FROM GEOGRAPHIC MEMORY TO SPATIAL GENERALIZATION IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) can associate places with geographic descriptions, but recovering an observed association does not establish an ability to generalize across geographic space. We distinguish geospatial memory, the recovery of location–description associations observed during training, from spatial generalization, the prediction of descriptions for locations in unobserved regions. We investigate whether continuous location representations improve this generalization and encourage geographic knowledge to become spatially organized within an LLM. We frame this organization as a latent spatial field: a learned mapping from locations to descriptions connected through spatial dependencies. Building on a continuous spatial encoder, we project location representations into the model's embedding space and supply them as spatial tokens. We evaluate the resulting models on two newly curated geo-text benchmarks for Los Angeles and London using two complementary assessments. Geospatial captioning tests both memory and spatial generalization by generating descriptions at observed locations and at locations beyond the spatial coverage of the training observations. A separate geospatial grounding model tests whether generated location–description pairs provide useful additional supervision for mapping text to geographic space. Across multiple LLM families, continuous location representations improve captioning at unobserved locations relative to textual coordinates, while their generated descriptions yield modest but consistent improvements in downstream grounding. Layer-wise probing and representation clustering provide additional evidence consistent with spatially organized geographic representations. Together, these findings show the value of evaluating spatial generalization separately from memory and of incorporating continuous spatial representations into language models.

1 INTRODUCTION

Large language models (LLMs) can answer questions about geographic places, but correct answers alone do not reveal how their geographic knowledge is organized (Li et al., 2025; Manvi et al., 2024a; Feng et al., 2025b). Consider a model trained on descriptions of several locations near a beach. At a training location, generating "a short walk from the beach" may simply recover an observed location–description association. Generating an appropriate description for a location in a nearby region not covered during training tests whether the model can generalize beyond those associations. This distinction motivates our central question: *Can LLMs organize geographic knowledge spatially enough to generalize to locations without direct training observations, and how does the representation of location affect this ability?*

This question matters because geographic observations are sparse and unevenly distributed samples of continuous space (Lv et al., 2026; Wang et al., 2026). Well-documented locations provide abundant information, while other areas have limited coverage (Feng et al., 2025a). Geographic knowledge nevertheless contains dependencies that can support inference across these gaps: nearby locations often share environmental characteristics, and descriptions of places depend on their relationships to landmarks, transportation networks, and surrounding areas (Wu et al., 2026). We distinguish two capabilities. **Geospatial memory** is the recovery of location–description associations observed during training. **Geospatial reasoning** is spatial generalization to regions beyond the

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.

spatial coverage of the training observations. This second setting tests whether learned geographic relationships support predictions beyond the observed pairs.

How locations are represented may influence both capabilities. Place names and coordinates represented as character strings fit naturally into an LLM's text interface, but their token representations do not explicitly encode geographic proximity (Chu et al., 2026). Existing location encoders (Mai et al., 2020; Chu & Shahabi, 2026) provide continuous representations designed to introduce spatial structure. These methods provide the foundation for our approach. We investigate whether incorporating such representations into an LLM improves its ability to learn geographic dependencies, generate in unobserved regions, and organize geographic information within its hidden states.

We develop a **spatially explicit LLM** by projecting a continuous location encoding into the model's input embedding space and supplying it as a spatial token. The model learns to generate geographic descriptions conditioned on this representation. We interpret the resulting location-to-description mapping as a **latent spatial field**: geographic knowledge associated with different locations is connected through learned spatial dependencies. This perspective provides a hypothesis about how geographic knowledge can be organized, without requiring every geographic property to vary smoothly. Our experiments examine whether continuous location encodings encourage this organization more effectively than textual coordinates, discrete identifiers and hierarchical spatial indexes.

We evaluate the approach using coordinate–description datasets constructed from Airbnb listings in Los Angeles and London. **Geospatial captioning** asks the model to generate a geographic description from a location. Captioning at observed training locations evaluates memory; captioning in withheld spatial regions evaluates reasoning under our definition. Using the same task for both settings allows us to distinguish the ability to recover observed associations from the ability to generalize across gaps in the training data. Comparisons with nearest-neighbor retrieval further assess whether generated descriptions provide benefits beyond reusing nearby observations.

We then test whether these descriptions contain useful geographic information through **geospatial grounding**, a separate task that maps a description to a distribution over possible locations. We use the captioning model to generate additional location–description pairs in data-sparse areas and evaluate whether those pairs improve an independently trained grounding model. Grounding therefore provides a downstream utility test for the generated knowledge, complementing direct comparisons between generated captions and withheld reference descriptions. Finally, layer-wise probing and clustering examine how geographic information is represented within the captioning model.

Across the tested LLM families and both cities, textual coordinates support strong recovery of observed associations, while continuous location representations improve captioning in withheld regions. Generated descriptions can also provide modest improvements in downstream grounding. Probing and clustering reveal patterns consistent with more spatially organized hidden representations. Together, these findings support the value of continuous location encodings for geographic generalization, while providing complementary behavioral and representational evidence for the latent-spatial-field perspective. Our contributions are threefold:

- **A framework for evaluating geographic generalization in LLMs.** We distinguish recovery of observed associations from prediction in spatially uncovered regions, using captioning to evaluate both and independent grounding to assess downstream utility.
- **An integration of continuous location representations into LLMs.** We systematically compare continuous encodings with textual, grid-based, and hierarchical representations.
- **An empirical investigation of geographic knowledge organization.** Experiments on Los Angeles and London connect location representation, generalization performance, downstream usefulness, and the spatial structure of hidden representations.

2 Related Work

Geospatial Reasoning in LLMs Spatial reasoning is typically formulated as inference from explicitly provided relations, such as deriving the relation between A and C from 'A is north of B" and 'B is north of C" (Li et al., 2024; Cheng et al., 2024), or decomposed into atomic functions (Chu et al., 2026; Bao et al., 2026). Real-world geographic dependencies, however, are too complex to

enumerate symbolically (Brunsdon et al., 1996), while supplying all relevant observations is impractical. We study geospatial reasoning as inference from knowledge internalized by an LLM.

Geospatial Knowledge in LLMs Prior work shows that LLMs encode geographic knowledge useful for downstream prediction tasks (Feng et al., 2025b; Manvi et al., 2024b), while Gurnee & Tegmark (2024) reveal decodable location representations across multiple spatial scales. These studies primarily examine entities and places assumed to be represented in the training data. We instead investigate whether learned geographic knowledge reflects surface-level text–location associations encountered during training or a coherent internal spatial organization that supports generalization to unobserved locations.

Geospatial Representation Geospatial representation learning maps locations or entities into a latent space while preserving properties such as distance, direction, periodicity, and multi-scale structure (Chu & Shahabi, 2026; Siampou et al., 2025). It introduces inductive bias of location into various downstream models (Klemmer et al., 2025; Mai et al., 2023; 2020). We integrate continuous location encodings directly into an LLM and investigate whether they encourage geographic knowledge to form a coherent latent spatial field that supports both memory and reasoning.

3 Geospatial Memory and Reasoning over Latent Spatial Field

Given the largely persistent and continuous nature of geospatial knowledge, we propose to view its internal organization within a language model as a latent spatial field. Under this view, geographic knowledge is not merely stored as a collection of independent location–value associations. Instead, these associations form a structured field in which the knowledge associated with different locations is connected through implicit spatial dependencies.

3.1 Geospatial Knowledge as a Latent Spatial Field

Let $\Omega \subseteq \mathbb{R}^2$ denote a continuous geographic domain, and let $\mathcal{V}$ denote a general space of geographic values. A value $v \in \mathcal{V}$ may represent a natural-language description, a land-use category, or any other information associated with a location. We represent the observed geospatial data as $\mathcal{D} = \big\{(\mathbf{x}_i, v_i)\big\}_{i=1}^{N}$, where $\mathbf{x}_i \in \Omega$ is a geographic location and $v_i \in \mathcal{V}$ is its observed value. These observations can be viewed as sparse samples from an underlying spatial field $F : \Omega \to \mathcal{V}$, where $F(\mathbf{x})$ represents the geographic knowledge associated with location $\mathbf{x}$. In practice, these observations usually are not collected uniformly. We express the data-generating process as

$$(\mathbf{x}, v) \sim \pi_{\text{obs}}(\mathbf{x}) p\big(v \mid \mathbf{x}\big), \tag{1}$$

where $\pi_{\text{obs}}(\mathbf{x})$ denotes the spatial observation density and $p(v \mid \mathbf{x})$ describes the distribution of geographic values at location $\mathbf{x}$. Geographic values are also not independent across locations. Consequently, inferring $F(\mathbf{x})$ at an unobserved location requires more than applying an explicitly given symbolic rule: the model must first internalize the latent dependencies F between space and geographic values from the observed data.

3.2 Memory and Reasoning

Given the training observations in $\mathcal{D}_{\text{train}}$, a model approximates the underlying field F by learning $\theta^* = \arg\min_{\theta} \sum_{(\mathbf{x}_i, v_i) \in \mathcal{D}_{\text{train}}} \mathcal{L}\big(F_\theta(\mathbf{x}_i), v_i\big)$, where $\mathcal{L}$ denotes the training loss. The learned field is F_{θ^*}, and inference at location $\mathbf{x}$ is given by $\hat{v} = F_{\theta^*}(\mathbf{x})$. We distinguish geospatial memory from geospatial reasoning according to whether the target location–value association was observed during training. Let $\mathcal{X}_{\text{obs}} = \big\{\mathbf{x}_i : (\mathbf{x}_i, v_i) \in \mathcal{D}_{\text{train}}\big\}$ denote the set of observed training locations. We define **geospatial memory** as the ability to retain values associated with locations in $\mathcal{X}_{\text{obs}}$:

$$F_{\theta^*}\big(\mathbf{x}_i\big) \approx v_i, \qquad \mathbf{x}_i \in \mathcal{X}_{\text{obs}}. \tag{2}$$

Geospatial memory therefore measures whether the model has internalized the observed geographic knowledge so that it can be directly retrieved from its parameters.

To define geospatial reasoning, let $\mathcal{H} \subseteq \Omega$ denote a collection of regions, not covered by the training observations. We define the corresponding dataset as $\mathcal{D}_{\text{h}} = \big\{(\mathbf{x}_i, v_i) \in \mathcal{D} : \mathbf{x}_i \in \mathcal{H}\big\}$, where $\mathcal{H} \cap$

$\mathcal{X}_{\text{obs}} = \varnothing$. The values in $\mathcal{D}_{\text{h}}$ remain unavailable during training. We define **geospatial reasoning** as the ability to infer the geographic value v_* at an unobserved location $\mathbf{x}_* \in \mathcal{H}$:

$$\hat{v}_* = F_{\theta^*}(\mathbf{x}_*) \approx v_*, \qquad \mathbf{x}_* \in \mathcal{H}. \tag{3}$$

Equivalently, geospatial reasoning estimates the posterior predictive distribution $p(v_* \mid \mathbf{x}_*, \mathcal{D}_{\text{train}})$ for a location whose value was not observed. Although $\mathcal{D}_{\text{train}}$ is not explicitly supplied as context at inference time, its spatial dependencies have been internalized in the learned parameters θ^*. The model must use this learned structure to bridge the missing portion of the geographic field.

4 Intrinsic and Extrinsic Evaluation of Memory and Reasoning

We employ two complementary evaluation tasks. Geospatial captioning directly evaluates whether a model can recover geographic knowledge at observed locations and reason about unobserved ones. Geospatial grounding, implemented using a separate model, provides an extrinsic evaluation of reasoning by testing whether knowledge inferred at unobserved locations benefits a downstream task.

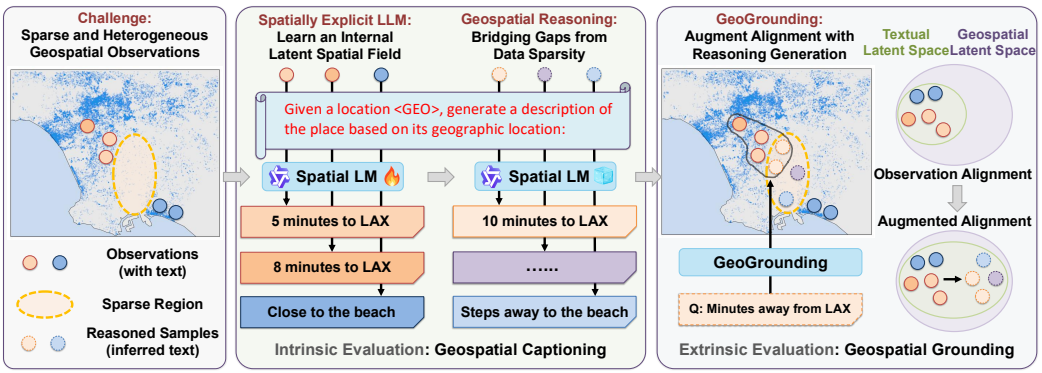


Figure 1: **Overview of geospatial captioning and grounding.** Geo-text observations are spatially heterogeneous. A spatially explicit LLM organizes geographic knowledge as a latent spatial field that supports generalization across space. Through **geospatial captioning**, it generates geo-text pairs that augment the observed data and improve text–space alignment in **geospatial grounding**.

4.1 Geospatial Captioning

As shown in Figure 1, geospatial captioning instantiates our general formulation by defining the geographic value space $\mathcal{V}$ as the space of natural-language descriptions of locations $\mathcal{Y}$. A captioning model learns the conditional mapping $c_\psi : \Omega \to \mathcal{P}(\mathcal{Y})$, where $\mathcal{P}(\mathcal{Y})$ denotes the space of probability distributions over possible descriptions. Given a location $\mathbf{x}$, the model generates

$$\hat{y} \sim p_\psi(y \mid \mathbf{x}). \tag{4}$$

At an observed location $\mathbf{x} \in \mathcal{X}_{\text{obs}}$, the generated caption $\hat{y}$ evaluates geospatial memory. At an unobserved location $\mathbf{x} \in \mathcal{H}$, it evaluates geospatial reasoning by testing whether the model can infer an unobserved geo-text association. Because multiple descriptions may be valid for the same location, the intrinsic evaluation measures the semantic and lexical alignment between the generated description $\hat{y}$ and the reference description y.

4.2 Extrinsic Evaluation through Geospatial Grounding

Intrinsic similarity to a reference does not fully establish whether generated descriptions contain useful geographic knowledge. We therefore introduce geospatial grounding as an independent extrinsic evaluation task. Given a natural-language description, a separate grounding model predicts a probability distribution over geographic space:

$$g_\phi : \mathcal{Y} \to \mathcal{P}(\Omega), \qquad g_\phi(y) = q_\phi(\cdot \mid y), \tag{5}$$

where locations consistent with the query receive higher probabilities. This formulation supports both precise descriptions of individual locations and spatially ambiguous descriptions of regions.

We query the captioning model at augmentation locations $\mathcal{X}_{\text{aug}}$ in data-sparse regions and construct the augmented dataset $\widetilde{\mathcal{D}}_{\text{aug}} = \{(\tilde{\mathbf{x}}_j, \tilde{y}_j) : \tilde{\mathbf{x}}_j \in \mathcal{X}_{\text{aug}},\ \tilde{y}_j \sim p_\psi(y \mid \tilde{\mathbf{x}}_j)\}$. We then compare a grounding model trained on $\mathcal{D}_{\text{train}}$ with one trained on $\mathcal{D}_{\text{train}} \cup \widetilde{\mathcal{D}}_{\text{aug}}$. Improved grounding performance indicates that the LLM's generalization provides useful and geographically consistent supervision rather than merely plausible text. By filling gaps in the observed geo-text distribution, such improvements provide evidence that the captioning model has learned a coherent latent spatial field and can infer meaningful geographic knowledge at previously uncovered locations.

5 Spatially Explicit LLMs

Having formulated geospatial reasoning as reconstructing an unobserved portion of a latent spatial field, we next consider how locations should be represented within an LLM. The input representation determines the geometry through which the model observes space and, consequently, the spatial field it can learn. When a coordinate such as `(34.0° N, 118.2° W)` is represented as text, it is decomposed into a sequence of discrete tokens whose embedding similarity does not necessarily preserve geographic proximity: nearby coordinates may differ in multiple tokens, while distant coordinates may share many. The LLM must therefore recover the continuous structure of space while simultaneously learning how geographic values vary over it.

We instead encode each location continuously and inject it into the LLM as a spatial token, providing an explicit inductive bias for the continuity and multi-scale structure of geographic space. Let $PE_{\text{geo}} : \Omega \to \mathbb{R}^{d_{\text{geo}}}$ denote a spatial encoder whose representation may capture absolute location, direction, and multi-scale spatial patterns. To align this representation with the dimension and distribution of the LLM token embeddings, we introduce a learned adapter $A \in \mathbb{R}^{d_{\text{geo}} \times d_{\text{LLM}}}$:

$$\mathbf{u}_{\mathbf{x}} = PE_{\text{geo}}(\mathbf{x}), \qquad \mathbf{Z}_{\mathbf{x}} = \mathbf{u}_{\mathbf{x}} A \in \mathbb{R}^{d_{\text{LLM}}}. \tag{6}$$

We replace the textual coordinate tokens with $\mathbf{Z}_{\mathbf{x}}$ and refer to the resulting architecture as a **spatially explicit LLM**, in which location is represented as a continuous modality that preserves spatial relationships. Our central hypothesis is that this representation encourages the LLM to organize geographic knowledge as a smoother and more generalizable latent spatial field.

6 Experiments

6.1 Datasets and Settings

We construct paired location–description datasets from Airbnb listings. Each listing contains a human-written property overview that describes its location and other context. We preprocess these overviews to extract self-contained, location-specific descriptions, forming overall data for captioning and grounding. To evaluate spatial generalization, we partition each city into **disjoint** training and evaluation regions and curate benchmarks for both tasks. The grounding benchmark includes two settings: **point-level grounding** for descriptions associated with specific coordinates and **region-level grounding** for descriptions associated with a region. We conduct experiments independently on Los Angeles and London. Further details are provided in Appendices A.1 and A.2.

We evaluate captioning using BERTScore F1 and ROUGE-L F1. For point-level grounding, we report Hit Rate@1% Area and Hit Rate@5% Area, which measure the percentage of ground-truth locations that fall within the highest-scoring region covering 1% or 5% of the domain. For region-level grounding, coverage rate, Jensen–Shannon divergence (JSD) and Wasserstein distance measure the alignment between predicted and ground-truth distributions. Details are provided in Appendix A.3.

6.2 Spatial Representations and LLMs

We compare four spatial representation methods and one hybrid variant:

- **GRID**: The geographic domain is divided into an $n \times n$ grid. Each location is assigned to a unique grid cell, and the corresponding grid identifier is provided to the LLM (Li et al., 2025).

- **HASH**: Each coordinate is encoded as a length-9 Base32 geohash, providing a hierarchical spatial index with approximately 2.4 meters of positional precision (Cui et al., 2026).
- **TEXT**: The coordinate pair is directly represented as text and provided to the LLM as part of the input prompt (Ji et al., 2025).
- **LATENT**: Each coordinate is encoded as a 144-dimensional continuous spatial representation using the positional encoder introduced in Geo2Vec (Chu & Shahabi, 2026).
- **HYBRID**: The textual coordinate and latent representation are provided together. The **TEXT** representation is followed by the corresponding **LATENT** representation in the input sequence.

We evaluate these representations across three LLM families. **GPT-OSS** denotes `gpt-oss-20b`[1], **Qwen3.5** denotes `Qwen3.5-9B`[2], and **Gemma4** denotes `Gemma-4-12B`[3]. Qwen3.5 and Gemma4 natively support multimodal inputs, whereas GPT-OSS is a text-only mixture-of-experts model. We evaluate each LLM using all spatial representation methods. For a fair comparison, all models are trained separately on each dataset using the same training samples, experimental procedures, and random seeds. Further training details are provided in Appendix A.4.

6.3 Evaluation of Geospatial Memory

We first evaluate the LLMs' zero-shot performance on the observed locations. In Table 1, rows labeled only with the model name report zero-shot results without tuning. `Random` is a baseline that assigns each query a description sampled randomly from the training descriptions.

Table 1: **Geospatial captioning performance on observed locations.** Evaluation of memory. Bold and underlined denote the best and second-best results.

	Los Angeles		London	
Method	BERT ↑	ROUGE ↑	BERT ↑	ROUGE ↑
Random	0.4410	0.2070	0.4664	0.2282
GPT-OSS	0.4048	0.1155	0.4158	0.1523
Qwen3.5	0.4255	0.1477	0.4555	0.2094
Gemma4	0.4392	0.2245	0.5012	0.2654
GPT-OSS-GRID	0.5125	0.2884	0.5449	0.3299
GPT-OSS-HASH	0.5376	0.3190	0.5568	0.3414
GPT-OSS-TEXT	0.5582	0.3462	0.5657	0.3548
GPT-OSS-HYBRID	0.5511	0.3322	0.5728	0.3627
GPT-OSS-LATENT	0.5562	0.3409	0.5730	0.3630
Qwen3.5-GRID	0.5593	<u>0.3566</u>	0.5694	0.3604
Qwen3.5-HASH	0.5588	0.3520	0.5822	0.3793
Qwen3.5-TEXT	**0.5623**	**0.3585**	**0.5849**	**0.3821**
Qwen3.5-HYBRID	0.5565	0.3444	0.5795	0.3752
Qwen3.5-LATENT	0.5536	0.3401	0.5771	0.3711
Gemma4-GRID	0.5464	0.3363	0.5640	0.3537
Gemma4-HASH	<u>0.5619</u>	0.3513	0.5812	0.3805
Gemma4-TEXT	0.5567	0.3478	<u>0.5833</u>	<u>0.3817</u>
Gemma4-HYBRID	0.5580	0.3500	0.5770	0.3746
Gemma4-LATENT	0.5524	0.3393	0.5678	0.3591

First, zero-shot performance is poor. GPT-OSS and Qwen3.5 underperform random sampling, while Gemma4 performs only slightly better; all three models remain substantially below their fine-tuned variants. These results suggest that the pretrained models contain limited geospatial knowledge about the queried cities. The substantial improvements after fine-tuning can therefore be attributed to the acquisition of geospatial memory.

Among the fine-tuned models, **TEXT** achieves the strongest overall memory performance, followed by the **HASH** and **HYBRID** representations. This result is expected because an exact textual coordinate serves as a unique identifier for an observed location, and LLMs are effective at learning direct token-to-token associations. In contrast, **GRID** generally performs worse because multiple locations share the same grid identifier, limiting the accurate retrieval of location-specific knowledge.

6.4 Intrinsic Evaluation of Geospatial Reasoning

We introduce `k-NN` as a retrieval-based baseline. For each test location, `k-NN` retrieves the descriptions associated with its $k = 4$ nearest training locations as predictions. This baseline tests whether LLMs can infer geographic knowledge beyond simply copying descriptions from nearby observations. Distance-stratified evaluation can be found in Appendix A.5.

[1] `https://huggingface.co/openai/gpt-oss-20b`
[2] `https://huggingface.co/Qwen/Qwen3.5-9B`
[3] `https://huggingface.co/google/gemma-4-12B`

As shown in Table 2, **GRID** consistently underperforms all methods, including `k-NN`, because the grid identifiers encode no spatial relationship at all. In contrast, all other fine-tuned representations outperform nearest-neighbor retrieval, indicating that spatially tuned LLMs can learn geographic patterns beyond simply copying nearby observations.

Table 2: **Geospatial captioning performance across unobserved regions.** Evaluation of spatial generalization.

	Los Angeles		London	
Method	BERT ↑	ROUGE ↑	BERT ↑	ROUGE ↑
k-NN	0.5213	0.2931	0.5306	0.2980
GPT-OSS-GRID	0.4921	0.2595	0.5084	0.2871
GPT-OSS-HASH	0.5320	0.3109	0.5348	0.3162
GPT-OSS-TEXT	0.5329	0.3078	0.5524	0.3369
GPT-OSS-HYBRID	<u>0.5503</u>	<u>0.3289</u>	0.5597	0.3447
GPT-OSS-LATENT	**0.5524**	**0.3295**	0.5606	<u>0.3459</u>
Qwen3.5-GRID	0.4929	0.2663	0.5242	0.2959
Qwen3.5-HASH	0.5385	0.3167	0.5425	0.3245
Qwen3.5-TEXT	0.5345	0.3243	0.5580	0.3377
Qwen3.5-HYBRID	0.5441	0.3240	<u>0.5613</u>	0.3449
Qwen3.5-LATENT	0.5432	0.3212	0.5607	0.3455
Gemma4-GRID	0.4888	0.2602	0.5218	0.2973
Gemma4-HASH	0.5435	0.3225	0.5425	0.3245
Gemma4-TEXT	0.5423	0.3195	0.5599	0.3434
Gemma4-HYBRID	0.5439	0.3237	0.5609	0.3451
Gemma4-LATENT	0.5453	0.3287	**0.5617**	**0.3475**

More importantly, the generalization results reverse the pattern observed for geospatial memory, where **TEXT** achieves the strongest overall performance. For generalization, **LATENT** and **HYBRID** consistently outperform **TEXT** variants across almost all LLM families and both datasets. This contrast empirically supports our distinction between geospatial memory and reasoning. The gains also extend to GPT-OSS, despite its text-only architecture, demonstrating that continuous spatial representations can be effectively integrated into different LLM families. Their consistent improvements suggest that spatially explicit representations provide a broadly applicable foundation for strengthening geospatial awareness in LLMs.

6.5 Extrinsic Evaluation of Geospatial Reasoning

We further evaluate the inferred geographic knowledge through the geospatial grounding framework. For each augmentation setting, we train a separate CLIP-based grounding model over five runs, using different seeds but otherwise identical settings across representation methods. `GeoGrounding` denotes the baseline trained only on $\mathcal{D}_{\text{train}}$, whereas `w/` denotes models trained with augmented dataset generated by the corresponding captioning model. More details are provided in Appendix A.6.

Tables 3 and 4 present the point- and region-level grounding results, respectively. Although **TEXT** substantially outperforms `k-NN` on BERTScore and ROUGE-L, its descriptions do not consistently improve either grounding task, showing that reference similarity does not guarantee useful or spatially coherent knowledge. Grounding therefore provides complementary evidence beyond conventional NLP metrics. Meanwhile, **GRID** augmentation consistently degrades both tasks, since its less accurate descriptions introduce noisy supervision and distort text–space alignment. Thus, augmentation is beneficial only when the generated descriptions preserve meaningful geographic structure.

LATENT provides the strongest overall augmentation performance. Gemma4-LATENT achieves the best mean on all four point-level metrics and four of the six region-level metrics. Although modest, these consistent gains, particularly in region-level distributional alignment, provide functional evidence that continuous representations promote a smoother and more structured latent spatial field, enabling more geographically coherent generation at unobserved locations.

In sum, **HYBRID** generally improves captioning and grounding over **TEXT** while retaining competitive memory performance, offering a practical balance between memorizing observed textual associations and generalizing over unobserved regions. Overall, the grounding results support our argument that learning a continuous latent spatial field enables stronger geospatial generalization.

6.6 Evidence of Spatial Latent Field

To examine whether LLM hidden states encode geographic structure, we conduct layer-wise probing. For each location description, we extract its generation hidden state from five transformer layers and train an MLP to predict the corresponding latitude and longitude. Further details are provided in Appendix A.7. **NAIVE** denotes models without task-specific tuning.

Table 3: **Point-level geospatial grounding performance under different data augmentations.** Values following ± denote standard deviations. ↑ indicates that higher values are better.

	Los Angeles		London	
Method	Hit Rate@1% ↑	Hit Rate@5% ↑	Hit Rate@1% ↑	Hit Rate@5% ↑
GeoGrounding	34.90% ± 0.52%	<u>54.81%</u> ± 0.57%	55.07% ± 0.33%	69.26% ± 0.70%
w/ k-NN	35.05% ± 0.46%	54.57% ± 0.41%	55.52% ± 0.25%	69.52% ± 0.32%
w/ GPT-OSS-GRID	30.70% ± 0.26%	51.39% ± 0.78%	53.24% ± 0.56%	67.41% ± 0.92%
w/ GPT-OSS-HASH	34.79% ± 0.45%	54.46% ± 0.43%	55.10% ± 0.66%	69.04% ± 0.99%
w/ GPT-OSS-TEXT	34.84% ± 0.26%	54.60% ± 0.31%	55.34% ± 0.63%	69.16% ± 0.66%
w/ GPT-OSS-HYBRID	35.09% ± 0.41%	54.62% ± 0.68%	55.54% ± 0.37%	69.43% ± 0.21%
w/ GPT-OSS-LATENT	35.00% ± 0.44%	54.65% ± 0.24%	55.52% ± 0.28%	69.40% ± 0.45%
w/ Qwen3.5-GRID	31.92% ± 0.31%	52.47% ± 0.50%	53.98% ± 0.73%	67.99% ± 0.78%
w/ Qwen3.5-HASH	34.96% ± 0.47%	54.60% ± 0.42%	55.19% ± 0.80%	69.17% ± 0.89%
w/ Qwen3.5-TEXT	34.98% ± 0.50%	54.54% ± 0.49%	55.43% ± 0.75%	69.19% ± 0.72%
w/ Qwen3.5-HYBRID	35.07% ± 0.35%	54.41% ± 0.54%	55.30% ± 0.41%	69.25% ± 0.57%
w/ Qwen3.5-LATENT	<u>35.40%</u> ± 0.47%	54.76% ± 0.43%	<u>55.91%</u> ± 0.40%	<u>69.88%</u> ± 0.45%
w/ Gemma4-GRID	30.57% ± 0.71%	51.27% ± 0.97%	53.42% ± 0.78%	67.97% ± 1.12%
w/ Gemma4-HASH	34.92% ± 0.23%	54.77% ± 0.26%	55.67% ± 0.33%	69.76% ± 0.48%
w/ Gemma4-TEXT	34.96% ± 0.39%	54.48% ± 0.40%	55.58% ± 0.37%	69.68% ± 0.59%
w/ Gemma4-HYBRID	35.03% ± 0.53%	54.47% ± 0.60%	55.28% ± 0.77%	69.30% ± 0.89%
w/ Gemma4-LATENT	**35.89%** ± 0.39%	**55.05%** ± 0.52%	**55.96%** ± 0.25%	**70.11%** ± 0.43%

Table 4: **Region-level geospatial grounding performance under different data augmentations.** ↓ indicates that lower values are better.

	Los Angeles			London		
Method	Coverage@1% ↑	JSD ↓	Wasserstein ↓	Coverage@1% ↑	JSD ↓	Wasserstein ↓
GeoGrounding	<u>60.22%</u> ± 0.86%	0.4747 ± 0.0019	7459.29 ± 116.46	74.96% ± 0.31%	0.4632 ± 0.0079	6202.41 ± 341.13
w/ k-NN	60.06% ± 0.74%	0.4734 ± 0.0029	7394.78 ± 71.03	75.09% ± 0.39%	0.4693 ± 0.0046	6120.26 ± 239.72
w/ GPT-OSS-GRID	58.57% ± 0.47%	0.4889 ± 0.0047	7743.80 ± 57.20	72.59% ± 0.33%	0.4895 ± 0.0067	6629.12 ± 300.62
w/ GPT-OSS-HASH	59.72% ± 0.72%	0.4716 ± 0.0027	7386.82 ± 62.27	74.82% ± 0.67%	0.4645 ± 0.0098	6075.04 ± 334.89
w/ GPT-OSS-TEXT	60.07% ± 0.51%	0.4690 ± 0.0048	<u>7327.45</u> ± 115.96	74.81% ± 0.66%	0.4690 ± 0.0064	6031.12 ± 249.98
w/ GPT-OSS-HYBRID	59.11% ± 0.55%	0.4709 ± 0.0018	7459.86 ± 64.31	74.91% ± 0.44%	<u>0.4536</u> ± 0.0041	5984.08 ± 215.11
w/ GPT-OSS-LATENT	59.24% ± 0.59%	<u>0.4672</u> ± 0.0063	**7308.13** ± 127.33	74.76% ± 0.32%	0.4549 ± 0.0039	6006.23 ± 168.65
w/ Qwen3.5-GRID	59.04% ± 0.53%	0.4863 ± 0.0049	7756.12 ± 120.81	73.46% ± 0.60%	0.4783 ± 0.0071	6707.53 ± 275.28
w/ Qwen3.5-HASH	60.11% ± 0.50%	0.4736 ± 0.0037	7438.10 ± 76.03	74.92% ± 0.87%	0.4635 ± 0.0087	6002.55 ± 257.31
w/ Qwen3.5-TEXT	60.21% ± 0.58%	0.4728 ± 0.0019	7383.66 ± 82.94	75.02% ± 0.71%	0.4653 ± 0.0083	5999.37 ± 336.68
w/ Qwen3.5-HYBRID	59.79% ± 0.60%	0.4720 ± 0.0073	7490.35 ± 104.48	74.50% ± 0.54%	0.4590 ± 0.0095	6165.56 ± 288.68
w/ Qwen3.5-LATENT	59.88% ± 0.80%	0.4715 ± 0.0009	7394.02 ± 37.57	75.12% ± 0.57%	0.4555 ± 0.0067	6069.78 ± 254.73
w/ Gemma4-GRID	57.93% ± 0.65%	0.4908 ± 0.0028	7768.11 ± 84.88	73.34% ± 0.67%	0.4735 ± 0.0100	6253.76 ± 320.27
w/ Gemma4-HASH	60.01% ± 0.20%	0.4719 ± 0.0020	7389.10 ± 72.46	**75.34%** ± 0.41%	0.4610 ± 0.0068	<u>5882.12</u> ± 293.23
w/ Gemma4-TEXT	59.98% ± 0.37%	0.4743 ± 0.0032	7429.47 ± 89.79	74.91% ± 0.24%	0.4563 ± 0.0048	5931.76 ± 210.17
w/ Gemma4-HYBRID	59.53% ± 0.87%	0.4707 ± 0.0076	7435.79 ± 136.56	74.60% ± 0.10%	0.4589 ± 0.0107	6017.19 ± 280.55
w/ Gemma4-LATENT	**60.40%** ± 0.74%	**0.4650** ± 0.0060	7350.43 ± 50.92	<u>75.33%</u> ± 0.23%	**0.4513** ± 0.0077	**5870.19** ± 285.26

As shown in Figure 2, all tuned models achieve lower probing errors than their **NAIVE** counterparts, indicating that explicit location conditioning produces more spatially structured hidden states. More importantly, **LATENT** is competitive with and often outperforms **TEXT** and **HYBRID** across model families and cities, particularly in intermediate and later layers. This is notable because the probing setup naturally favors **TEXT** and **HYBRID**: during captioning training, both **leak** the exact latitude and longitude alongside their corresponding descriptions, providing a direct shortcut for learning text–coordinate associations. If geographic knowledge were organized merely through such co-occurrence, these representations should produce the most decodable states. Instead, **LATENT** achieves comparable or better coordinate recoverability without exposing raw coordinates as text, suggesting that continuous encodings encourage LLMs to organize geographic information structurally by forming a continuous latent spatial field.

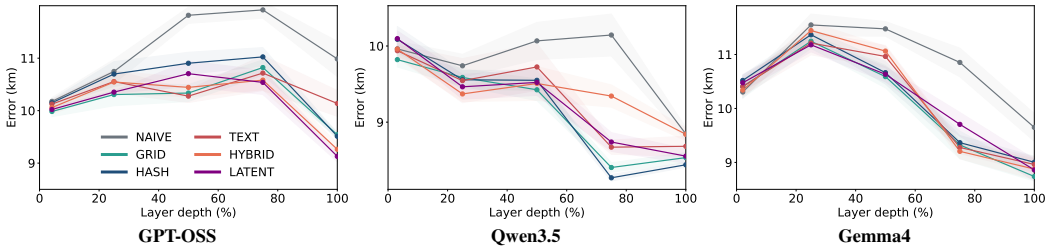


Figure 2: **Layer-wise geospatial probing across models on LA.** Lower error indicates that location is more explicitly encoded. Curves and shaded regions show means and standard deviations.

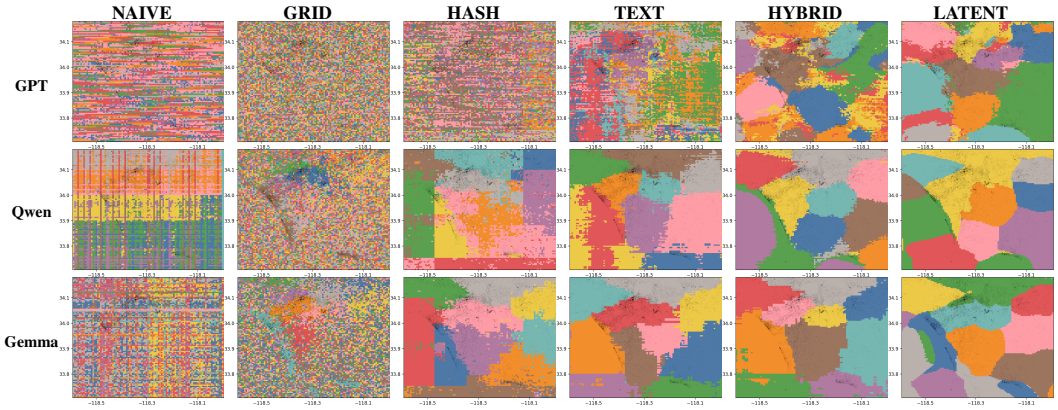


Figure 3: **Captioning model's hidden representation cluster maps for the Los Angeles dataset.** Rows correspond to the GPT, Qwen, and Gemma model families, while columns correspond to **NAIVE**, **GRID**, **HASH**, **TEXT**, **HYBRID**, and **LATENT** spatial representations.

We further cluster the hidden states of the generation token across query locations and map the assignments back to space. Details and London results are provided in Appendix A.8. As shown in Figure 3, **NAIVE** models exhibit little coherent spatial organization, with cluster assignments distributed almost randomly. Despite task-specific tuning, **GRID** remains spatially fragmented. The **HASH** and **TEXT** maps also exhibit abrupt transitions and numerous isolated regions. In contrast, **LATENT** produces more contiguous regions with clearer boundaries. This difference is especially evident for the text-only GPT-OSS model, where **TEXT** fails to organize hidden features coherently, whereas **LATENT** and **HYBRID** recover continuous regions reflecting geographic heterogeneity.

Together with the stronger distributional alignment observed in grounding, the probing and clustering results support our hypothesis that continuous spatial encoding promotes a coherent latent spatial field for organizing geographic knowledge and enables spatial generalization.

7 Conclusion

In this work, we conceptualize geospatial knowledge in LLMs as a latent spatial field and formulate geospatial awareness through two capabilities: memory, which recovers observed location–knowledge associations, and reasoning, which generalizes such associations to unobserved regions. Based on this formulation, we introduce spatially explicit LLMs that incorporate continuous location encodings into LLM training. Through comprehensive experiments on two newly constructed benchmarks, we demonstrate that continuous spatial representations enable models to generalize more effectively across geographic space and generate useful information that benefits downstream tasks, suggesting that textual and continuous representations are complementary. Our probing and clustering analyses further show that LLMs develop spatially organized internal representations of geographic knowledge. These findings support our hypothesis that LLMs can organize geographic knowledge as a latent spatial field rather than relying solely on textual location–knowledge correlations. Together, our findings position continuous spatial representations as a promising foundation for building more geospatially aware and generalizable language models.

AI USE STATEMENT

In this work, we used generative AI tools for implement methods, clean and reformat dataset, design or provide feedback on research methodology or experiments. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, assist with translation, support qualitative And thematic data analysis, interpret results. And generate synthetic data sets, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, are not applicable to this work. Additionally, we used generative AI tools for create or edit software code, draft parts of a research paper, brainstorming, edit a research paper to improve readability, propose a title or keywords for a research paper. We have reviewed all AI-assisted work. We checked LLM-generated code was verified and tested for correctness, LLM-polished paragraphs are carefully checked by all authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All the datasets involved in the paper are publicly available. The Appendix provides complete descriptions of the data sources, dataset construction, preprocessing, benchmark splits, model training, and evaluation procedures. The two benchmarks introduced in this work and the source code required to reproduce our experiments are included in the supplementary material. The datasets and code will be released publicly on GitHub upon acceptance, and the trained model checkpoints will be released through Hugging Face.

REFERENCES

Riyang Bao, Cheng Yang, Dazhou Yu, Zhexiang Tang, Gengchen Mai, and Liang Zhao. Spatial-agent: Agentic geo-spatial reasoning with scientific core concepts. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14896–14911, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.679. URL https://aclanthology.org/2026.acl-long.679/.

Chris Brunsdon, A. Stewart Fotheringham, and Martin E. Charlton. Geographically weighted regression: A method for exploring spatial nonstationarity. *Geographical Analysis*, 28(4):281–298, 1996. doi: 10.1111/j.1538-4632.1996.tb00936.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1538-4632.1996.tb00936.x.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 135062–135093. Curran Associates, Inc., 2024. doi: 10.52202/079017-4293. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f38cb4cf9a5eaa92b3cfa481832719c6-Paper-Conference.pdf.

Chen Chu and Cyrus Shahabi. Geo2vec: Shape- and distance-aware neural representation of geospatial entities. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(23):18985–18993, Mar. 2026. doi: 10.1609/aaai.v40i23.38970. URL https://ojs.aaai.org/index.php/AAAI/article/view/38970.

Chen Chu, Bita Azarijoo, Li Xiong, Khurram Shafique, and Cyrus Shahabi. From symbolic to geometric: Enabling spatial reasoning in large language models, 2026. URL https://arxiv.org/abs/2606.04381.

Murray Cox. Inside airbnb. https://insideairbnb.com/, 2026. Accessed: 2026-09-15.

Jian Cui, Zhiyuan Ren, Desheng Weng, Yongqi Zhao, Gong Wenbin, Yu Lei, and Zhenning Dong. ReaGeo: Reasoning-enhanced end-to-end geocoding with LLMs. In Maria Liakata, Viviane P.

Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 28012–28023, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1396. URL https://aclanthology.org/2026.findings-acl.1396/.

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*, KDD '25, pp. 591–602, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3736878. URL https://doi.org/10.1145/3711896.3736878.

Jie Feng, Jun Zhang, Tianhui Liu, Xin Zhang, Tianjian Ouyang, Junbo Yan, Yuwei Du, Siqi Guo, and Yong Li. Citybench: Evaluating the capabilities of large language models for urban tasks. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 5413–5424, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737375. URL https://doi.org/10.1145/3711896.3737375.

Wes Gurnee and Max Tegmark. Language models represent space and time. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 2483–2503, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/0a6059857ae5c82ea9726ee9282a7145-Paper-Conference.pdf.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yuhan Ji, Song Gao, Ying Nie, Ivan Majić, and Krzysztof Janowicz. Foundation models for geospatial reasoning: assessing the capabilities of large language models in understanding geometries and topological spatial relations. *International Journal of Geographical Information Science*, 39(9):1866–1903, 2025. doi: 10.1080/13658816.2025.2511227. URL https://doi.org/10.1080/13658816.2025.2511227.

Konstantin Klemmer, Esther Rolf, Caleb Robinson, Lester Mackey, and Marc Rußwurm. Satclip: global, general-purpose location embeddings with satellite imagery. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i4.32457. URL https://doi.org/10.1609/aaai.v39i4.32457.

Fangjun Li, David C. Hogg, and Anthony G. Cohn. Advancing spatial reasoning in large language models: an in-depth evaluation and enhancement using the stepgame benchmark. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i17.29811. URL https://doi.org/10.1609/aaai.v38i17.29811.

Siyu Li, Toan Tran, Haowen Lin, John Krumm, Cyrus Shahabi, Lingyi Zhao, Khurram Shafique, and Li Xiong. Geo-llama: Leveraging llms for human mobility trajectory generation with spatiotemporal constraints, 2025. URL https://arxiv.org/abs/2408.13918.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Dongyi Lv, Qiuyu Ding, Heng-Da Xu, Zhaoxu Sun, Zhi Wang, Feng Xiong, and Mu Xu. Reasoning over space: Enabling geographic reasoning for LLM-based generative next POI recommendation.

In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7322–7336, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.332. URL https://aclanthology.org/2026.acl-long.332/.

Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. Multi-scale representation learning for spatial feature distributions using grid cells, 2020. URL https://arxiv.org/abs/2003.00824.

Gengchen Mai, Ni Lao, Yutong He, Jiaming Song, and Stefano Ermon. CSP: Self-supervised contrastive spatial pre-training for geospatial-visual representations. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23498–23515. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/mai23a.html.

Rohin Manvi, Samar Khanna, Marshall Burke, David Lobell, and Stefano Ermon. Large language models are geographically biased. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David B. Lobell, and Stefano Ermon. GeoLLM: Extracting geospatial knowledge from large language models. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=TqL2xBwXP3.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Maria Despoina Siampou, Jialiang Li, John Krumm, Cyrus Shahabi, and Hua Lu. Poly2Vec: Polymorphic Fourier-based encoding of geospatial objects for GeoAI applications. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55511–55532. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/siampou25a.html.

Yu Wang, Yi Wang, Rui Dai, Yujie Wang, Kaikui Liu, Xiangxiang Chu, and Yansheng Li. Urban socio-semantic segmentation with vision-language reasoning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 74495–74518, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/792845ddfe4047d7066348e52e46b74d-Paper-Conference.pdf.

Meiliu Wu, Qunying Huang, and Song Gao. Advancing vision-language models with spatial-context prompt tuning: a case study in geoai-empowered image geo-localization. *Journal of Location Based Services*, 20(1-2):24–44, 2026. doi: 10.1080/17489725.2025.2599302. URL https://doi.org/10.1080/17489725.2025.2599302.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

A Appendix

A.1 Dataset Construction

Data source. In this research, we exploited a new source of geospatial-textual coupled dataset, the Airbnb listings dataset (Cox, 2026). Each listing is associated with geographic coordinates and a

human-written description provided by its host. These descriptions not only introduce the property, but also contain detailed geographic context, including nearby landmarks, neighborhood characteristics, transportation facilities, walking distances, and travel times. They therefore provide naturally occurring associations between geographic locations and textual descriptions of their surroundings. We independently process listings from Los Angeles and London to construct the two datasets used in our experiments. The following example illustrates the geographic information contained in a typical raw listing description.

Example raw Airbnb description

It's all about location with this stunning 1 Bedroom at a 5 min walk to the Sand! Be 3 blocks away from the nearest Beach. Be steps away from the popular Main Street Santa Monica and all the best restaurants & bars the neighborhood has to offer. The area is walkable with a close-by convenience & liquor store, produce market, bike rental shops, Urth Caffé, and ocean-view parks, to name a few. Venice is about 1 mile away, a straight shot down on Main St. Walk, bike, surf, skate and live like a local!

Geographic-description extraction. Raw listing descriptions also contain substantial information unrelated to location, such as room layouts, furniture, and property amenities. We therefore use `Llama-3.3-70B-Instruct`[4] to extract only explicit geographic relations. The model is instructed to preserve stated place names, transportation modes; exclude property-internal information; and avoid introducing facts not present in the original description.

Prompt for geographic-description extraction

System prompt:
You are a helpful assistant that extracts location descriptions from Airbnb property overviews.
Focus solely on identifying phrases that describe the property's location, such as proximity to landmarks, transportation, or neighborhood features.
Ignore any information about the property's amenities, host details, or other non-location content.
Your output should be a JSON object with a single key `"locations"`, which contains a list of concise location phrases. For example:

```
{"locations": ["near the beach",
               "10-minute walk to the metro",
               "close to the London Eye",
               "in Kennington"]}
```

Each phrase should be self-contained and directly related to the property's location.
Do not include any explanations, reasoning, or additional text in your response—only the JSON object.
If no location information is present in the overview, return:

```
{"locations": []}
```

User prompt:

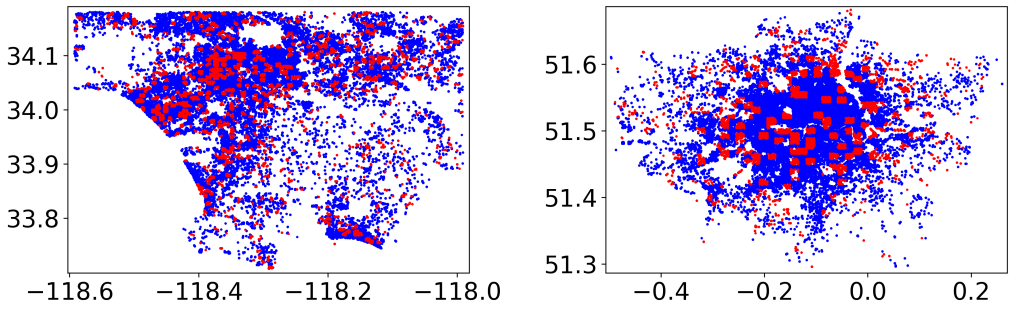
```
Airbnb Overview: '<DESCRIPTION>'
Extract location phrases and return the JSON object as
specified.
```

The structured output is parsed and converted into independent, self-contained descriptions. Empty or malformed outputs are removed, and repeated descriptions associated with the same listing are deduplicated. The example above produces descriptions such as:

Extracted self-contained geographic descriptions

- Five-minute walk from the beach.
- Three blocks from the nearest beach.
- Steps away from Main Street in Santa Monica.
- Restaurants and bars are within walking distance.
- A convenience store is nearby.
- Venice is approximately one mile away via Main Street.

[4] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct



(a) Los Angeles (b) London

Figure 4: **Spatial distribution of the training and test samples.** Training samples colored in blue, testing samples colored in red.

For a listing at location $\mathbf{x}_i$, each extracted description $y_{i,j}$ forms an independent geo-text pair $(\mathbf{x}_i, y_{i,j})$. This procedure transforms the original long-form listings into concise location–description associations suitable for geospatial captioning and grounding. Importantly, the extraction model is used only to filter and restructure information explicitly contained in each listing; it is not asked to generate additional geographic knowledge.

Spatial train–test split. After preprocessing, the Los Angeles and London datasets contain 226,250 and 220,183 independent location–description pairs, respectively. To evaluate whether an LLM can infer knowledge in spatially unobserved regions, we use a spatial block split rather than randomly dividing individual samples. Specifically, we partition each geographic domain into an $N \times N$ grid. We randomly select 20% of the grid cells as held-out test regions and use the remaining 80% for training. Each location–description pair is assigned to the training or test set according to the cell containing its coordinate. Consequently, this procedure yields spatially separated training and test sets, with the training data exhibiting substantial heterogeneity and sparsity in its geographic distribution.

To evaluate reasoning across spatial gaps of different scales, we set $N_{\text{LA}} = 100$ and $N_{\text{London}} = 50$, corresponding to cell sizes of approximately 555.3×526.6 meters and $1{,}119.3 \times 1{,}260.1$ meters, respectively. The coarser London partition creates larger data-sparse regions, requiring models to generalize across wider spatial gaps. The resulting Los Angeles split contains 179,801 training samples and 46,449 test samples, while the London split contains 177,511 training samples and 42,672 test samples. Figure 4 visualizes these distributions.

A.2 Geospatial Captioning and Grounding Benchmarks

Based on the spatially partitioned datasets introduced above, we construct a geospatial captioning benchmark and a geospatial grounding benchmark. Following the notation in Section 3, we set $\mathcal{V} = \mathcal{Y}$ and write the processed dataset as

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}. \tag{7}$$

where $\mathbf{x}_i \in \Omega$ is the listing location and $y_i \in \mathcal{Y}$ is an extracted location description. The benchmarks use the same underlying pairs and spatial split but evaluate different directions of geospatial understanding.

A.2.1 Geospatial Captioning

Geospatial captioning maps a location $\mathbf{x}_i$ to a natural-language description yi. A captioning model estimates $p_\psi(y \mid \mathbf{x})$ and generates a description for the queried location. The model is trained using $\mathcal{D}_{\text{train}}$ and evaluated at locations in the testing regions $\mathcal{H}$. Because no location–description pairs from $\mathcal{H}$ are available during training, performance in these regions measures whether the model can infer geographic knowledge beyond its observed spatial support.

A.2.2 POINT-LEVEL GEOSPATIAL GROUNDING

Point-level grounding reverses the captioning direction. Given an individual description y_i, the grounding model estimates $q_\phi(\mathbf{x} \mid y_i)$ over the geographic domain and retrieves the associated listing location $\mathbf{x}_i$. Each query therefore has one point-level target. This benchmark evaluates fine-grained text-to-location correspondence by measuring whether $\mathbf{x}_i$ falls within the highest-scoring 1% or 5% of spatial candidates.

A.2.3 REGION-LEVEL GEOSPATIAL GROUNDING

Many location descriptions refer to an area rather than a unique point. Moreover, free-text descriptions are highly redundant: phrases such as 'near the beach" and 'close to the beach" may express the same geographic relation. Semantic similarity alone, however, does not imply geographic specificity. For example, "near restaurants" may be semantically coherent but distributed throughout an entire city. We therefore construct region-level grounding labels in two stages: semantic clustering groups paraphrastic descriptions, while spatial filtering retains only clusters whose occurrences form a localized region.

Semantic clustering of location descriptions. Let $\mathcal{U} \subseteq \mathcal{Y}$ be the set of unique descriptions after exact duplicates have been collapsed, and let $n(u)$ denote the number of samples containing $u \in \mathcal{U}$. We encode each unique description using a pretrained sentence encoder E_{text}, specifically Sentence-BERT with `all-MiniLM-L6-v2`[5], and normalize its representation as

$$\mathbf{z}(u) = \frac{E_{\text{text}}(u)}{|E_{\text{text}}(u)|_2}. \tag{8}$$

Cosine similarity s and distance d are then defined as

$$s(u, u') = \mathbf{z}(u)^\top \mathbf{z}(u'), \qquad d(u, u') = 1 - s(u, u'). \tag{9}$$

Descriptions in $\mathcal{U}$ are grouped using average-linkage agglomerative clustering with cosine-similarity threshold $\tau = 0.95$, or equivalently cosine-distance threshold $1 - \tau$. The number of clusters is determined by this threshold rather than specified in advance.

Assignment of infrequent descriptions. Let $C_1, \ldots, C_J$ denote the clusters constructed from $\mathcal{U}$. For each cluster C_j, we compute the normalized centroid

$$\boldsymbol{\mu}_j = \frac{\sum_{u \in C_j} \mathbf{z}(u)}{\left| \sum_{u \in C_j} \mathbf{z}(u) \right|_2}. \tag{10}$$

For each $u \in \mathcal{U}$, its closest centroid is

$$j^\star(u) = \arg\max_{1 \le j \le J} \mathbf{z}(u)^\top \boldsymbol{\mu}_j. \tag{11}$$

The description is assigned to this cluster only when it satisfies the same similarity threshold:

$$\text{Assign}(u) = \begin{cases} C_{j^\star(u)}, & \text{if } \mathbf{z}(u)^\top \boldsymbol{\mu}_{j^\star(u)} \ge \tau, \\ \{u\}, & \text{otherwise.} \end{cases} \tag{12}$$

An unmatched description therefore forms a singleton cluster. Let $\mathcal{C}$ denote the resulting collection of semantic clusters. Each cluster $C \in \mathcal{C}$ is assigned a canonical description equal to its most frequent member:

$$y_C = \arg\max_{u \in C} n(u). \tag{13}$$

[5] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

Spatial compactness filtering. As discussed above, descriptions such as "near restaurants" may occur throughout an entire city and therefore lack a distinguishable geographic region for grounding. To remove such spatially diffuse queries, we apply a spatial compactness filter that measures how tightly the locations associated with each grounding label are clustered.

For each cluster $C \in \mathcal{C}$, define the indices of its associated records as

$$\mathcal{I}_C = \{i \in \{1, \dots, N\} : y_i \in C\}, \qquad n_C = |\mathcal{I}_C|. \tag{14}$$

We first retain only clusters satisfying

$$n_{\min} \le n_C \le n_{\max}. \tag{15}$$

where $n_{\min} = 40$ and $n_{\max} = 500$. The lower bound removes clusters with insufficient observations to define a reliable region, while the upper bound removes broad, city-level descriptions.

For each remaining cluster, we compute the haversine distance d_{hav} from every location to its nearest distinct neighbor within the same cluster. For $i \in \mathcal{I}_C$, this distance is

$$d_C^{\text{NN}}(i) = \min_{j \in \mathcal{I}_C \; j \neq i} d_{\text{hav}}(\mathbf{x}_i, \mathbf{x}_j), \tag{16}$$

where the haversine distance is computed using Earth radius $R = 6{,}371\text{km}$. We summarize the within-cluster distances as

$$\bar{d}_C = \frac{1}{n_C} \sum_{i \in \mathcal{I}_C} d_C^{\text{NN}}(i), \qquad \widetilde{d}_C = \operatorname*{median}_{i \in \mathcal{I}_C} d_C^{\text{NN}}(i). \tag{17}$$

A cluster is retained only when

$$\bar{d}_C \le d_{\text{mean}} \qquad \text{and} \qquad \widetilde{d}_C \le d_{\text{med}}, \tag{18}$$

where $d_{\text{mean}} = 500\text{m}$ and $d_{\text{med}} = 300\text{m}$.

Final region-level labels. Each surviving cluster defines one region-level query. Its label consists of the canonical description y_C and the locations of all records assigned to the cluster:

$$(y_C, \{\mathbf{x}_i : i \in \mathcal{I}_C\}). \tag{19}$$

After this process, 334 and 625 region-level grounding queries remain in the Los Angeles and London datasets, respectively. On average, each query is grounded by 52.61 unique locations in Los Angeles and 41.77 in London. A few sample queries and their corresponding samples are presented in Figure 5 and Figure 6.

Together, these benchmarks evaluate complementary aspects of geospatial understanding. Captioning tests inference from location to geographic knowledge, point-level grounding evaluates precise text-to-location retrieval, and region-level grounding evaluates the recovery of localized spatial distributions. Their shared spatial split also allows captions generated in $\mathcal{H}$ to augment a separately trained grounding model, providing an extrinsic evaluation of whether the inferred descriptions contain useful geographic knowledge.

Two benchmarks are included in the supplementary material and will be publicly released.

A.3 Evaluation Metrics

A.3.1 Metrics for Geospatial Captioning

Geospatial captioning is inherently open-ended: the same location can be correctly described through different landmarks, neighborhoods, transportation options, or spatial relations. The available human-written descriptions should therefore be treated as valid references rather than as unique ground-truth answers. We consequently evaluate captioning through many-to-many alignment between the generated and reference descriptions associated with each location. For each evaluated location $\mathbf{x}_i$, let

$$\widehat{\mathcal{Y}}_i = \{\widehat{y}_{i,r}\}_{r=1}^{N_g}, \qquad \mathcal{Y}_i = \{y_{i,j}\}_{j=1}^{M_i}, \tag{20}$$



Figure 5: **Examples from the Los Angeles geospatial grounding benchmark.** Orange points denote target locations and gray points denote the underlying data distribution; labels give the ground-truth descriptions.

denote its generated and reference description sets, respectively. We generate $N_g = 4$ descriptions for every location, while M_i is the number of available references for $\mathbf{x}_i$. To augment the grounding model, each generated description is paired with its corresponding location $\mathbf{x}_i$. Using the same augmentation locations across methods isolates performance differences arising from the generated descriptions rather than from location selection.

Let $s(\hat{y}, y)$ denote a pairwise similarity function between a generated description and a reference. We instantiate s separately using BERTScore (Zhang* et al., 2020) (`sentence-transformers/all-MiniLM-L6-v2`[6]) F1 and ROUGE-L (Lin, 2004) F1, two commonly used metrics in open-ended text generation. These metrics capture complementary aspects of caption quality. BERTScore is tolerant of semantic paraphrases, whereas ROUGE-L measures lexical and sequence-level agreement. For example, 'near the beach' and 'close to the coast' may express similar geographic meaning despite limited token overlap, while exact landmark names, neighborhood names, and distance expressions remain important lexical evidence. For each metric,

[6] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`



Figure 6: **Examples from the London geospatial grounding benchmark.** Red points denote target locations and gray points denote the underlying data distribution; labels give the ground-truth descriptions.

we construct the similarity matrix

$$S_{i,r,j} = s\left(\widehat{y}_{i,r}, y_{i,j}\right). \tag{21}$$

The set-level precision measures whether each generated description is supported by at least one reference. Each generated description is matched to its most similar reference, and the resulting similarities are averaged:

$$P_i = \frac{1}{N_g} \sum_{r=1}^{N_g} \max_{1 \le j \le M_i} S_{i,r,j}. \tag{22}$$

Conversely, set-level recall measures how well the generated descriptions cover the available references. Each reference is matched to its most similar generated description:

$$R_i = \frac{1}{M_i} \sum_{j=1}^{M_i} \max_{1 \le r \le N_g} S_{i,r,j}. \tag{23}$$

Their harmonic mean gives the alignment F1 score for location $\mathbf{x}_i$:

$$F_{1,i} = \frac{2P_iR_i}{P_i + R_i + \varepsilon}, \tag{24}$$

Algorithm 1 Construction of semantically and spatially grounded region labels

Require: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$; encoder E_{text}; head size K; similarity threshold τ; support bounds $[n_{\min}, n_{\max}]$; spatial thresholds d_{mean} and d_{med}

Form the unique-description set $\mathcal{U}$ with frequencies $n(u)$; compute normalized embeddings $\mathbf{z}(u)$
Split $\mathcal{U}$ into the frequent set $\mathcal{U}_{\text{freq}}$ containing the top-K descriptions and $\mathcal{U}_{\text{rare}} = \mathcal{U} \setminus \mathcal{U}_{\text{freq}}$
Cluster $\mathcal{U}_{\text{freq}}$ using average-linkage agglomerative clustering with cosine-distance threshold $1 - \tau$, producing $\{C_1, \ldots, C_J\}$
Compute the normalized centroid $\boldsymbol{\mu}_j$ of each cluster C_j
for $u \in \mathcal{U}_{\text{rare}}$ **do**
    Find $j^\star(u) = \arg\max_j \mathbf{z}(u)^\top \boldsymbol{\mu}_j$
    Assign u to $C_{j^\star(u)}$ if $\mathbf{z}(u)^\top \boldsymbol{\mu}_{j^\star(u)} \geq \tau$; otherwise, create a singleton cluster
end for
Let $\mathcal{C}$ denote the resulting semantic clusters and set $y_C = \arg\max_{u \in C} n(u)$ for each $C \in \mathcal{C}$
Define $\mathcal{I}_C = \{i \in \{1, \ldots, N\} : y_i \in C\}$ and $n_C = |\mathcal{I}_C|$
Discard clusters for which $n_C \notin [n_{\min}, n_{\max}]$
For each remaining cluster, compute the within-cluster haversine nearest-neighbor distances $d_C^{\text{NN}}(i)$
Retain C if $\overline{d}_C \leq d_{\text{mean}}$ and $\widetilde{d}_C \leq d_{\text{med}}$
return Retained labels $\{(y_C, \{\mathbf{x}_i : i \in \mathcal{I}_C\})\}$

where $\varepsilon = 10^{-12}$ ensures numerical stability.

Let $\mathcal{I}_{\text{eval}}$ denote the indices of the evaluated locations. We first compute the three scores independently for each location and then average them across the evaluation set:

$$\overline{P} = \frac{1}{|\mathcal{I}_{\text{eval}}|} \sum_{i \in \mathcal{I}_{\text{eval}}} P_i, \qquad \overline{R} = \frac{1}{|\mathcal{I}_{\text{eval}}|} \sum_{i \in \mathcal{I}_{\text{eval}}} R_i, \tag{25}$$

and

$$\overline{F}_1 = \frac{1}{|\mathcal{I}_{\text{eval}}|} \sum_{i \in \mathcal{I}_{\text{eval}}} F_{1,i}. \tag{26}$$

This aggregation is applied independently to BERTScore F1 and ROUGE-L F1, producing semantic and lexical set-alignment scores, respectively. The many-to-many evaluation rewards generated descriptions that are supported by the references while also measuring how well the generated set covers their diversity. Nevertheless, it remains a reference-based intrinsic evaluation and cannot by itself establish whether the generated descriptions contain spatially useful knowledge, motivating the extrinsic grounding evaluation described next.

A.3.2 Metrics for Geospatial Grounding

Predefined Grid Over Space For each city, longitude and latitude are independently rescaled to $[-1, 1]$ using the bounding box of the coordinates. We then divide the whole region into a uniform grid $\mathcal{G}$ with resolution $M \times M$. We use $M = 100$, giving $|\mathcal{G}| = 10{,}000$ grid cells.

For each grid cell $\mathbf{x} \in \mathcal{G}$ and description $y \in \mathcal{Y}$, the grounding model produces a location representation $E_\phi^{\text{loc}}(\mathbf{x})$ and a text representation $E_\phi^{\text{text}}(y)$. Their cosine similarity is

$$s_\phi(\mathbf{x}, y) = \frac{\left\langle E_\phi^{\text{text}}(y), E_\phi^{\text{loc}}(\mathbf{x}) \right\rangle}{\left\| E_\phi^{\text{text}}(y) \right\|_2 \left\| E_\phi^{\text{loc}}(\mathbf{x}) \right\|_2}. \tag{27}$$

For a coverage ratio $\alpha \in (0, 1]$, let $k_\alpha = \lceil \alpha |\mathcal{G}| \rceil$, and let $s_\phi^{[k_\alpha]}(y)$ denote the k_α-th largest value in $\{s_\phi(\mathbf{x}, y) : \mathbf{x} \in \mathcal{G}\}$. The predicted high-similarity region is

$$\mathcal{R}_\alpha(y) = \left\{ \mathbf{x} \in \mathcal{G} : s_\phi(\mathbf{x}, y) \geq s_\phi^{[k_\alpha]}(y) \right\}. \tag{28}$$

Ties at the threshold are broken deterministically so that $|\mathcal{R}_\alpha(y)| = k_\alpha$. We denote by $\pi_{\mathcal{G}}(\mathbf{x})$ the grid cell nearest to a continuous coordinate $\mathbf{x}$. All reported results are means over five independent training runs, with standard deviations reported after $\pm$.

Point-Level Grounding Point-level grounding evaluates whether the location paired with a description falls inside its predicted high-similarity region. Let $\mathcal{I}_{\text{test}}$ denote the indices of the in-bounds test pairs $(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{h}}$. The hit rate at coverage α is

$$\text{HitRate}\,@\alpha = \frac{1}{|\mathcal{I}_{\text{test}}|} \sum_{i \in \mathcal{I}_{\text{test}}} \mathbf{1}\left[\pi_{\mathcal{G}}(\mathbf{x}_i) \in \mathcal{R}_\alpha(y_i)\right]. \tag{29}$$

We report $\text{HitRate}\,@1\%$ and $\text{HitRate}\,@5\%$, corresponding to $\alpha = 0.01$ and $\alpha = 0.05$. Higher values indicate better point-level grounding.

Region-Level Grounding For a region-level grounding label $C \in \mathcal{C}$, let y_C be its canonical description and let $\mathcal{I}_C$ index its $n_C = |\mathcal{I}_C|$ support locations. Region-level grounding is evaluated using both coverage rate and alignment between the complete predicted and empirical spatial distributions.

Coverage rate The coverage score measures the fraction of support locations captured by the predicted high-similarity region, averaged equally across grounding labels:

$$\text{Coverage}\,@\alpha = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} \frac{1}{n_C} \sum_{i \in \mathcal{I}_C} \mathbf{1}\left[\pi_{\mathcal{G}}(\mathbf{x}_i) \in \mathcal{R}_\alpha(y_C)\right]. \tag{30}$$

We use $\alpha = 0.01$ by default. Higher values indicate that the predicted region covers a larger proportion of the locations associated with each description.

Empirical Target Distribution We construct a discrete target distribution p_C over $\mathcal{G}$ using isotropic Gaussian kernel density estimation. With bandwidth $\sigma = 0.05$ in normalized coordinates,

$$\widetilde{p}_C(\mathbf{x}) = \sum_{i \in \mathcal{I}_C} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|_2^2}{2\sigma^2}\right), \qquad \mathbf{x} \in \mathcal{G}, \tag{31}$$

and

$$p_C(\mathbf{x}) = \frac{\widetilde{p}_C(\mathbf{x})}{\sum_{\mathbf{x}' \in \mathcal{G}} \widetilde{p}_C(\mathbf{x}')}. \tag{32}$$

Predicted Distribution To obtain the discrete form of the grounding distribution $q_\phi(\mathbf{x} \mid y_C)$ defined in the main text, we floor the cosine-similarity field at a small constant $\varepsilon > 0$ and normalize it over $\mathcal{G}$:

$$\widetilde{q}_{\phi,C}(\mathbf{x}) = \max\left\{s_\phi(\mathbf{x}, y_C), \varepsilon\right\}, \tag{33}$$

$$q_{\phi,C}(\mathbf{x}) = \frac{\widetilde{q}_{\phi,C}(\mathbf{x})}{\sum_{\mathbf{x}' \in \mathcal{G}} \widetilde{q}_{\phi,C}(\mathbf{x}')}. \tag{34}$$

This normalization preserves relative similarity across the complete geographic domain rather than only within $\mathcal{R}_\alpha(y_C)$.

Jensen–Shannon Divergence Given the target distribution p_C and predicted distribution $q_{\phi,C}$, we use Jensen–Shannon divergence (JSD) to measure their cell-wise distributional agreement. JSD is symmetric, bounded, and well defined for distributions with partially non-overlapping support, making it suitable for comparing spatial probability fields. Let

$$m_C(\mathbf{x}) = \frac{1}{2}\left(p_C(\mathbf{x}) + q_{\phi,C}(\mathbf{x})\right). \tag{35}$$

For discrete distributions a and b over $\mathcal{G}$, the Kullback–Leibler divergence is

$$\mathrm{KL}(a\|b) = \sum_{\mathbf{x}\in\mathcal{G}} a(\mathbf{x}) \log \frac{a(\mathbf{x})}{b(\mathbf{x})}. \tag{36}$$

The Jensen–Shannon divergence for cluster C is

$$\mathrm{JSD}_C = \frac{1}{2}\,\mathrm{KL}(p_C\|m_C) + \frac{1}{2}\,\mathrm{KL}(q_{\phi,C}\|m_C). \tag{37}$$

Using the natural logarithm, JSD is bounded by

$$0 \le \mathrm{JSD}_C \le \log 2. \tag{38}$$

A value of zero indicates identical distributions, while larger values indicate weaker overlap between their probability mass. We report the mean across region-level labels:

$$\overline{\mathrm{JSD}} = \frac{1}{|\mathcal{C}|}\sum_{C\in\mathcal{C}} \mathrm{JSD}_C, \qquad \widetilde{\mathrm{JSD}} = \operatorname*{median}_{C\in\mathcal{C}} \mathrm{JSD}_C\,. \tag{39}$$

JSD captures differences in the shape and concentration of the two distributions but does not consider the distance between mismatched regions. For example, two predictions displaced by different geographic distances may receive similar JSD values if their overlap with the target is comparable.

Rasterized Wasserstein Distance We additionally report Wasserstein distance as a complement to JSD. While JSD primarily measures probability overlap, Wasserstein distance measures how far probability mass must be shifted to align the predicted and target distributions. To match our evaluation implementation, we enumerate the cells of $\mathcal{G}$ in row-major raster order as $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(|\mathcal{G}|)}$. For any distribution a over $\mathcal{G}$, its cumulative mass along this ordering is

$$F_a(k) = \sum_{r=1}^{k} a\left(\mathbf{x}^{(r)}\right). \tag{40}$$

The rasterized Wasserstein-1 distance for cluster C is

$$W_{1,C}^{\mathrm{raster}} = \sum_{k=1}^{|\mathcal{G}|} \left|F_{p_C}(k) - F_{q_{\phi,C}}(k)\right|. \tag{41}$$

We report its mean across region-level labels:

$$\overline{W}_1^{\mathrm{raster}} = \frac{1}{|\mathcal{C}|}\sum_{C\in\mathcal{C}} W_{1,C}^{\mathrm{raster}}, \qquad \widetilde{W}_1^{\mathrm{raster}} = \operatorname*{median}_{C\in\mathcal{C}} W_{1,C}^{\mathrm{raster}}. \tag{42}$$

Lower values indicate that the predicted probability mass is more closely aligned with the target field along the raster ordering.

Together, the two metrics provide complementary information: JSD evaluates whether probability mass is assigned to the same grid cells with a similar distributional shape, whereas rasterized Wasserstein distance measures the extent of distribution between the complete fields. A model must perform well on both metrics to achieve strong overlap without substantially shifting the predicted spatial distribution.

Metric Complementarity $\mathrm{HitRate}\,@\alpha$ and $\mathrm{Coverage}\,@\alpha$ evaluate whether annotated locations occur within a binary high-similarity region. In contrast, JSD and W_1^{raster} compare how probability mass is distributed across the complete geographic domain. A model may therefore achieve high coverage while still obtaining poor distributional alignment.

A.4 Geospatial Captioning Model

We fine-tune each LLM to approximate the conditional distribution $p_\psi(y \mid \mathbf{x})$, where $\mathbf{x}$ is a geographic location and y is a natural-language description of that location. To ensure a controlled comparison, all model–representation combinations use identical instruction content while retaining each model's native chat template and special tokens.

System: You are a helpful assistant for geo-localization. Always use only the information contained in the provided representation to generate the description.

User: You are at location `<GEO>`. Generate a detailed description of the location.

Table 5: Training configurations.

Configuration	Value
Base model	Frozen
Token embeddings	Frozen
Batch size	16
Gradient accumulation	12
LoRA learning rate	1×10^{-5}
Geo-module learning rate	1×10^{-4}
Weight decay	0
Learning-rate scheduler	Cosine
Warmup ratio	0.01
Encoder-only epochs	1
Maximum gradient norm	1.0
Samples per epoch	$10,000$
Training epochs	100

The `<GEO>` placeholder is replaced by the corresponding location representation. For **GRID** (Li et al., 2025), we use the same spatial partition as the train–test split, with $N_{\mathrm{LA}} = 100$ and $N_{\mathrm{London}} = 50$, and replace `<GEO>` with the identifier of the cell containing the location, such as `grid_ID`. For **HASH** (Cui et al., 2026), Geohash, as one of the most typical geospatial indexing method, converts geographic coordinates into a short Base32 string by using a hierarchical grid structure that repeatedly bisects latitude and longitude ranges into smaller sub-intervals and assigns binary bits based on the coordinate's location, producing strings such as `gcpugf00y`. For **TEXT** (Ji et al., 2025), the coordinate is provided directly as a textual string, we use six decimals for the representation, such as `(34.062403, -118.372775)`. For **LATENT**, the placeholder is replaced in the input embedding sequence by the continuous spatial representation $\mathbf{z_x}$. For **HYBRID**, the textual coordinate and continuous spatial representation are provided together as `(34.062403, -118.372775)` $\mathbf{z_x}$. The latent representation is inserted through the model's input embedding interface rather than converted into a textual token.

Given a target description tokenized as $y = (y_1, \ldots, y_T)$ and the representation-dependent prompt $c(\mathbf{x})$, we optimize the standard supervised fine-tuning objective

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{t=1}^{T} \log p_\psi \left(y_t \mid c(\mathbf{x}), y_{<t} \right). \tag{43}$$

The pretrained LLM parameters remain frozen, and the model is adapted using LoRA (Hu et al., 2022). The original token embeddings are also frozen. We fine-tune each backbone while freezing its pretrained weights and applying LoRA to the same attention projections, with rank $r = 64$, scaling $\alpha = 128$, and dropout 0.05. For the dense models, adapters are also applied to the SwiGLU feed-forward projections implemented as standard linear layers. This configuration yields 262.28M trainable LLM parameters for Gemma-4-12B and 116.39M for Qwen3.5-9B. Qwen's linear-attention blocks remain unchanged. GPT-OSS uses a mixture-of-experts architecture. To match Qwen's trainable-parameter count, adapters are instantiated for all 32 experts, we assign the expert adapters rank $r_{\mathrm{MoE}} = 8$ and scaling $\alpha_{\mathrm{MoE}} = 16$, preserving the update ratio $\alpha_{\mathrm{MoE}}/r_{\mathrm{MoE}} = \alpha/r = 2$. Each expert's fused gate–up and down projections receives an adapter, while the router remains frozen and expert dropout is set to 0. The resulting GPT-OSS adapter contains 120.32M trainable parameters, comparable to Qwen.

For **LATENT** and **HYBRID**, the geospatial encoder and projection modules are trained from scratch and therefore use a learning rate ten times larger than that of the LoRA parameters. We first perform one encoder-only epoch to initialize the geospatial modules before jointly optimizing them with the LoRA adapters.

At inference time, we generate four descriptions for each location using temperature 0.8, top-k sampling with $k = 30$, and nucleus sampling with $p = 0.95$. These four descriptions form the generated set $\widehat{\mathcal{Y}}_i$ used by the captioning evaluation described in Appendix A.3.1.

A.5 Distance-Stratified Evaluation

The aggregate captioning scores defined in Appendix A.3.1 may obscure how geospatial reasoning changes with a test location's distance from the observed training data. We therefore stratify locations in unobserved regions by their distance to $\mathcal{X}_{\mathrm{obs}}$ and compare spatial generalization ability of different methods within each distance range, separately for each city and LLM backbone.

Distance to Observed Locations. For this analysis, we restrict $\mathcal{I}_{\mathrm{eval}}$ to unique test locations evaluated by every method, ensuring a paired comparison across representations. For each $\mathbf{x}_i \in \mathcal{H}$, we

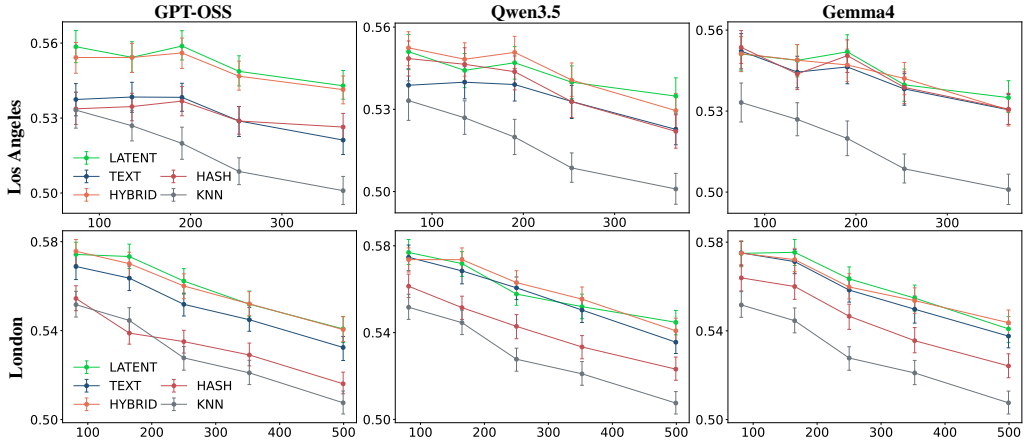


Figure 7: **Distance-stratified geospatial captioning performance.** Rows correspond to Los Angeles and London, and columns correspond to the GPT-OSS, Qwen3.5, and Gemma4 backbones. The horizontal axis reports the median nearest-observation distance d_i, in meters, within each equal-count distance bin B_b. The vertical axis reports $\overline{F}^m_{1,b}$, the mean per-location BERTScore F1 achieved by representation method m within bin B_b. Curves denote representation methods, and error bars indicate 95% bootstrap confidence intervals.

calculate its distance to the observed training data in meters, to the nearest location in $\mathcal{X}_{\text{obs}}$:

$$d_i = \min_{\mathbf{x}' \in \mathcal{X}_{\text{obs}}} \text{hav}\left(\mathbf{x}_i, \mathbf{x}'\right). \tag{44}$$

Here, $\text{hav}(\cdot,\cdot)$ denotes the haversine distance. Larger d_i indicates that the model must bridge a wider spatial gap from the observed training locations.

Per-Location Scores. For each representation method m, let $F^m_{1,i}$ denote the per-location many-to-many alignment score computed from $\widehat{\mathcal{Y}}^m_i$ and $\mathcal{Y}_i$ using the procedure defined in Appendix A.3.1. We perform the distance-stratified analysis independently for BERTScore F1.

Distance Binning and Paired Contrasts. We divide $\mathcal{I}_{\text{eval}}$ into $K = 5$ equal-count bins $\{B_b\}_{b=1}^K$ according to the quintiles of $\{d_i : i \in \mathcal{I}_{\text{eval}}\}$. Equal-count binning avoids sparsely populated long-distance intervals because the distribution of d_i is right-skewed. For representation method m, the mean score within bin B_b is

$$\overline{F}^m_{1,b} = \frac{1}{|B_b|} \sum_{i \in B_b} F^m_{1,i}. \tag{45}$$

$\overline{F}^m_{1,b}$ is then plotted against the median value of d_i within each bin.

Uncertainty. We construct 95% percentile bootstrap confidence intervals by resampling locations with replacement within each bin using $N_{\text{boot}} = 1000$ replicates. The interval bounds are the 2.5th and 97.5th percentiles of the resulting bootstrap means. These intervals capture variation from sampling test locations in unobserved regions, but not randomness from decoding or from the fixed $\texttt{k-NN}$ sampling procedure.

Results To interpret Figure 7, we focus on the rate at which performance decreases as the distance from observed training locations increases. A flatter curve, corresponding to a smaller negative slope, indicates stronger spatial generalization because the model maintains higher captioning accuracy across wider spatial gaps. For visual clarity, we omit **GRID** because of its consistently poor performance. Across nearly all model–city settings, the plotted LLM-based methods degrade more gradually than $\texttt{k-NN}$, which shows a significant performance decrease after 200 meters, indicating that spatially explicit tuning enables reasoning beyond nearest-neighbor retrieval. In the farthest-distance bin, **LATENT** achieves the highest performance in five of the six settings, while **HYBRID**,

which also incorporates the continuous latent encoding, performs best in the remaining setting. The difference between **LATENT** and the remaining methods becomes larger at the farthest bins. These results suggest that continuous spatial representations improve generalization across varying distances and that their advantage becomes more pronounced as the spatial gap increases.

A.6 GeoGrounding Model

GeoGrounding aligns natural-language descriptions with geographic locations through a CLIP-style contrastive objective (Radford et al., 2021). Given a location–description pair $(\mathbf{x}_i, y_i)$, we use `Qwen/Qwen3-Embedding-4B`[7] as the text encoder and take the hidden state of the final token as the pooled description representation. The text encoder remains frozen during training, while a trainable MLP projects its output into the shared embedding space. The location branch encodes $\mathbf{x}_i$ using the Geo2Vec position encoding PE (Chu & Shahabi, 2026), followed by a separate trainable MLP. The normalized representations are

$$E_\phi^{\text{text}}(y_i) = \text{Norm}\left(P_\phi^{\text{text}}\left(H_{\text{Qwen}}(y_i)\right)\right), \tag{46}$$

$$E_\phi^{\text{loc}}(\mathbf{x}_i) = \text{Norm}\left(P_\phi^{\text{loc}}\left(PE(\mathbf{x}_i)\right)\right), \tag{47}$$

where P_ϕ^{text} and P_ϕ^{loc} are the trainable projection MLPs and $\text{Norm}(\cdot)$ denotes ℓ_2 normalization.

For a batch of B paired descriptions and locations, their temperature-scaled similarity is

$$s_{ij} = \frac{E_\phi^{\text{text}}(y_i)^\top E_\phi^{\text{loc}}(\mathbf{x}_j)}{\tau}, \tag{48}$$

where $\tau = 1.0$ is the contrastive temperature. We optimize the symmetric text-to-location and location-to-text losses

$$\mathcal{L}_{\text{text}\to\text{loc}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(s_{ii})}{\sum_{j=1}^{B}\exp(s_{ij})}, \tag{49}$$

$$\mathcal{L}_{\text{loc}\to\text{text}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(s_{ii})}{\sum_{j=1}^{B}\exp(s_{ji})}, \tag{50}$$

and

$$\mathcal{L}_{\text{ground}} = \frac{1}{2}\left(\mathcal{L}_{\text{text}\to\text{loc}} + \mathcal{L}_{\text{loc}\to\text{text}}\right). \tag{51}$$

At inference time, we precompute $E_\phi^{\text{loc}}(\mathbf{x})$ for every location $\mathbf{x}$ in the predefined $M \times M$ grid $\mathcal{G}$. Given a textual query y, we calculate its representation once and compare it with every grid-cell representation:

$$s_\phi(\mathbf{x}, y) = E_\phi^{\text{text}}(y)^\top E_\phi^{\text{loc}}(\mathbf{x}), \qquad \mathbf{x} \in \mathcal{G}. \tag{52}$$

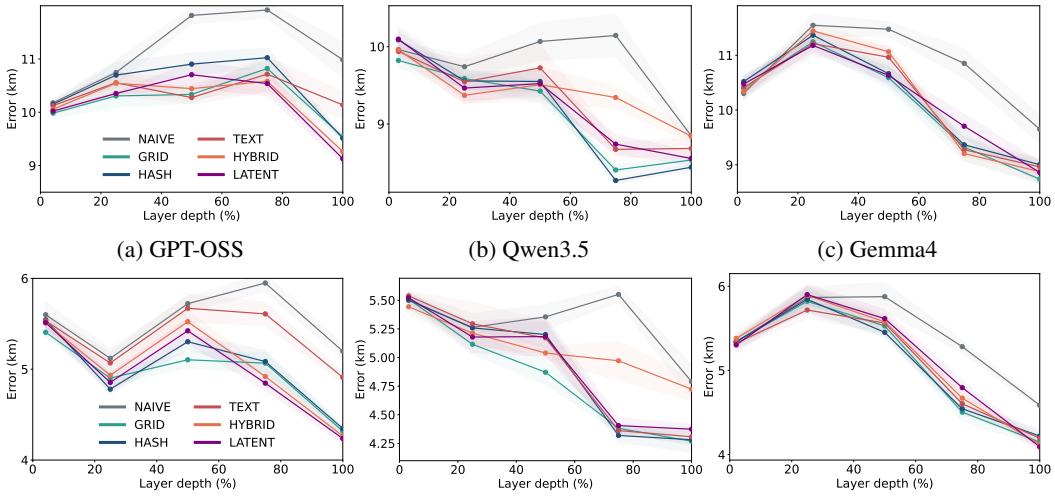
The resulting similarity field provides a dense grounding map over the geographic domain, from which the highest-scoring locations or regions associated with the query can be retrieved.

A.7 Layer-Wise Probing

Prior work uses Ridge linear probes to examine whether pretrained LLMs encode space and time in their hidden representations (Gurnee & Tegmark, 2024). Given place names and their corresponding coordinates, hidden states are extracted using last-token pooling and used to predict geographic coordinates. The underlying intuition is that if geographic structure is encoded in the model, location information should be recoverable from its hidden representations using a lightweight probe.

We extend this approach to examine how precisely our tuned LLMs encode geographic space and how this information evolves across layers. For each description y_i, we prompt the model to predict its location and manually prefill the beginning of the assistant response as follows:

[7] `https://huggingface.co/Qwen/Qwen3-Embedding-4B`



(a) GPT-OSS (b) Qwen3.5 (c) Gemma4

(d) GPT-OSS (e) Qwen3.5 (f) Gemma4

Figure 8: **Layer-wise geospatial probing performance across model families.** The top row reports results on Los Angeles, and the bottom row reports results on London. Lower probing error indicates that geographic location is more explicitly encoded in the hidden representations. The curves report mean performance, while the shaded regions indicate standard deviations.

System: You are a helpful assistant for geo-localization. Always use only the information contained in the provided representation to generate the description.
User: Answer the question using your geospatial knowledge and memory regarding the given description. What are the latitude and longitude coordinates of: `<DESCRIPTION>`?
Assistant: `<DESCRIPTION>` is at `<TOKEN_EXTRACTED>`.

Here, `<DESCRIPTION>` is replaced by y_i, and `<TOKEN_EXTRACTED>` marks the first autoregressive position after the prefixed response, before any coordinate is generated. We extract the hidden state at this position from five selected transformer layers. This design is intended to elicit the geographic information associated with each description, allowing every model to express its strongest recoverable spatial awareness under a consistent prompting protocol.

Let $\mathbf{h}_i^{(\ell)}$ denote the extracted representation at layer ℓ. We train a separate multilayer perceptron r_ℓ for each layer to predict the corresponding coordinate $\mathbf{x}_i$:

$$\widehat{\mathbf{x}}_i^{(\ell)} = r_\ell\left(\mathbf{h}_i^{(\ell)}\right). \tag{53}$$

Unlike a linear probe, the MLP can recover nonlinearly encoded location information, allowing us to assess the amount of geographic information available in each representation rather than only its linear separability.

For each representation method and LLM, we randomly divide $\mathcal{D}_{\text{train}}$ into probing-training and probing-testing subsets. Each probe is trained for 10 epochs using the mean squared error objective

$$\mathcal{L}_{\text{probe}}^{(\ell)} = \frac{1}{|\mathcal{I}_{\text{probe}}|} \sum_{i \in \mathcal{I}_{\text{probe}}} \left\| r_\ell\left(\mathbf{h}_i^{(\ell)}\right) - \mathbf{x}_i \right\|_2^2, \tag{54}$$

with a learning rate of 1×10^{-3}. We repeat the training and evaluation pipeline five times using different random splits and report the mean regression error with its standard deviation.

If an LLM organizes geographic knowledge as a latent spatial field, its hidden representations should retain information about where each description is geographically situated. Lower testing regression error therefore indicates that location information is more accurately recoverable from the corresponding layer. This analysis provides evidence about the internal organization of geographic knowledge.

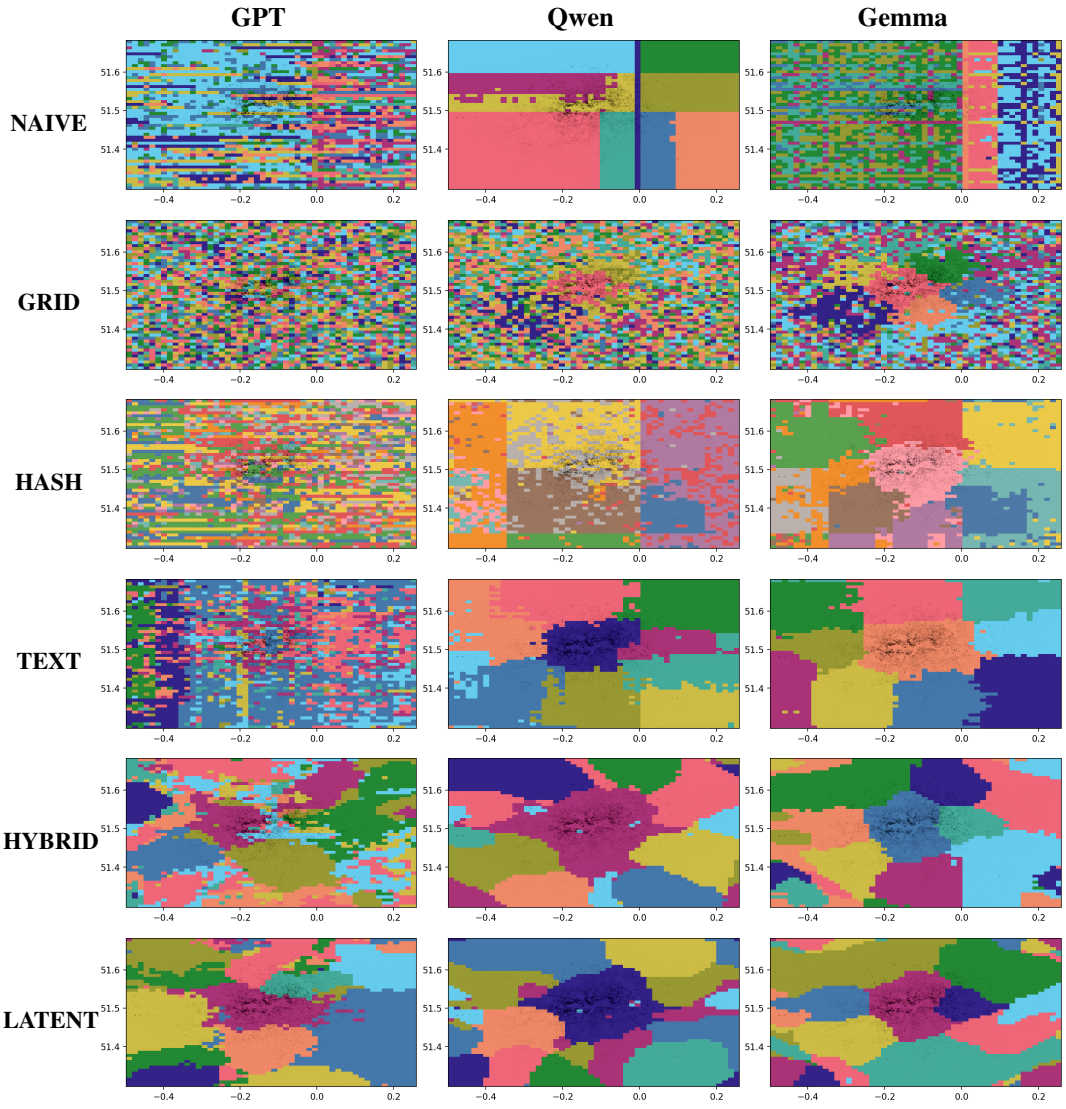


Figure 9: **Cluster maps for the London dataset across model families and spatial representations.** Columns correspond to GPT, Qwen, and Gemma, while rows correspond to **NAIVE**, **GRID**, **HASH**, **TEXT**, **HYBRID**, and **LATENT**.

In Figure 8, across most settings, probing error also decreases substantially toward the later layers and reaches its minimum near the final layer. This trend suggests that geographic information is progressively organized across the model depth, with later representations expressing the most precise spatial awareness.

A.8 Organization of Space

To further investigate how LLMs internally organize geographic space and whether the learned spatial field is continuous, we cluster hidden representations elicited at different query locations and map the resulting assignments back to geographic space. Unlike the layer-wise probing analysis, this experiment does not train a decoder using geographic coordinates. Instead, it directly examines whether the representational similarity learned by the LLM corresponds to coherent spatial regions.

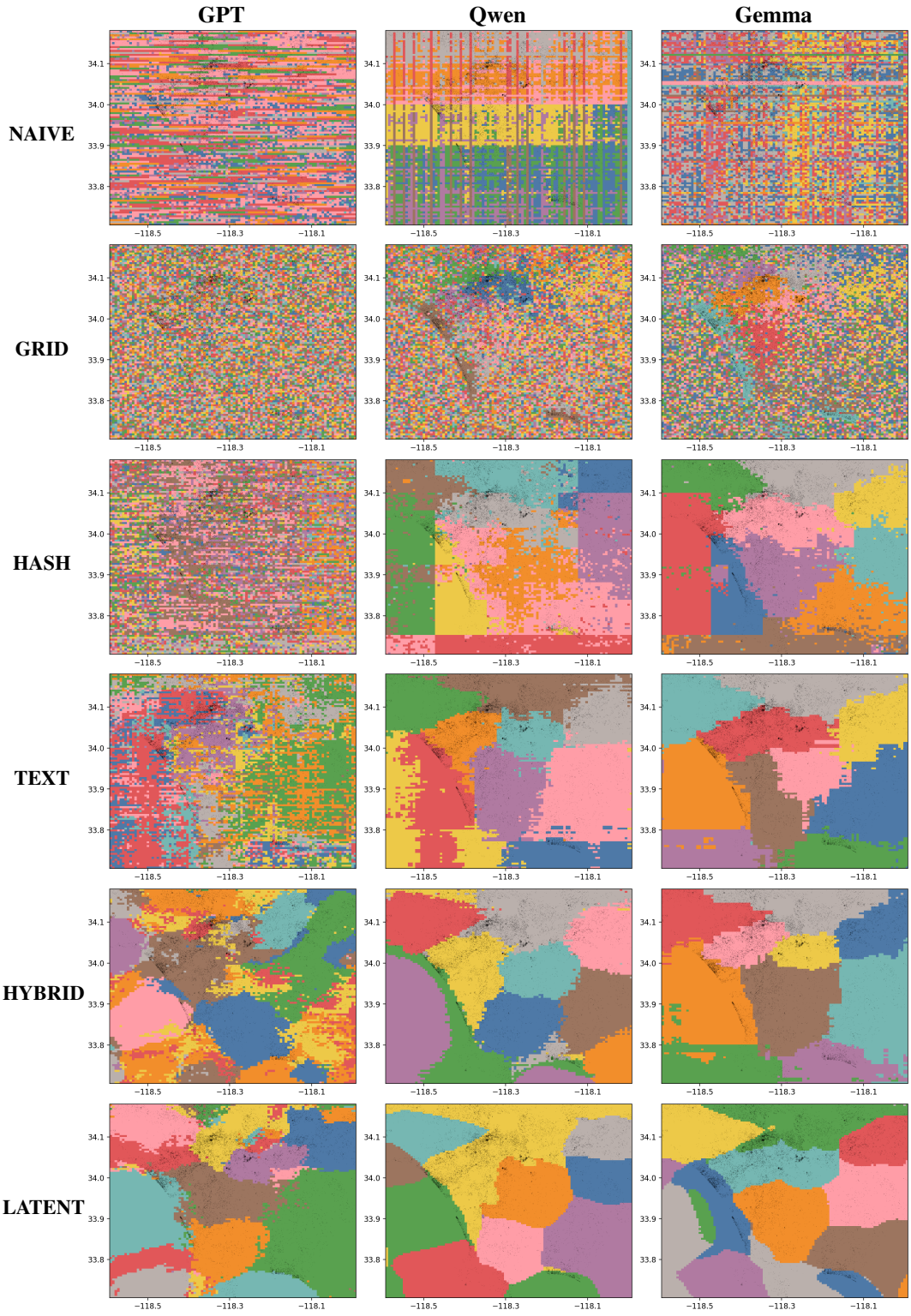


Figure 10: **Cluster maps for the Los Angeles dataset across model families and spatial representations.** Columns correspond to GPT, Qwen, and Gemma, while rows correspond to **NAIVE**, **GRID**, **HASH**, **TEXT**,**HYBRID**, and **LATENT**.

Feature extraction. Similar to the grid we used in evaluation, we uniformly sample a query grid $\mathcal{G} = \{\mathbf{x}_j\}_{j=1}^{N_q \times N_q}$ over each city's geographic domain, N_q follows the training-split settings. For every query location $\mathbf{x}_j$, its corresponding <GEO> representation is inserted into the following prompt:

> **System:** You are a helpful assistant for geo-localization. Always use only the information contained in the provided representation to generate the description.
>
> **User:** You are at location <GEO>. Generate a detailed description of the location.
>
> **Assistant:** <TOKEN_EXTRACTED>

Here, <TOKEN_EXTRACTED> marks the first assistant-generation position and is not provided as literal input to the model. We extract the hidden state at this position, before any location description has been generated. Let $\mathbf{h}_m(\mathbf{x}_j) \in \mathbb{R}^d$ denote the resulting hidden representation for model configuration m at location $\mathbf{x}_j$.

Representation clustering. Before clustering, each hidden representation is ℓ_2-normalized. This normalization reduces the influence of representation magnitude and emphasizes differences in representational direction.

For each model configuration, we independently apply K-means clustering to the normalized hidden states. We use $K = 10$ clusters and $n_{init} = 10$ centroid initializations and a fixed random seed for all model configurations. The resulting assignments are reshaped to the original $N_q \times N_q$ query grid and visualized over geographic space.

Interpretation. The clustering algorithm receives only hidden representations and does not use geographic coordinates when forming the clusters. Therefore, spatially compact clusters and smooth transitions between neighboring regions must arise from geographic structure encoded in the LLM's representations themselves. A fragmented map indicates that nearby locations are mapped to dissimilar hidden states, whereas contiguous regions and coherent boundaries indicate a smoother latent spatial field. Because clustering is performed independently for each configuration, cluster indices and colors do not correspond across models. The Los Angeles results are presented in Figure 10, and the corresponding London results are shown in Figure 9.

Without spatially explicit tuning, **NAIVE** exhibits only weak horizontal and vertical bands, likely because its textual coordinate input varies systematically with latitude and longitude. These coarse patterns do not separate the cities into sufficiently coherent regions to support useful spatially aware geographic inference. **GRID** remains largely fragmented because its identifiers encode no relationships between cells; the model must reconstruct these relationships solely from the similarity of training observations, imposing an additional learning burden. Very few coherent regions appear primarily in areas with exceptionally high observation density, consistent with its weak spatial generalization. **HASH** provides a hierarchical spatial index whose shared prefixes expose structure at multiple scales, reducing dependence on a single grid resolution. It consequently produces clearer spatial patterns than **GRID** for Qwen and Gemma, although the maps remain fragmented, particularly near cluster boundaries. For the text-only GPT-OSS model, **HASH** still fails to produce a well-organized spatial structure.

The **TEXT** representation similarly fails to organize space coherently for GPT-OSS in either city. For Qwen and Gemma, its clusters broadly follow spatial proximity but contain fragmented components near their boundaries as well. Its spatial organization is largely constrained by absolute position. In contrast, **LATENT** produces smoother transitions, more contiguous regions, and clearer spatial boundaries. It can also connect spatially disjoint but semantically similar regions. For example, the coastal areas around Santa Monica and Long Beach are grouped despite their geographic separation. Although this connectivity does not imply identical generated descriptions, it suggests that the learned field captures shared semantic–geographic structure rather than spatial proximity alone. Finally, **HYBRID** combines the local specificity of **TEXT** with the continuity of **LATENT**, largely preserving proximity-based organization while producing smoother cluster boundaries. This effect is particularly clear for GPT-OSS, where **HYBRID** recovers coherent spatial structure absent from the corresponding **TEXT** map.

Table 6: **Generated distances to LAX under different spatial representations.** All distances are reported in miles. Bold values indicate the closest prediction to the measured distance for each query.

ID	Measured Distance	NAIVE	TEXT	GRID	LATENT	HYBRID
0	1.3	14.1	15.0	12.0	**2.5**	10.0
1	7.3	23.1	15.0	10.5	**10.0**	15.0
2	9.6	15.1	15.0	**12.5**	20.0	20.0
3	14.0	18.1	20.0	**12.5**	18.0	17.0
4	14.0	15.1	30.0	**14.0**	25.0	30.0
5	18.0	**18.1**	25.0	12.5	25.0	20.0
6	21.3	15.1	25.0	12.5	**24.0**	30.0

A.9 Qualitative Analysis

We present qualitative case studies for **Geospatial Captioning** and **Geospatial Grounding** to illustrate how different spatial representations affect model behavior.

Geospatial Captioning. We evaluate the spatial awareness of the Gemma4-based models through response prefilling. Given a query location, we manually specify the beginning of the response and ask each model to complete the distance to Los Angeles International Airport (LAX):

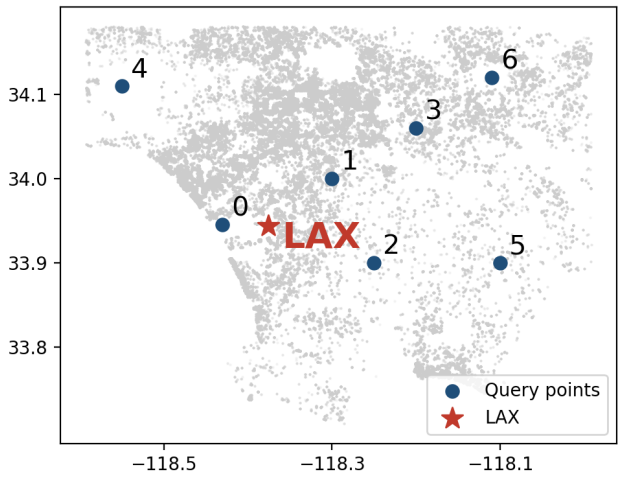


Figure 11: **Query locations used for the LAX distance-estimation case study.** The seven locations are selected from spatially held-out regions and indexed from 0 to 6.

System: You are a helpful assistant for geo-localization. Always use only the information contained in the provided representation to generate the description.

User: You are at location <GEO>. Generate a detailed description of the location.

Assistant: The distance to LAX airport, in mile, is

We deliberately select seven query locations from data-sparse held-out regions containing no training observations. As shown in Figure 11, these locations cover different directions and distances from LAX. Table 6 reports the numerical distance extracted from each model's completed response. Ground-truth values are calculated using the great-circle distance between each query location and LAX.

The examples reveal clear differences in how the representations respond to spatial variation. For the closest location, ID 0, **LATENT** produces the closest estimate, while the other models substantially overestimate the distance. **LATENT** also produces the closest estimates for IDs 1 and 6, distin-

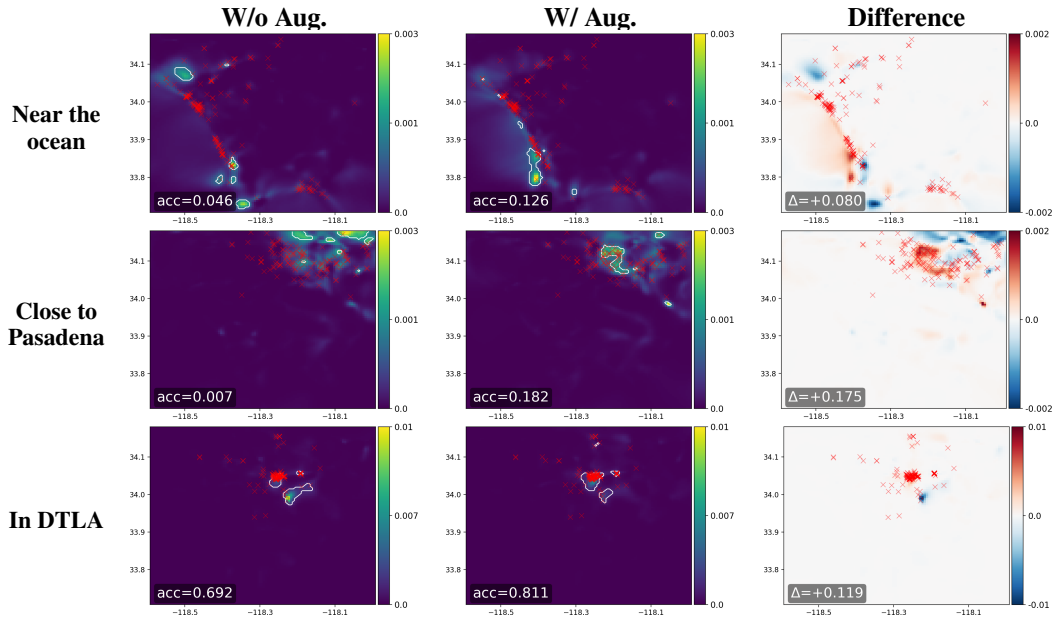


Figure 12: **Qualitative examples of geospatial grounding before and after caption augmentation.** Each row corresponds to a region-level grounding query, with red crosses indicating its ground-truth support locations. The first two columns show the predicted spatial similarity fields without and with gemma generated-caption augmentation, respectively, while white contours outline the highest-scoring 1% of the geographic domain. The final column shows their difference, computed as the augmented field minus the baseline field; red indicates increased similarity and blue indicates decreased similarity.

guishing locations near LAX from those substantially farther away. In contrast, **NAIVE** predictions remain within a relatively narrow range, while **GRID** produces nearly constant values between 10.5 and 14.0 miles, suggesting that it does not reliably capture continuous changes in distance. **TEXT** and **HYBRID** exhibit greater sensitivity to location, but their estimates remain less consistent.

Most predictions, particularly those from **TEXT**, **LATENT**, and **HYBRID**, overestimate the distance. One plausible explanation is a mismatch between the evaluation target and the geographic expressions in the training data. Our measured distance measures straight-line distance, whereas Airbnb descriptions often express approximate driving distances or travel times along road networks. The models may therefore inherit a systematic travel-distance bias from the training corpus. While these examples are qualitative, they illustrate that continuous spatial representations allow the generated values to respond more meaningfully to changes in geographic location. The smooth variation across space also indicates that they are learning a continuous spatial latent field to support their understanding of space.

Geospatial Grounding. Figure 12 illustrates how caption augmentation changes the spatial field learned by the grounding model. For "near the ocean," the baseline produces several disconnected high-similarity regions, including an incorrectly aligned inland region. After augmentation, the response becomes more continuous along the coastline while the incorrect inland mode is suppressed, increasing Top-1% coverage from 0.046 to 0.126.

For "close to Pasadena," the second row of Figure 5 shows the distribution of another semantically related query "in Pasadena." The two similar expressions have distinct but spatially consistent ground-truth distributions: locations described as "in Pasadena" form a compact central region, while those described as "close to Pasadena" surround it. This pattern demonstrates that our grounding benchmark preserves fine-grained geographic relations between similar expressions. Without LLM-generated reasoning augmentation, the grounding model incorrectly aligns "close to Pasadena" with the central Pasadena region, effectively conflating the two phrases. Augmentation

corrects this mismatch by shifting similarity toward the surrounding support locations, improving Top-1% coverage from 0.007 to 0.182.

The third example shows that augmentation can also refine an already meaningful spatial field. For "in DTLA," the baseline identifies the general area but assigns substantial similarity to a displaced region south of the primary support. Augmentation suppresses this incorrect response and strengthens the spatially continuous region around the DTLA observations, increasing coverage from 0.692 to 0.811. The difference maps further show that augmentation does not simply increase similarity across the entire domain; instead, it redistributes similarity toward geographically consistent regions while reducing incorrectly aligned responses. These cases provide qualitative evidence that captions inferred at unobserved locations can fill gaps in the grounding data and promote a more continuous and accurately aligned spatial field.