, pp. 1–8. IEEE, 2008.

Yuxiang Ji, Yong Wang, Ziyu Ma, Yiming Hu, Hailang Huang, Xuecai Hu, Guanhua Chen, Liaoni Wu, and Xiangxiang Chu. Thinking with map: Reinforced parallel map-augmented agent for geolocalization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17991–18004, 2026.

Pengyue Jia, Yingyi Zhang, Xiangyu Zhao, and Sharon Li. Geoarena: Evaluating open-world geographic reasoning in large vision-language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20886–20901, 2026.

Yutian Jiang, Ruijie Li, Sisuo Lyu, Xixuan Hao, Qingxiang Liu, Yongzi Yu, and Yuxuan Liang. Perceive to hypothesize, verify to ground: An agentic reasoning framework for open-world geolocalization. *arXiv preprint arXiv:2608.29880*, 2026.

Modi Jin, Yiming Zhang, Boyuan Sun, Dingwen Zhang, Ming-Ming Cheng, and Qibin Hou. Geoagent: Learning to geolocate everywhere with reinforced geographic characteristics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41352–41364, 2026.

Tianjie Ju, Zheng Wu, Yueqing Sun, Yuhan Cui, Bobo Li, Shengqiong Wu, Pengzhou Cheng, Haodong Zhao, Zongru Wu, Xinbei Ma, et al. Urbanground: From local perception to spatial agency in a real-scale city. *arXiv preprint arXiv:2608.27456*, 2026.

Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. Correlation verification for image retrieval. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5364–5374. IEEE, 2022.

Ling Li, Yao Zhou, Yuxuan Liang, Fugee Tsung, and Jiaheng Wei. Recognition through reasoning: Reinforcing image geo-localization with large vision-language models. *Advances in Neural Information Processing Systems*, 38:62132–62159, 2026.

Zongxia Li, Xiyang Wu, Hongyang Du, Fuxiao Liu, Huy Nghiem, and Guangyao Shi. A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1578–1597. IEEE, 2025.

Gionnieve Lim and Simon T Perrault. Rapid aideation: Generating ideas with the self and in collaboration with large language models. *arXiv preprint arXiv:2403.12928*, 2024.

Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5617–5626. IEEE, 2019.

Shraman Pramanick, Ewa M Nowara, Joshua Gleason, Carlos D Castillo, and Rama Chellappa. Where in the world is this image? transformer-based geo-localization in the wild. In *European Conference on Computer Vision*, pp. 196–215. Springer, 2022.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations*, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Shihao Shao, Kaifeng Chen, Arjun Karpur, Qinghua Cui, André Araujo, and Bingyi Cao. Global features are all you need for image retrieval and reranking. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11002–11012. IEEE, 2023.

Mohit Talreja, Joshua Diao, Jim James, Radu Casapu, Tejas Santanam, Ethan Mendes, Alan Ritter, Wei Xu, and James Hays. Georc: A benchmark for geolocation reasoning chains. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40540–40564, 2026.

Nam Vo, Nathan Jacobs, and James Hays. Revisiting im2gps in the deep learning era. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2640–2649. IEEE, 2017.

Chun Wang, Xiaojun Ye, Xiaoran Pan, Zihao Pan, Haofan Wang, and Yiren Song. Gre suite: Geo-localization inference via fine-tuned vision-language models and enhanced reasoning chains. *Advances in Neural Information Processing Systems*, 38:92379–92409, 2026.

Yikun Wang, Zuyan Liu, Ziyi Wang, Han Hu, Pengfei Liu, and Yongming Rao. Geovista: Web-augmented agentic visual reasoning for geolocalization. *arXiv preprint arXiv:2511.15705*, 2025.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European conference on computer vision*, pp. 37–55. Springer, 2016.

Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *Proceedings of the IEEE international conference on computer vision*, pp. 3961–3969, 2015.

Jianfei Wu, Xubin Wang, and Weijia Jia. Enhancing text annotation through rationale-driven collaborative few-shot prompting. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10889661.

Jianfei Wu, Zhichun Wang, Zhensheng Wang, and Zhiyu He. Evgeoqa: Benchmarking llms on dynamic, multi-objective geo-spatial exploration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3809–3821, 2026.

Kaiwen Xue, Tao Wei, Zhonghong Ou, Guoxin Zhang, Kaoyan Lu, Yu Feng, Yifan Zhu, and Haoran Luo. Ergeobench: A comprehensive benchmark for embodied reasoning and geo-localization in multimodal large language models. In *Proceedings of the International Conference on Machine Learning*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2023.

Dazhou Yu, Riyang Bao, Ruiyu Ning, Jinghong Peng, Gengchen Mai, and Liang Zhao. Spatial-rag: Spatial retrieval augmented generation for real-world geospatial reasoning questions. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11094–11112, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Chenyue Zhou, Mingxuan Wang, Yanbiao Ma, Chenxu Wu, Wanyi Chen, Zhe Qian, Xinyu Liu, Yiwei Zhang, Junhao Wang, Hengbo Xu, et al. From perception to cognition: A survey of vision-language interactive reasoning in multimodal large language models. *arXiv preprint arXiv:2509.25373*, 2025.

A Appendix

A.1 Ablation Study and Tool-call-count Analysis

We study the contribution of each tool in our framework by ablating one at a time on Type 1 queries. The full framework, in which all three tools (*SearchPlaces*, *StreetView*, *SatelliteView*) are available, is compared against four ablated variants: *w/o SearchPlaces*, *w/o StreetView*, and *w/o SatelliteView*, in which the corresponding tool is removed while the other two remain available, and *w/o all tools*, in which no tool is available and the model answers directly from its parametric knowledge.

Metrics. We report the Success rate at multiple distance thresholds $\tau \in \{25, 50, 100, 500\}$ m (S@τ) and the median distance error (MDErr) over answered queries. Furthermore, we evaluate *efficiency* with the average number of calls to each tool (avg_search, avg_streetview, avg_satellite) and the total number of tool calls per query (sum_toolcall). All metrics in this section are computed independently for each language and city, and then averaged.

Table 7: Ablation Study and Tool-call-count Analysis on Type 1 queries across different MLLMs.

Model	Settings	S@25	S@50	S@100	S@500	MDErr (m)	avg_search	avg_streetview	avg_satellite	sum_toolcall
MiniMax-M3	*w/o all tools*	8%	13%	21%	53%	799	-	-	-	-
	w/o SearchPlaces	8%	13%	25%	42%	5,023	-	6.9	7.6	14.5
	w/o SatelliteView	28%	33%	43%	50%	582	5.6	6.2	-	11.7
	w/o StreetView	28%	33%	40%	58%	490	5.2	-	2.5	7.7
	full	**39%**	**44%**	**51%**	**64%**	**83**	4.7	2.6	2.1	9.3
Gemini-3.5-Flash	*w/o all tools*	19%	24%	40%	64%	200	-	-	-	-
	w/o SearchPlaces	18%	20%	29%	53%	243	-	11.6	3.1	14.7
	w/o SatelliteView	35%	40%	48%	58%	185	8.9	3.2	-	12.1
	w/o StreetView	38%	43%	48%	58%	133	11.8	-	2.3	14.0
	full	**47%**	**52%**	**59%**	**69%**	**14**	7.4	3.5	0.4	11.3
MiMo-V2.5	*w/o all tools*	6%	8%	14%	39%	897	-	-	-	-
	w/o SearchPlaces	5%	13%	20%	28%	1,012	-	8.5	6.7	15.2
	w/o SatelliteView	30%	30%	38%	50%	432	5.2	2.9	-	8.1
	w/o StreetView	18%	18%	25%	43%	623	3.3	-	2.0	5.2
	full	**39%**	**42%**	**48%**	**61%**	**74**	3.5	2.1	0.8	6.4
Qwen3.5-27B	*w/o all tools*	8%	11%	14%	36%	1,163	-	-	-	-
	w/o SearchPlaces	6%	9%	14%	33%	853	-	11.3	3.9	15.2
	w/o SatelliteView	33%	35%	**45%**	53%	578	5.2	2.0	-	7.1
	w/o StreetView	30%	33%	35%	47%	434	2.7	-	2.2	4.9
	full	**35%**	**38%**	45%	**60%**	**146**	4.3	2.4	0.7	7.5
Qwen3.5-Plus	*w/o all tools*	8%	11%	19%	49%	677	-	-	-	-
	w/o SearchPlaces	0%	6%	13%	31%	1,881	-	13.1	3.6	16.8
	w/o SatelliteView	38%	43%	**53%**	63%	120	6.7	3.0	-	9.7
	w/o StreetView	30%	35%	43%	65%	254	7.4	-	2.7	10.1
	full	**39%**	**44%**	51%	**64%**	**63**	4.8	2.7	1.3	8.8
Qwen3.8-Max	*w/o all tools*	9%	11%	19%	50%	494	-	-	-	-
	w/o SearchPlaces	3%	7%	22%	35%	645	-	5.4	2.5	8.0
	w/o SatelliteView	45%	47%	**60%**	65%	176	3.1	4.4	-	7.5
	w/o StreetView	30%	35%	48%	63%	253	3.0	-	1.8	4.8
	full	**46%**	**49%**	56%	**70%**	**97**	2.5	2.5	0.8	5.8

Table 7 reports the ablation results and tool-use statistics. We draw four main observations.

***SearchPlaces* is critical to both localization accuracy and tool-use efficiency.** Removing *SearchPlaces* produces the largest and most consistent degradation across all six models. Relative to the full framework, S@100 decreases by 26–38 percentage points, while MDErr increases from 14–146 m to 243–5,023 m. This result highlights the importance of explicit place retrieval for grounding predictions in plausible geographic candidates. In its absence, the models must rely more heavily on their parametric knowledge to generate candidate locations, which often leads to substantial localization errors. The removal also increases the total number of tool calls for every model, from 5.8–11.3 to 8.0–16.8 calls per query, indicating that, without reliable candidate coordinates, the models resort to more frequent but poorly directed trial-and-error exploration.

***StreetView* provides an important ground-level verification signal.** Removing *StreetView* reduces S@100 by 8–23 percentage points across models and consistently increases MDErr. For example, MDErr increases from 83 to 490 m for MiniMax-M3 and from 14 to 133 m for Gemini-3.5-Flash. These results indicate that place names, addresses, and overhead imagery alone are often insufficient to distinguish geographically plausible candidates. Ground-level observations provide complementary visual evidence that helps verify local appearance and reject incorrect candidates. Among the two visual tools, *StreetView* is also used more frequently in the full framework, with 2.1–3.5 calls per query, compared with 0.4–2.1 calls to *SatelliteView*.

***SatelliteView* primarily improves fine-grained localization precision.** The effect of removing *SatelliteView* on S@100 is model-dependent: S@100 decreases by 8–11 percentage points for the first three models, remains unchanged for Qwen3.5-27B, and increases slightly for Qwen3.5-Plus and Qwen3.8-Max. Nevertheless, MDErr increases consistently across all models, from 14–146 m in the full setting to 120–582 m without *SatelliteView*. This pattern suggests that aerial imagery is particularly useful for refining predictions that are already near the target, even when it does not consistently improve the binary success rate at the 100 m threshold. Moreover, removing *SatelliteView* increases the average number of *StreetView* calls for four of the six models, indicating that overhead imagery can sometimes eliminate mismatched candidates before ground-level verification.

Overall findings. The three tools therefore provide complementary forms of evidence. *SearchPlaces* grounds the search in explicit candidate locations and yields the largest gains in both accuracy and efficiency. *StreetView* supports ground-level visual verification, whereas *SatelliteView* mainly improves fine-grained spatial precision and can reduce the need for repeated street-level inspection. When all tools are removed, S@100 falls to 14–40% and MDErr increases to 200–1,163 m, demonstrating the substantial limitations of relying exclusively on the models' parametric knowledge for precise geolocation.

A.2 Model Sources

Since the release format of the MLLM is inconsistent with the standard bibliography style, here we list the vendor, URL, and technical report (where available) of every model used in our experiments.

- **Gemini-3.5-Flash** (Google). `https://deepmind.google/models/model-cards/gemini-3-5-flash/`
- **Gemini-3.1-Pro** (Google). `https://deepmind.google/models/gemini/pro/`
- **GPT-5.6-Luna** (OpenAI). `https://openai.com/index/gpt-5-6/`
- **GPT-5.6-Sol** (OpenAI). `https://openai.com/index/gpt-5-6/`
- **GPT-5.6-Terra** (OpenAI). `https://openai.com/index/gpt-5-6/`
- **Qwen3.5-27B** (Alibaba). `https://huggingface.co/Qwen/Qwen3.5-27B` (Qwen Team, 2026a)
- **Qwen3.5-Plus** (Alibaba). `https://qwen.ai/blog?id=qwen3.5` (Qwen Team, 2026a)
- **Qwen3.8-Max** (Alibaba). `https://qwen.ai/blog?id=qwen3.8` (Qwen Team, 2026b)
- **GLM-4.6V** (Zhipu AI). `https://z.ai/blog/glm-4.6`

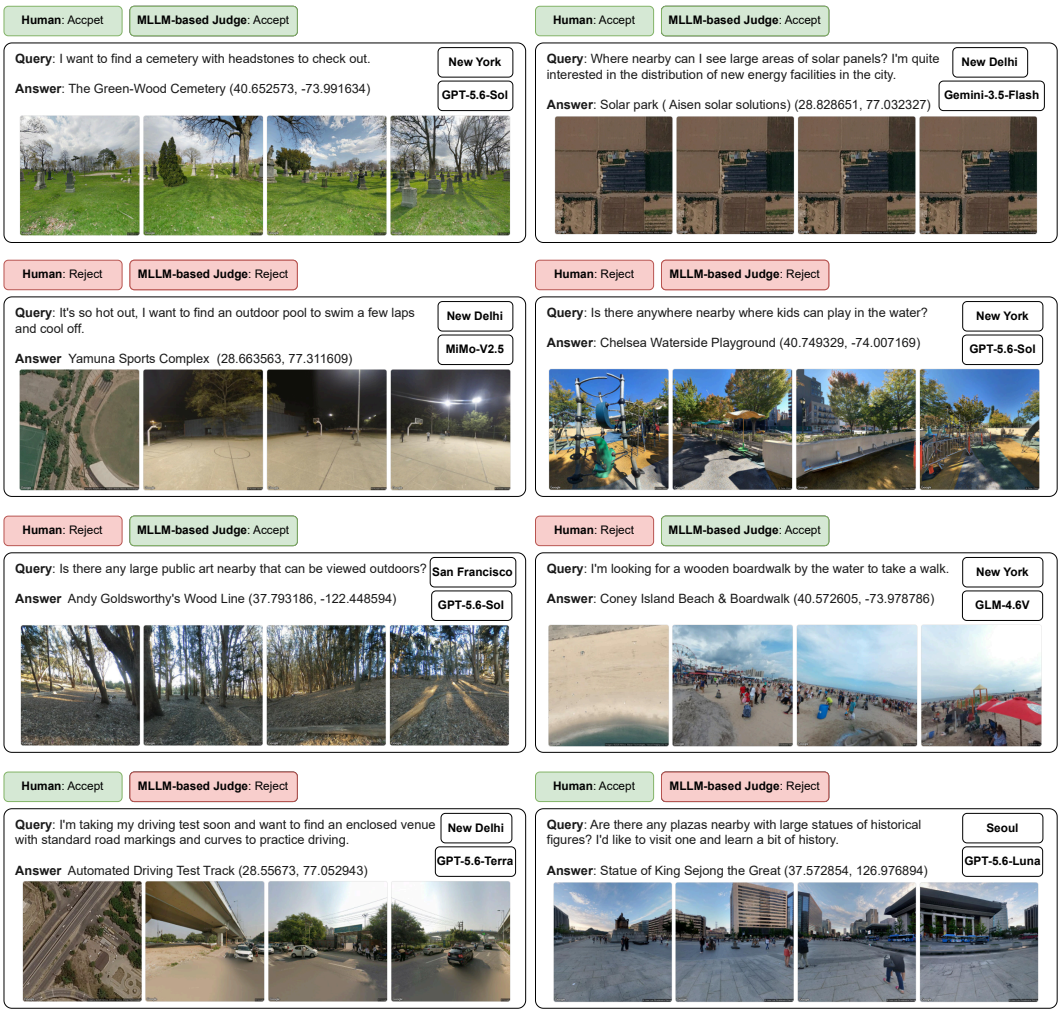

Figure 5: Representative cases of agreement and disagreement between human annotators and the Qwen3.8-Max judge.

- **GLM-5.3-Flash** (Zhipu AI). `https://z.ai/blog/glm-5.3-flash`
- **MiniMax-M3** (MiniMax). `https://www.minimax.io/blog/minimax-m3`
- **MiMo-V2.5** (Xiaomi). `https://mimo.xiaomi.com/mimo-v2-5`

A.3 MLLM-BASED JUDGE ANALYSIS

We manually inspect the disagreement cases from the human-labeled places described in Section 5.3. As shown in Figure 5, we group them by the pair of verdicts into four categories, both accept, both reject, false pass (the judge accepts a place the human rejects), and miss (the judge rejects a place the human accepts), and focus the analysis on the two disagreement categories, which reveal two recurring causes of judge error.

Insufficient fine-grained image understanding. The judge frequently overlooks the subtle visual cues a predicate describes, especially when they sit at the periphery of the retrieved imagery. For instance, in the bottom-left example of Figure 5, the judge failed to recognize the lane and curve markings of a driving-test ground and reported that no road markings were visible. This produces the miss errors in Table 5.

Function-induced hallucination. The MLLM-based judge infers a candidate's function from the imagery and once it has formed an expectation of what the place is, tends to hallucinate that the

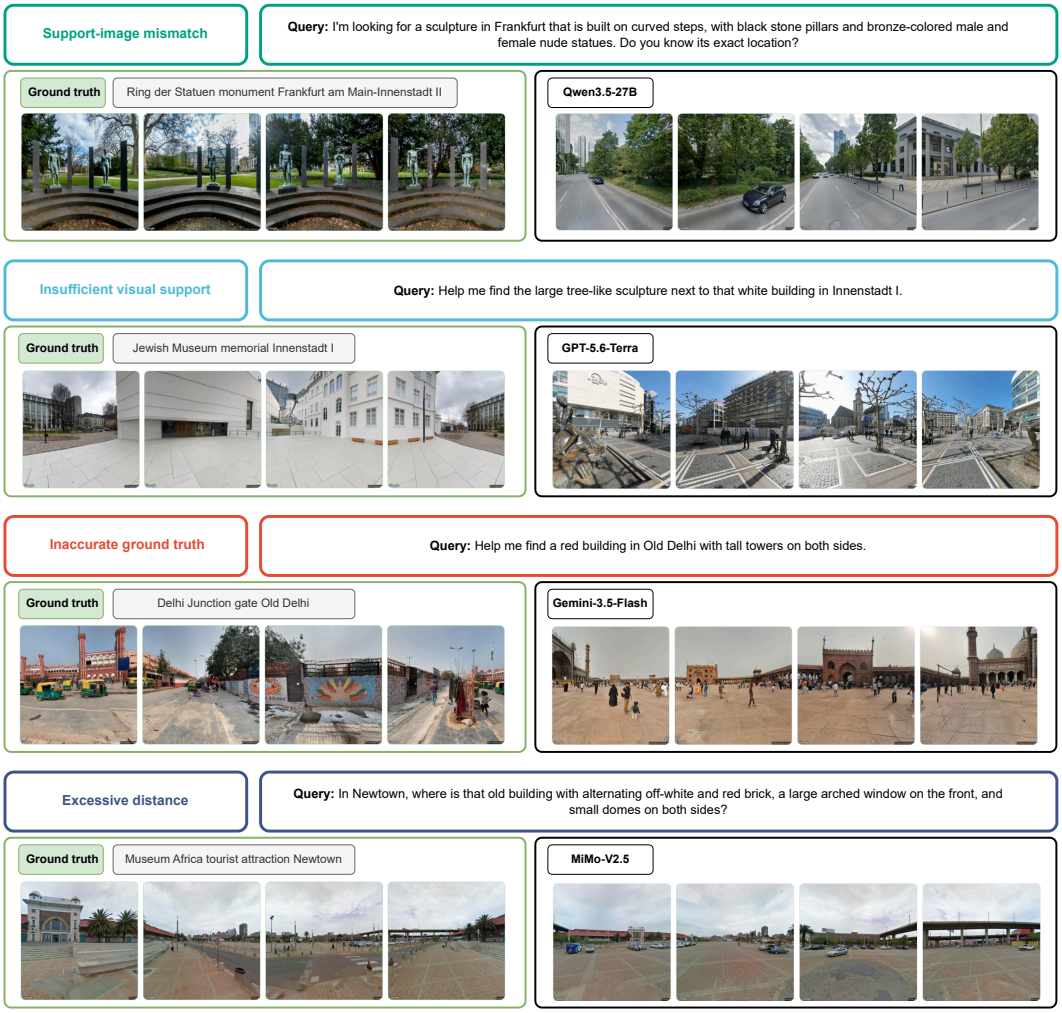

Figure 6: Representative cases of the four visual Type 1 failure causes.

place satisfies the query's requirements even when the specific visual features the query describes are absent from the retrieved views. This produces the false-pass errors in Table 5.

A.4 Case Studies of Type 1 Failures

Figure 6 shows representative examples of the four visual failure causes defined in Section 5.5. For each case we show the instruction, the ground-truth street views, and the street-level views the model retrieved to support its (incorrect) answer.

A.5 Dataset Construction and Human Review

We conduct all manual reviews during dataset construction using the Chinese versions of the queries. All reviewers are native Chinese speakers with at least a bachelor's degree. Reviewing the queries in a common language ensures that consistent linguistic criteria are applied before the queries are localized into the other six languages. After manual review, the accepted queries are translated into the other six languages by Claude-Opus-4.8 [3], a powerful LLM that is independent of all evaluated models.

For each query, reviewers jointly inspect the query text and its associated visual evidence according to the following protocol.

[3] https://www.anthropic.com/news/claude-opus-4-8

Table 8: Dataset-construction statistics for **San Francisco**.

Query Type	Construction Stage	Input Count	Retained Count
Type 1	Keyword retrieval	–	138 candidate places
	MLLM-based street-view screening	138 candidate places	105 candidate places
	Query generation	105 candidate places	$105 \times 3 = 315$ queries
	Human review	315 queries	127 queries
Type 2	MLLM-based predicate brainstorming	–	100 predicates
	Street-view and satellite verification	100 predicates	35 predicates
	Human review	35 predicates	27 queries

Linguistic quality. For both query types, reviewers first assess whether the query is fluent, natural, and realistic. In particular, it should resemble how a person informally asks about a real-world place, rather than read like an automatically generated image caption or a list of disconnected visual attributes.

Review of Type 1 queries. For each Type 1 query, reviewers jointly inspect the query and its corresponding street-view images. The depicted target must be outdoors and have distinctive visual features. These features should make the target plausibly unique within the city, with no obvious concentration of similar places nearby. The query must faithfully and sufficiently describe the target's salient visual characteristics without introducing nonexistent attributes or exaggerating visible ones. Reviewers reject queries that rely excessively on minor, inconspicuous details or, conversely, are too generic to distinguish the target from other places.

Review of Type 2 queries. For each Type 2 query, reviewers inspect the street-view and satellite imagery of multiple candidate places. The visual evidence must demonstrate that the predicate applies consistently to a class of places rather than to a single instance. Accordingly, the query should describe visual properties shared across the verified candidates while avoiding incidental details that appear at only one location. A query is retained only when imagery from multiple distinct places supports its stated visual requirements.

Only queries satisfying all applicable criteria are included in the final benchmark. Table 8 illustrates the filtering process using San Francisco as an example. For Type 1 queries, keyword retrieval produces 138 candidate places, of which 105 pass the MLLM-based street-view screening. The MLLM then generates three queries for each retained place, yielding 315 queries for manual review, of which 127 are accepted. For Type 2 queries, MLLM-based brainstorming generates 100 candidate predicates. Visual verification reduces this set to 35, and manual review retains 27 queries in the final dataset.

A.6 Tool Call Settings

We implement all three geospatial tools using the Google Maps Platform and apply the same tool configurations to every evaluated model. At each interaction step, the agent outputs a single JSON action specifying either a tool invocation or a final answer. The agent may invoke the tools in any order and may call the same tool multiple times, subject to a shared budget of 30 geospatial tool calls per query. Invalid or unparsable actions do not consume this budget, but evaluation terminates after three such parsing failures.

***SearchPlaces*.** The *SearchPlaces* tool retrieves candidate places from a free-form keyword query generated by the agent. Each call returns at most ten candidates. For every candidate, the tool obtains its place name, formatted address, latitude, longitude, and Google place identifier. The name, address, and coordinates are presented to the agent as a textual observation, while the complete structured response is retained in the interaction trajectory for subsequent analysis. We do not paginate the search results, so each call exposes only the first ten candidates returned by the API. The agent may issue additional or reformulated searches when these candidates are insufficient.

***SatelliteView*.** The *SatelliteView* tool acquires an overhead image centered at a coordinate specified by the agent. Each request uses the satellite map type, a zoom level of 19, and an image resolution

of 640×640 pixels. The image is retrieved in JPEG format and returned to the agent together with the queried coordinate.

***StreetView*.** The *StreetView* tool combines the Google Street View Metadata API and Street View Static API. Before requesting imagery, the tool queries the metadata endpoint to determine whether street-view coverage is available near the specified coordinate. If coverage is available, it retrieves four 640×640 JPEG images at headings of $0°$, $90°$, $180°$, and $270°$. Every image uses a horizontal field of view of $90°$ and a pitch of $0°$, thereby providing four complementary, ground-level views that jointly cover the surroundings of the queried location. If no imagery is available, the tool returns an explicit no-coverage observation rather than a placeholder image.

Observation and trajectory handling. Every tool invocation produces a textual summary, optional images, and structured metadata. The complete sequence of actions and textual observations remains available to the agent throughout the trajectory. To control the multimodal context size, only images produced by the immediately preceding tool invocation are attached to the next model call; images from earlier steps remain stored in the trajectory but are not repeatedly inserted into the model context. All retrieved images, tool arguments, structured responses, and model actions are saved for reproducibility and process-oriented evaluation. Geospatial API requests use a timeout of 30 seconds, and failures are returned to the agent as explicit tool errors so that it may revise its search strategy.

A.7 Category Keywords for Type 1 Query Generation

Table 9 lists the full set of category keywords used to retrieve Type 1 candidate places from the Google Maps Platform. The set was constructed manually to favor categories with prominent, visually distinctive appearances and low cross-instance repetition.

Table 9: The 13 category keywords used for Type 1 candidate retrieval.

landmark	monument	memorial	cathedral	clock tower
gate	statue	historic theater	mural	must see
iconic landmark	tourist attraction	point of interest		

A.8 PROMPT TEMPLATES

We list every prompt template used in the IDOL benchmark-generation and evaluation framework.

A.8.1 EVALUATION AGENT PROMPTS

System prompt for Type 1 agent. Instructs the agent to find the single place matching a vague description, by composing the three geospatial tools in any order within a fixed call budget.

System prompt for Type 2 agent. Instructs the agent to find one or more places satisfying an open-ended visual predicate. It differs from the Type 1 system prompt in that the answer action carries a `places` array and the prompt asks for at least two qualifying places.

Per-turn user message. Assembles the user message sent at each interaction step. It contains the query and, when available, the accumulated textual history and the images returned by the immediately preceding tool call.

Type 1 agent system prompt.

```
You are a local guide in {city}. The user will describe a place they want to find using a vague natural-language instruction. Use the available tools to find and verify the single place that best matches the description.

## Available tools (invoke exactly one per turn and output one JSON object)
1. Search for candidate places by keyword:
   {"action":"search","query":"English keywords"}
   Returns candidate place names, addresses, and coordinates.
2. View street-level imagery at coordinates:
   {"action":"streetview","lat":37.8,"lng":-122.4}
   Returns four street-view images facing front, right, back, and left.
3. View overhead satellite imagery at coordinates:
   {"action":"satellite","lat":37.8,"lng":-122.4}
   Returns an overhead satellite image centered at the coordinates.

## Suggested workflow (the order and number of calls are unrestricted)
You may search for candidates first, then use satellite imagery to inspect layouts or terrain and street-view imagery to inspect ground-level details as needed.
You may combine the tools in any order within the budget of {max_calls} tool calls.

## When to finish
When you believe you have found the correct place, output:
{"action":"answer","name":"place name","lat":37.8,"lng":-122.4,"reason":"one-sentence reason"}
An answer action ends the run. If you are uncertain, continue using the tools.

Important: output exactly one JSON object per turn and no additional text.
```

Type 2 agent system prompt.

```
You are a local guide in {city}. The user will vaguely describe a class of places they want to find rather than one specific place. Use the available tools to find and verify one or more places that satisfy the description; you may return multiple valid places.

## Available tools (invoke exactly one per turn and output one JSON object)
1. Search for candidate places by keyword:
   {"action":"search","query":"English keywords"}
   Returns candidate place names, addresses, and coordinates.
2. View street-level imagery at coordinates:
   {"action":"streetview","lat":37.8,"lng":-122.4}
   Returns four street-view images facing front, right, back, and left.
3. View overhead satellite imagery at coordinates:
   {"action":"satellite","lat":37.8,"lng":-122.4}
   Returns an overhead satellite image centered at the coordinates.

## Suggested workflow (the order and number of calls are unrestricted)
You may search for candidates first, then use satellite imagery to inspect layouts or terrain and street-view imagery to inspect ground-level details. Verify that each candidate truly satisfies the description; more than one place may qualify. You may combine the tools in any order within the budget of {max_calls} tool calls.

## When to finish
When you believe you have found qualifying places, output:
{"action":"answer","places":[{"name":"place name","lat":37.8,"lng":-122.4,"reason":"one-sentence reason"}]}
The places array must contain at least two places; include every qualifying place you found.
An answer action ends the run. If you are uncertain, continue using the tools.

Important: output exactly one JSON object per turn and no additional text.
```

Per-turn user message.

```
The user asks: "{query}"

Interaction history so far:
{transcript_lines}

The {n_images} attached images were returned by your previous tool call.

Output the next action as one JSON object.
```

A.8.2 Construction Prompts for Type 1 Queries

Landmark detection and query generation system prompt. Sets the annotator role and the anti-hallucination ground rule. Used for every Type 1 candidate place.

Landmark detection and query generation user prompt. Defines the three acceptance criteria, the anti-hallucination rule, and the exact output schema. The {region} placeholder is a coarse neighborhood-level region (not a full address) to avoid leaking the precise location to the generation model.

Landmark detection and query generation system prompt.

You are a rigorous geospatial annotator. You will receive four Street View images facing front, right, back, and left. Determine whether they contain a distinctive human-made scene or object, then write place-finding questions about it. Base every judgment strictly on visible image content; do not infer details from place names or prior knowledge.

Landmark detection and query generation user prompt.

Approximate area: {region}

Using the Street View images below, determine whether they contain a distinctive human-made scene, object, or structure, and write place-finding questions about it.

The feature may be any size: a building, statue, door, sign, or another object. It must be distinctive enough to recognize at a glance and satisfy all three criteria:
1. Distinctiveness: it is unusual rather than something found throughout any city (ordinary streetlights, storefronts, and apartment buildings do not qualify), and a single visual description can distinguish and locate it within its surroundings.
2. Measurability: at least one of its color, shape, or size is salient enough to serve as an identifying cue.
3. Human-made nature: it must be a building, statue, archway, sign, door, decoration, or other constructed object. Natural scenery such as mountains, trees, lakes, and sky does not qualify.

Anti-hallucination rule: use only content actually visible in the images. Do not infer anything from the place name or prior knowledge. If you cannot clearly point to the distinctive object in the images, set has_landmark to false.

Complete three tasks:
1. Decide whether the images contain a distinctive human-made scene or object as defined above.
2. If so, describe it in one sentence in Simplified Chinese. Include a concrete color, shape, or size attribute; avoid vague claims such as 'very distinctive.'
3. If so, write three colloquial place-finding questions in Simplified Chinese:
   - Phrase them as questions asking where a place is or asking for help finding it, not as navigation commands, and do not refer to 'this place' or 'this image.'
   - Do not include the place name or exact address.
   - Include both an approximate-area clue and visual clues.
   - Keep each question vague and clue-based, as a natural request for directions.

Output JSON: {"has_landmark": true/false, "visual_feature": "...", "queries": ["...", "...", "..."]}

A.8.3 Construction Prompts for Type 2 Queries

Predicate brainstorming system prompt. Sets the task: design open-ended place-finding requirements whose answer set is a class of places rather than a single instance.

Predicate brainstorming user prompt. Enforces the five acceptance rules, a hard exclusion list of sign-only businesses, and de-duplication against already-generated predicates.

Street-view verification system prompt. Sets the strict street-view annotator role for verifying whether a candidate place satisfies a predicate.

Satellite verification system prompt. Sets the remote-sensing annotator role for verifying a candidate place from an overhead satellite image.

Verification user prompt (street-view and satellite). Supplies the candidate place and predicate and requests a binary verdict with a one-sentence justification.

Predicate brainstorming system prompt.

You design Type 2 questions for a geospatial benchmark in which users describe a vague place-finding need. The answer set is open: multiple distinct places in the city should satisfy the same requirement.

Predicate brainstorming user prompt.

Design {num} mutually distinct Type 2 place-finding requirements for {city}. Each must satisfy all five rules:
1. Street-view verifiability: whether a place qualifies must be visible from ground-level Street View imagery, such as a sports court, beach, fountain, gas station, or steep road. Exclude properties that images cannot establish, such as quietness, price, Wi-Fi, ratings, or opening hours.
2. Multiple instances: at least three places in {city} should qualify. Use a general category, not a unique landmark.
3. General function: describe a function or form shared by a class of places, not a proper name.
4. Realistic need: formulate something an ordinary user would genuinely ask to find.
5. Recognition by form, not signage: the evidence must be the shape or structure of the object, terrain, or facility itself, such as basketball hoops and a hard court, fountain jets, a pier, cable-car tracks, fuel pumps, an outdoor pool, rows of picnic tables, or a steep road surface.

Hard exclusion: do not generate shops, restaurants, or institutions that can only be identified by reading storefront signs, facade text, or nameplates. This includes supermarkets, convenience stores, banks, pharmacies, hospitals, post offices, schools, libraries, museums, hotels, gyms, cinemas, universities, courts, police stations, specialty shops, and food or beverage businesses. They do not qualify even if their signs are visible in Street View.

Existing requirements (do not repeat them, including paraphrases or punctuation-only variants): {avoid}

Return three fields for each requirement:
- predicate: one precise judging criterion in Simplified Chinese that names the key visual shape or structural features needed to decide whether a place qualifies;
- search_queries: three English map-search queries for Google Places, each including the city name;
- query: one colloquial place-finding question in Simplified Chinese. Mention only the user's need or intended activity; keep it natural and vague and do not name a specific place. Do not mention Street View, visibility, recognition, passing by, seeing the place from outside, or photography. Those are evaluation criteria, not part of a natural user request. Good intents include finding an outdoor pool, a place with a fountain, somewhere a dog can run freely, or somewhere to fly a kite.

Output only a JSON array in this form: [{{"predicate":"...","search_queries":["...","...","..."],"query":"..."}}]

Street-view verification system prompt.

You are a rigorous Street View annotator. You will receive four images of a place, facing front, right, back, and left. Determine whether visible features establish that the place satisfies the specified requirement. Use only visible image content; do not infer anything from place names or prior knowledge.

Satellite verification system prompt.

You are a rigorous remote-sensing annotator. You will receive an overhead satellite image of a place. Determine whether visible features establish that the place satisfies the specified requirement. Use only visible image content; do not infer anything from place names or prior knowledge.

Verification user prompt (street-view and satellite).

Place: {name} ({address})
User requirement/predicate: {predicate}

Based only on the attached Street View images, are features that establish the place satisfies the requirement actually visible? Do not infer anything from the place name or prior knowledge; if the evidence is not visible, answer false.

Output JSON: {"satisfies": true/false, "reason": "one sentence stating what is or is not visible"}

Place: {name} ({address})
User requirement/predicate: {predicate}

The attached image is an overhead satellite view of this place. Based only on features visible from above, can you establish that the place satisfies the requirement? Do not infer anything from the place name or prior knowledge; if the evidence is not visible, answer false.
Output JSON: {"satisfies": true/false, "reason": "one sentence stating what is or is not visible"}

A.8.4 Evaluation Judge Prompts for Type 2 Queries

Judge system prompt. Sets the evaluator role and restricts judgment to the supplied images.

Judge user prompt. Presents the query, the predicate, the candidate place, and the supporting images, and requests a three-way verdict with a confidence score and evidence.

Judge system prompt.

You are a rigorous evaluator of visual geospatial predicates. Determine whether a place satisfies the predicate using only the supplied images, and never infer information outside the images. The images are Street View or satellite observations retrieved by the agent at the candidate place.

Judge user prompt.

User request: {query}

Required predicate: {predicate}

The attached images are Street View or satellite observations retrieved by the agent at the candidate place. Determine whether this place satisfies the predicate using only these images.

Output strict JSON with no additional text:
{"verdict":"satisfies|fails|indeterminate","confidence":0.0,"evidence":"one-sentence evidence"}
- satisfies: the visual features required by the predicate are clearly visible.
- fails: the scene is discernible but does not satisfy the predicate or is not the target type.
- indeterminate: the images are blurry, irrelevant, or provide insufficient evidence.