OmniLoc: Towards Leveraging Multiple Perspectives for Probabilistic Visual Geolocalization

Xiyan Shao
Stanford University
xiyan128@stanford.edu

Charlie Haywood
Stanford University
chaywood@stanford.edu

Abstract

Discriminative, single-image visual geolocalization often suffers from ambiguity. In this report, we present OmniLoc, a framework for Probabilistic Multiview Visual Geolocalization (PMVG) that leverages multiple, geographically coherent ground-level perspectives. We address the PMVG task by efficiently adapting the OpenStreetView dataset into a multiview scene dataset `osv5m-multi`*. The OmniLoc models use an attention-based mechanism to fuse information from multiple images. This fused representation conditions both a deterministic regression head for direct coordinate prediction (OmniLoc$_{regression}$) and a conditional generative head (OmniLoc$_{rfm}$) employing Riemannian Flow Matching to model location probability distributions, allowing for uncertainty-aware localization conditioning on the scene embedding. We systematically explored the design space of OmniLoc$_{regression}$ and OmniLoc$_{rfm}$, and achieved strong results on a challenging subset of* `osv5m-multi`*.*

1. Introduction

Visual geolocalization, the task of estimating the geographic origin of an image or a set of images using computer vision, is a fundamental capability with far-reaching implications across numerous real-world applications [17]. It underpins autonomous navigation systems, particularly where GPS reliability is compromised [3], enhances safety and security through intelligent surveillance and forensic analysis, aids in cultural heritage documentation by identifying and cataloging location-specific features, and serves as a challenging benchmark for assessing the contextual understanding of advanced vision models [1].

The Ambiguity of a Single Glimpse. Localizing an image is fundamentally a task of uncertainty. A solitary image usually lacks the comprehensive visual cues necessary for deterministic localization, especially in common urban settings or visually repetitive natural environments. This limitation highlights a discrepancy between current single-image systems and innate human spatial reasoning.

Towards Human-like Spatial Reasoning Humans, when faced with an unfamiliar environment, instinctively seek out multiple perspectives to orient themselves. Consider the popular game GeoGuessr: successful players rarely rely on the initial static view. Instead – in the gamemode where they can – they virtually "explore" their surroundings, panning and moving to gather contextual information from various vantage points. This accumulation of evidence from multiple, nearby perspectives is crucial for disambiguating the scene and forming a confident hypothesis about their location [6].

While recent advancements in visual geolocalization have explored the fusion of diverse information sources, such as combining ground-level imagery with overhead satellite views [2], the specific potential of leveraging multiple *ground-level perspectives* captured from a local vicinity remains a compelling and relatively underexplored avenue. Harnessing this rich, spatially coherent information, akin to human exploration, offers headroom for localization accuracy and the reliability of predictions.

OmniLoc: Probabilistic Multiview Visual Geolocation (PMVG). This work makes two primary contributions to the space of visual geolocalization:

1. **A New Task and a Dataset Adaptation Algorithm for Multiview Probabilistic Geolocation:** We formally define the task of *multiview probabilistic geolocalization*, which involves predicting a probability distribution over geographic locations given a set of images from a coherent local scene. To support research in this area, we introduce `osv5m-multi`, a dataset derived from `osv5m` [1], specifically curated to provide geographically coherent multiview scenes (Section 3.2). The proximity partitioning algorithm can be applied to any dense, GPS-tagged dataset.

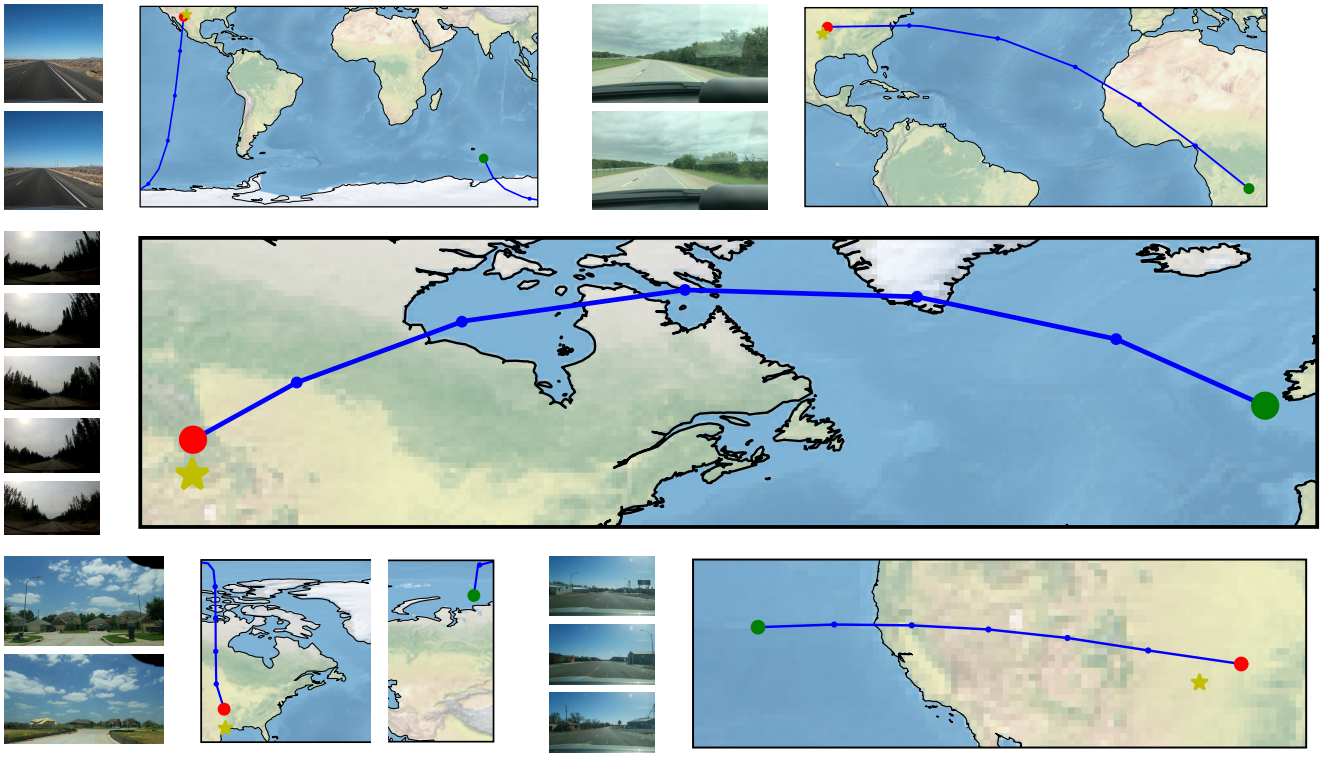

Figure 1. Visualizing the Riemannian Flow Matching process for geolocalization by OmniLoc. Given input street-view images group (shown on the periphery of map insets), the model learns a trajectory (blue line) on the Earth's manifold. This flow transports an initial noisy location (●) through intermediate steps (●) towards a final predicted geographic coordinate (●), aiming to match the ground truth location (⋆).

2. **OmniLoc: A Novel Multiview Probabilistic Geolocation Framework:** We present *OmniLoc*, a series of PMVG models designed for this task. OmniLoc effectively fuses information from multiple ground-level perspectives using an attention-based mechanism to create a unified scene representation. Along with a regression head, we also present a Riemannian Flow Matching (RFM) model [12] to learn a conditional probability distribution over geographic coordinates for uncertainty-aware localization (Section 4.3.2).

By addressing the limitations of single-view methods and providing a principled way to model uncertainty, OmniLoc, along with the proposed task and dataset, aims to advance the capabilities and reliability of visual geolocalization systems for real-world applications.

2. Related Work

Visual geolocalization research has evolved significantly, from foundational single-image approaches to more nuanced probabilistic and multi-modal methods. This section reviews key developments relevant to our work on multi-view probabilistic geolocalization.

2.1. Classical Visual Geolocation

Early and foundational work in visual geolocalization focused on localizing single images using large-scale datasets and powerful image representations. **OSV-5M** [1] stands as a critical resource in this area. It is a large-scale, crowd-sourced street-view dataset comprising 5.1 million geo-tagged images, uniformly sampled to cover 225 countries and territories with minimal geographic bias. Each image includes administrative metadata and auxiliary tags like land-cover type, making this dataset's scale and diversity well-suited for training and evaluating robust geolocation systems.

Building on such datasets, models like **StreetCLIP** [7] have advanced the state-of-the-art. StreetCLIP is a CLIP (Contrastive Language–Image Pretraining)-based vision-language foundation model fine-tuned for street-level geolocation. StreetCLIP is pre-trained by deriving image captions synthetically from image class labels using a domain-specific caption template. Built on a ViT-L/14 backbone, it was pretrained on 1.1 million GeoGuessr images across 101 countries, each paired with a synthetic caption of the form "A Street View photo close to the town of city in the region of region in country." As a foundation model,

StreetCLIP achieves state-of-the-art performance on multiple open-domain image geolocalization benchmarks and does so in a zero-shot setting without additional fine-tuning, outperforming supervised models trained on more than 4 million images.

2.2. Probabilistic Visual Geolocation

Recognizing the inherent ambiguity in visual scenes, some research has shifted towards probabilistic visual geolocation, where models predict a probability distribution over possible locations rather than a single point estimate. An early example, **Im2GPS** [8], produced a probabilistic-like output by considering k-nearest neighbors (kNN) of feature distances and performing mean-shift clustering on their GPS coordinates, often visualized as a density map.

More recently, generative modeling techniques have offered sophisticated ways to model these distributions. **Flow Matching (FM)** [11], for instance, learns a continuous-time velocity field to transport samples from a simple prior distribution to a target data distribution by regressing instantaneous velocities of predefined probability paths. Approaches employing generative techniques like diffusion and **Riemannian Flow Matching** [5] further refine this by explicitly modeling probability densities over geographic locations, learning denoising trajectories on the Earth's surface. This allows for robust location estimation and the quantification of inherent localizability. Our work on probabilistic modeling draws inspiration from these methods, particularly aiming to adapt techniques similar to those used for $\mathbb{S}^2$ manifold modeling as seen in [5].

2.3. Visual Geolocation Beyond Single Image

The intuition that multiple perspectives improve localization, as commonly observed in games like GeoGuessr, has motivated research into methods that utilize more than a single image. Some approaches have explored **multimodal fusion**, such as models [15] that combine convolutional visual features with textual context from news articles. Experiments on news photo geolocation demonstrate that jointly modeling both modalities significantly outperforms single-modality baselines, highlighting the value of diverse information sources.

Other works, like Bianco et al. [2], have focused on different forms of supplementary information. They introduced a retrieval-inspired metric, Recall vs. Area (RvA), and ensembled geolocation models (GeoEstimation, GeoCLIP) with satellite-derived attribute predictors (e.g., population density, land-cover). This strategy yielded significant accuracy gains, especially in underrepresented rural and wilderness areas. These efforts underscore the benefits of incorporating diverse data beyond a single query image for more robust global visual geolocalization, aligning with our goal of leveraging multiple ground-level views.

3. Problem Statement

3.1. Problem Formalization

The traditional single-image geolocalization task is to learn a mapping $f : \mathcal{I} \to \mathbb{S}^2$. In contrast, we formulate the problem as **multiview geolocalization**. Our goal is to learn a function $f_{\text{multi}} : \mathcal{P}(\mathcal{I}) \to \mathbb{S}^2$, where we impose proximity constraints on $\mathcal{P}(\mathcal{I})$. The input is a set of M images, $I = \{i_1, i_2, \ldots, i_M\}$, that constitute a single geographic scene. The output is a single, unified point estimate $\hat{L} \in \mathbb{S}^2$ that represents the location of the entire scene.

We also explore a conditional generative formulation. In this setting, the goal is to learn a mapping $f_{\text{gen}} : \mathcal{P}(\mathcal{I}) \to \text{Prob}(\mathbb{S}^2)$, which outputs a full probability distribution $p(L|I)$ over the sphere, allowing us to explicitly model location uncertainty.

3.2. Dataset

To support our multiview geolocalization tasks, we adapt the OpenStreetView-5M (`osv5m`) dataset [1] (∼5 million images with GPS coordinates) to generate geographically coherent "scenes" for multiview geolocation. The graph-based method imposes proximity constraints on $\mathcal{I}$ such that it has proximity structure ($I \in \text{CCs}(G_\Delta) \quad \wedge \quad \text{diam}(I) \leq D_{\max}$). Figure 1 shows several examples of multiview scenes from `osv5m-multi`.

Efficient Graph-based Partitioning. To create these multiview scenes, we developed an efficient graph-based partitioning algorithm. This method first indexes images spatially, then constructs a proximity graph by connecting images within a defined distance Δ. Finally, scenes are formed from the connected components of this graph. A detailed description of this algorithm can be found in Appendix 6.1.

This algorithm is computationally tractable for large datasets like `osv5m` ($N \approx 5 \times 10^6$). The Δ parameter defines guarantees of how far away images are in a partition. We explored different choices of Δ (Figure 4) and found $\Delta = 0.5$km with a maximum scene diameter of 8km gives a good 39.97% coverage of the `osv5m` [1] training set. We provide this multiview version of `osv5m`, named `osv5m-multi`, as a benchmark for the multiview geolocalization tasks we propose. `osv5m-multi`'s train set contains 1,956,167 images with an average scene size of 7.0.

3.3. Evaluation Metrics

We evaluate our models using several quantitative metrics, some specific to the formulations, averaged over the test set. For point-estimate predictions (the direct output of deterministic models, or samples from generative models), we use standard accuracy metrics: Accuracy@R (km), Mean/Median Distance (km) [1], and GeoScore

$(5000 \exp(-d/1492.7)$, where higher is better). To assess the quality of conditional generative models, we evaluate the full output distribution using the Negative Log-Likelihood (NLL) of the true location under the predicted distribution.

4. Methods

OmniLoc models are designed to transform a set of images from a single scene into a precise geographic prediction. We investigate two distinct prediction modalities: a deterministic **regression** approach that outputs a single coordinate point, and a **conditional generative** approach that conditionally models distributions of likely locations. Both modalities share a common front-end architecture consisting of a **visual backbone** and a **fusion** module but employ specialized prediction heads tailored to their respective tasks. Figure 2 shows the architecture of the $\text{OmniLoc}_{\text{regression}}$ and $\text{OmniLoc}_{\text{rfm}}$ models.

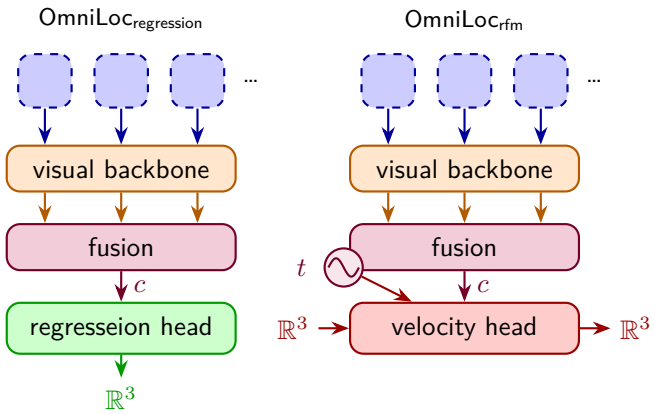

Figure 2. Overview of the $\text{OmniLoc}_{\text{regression}}$ and $\text{OmniLoc}_{\text{rfm}}$ model architectures. Both models share a common visual backbone and fusion module, which processes multiple input images into a scene context vector $\mathbf{c}$. This vector then conditions specialized prediction heads: a regression head for direct coordinate prediction, and a velocity head for conditional generative modeling of location distributions using Riemannian Flow Matching.

4.1. Visual Backbone

To extract robust visual features, we leverage large, pretrained vision models as frozen feature extractors. This allows us to build upon robust, general-purpose visual knowledge and train the prediction network more efficiently. We investigate two state-of-the-art backbones:

- **StreetCLIP:** A vision-language model based on CLIP, specifically fine-tuned for street-level visual geolocalization [7]. It was trained using synthetic captions (e.g., "A Street View photo close to the town of city in the region of region in country.") derived from GeoGuessr images, enabling it to learn strong geographically-aware visual features. We utilize its ViT-L/14 image encoder.
- **SigLIP 2:** A more recent family of powerful multilingual vision-language encoders that improve upon the original SigLIP [16]. SigLIP 2 incorporates techniques like captioning-based pretraining and self-supervised losses, resulting in enhanced semantic understanding and localization. We employ its ViT (So400m/14) image encoder.

For any input image i_j, the backbone outputs a 768/1,152-dimensional embedding $\mathbf{e}_j$.

4.2. Fusion Module

The fusion moduleś task is to map the variable-sized set of image embeddings $\{\mathbf{e}_1, \ldots, \mathbf{e}_M\}$ to a single, fixed-size scene vector $\mathbf{c}$ that represents the scene. We explored several architectures for this task:

- **Mean Pooling Baseline:** Simple, parameter-free method. We evaluated element-wise `Mean Pooling` ($\mathbf{c} = \frac{1}{M}\sum_{j=1}^{M} \mathbf{e}_j$).
- **Query Attention Fusion:** A single learnable query vector attends to all image embeddings using multi-head attention to produce the scene vector.
- **Self-Attention Fusion:** Image embeddings first undergo self-attention. A learnable token, prepended to the sequence (a.k.a. CLS token), is then used as the final scene vector after attention.

4.3. Prediction Heads

The fused scene vector $\mathbf{c}$ is passed to one of two specialized heads, corresponding to our two prediction modalities.

4.3.1. Deterministic Regression Head

For deterministic prediction, we use a regression head to map the scene vector $\mathbf{c}$ to a single 3D coordinate vector on the unit sphere.

Architecture The head is a MLP conditioned on the scene vector $\mathbf{c}$ with the option to use AdaLN and residual connections. It employs ReLU or GELU activations and a final linear layer that outputs a 3D vector $\hat{\mathbf{y}} \in \mathbb{R}^3$. This vector is L2-normalized to lie on the sphere $\mathbb{S}^2$, yielding our final location estimate $\hat{L}$. The architectural details are explained in 5.4.

Loss Function To handle the spherical topology correctly, we explored several loss functions, defaulting to the Huber geodesic loss for its robustness to outliers. The primary losses considered, operating on the predicted 3D vector $\hat{\mathbf{y}}$ and the ground truth vector $\mathbf{y}$ (both on or normalized to the unit sphere $\mathbb{S}^2$), are:

- **Cosine Similarity Loss**: Minimizes the angular distance. Defined as $\mathcal{L}_{\text{reg}} = 1 - \frac{\hat{\mathbf{y}} \cdot \mathbf{y}}{\|\hat{\mathbf{y}}\|_2 \|\mathbf{y}\|_2}$.
- **Angular Loss**: Directly computes the angle (in radians) between L2-normalized predicted vector $\hat{\mathbf{y}}$ and ground truth vector $\mathbf{y}$. Assuming inputs are normalized, it is

$\mathcal{L}_{\text{ang}} = \text{acos}(\hat{\mathbf{y}} \cdot \mathbf{y})$. During computation, the argument to acos is clamped to the range $[-1, 1]$ for numerical stability.

- **Huber Geodesic Loss**: Operates on the great-circle distance d_{km} (in kilometers) between the predicted and ground truth locations. It is quadratic for errors smaller than a threshold β and linear otherwise:

$$\mathcal{L}_{\text{Huber}}(d_{\text{km}}, \beta) = \begin{cases} 0.5 \cdot d_{\text{km}}^2 / \beta & \text{if } d_{\text{km}} < \beta \\ d_{\text{km}} - 0.5 \cdot \beta & \text{otherwise} \end{cases} \quad (1)$$

Our default configuration uses $\beta = 250$ km.

4.3.2. Velocity Head for Conditional Generative Modeling with Riemannian Flow Matching

For conditional generative modeling, we learn a distribution $p(L|\mathbf{c})$ over likely geographic locations L on the sphere $\mathbb{S}^2$, conditioned on the scene vector $\mathbf{c}$. This is achieved using Riemannian Flow Matching (RFM) [4, 11], a technique for learning generative models on manifolds. RFM learns a time-dependent vector field (velocity field) that transports samples from a simple prior distribution (uniform on $\mathbb{S}^2$) to samples from the target data distribution (the ground truth locations). The "Riemannian" aspect ensures that the learned flow respects the geometry of the sphere. This method was proposed in [5] and we reimplemented it in our work in conjunction with the explored visual backbones and fusion modules.

Architecture The core of this head is a **velocity network**, denoted $\mathbf{u}_\theta$. This network is tasked with approximating the true velocity field. We explored FiLM conditioning and AdaLN conditioning and explain the details in 5.4. The network $\mathbf{u}_\theta$ takes three inputs: a point on the sphere $\mathbf{x}_t \in \mathbb{S}^2$, a scalar time step $t \in [0, 1]$, and the scene conditioning vector $\mathbf{c}$. The time t is first embedded into a high-dimensional representation using Gaussian Fourier Projection. The AdaLN conditioning and Fourier embedding tricks are similar to the original implementation in [5], but we reimplemented them in our work. We also heavily used the `flow-matching` library from [12] to implement the Riemannian Flow Matching models.

The output of this network is an ambient vector in $\mathbb{R}^3$. This vector is then projected onto the tangent space $T_{\mathbf{x}_t}\mathbb{S}^2$ at the point $\mathbf{x}_t$ to ensure the manifold structure.

Loss Function The velocity network $\mathbf{u}_\theta$ is trained using the conditional Optimal Transport (OT) Flow Matching objective on the Riemannian manifold $\mathbb{S}^2$. The specific path between a prior sample and a data sample is chosen to be the geodesic path on the sphere. The loss encourages the learned velocity field $\mathbf{u}_\theta(\mathbf{x}_t, t, \mathbf{c})$ to match a target velocity field $\mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1)$ that deterministically transports samples $\mathbf{x}_0$ from a simple prior distribution $p_0(\mathbf{x})$ (uniform on $\mathbb{S}^2$) to the ground truth location $\mathbf{x}_1 = L_{GT}$, conditioned on the scene vector $\mathbf{c}$. The path sample $\mathbf{x}_t$ at time t is given by $\mathbf{x}_t = \text{Path}(t; \mathbf{x}_0, \mathbf{x}_1)$, which is the point along the geodesic from $\mathbf{x}_0$ to $\mathbf{x}_1$ at fraction t of the path. The target velocity is then $\mathbf{v}_t = \frac{d}{dt}\mathbf{x}_t$. The loss, fully conditioned on our scene vector $\mathbf{c}$, is:

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t\sim U[0,1], \mathbf{x}_1 \sim p_{data}(\cdot|\mathbf{c}), \mathbf{x}_0 \sim p_0(\cdot)} \left\| \mathbf{u}_\theta(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1) \right\|_2^2 \quad (2)$$

Here, $\mathbf{x}_1$ is the ground truth location (converted to a 3D vector on $\mathbb{S}^2$) and $\mathbf{x}_0$ is a random sample from the uniform distribution on $\mathbb{S}^2$. The expectation is taken over uniformly sampled time $t \in [0, 1]$, ground truth locations $\mathbf{x}_1$ (whose distribution depends on $\mathbf{c}$), and prior samples $\mathbf{x}_0$.

To enable Classifier-Free Guidance (CFG) [9], during training, we randomly replace the scene conditioning $\mathbf{c}$ with a learned null embedding (a zero vector in our case) with a probability `cfg_dropout_prob` (e.g., 0.1). This allows the model to learn both conditionally and unconditionally. We examine the effect of CFG in 5.5.

Inference and Sampling During inference, we generate samples from the learned conditional distribution $p(L|\mathbf{c})$ by solving an ordinary differential equation (ODE) on the sphere $\mathbb{S}^2$, typically using numerical solvers. Classifier-Free Guidance (CFG) can be optionally employed to enhance the conditioning. This generative process enables the visualization of the full distribution of likely locations and the estimation of its properties (e.g., mode, uncertainty), offering a more comprehensive understanding than a single point estimate. Full details of the inference and sampling procedure, including the ODE formulation and CFG application, are provided in Appendix 6.2.

5. Experiments and Results

In this section, we detail the experimental setup and present the results of our investigations aimed at answering the following key research questions. We evaluate our methods on the `osv5m-multi` dataset, focusing on the metrics defined in Section 3.3.

5.1. Implementation Details

We implemented variants of the $\text{OmniLoc}_{\text{regression}}$ and $\text{OmniLoc}_{\text{rfm}}$ models, with different choices of visual backbones, fusion modules, and prediction heads. We use a single `g4dn.xlarge` instance with T4 GPU, 4 vCPU, and 16 GB memory. Because of the limited compute resource, we trained the models on a subset of the `osv5m-multi` dataset consisting of 129k examples from 1593 unique cities from the west and southwest of the US; we use the partition algorithm described in Section 3.2. This is a challenging subset because the images share regional features, and the distances

have a mean of 1244.82 km and a median of 1183.65 km. For computational efficiency, we precompute image embeddings using the visual backbones described in Section 4.1.

Training Details Unless otherwise specified for a particular experiment, all models were trained using the AdamW optimizer [13] with a weight decay of 1e-5. We employed a learning rate schedule with a linear warmup phase for the first 5 epochs (1% of total epochs, with an initial learning rate factor of 0.1), followed by a cosine annealing decay down to 1e-6 (LR/100). Models are trained for a maximum of 500 epochs (with early stopping using median val distance with a patience of 10 epochs), using mixed precision (bf16-mixed) for efficiency. We use the Huber geodesic loss (β=250km) (which performs the best within the three candidates) for deterministic regression and the Riemannian Flow Matching loss for generative models (Section 4.3.2), both operating on the sphere $\mathbb{S}^2$. We use batch size of 512 for all experiments, and a 9:1 train-validation split. We performed 10-fold cross-validation for shallower networks for hyperparameter tuning, and inherited applicable settings for larger models where we couldn't. We swept learning rates before training each model.

5.2. RQ1: Impact of Visual Backbone Embeddings

We first explore how different visual backbone embeddings affect the performance of our geolocalization models. We compare the effectiveness of StreetCLIP [7] and SigLIP 2 [16] as feature extractors. Specifically, we used the so400m/14 variant of SigLIP 2 and the ViT-L/14 variant of StreetCLIP (largest models that we can efficiently use with our GPU memory).

Preliminary t-SNE Analysis To guide our selection of visual backbones, we first performed a preliminary t-SNE analysis on a sample of 5000 SigLIP and StreetCLIP embeddings (Figure 5). The resulting visualizations show that SigLIP embeddings could form more distinct and well-structured clusters compared to StreetCLIP. SigLIP embeddings showed a lower coefficient of variation in pairwise t-SNE distances (0.474 vs. 0.544 for StreetCLIP) and a higher point density in the 2D projection (0.346 vs. 0.319 points/unit^2). This observation, suggesting potentially better separability and feature representation with SigLIP, motivates further investigation in our subsequent experiments, despite StreetCLIP's specialization for street-level scenes.

Experimental Setup To understand the impact of the visual backbones, we train a simple regression model using both StreetCLIP and SigLIP 2 embeddings, keeping all other hyperparameters and architectural details consistent (mean fusion and simple MLP head described below).

Table 1. Performance comparison of StreetCLIP and SigLIP 2 embeddings.

Embedding Type	Median Dist. (km) ↓	Mean Dist. (km) ↓	Acc@250km (%) ↑	GeoScore ↑
StreetCLIP	1013.67	1129.19	7.04	2,534.23
SigLIP 2	543.89	708.00	18.16	3,474.39

Analysis Our results indicate that SigLIP 2 provides stronger conditioning than StreetCLIP, presumably because its WebLI pre-training corpus (≈10 B images) captures richer geographic and scene diversity than the 1 M street-view photos used for StreetCLIP, and because its contrastive objective is augmented with caption-grounding and masked-patch tasks that preserve fine spatial cues critical for localization [16]. The results also validate our findings in the preliminary t-SNE experiment. Because of the strong result, we default to using SigLIP 2 for the rest of the experiments.

5.3. RQ2: Evaluating Pooling Mechanisms for Multiview Fusion

This study examines the influence of various pooling mechanisms in the fusion module for aggregating multiview image embeddings into a unified scene vector. We compare Mean Pooling and several attention-based mechanisms, including Self-Attention and Query Attention (e.g., using a learned query vector).

Experimental Setup Using the best performing embedding from 5.2, SigLIP 2, we train simple regression models with different pooling strategies: Mean Pooling, Self-Attention, and Query Attention. This helps us understand the impact of the fusion module on the performance of the model, and its scalability with the number of images in a scene. Training parameters are kept consistent across these experiments.

Table 2. Comparison of different pooling mechanisms.

Pooling Mechanism	Median Dist. (km) ↓	Mean Dist. (km) ↓	Acc@250km (%) ↑	Geoscore ↑
Mean Pooling	543.89	708.00	18.16	3472.32
Self-Attention	458.87	646.70	27.52	3675.98
Query Attention	465.72	677.52	24.23	3659.13

Analysis Our results in 2 show that attention-based fusion outperforms simple pooling strategies. This difference stems from how each mechanism aggregates multiview information. Mean Pooling, while simple and efficient, treats all image features equally, failing to emphasize geometrically or semantically salient views. Query Attention introduces a learnable query for some selectivity but lacks the full pairwise context modeling of Self-Attention. Transformer-style Self-Attention explicitly models inter-image relationships, enabling the network to focus on the most informative perspectives (e.g., distinctive landmarks or unique street patterns), leading to better geolocation performance.

We also investigated scalability with the number of images (group size). Attention-based mechanisms (query and self-attention) generally outperform mean pooling as group sizes increase, supporting the hypothesis that complex fusion better captures salient features from multiple views. Mean fusion shows erratic behavior and low performance at larger scales, suggesting ineffective aggregation. Self-attention benefited from scaling, peaking at a group size of 3 (0.28 acc@250km), but performance declined at larger group sizes, indicating potential limitations in handling many images. Query attention, however, demonstrated the most consistent performance across group sizes, with strong performance at group size 5 (0.52 acc@250km), suggesting better scalability (for transparency, however, group size 5 sample size is relatively small, see 4). This may be due to its learned query vector selectively attending to relevant parts of each view, ignoring redundancy. With more images, it is more probable that some views will contain discriminative features for the query.

We acknowledged that more experiments are needed to confirm these findings, as our dataset's training mixture is skewed towards smaller group sizes. We added group size jittering to the training set to attempt to address sequence length generalization.

5.4. RQ3: Optimizing Prediction Head Architecture

Since we use frozen image encoders from strong visual backbones, we wanted to determine whether prediction networks benefit from deeper architectures or if shallower probing networks offer better quality-cost trade-offs. This section investigates the prediction head architecture for both deterministic regression and the RFM model's velocity network. All models in these experiments use SigLIP 2 embeddings with self-attention pooling. We use the best performing regression model from 5.3 as the base model, and implemented the baseline/deeper RFM models described in 4.3.2.

Deterministic regression.

- **Baseline:** a three-layer MLP (hidden width 512, `ReLU`) that maps the fused embedding to latitude-longitude. A LayerNorm is applied after each layer.
- **Deeper MLP with AdaLN:** eight AdaLN–MLP blocks operating on a learnable register; the fused embedding is linearly projected to the hidden dimension ($d_h = 256$) and supplies per-block triplets (γ, μ, σ) that adaptively scale and shift the LayerNorm-centred activations. A final AdaLN modulation and a tiny MLP predictor output the coordinates. We expect the AdaLN head to be more expressive and stable.

Riemannian Flow-Matching (RFM) velocity network.

- **Baseline with FiLM conditioning:** three FiLM-conditioned [14] residual blocks where time and image embeddings generate additive (γ, β) parameters that modulate hidden features.
- **Deeper with AdaLN:** twelve AdaLN–MLP blocks applied to the projected spherical state; the shared conditioning vector controls per-block (γ, μ, σ), followed by a final AdaLN layer and linear read-out. The scale-invariant design is expected to yield smoother, more accurate velocity fields while preserving tangent–space equivariance.

Table 3. Comparison of deterministic regression and RFM velocity network designs.

Model	NLL ↓	Median Dist. (km) ↓	Mean Dist. (km) ↓	Acc@250km (%) ↑	GeoScore ↑
Regression (Simple)	N/A	458.87	646.69879	24.68	3675.97
Regression (Deeper)	N/A	358.96	604.89545	37.54	3930.61
RFM (Simple)	5.41	825.12	979.53	8.13	2875.89
RFM (Deeper)	2.08	709.08	876.02	14.91	3108.32

Analysis The results in Table 3 show that deeper prediction head architectures incorporating AdaLN yield better performance for both deterministic regression and Riemannian Flow Matching (RFM). These findings validate our hypothesis that the deeper networks can acquire more complex mappings for geospatial localization, beyond probing the pre-trained scene embeddings.

We did notice, however, that the generative RFM models' accuracy performance, as measured by median distance and Acc@250km, is generally lower than their deterministic regression counterparts when trained with the same data and number of epochs. This is likely because RFM learns the entire conditional distribution on $\mathbb{S}^2$. Inference involves sampling this distribution, so variance inevitably introduces inaccuracies in deterministic metrics. The additional modality of being able to model the conditional distribution comes at a cost where future works should consider leverage more data to learn the distribution. We expect training with the full `osv5m-multi` will help to reduce the gap. In 5.5, we

explore a method to improve the RFM model's performance by using Classifier-Free Guidance (CFG).

5.5. RQ4: Enhancing RFM Performance with CFG

Experimental Setup Using the best performing RFM model architecture identified in the 5.4 , we conduct experiments with and without CFG (varying guidance scales) and study its effect on the model's performance.

We examine the effect of CFG on the RFM model's performance. Figure 3 shows the validation accuracy (Acc@250km) curves during training for the baseline RFM model and RFM with Classifier-Free Guidance (CFG, scale=2.0). Table 4 shows the performance metrics for the two models.

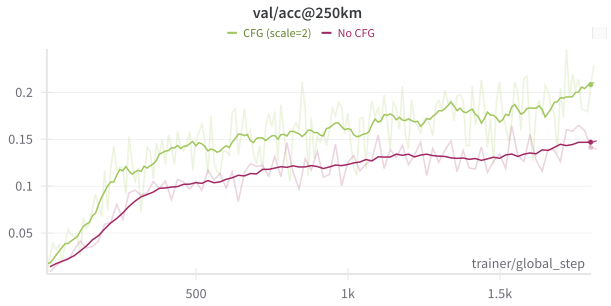

Figure 3. Comparison of validation accuracy (Acc@250km) curves during training for the baseline RFM model and RFM with Classifier-Free Guidance (CFG, scale=2.0). CFG demonstrates improved accuracy throughout training.

Table 4. Impact of Classifier-Free Guidance (CFG) on on RFM performance.

RFM Config.	NLL ↓	Median Dist. (km) ↓	Mean Dist. (km) ↓	Acc@250km (%) ↑	GeoScore ↑
Baseline RFM	2.08	709.08	876.02	14.91	3108.52
CFG (scale=2.0)	7.13	576.84	713.13	19.98	3396.51

Analysis The results show that CFG significantly improves the RFM models predictive accuracy, with better distance metrics and accuracy within 250 km. This shows CFG effectively steers the generative process toward more precise location predictions. However, this adversely affects probabilistic calibration, as shown by the significant rise in NLL. This is a known trade-off between sample quality and diversity [9].

5.6. Overall Performance and Future Work

Our experiments indicate that the optimal regression model utilizes a deeper MLP architecture with AdaLN, achieving a median distance error of 358.96 km and an accuracy of 37.54% within 250 km. For the generative approach, the Riemannian Flow Matching model enhanced with Classifier-Free Guidance (scale=2.0) performed best, yielding a median distance error of 576.84 km and an accuracy of 19.98% within 250 km. We systematically explored the impact of the visual backbones, fusion modules, and prediction heads on the performance of the models. We also explored the impact of Classifier-Free Guidance (CFG) on the RFM model's performance. For better performing models, we would like to train using the full `osv5m-multi` dataset. We expect this to, particularly, improve the RFM model's performance. We could use EMA to stabilize the training of the RFM model as well, a standard practice in generative models albeit requiring more compute.

6. Conclusion

This work introduced OmniLoc, a framework for multiview probabilistic visual geolocalization. We extended visual geolocation to the multiview setting by extracting features with strong visual backbones, fusing them with attention, and then either regressing coordinates ($\text{OmniLoc}_{\text{regression}}$) or modeling location probability distributions on the sphere ($\text{OmniLoc}_{\text{rfm}}$).

Key contributions include: (1) the `osv5m-multi` dataset, created with an efficient proximity-based partitioning algorithm that transforms dense GPS-tagged imagery into geographically coherent scenes for multiview research; and (2) an exploration of the OmniLoc design space, demonstrating strong performance for both $\text{OmniLoc}_{\text{regression}}$ and $\text{OmniLoc}_{\text{rfm}}$ on a challenging `osv5m-multi` subset.

Experiments revealed that: modern vision-language models (SigLIP 2) offer more discriminative features than specialized ones (StreetCLIP); attention-based fusion surpasses simple pooling, especially for complex scenes; deeper, adaptively normalized architectures improve both deterministic and generative models; and Classifier-Free Guidance enhances generative model accuracy at the cost of calibration.

We think OmniLoc provides a valuable foundation for future multiview geolocalization and uncertainty modeling. Future work includes scaling to the full `osv5m-multi` dataset, exploring advanced fusion and generative techniques for better calibration, and adapting OmniLoc for dynamic or sparse environments. Developing spatially reasoning agents that actively select viewpoints to reduce images needed for localization is a particularly exciting direction.

References

[1] Guillaume Astruc, Nicolas Dufour, Ioannis Siglidis, Constantin Aronssohn, Nacim Bouia, Stephanie Fu,

Romain Loiseau, Van Nguyen Nguyen, Charles Raude, Elliot Vincent, Lintao XU, Hongyu Zhou, and Loic Landrieu. Openstreetview-5m: The many roads to global visual geolocation. arXiv preprint arXiv:2404.18873, 2024. 1, 2, 3

[2] Michael J. Bianco, David Eigen, and Michael Gormish. Enhancing worldwide image geolocation by ensembling satellite-based ground-level attribute predictors. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 535–543, 2025. 1, 3

[3] Marcus A. Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3057–3064, 2013. 1

[4] Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries, 2024. 5

[5] Nicolas Dufour, David Picard, Vicky Kalogeiton, and Loic Landrieu. Around the world in 80 timesteps: A generative approach to global visual geolocation. arXiv preprint arXiv:2402.06781, 2024. 3, 5

[6] A. D. Ekstrom and E. A. Isham. Human spatial navigation: Representations across dimensions and scales. *Current Opinion in Behavioral Sciences*, 17:84–89, 2017. 1

[7] L. Haas, S. Alberti, and M. Skreta. Learning generalized zero-shot learners for open-domain image geolocalization. arXiv:2302.00275 [cs.CV], 2023. 2, 4, 6

[8] James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 3

[9] Jonathan Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022. 5, 8

[10] International Computer Science Institute and S.M. Omohundro. Five balltree construction algorithms. Technical report, International Computer Science Institute, 1989. 1

[11] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. arXiv preprint arXiv:2210.02747, 2023. 3, 5

[12] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. arXiv preprint arXiv:2402.06264, 2024. 2, 5

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[14] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. *CoRR*, abs/1709.07871, 2017. 7

[15] Golsa Tahmasebzadeh, Sherzod Hakimov, Ralph Ewerth, and Eric Müller-Budack. Multimodal geolocation estimation of news photos. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part II*, pages 204–220. Springer Cham, 2023. 3

[16] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. 4, 6

[17] Daniel Wilson, Xiaohan Zhang, Waqas Sultani, and Safwan Wshah. Visual and object geolocalization: A comprehensive survey. arXiv preprint arXiv:2112.15202, 2023. 1

OmniLoc: Towards Leveraging Multiple Perspectives for Probabilistic Visual Geolocalization

Supplementary Material

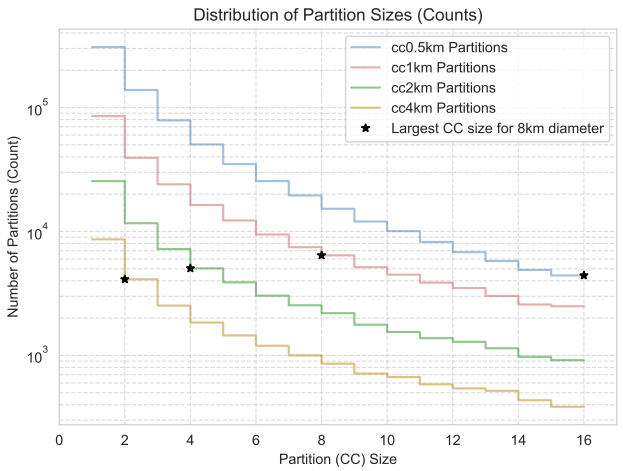

Figure 4. Effect of the Δ threshold on the distribution of multi-view scene sizes (Connected Components) from `osv5m`. Larger Δ thresholds produce fewer but larger scenes. The Y-axis is on a log scale. Stars denote the maximum scene size observed for each Δ setting, given an 8km maximum diameter constraint for scenes.

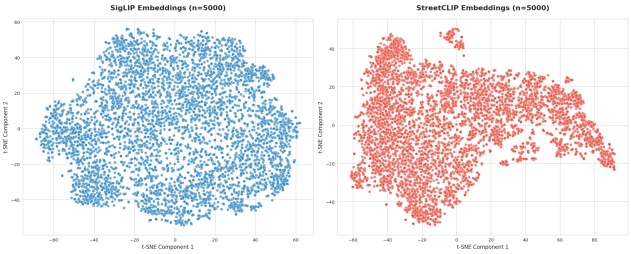

Figure 5. t-SNE visualization of StreetCLIP and SigLIP 2 embeddings (n=5000 each) from a subset (individual images) of the `osv5m-multi` dataset.

6.1. Efficient Graph-based Partitioning for Scene Creation

Identifying optimally "dense" visual clusters in large image datasets can be computationally challenging. For instance, framing this as finding maximal cliques in a proximity graph is an NP-hard problem, making it infeasible for datasets with millions of images. To address this, we employ an efficient graph-based approach. This method identifies connected components (CCs) in a proximity graph, which naturally represent explorable scenes where any image is reachable from another via a path of nearby observations. The algorithm proceeds as follows:

1. **Spatial Indexing:** Image coordinates (latitude, longitude) are indexed using a BallTree [10] with the Haversine metric. BallTree construction is $O(M \log M)$ for M images.
2. **Proximity Graph Construction:** For each image, a radius query on the `BallTree` identifies all neighbors within a Δ threshold (in kilometers). This defines an adjacency list for the graph. For sparse graphs, this step is typically efficient (amortized $O(M \log M)$ or $O(M \cdot k_{\text{avg}})$ where k_{avg} is the average number of neighbors per image).
3. **Scene Formation:** Multiview scenes are formed by computing the CCs of this graph. A Breadth-First Search (BFS) on the sparse graph representation (M nodes, E edges) achieves this in $O(M + E)$ time.

6.2. Detailed Inference and Sampling for Riemannian Flow Matching

At inference, to draw samples from the learned conditional distribution $p(L|\mathbf{c})$, we solve the ordinary differential equation (ODE) $\frac{d\mathbf{x}}{dt} = \mathbf{u}_\theta(\mathbf{x}_t, t, \mathbf{c})$ from $t = 0$ to $t = 1$, starting with an initial sample $\mathbf{x}_0$ drawn from the prior distribution (uniform on $\mathbb{S}^2$). This is performed on the manifold $\mathbb{S}^2$ using numerical ODE solvers such as Euler, Midpoint, or RK4, with a configurable number of sampling steps. In practice, we use 100 steps and midpoint solver.

If Classifier-Free Guidance (CFG) is used, the velocity function during sampling is modified to:

$$\hat{\mathbf{u}}_\theta(\mathbf{x}_t, t, \mathbf{c}) = \mathbf{u}_\theta(\mathbf{x}_t, t, \emptyset) + s \cdot (\mathbf{u}_\theta(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{u}_\theta(\mathbf{x}_t, t, \emptyset))$$

where $\emptyset$ denotes the null conditioning and s is the `cfg_scale` (guidance scale). A scale $s > 1$ amplifies the conditioning.