RAFMa: Retrieval-Augmented Flow Matching for Global Visual Geolocation

Anonymous authors
Paper under double-blind review

Abstract

Global visual geolocation predicts where on Earth an image was taken. Classification and generative models store all world knowledge in their parameters, cannot be updated without retraining, and are known to generalize well at coarse scales, such as country or region. In contrast, retrieval-based systems compare the image against a database of geotagged images and excel at fine scales, such as street or building, where memorization matters; moreover, their database can be updated at any time as new geotagged images become available. We combine the two families in a flow matching model conditioned jointly on the query image and on retrieved visually similar images together with their geotags, a form of in-context learning. Starting from a random location on the sphere, the model iteratively moves it along a learned velocity field, while an attention module learns to weight each retrieved image according to its appearance and geodesic distance to the current estimate. The model achieves a good trade-off between generalization and memorization, and effectively benefits from knowledge updates at inference time by changing the database. On street-view images (OSV-5M), RAFMa sets a new state of the art, clearly outperforming prior work. On unconstrained user photographs (YFCC4k, Im2GPS3k), it is competitive with systems built on multimodal LLMs of several billion parameters, while training only 135M parameters.

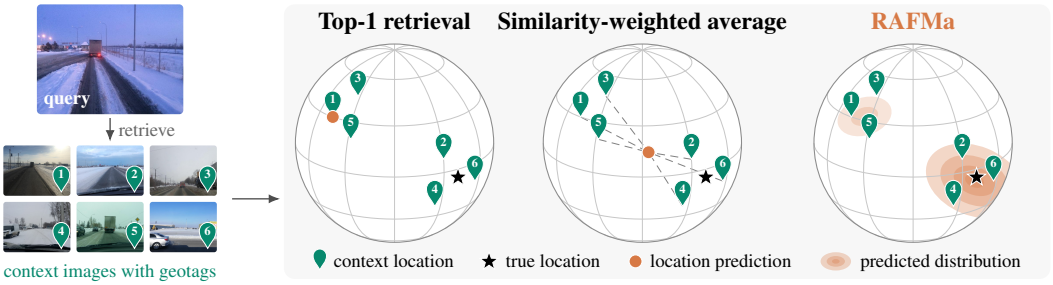


Figure 1: **Retrieved images as context for generative geolocation.** The images retrieved for the query look alike but lie in two distant places. Selecting the location of the most similar image predicts the wrong place, while averaging the retrieved locations lands between the two places, far from both. Our model instead predicts a distribution over locations, via flow matching conditioned on the query image and its context.

1 Introduction

Most deep learning models are trained once and then frozen. Their knowledge is fixed at training time, and new data can only be exploited by retraining. For global visual geolocation, which predicts where on Earth a query image was taken, this is a limitation. The world keeps changing, new geotagged images arrive every day, and each of them could help locate the next query.

Existing methods fall into two families. Parametric models, based on classification, regression, or generation (Weyand et al., 2016; Müller-Budack et al., 2018; Astruc et al., 2024; Dufour et al., 2025), absorb the knowledge of millions of training images into their parameters and generalize to unseen

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.

places, but new geotagged images cannot affect their predictions without retraining. Retrieval methods (Hays & Efros, 2008; Vo et al., 2017) instead keep geotagged images in an external database that can be extended at any time, and retrieve the most similar ones for each query. Retrieval memorizes rather than generalizes, and is thus limited by the density of the database around the true location.

We bridge the two worlds with a generative geolocation model that conditions on an external memory. We rely on a generative model because the task is intrinsically ambiguous; a dry coastal road with sparse vegetation may be compatible with Greece, California, Chile, South Africa, or Australia. Rather than regressing a single coordinate, generative geolocation learns a distribution over locations and samples from it (Dufour et al., 2025; Wang et al., 2025), and can thus represent several distant hypotheses. We provide the generator with retrieved image–location pairs as *in-context demonstrations*, borrowing from retrieval-augmented generation (Lewis et al., 2020; Borgeaud et al., 2022) and in-context learning (Brown et al., 2020). The model can thus exploit labeled images it never saw during training, and requires fewer training images when the remaining data stays available in the database. This raises the central question of our work: *how can a generator exploit retrieved images that may conflict, while its own geographic hypothesis is still forming?* (Figure 1)

We build on spherical flow matching (Dufour et al., 2025), which generates a location by transporting a sample from uniform noise on the sphere to a geographic prediction, and introduce a new contextual conditioning mechanism. Our key idea is to model, at every step of the generative process, both the interactions among the context images and those between the context and the current prediction. Both visual and geographic information are essential to this process. In particular, we inject a strong inductive bias by feeding the geodesic distances between the current sample and the retrieved locations into the attention mechanism, so that the role of the context evolves together with the prediction.

Our experiments show that contextual conditioning consistently improves over the same generator conditioned on the query image alone, and that performance keeps increasing as the database grows. The model exploits new geotagged images without retraining, even when only a small fraction of the context lies close to the true location. On OSV-5M (Astruc et al., 2024), it sets a new state of the art, and on YFCC4k (Vo et al., 2017), it is competitive with systems built on multimodal large language models (MLLMs). Notably, retrieval improves our generator at all scales, from city to continent level, and not only at fine scales, where retrieval is traditionally most effective.

Our contributions include a new formulation of retrieval-augmented generative geolocation as in-context learning from image–location pairs, a geodesic distance bias in attention that makes the weighting of the context depend on the evolving generated location, and a demonstration that external geotagged memory can both improve a fixed model after training and substantially reduce the amount of data that must be absorbed into its parameters.

2 Related Work

Geolocation as classification or regression. Classification approaches partition the Earth's surface into cells and predict the cell that contains the image location (Weyand et al., 2016). Later work uses grids at multiple granularities or along a geographic hierarchy (Müller-Budack et al., 2018; Seo et al., 2018; Pramanick et al., 2022; Clark et al., 2023), or replaces the cells with interpretable administrative regions (Theiner et al., 2022). The partition bounds the resolution of the prediction, unless an offset is regressed within the cell (Astruc et al., 2024) or the partition is replaced by a continuous density on the sphere (Izbicki et al., 2019). Contrastive methods instead learn a joint embedding of images with text, locations, or geographic entities (Haas et al., 2023; Vivanco Cepeda et al., 2023; Daruna et al., 2026; Gadi et al., 2026), and predict by matching the query against toponyms, candidate coordinates, or entities rather than against images with known locations. Recent approaches employ MLLMs that explicitly reason about the location from the image alone (Wu et al., 2026; Liu et al., 2025; Campos et al., 2026). Dufour et al. (2025) instead predict a continuous distribution over coordinates, treating the task as regression in a continuous space rather than classification over a quantized one, which fits the task definition more naturally. They learn this distribution with diffusion and flow matching (Lipman et al., 2023), including a Riemannian formulation (Chen & Lipman, 2024) that generates locations directly on the sphere. Similarly, LocDiff (Wang et al., 2025) diffuses spherical-harmonic coefficients of the location.

Geolocation with retrieval. Retrieval is the earliest formulation of the task (Hays & Efros, 2008), later improved with deep descriptors (Vo et al., 2017), and its accuracy depends on database coverage and retrieval quality. The locations of the retrieved images are converted into a prediction by top-1 selection, averaging, or Kernel Density Estimation (KDE) (Vo et al., 2017). Other methods refine a prediction originally made without retrieval, by searching within the predicted cells (Kordopatis-Zilos et al., 2021; Haas et al., 2024) or around sampled locations (Wang et al., 2025); retrieval thus corrects an existing estimate instead of shaping it from the start. Img2Loc (Zhou et al., 2024) and G3 (Jia et al., 2024) write the coordinates of the retrieved images into the prompt of an MLLM, which answers with coordinates as text; the MLLM is not trained for the task and relies on its general knowledge. GeoRanker (Jia et al., 2025) instead fine-tunes an MLLM to rerank candidate images and locations and keeps the top-ranked one, discarding the rest; a major limitation is requiring raw images at inference. Closest to our work, GeoToken (Ghasemi et al., 2025) trains a transformer on the query and its retrieved images, and decodes the location autoregressively as a sequence of hierarchical cell tokens. In contrast, we generate locations in the continuous space of the sphere, jointly exploiting all retrieved images and their geographic relations to the evolving prediction.

3 METHOD

Given a query image, the goal is to predict its geographic location. We extend the flow-matching model of Dufour et al. (2025) to support contextual conditioning, where the context is a set of visually similar geotagged images retrieved from a geotagged image database.

3.1 PROBLEM FORMULATION

Let x be a query image, represented by descriptor $\mathbf{x} \in \mathbb{R}^D$, whose ground-truth geographic location is $\mathbf{y}_x \in \mathbb{S}^2 \subset \mathbb{R}^3$, expressed as a unit vector. The goal is to predict the location $\hat{\mathbf{y}}_x$ of x. In addition to the query image, during inference, we are given a *database* $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{M}$ of M geotagged images, each represented by its descriptor $\mathbf{x}^{(i)}$ and its known location $\mathbf{y}^{(i)}$. A training set $\mathcal{T} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$ of N geotagged images is available for learning the model parameters. The database $\mathcal{D}$ may coincide with or differ from the training set, and it can be extended or replaced at any time without retraining.

Method overview. We extend Dufour et al. (2025), who model the conditional distribution $p_\theta(\mathbf{y}_x \mid \mathbf{x})$, learn θ on $\mathcal{T}$, and obtain a point prediction $\hat{\mathbf{y}}_x$ as a sample from it. Our pipeline has two stages, illustrated in Figure 2. First, we retrieve from the database a small ranked list of images visually similar to the query, which, together with their known locations, forms the context $\mathcal{C}_x$. Second, we model $p_\theta(\mathbf{y}_x \mid \mathbf{x}, \mathcal{C}_x)$, *i.e.* the generative model is conditioned on the context, which it learns to exploit as demonstrations of the task, a form of in-context learning. During training, $\mathcal{T}$ serves as database, and each query is excluded from its context to avoid trivial solutions. At inference, the context is retrieved from $\mathcal{D}$.

3.2 GENERATIVE GEOLOCATION WITHOUT CONTEXT

We adopt the generative formulation of Dufour et al. (2025), which applies flow matching (Lipman et al., 2023) on the sphere through its Riemannian extension (Chen & Lipman, 2024). In this section, the model is conditioned on the image descriptor only, *i.e.* $p_\theta(\mathbf{y}_x \mid \mathbf{x})$ is modeled.

Training objective. Flow matching learns a time-dependent velocity field that transports samples from a simple base distribution to samples from the data distribution. Let $\mathbf{y}_t \in \mathbb{S}^2$ denote the sample at time $t \in [0, 1]$, where, following Lipman et al. (2023), time runs from noise at $t = 0$ to data at $t = 1$. Given uniform noise $\mathbf{y}_0 \sim \mathcal{U}(\mathbb{S}^2)$ and the target location $\mathbf{y}_1$ set to the ground-truth location $\mathbf{y}_x$, the conditional path follows the geodesic between them,

$$\mathbf{y}_t = \mathrm{Exp}_{\mathbf{y}_0}\big(t \, \mathrm{Log}_{\mathbf{y}_0}(\mathbf{y}_1)\big) \quad , \qquad \mathbf{u}_t = \frac{\mathrm{d}\mathbf{y}_t}{\mathrm{d}t} \quad , \tag{1}$$

where Exp and Log denote the exponential and logarithmic maps of the sphere, and the velocity $\mathbf{u}_t$ lies in the tangent space at $\mathbf{y}_t$.

A network $\mathbf{v}_\theta(\mathbf{y}_t \mid \mathbf{c}_t)$ is used to regress this velocity, where $\mathbf{c}_t \in \mathbb{R}^d$ is a conditioning vector. It is given by $\mathbf{c}_t = f_\theta(\mathbf{x}, t) = \phi(\mathbf{x}) + \boldsymbol{\tau}_t$, where $\phi : \mathbb{R}^D \to \mathbb{R}^d$ is a learned linear projection and $\boldsymbol{\tau}_t \in \mathbb{R}^d$ is a learned embedding of sinusoidal features of t. The conditioning vector depends on both the image descriptor $\mathbf{x}$ and time t, but is subscripted with t only for simplicity.

This network is trained by minimizing

$$\mathcal{L}(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{y}_1)\sim\mathcal{T},\, \mathbf{y}_0\sim\mathcal{U}(\mathbb{S}^2),\, t}\left[\|\mathbf{v}_\theta(\mathbf{y}_t \mid \mathbf{c}_t) - \mathbf{u}_t\|_2^2\right] \quad . \tag{2}$$

Inference. A location is generated by drawing $\mathbf{y}_0 \sim \mathcal{U}(\mathbb{S}^2)$ and integrating the learned velocity field from $t = 0$ to $t = 1$. We take L Euler steps, $0 = t_0 < t_1 < \cdots < t_L = 1$, on a time grid in $[0, 1]$ and iteratively refine the predicted coordinate by

$$\mathbf{y}_{t_{n+1}} = \Pi(\mathbf{y}_{t_n} + (t_{n+1} - t_n)\, \mathbf{v}_\theta(\mathbf{y}_{t_n} \mid \mathbf{c}_{t_n})) \quad , \qquad \Pi(\mathbf{y}) = \frac{\mathbf{y}}{\|\mathbf{y}\|_2} \quad . \tag{3}$$

Since the network output is not constrained to the tangent space, the projection Π maps the sample back onto the sphere after every step. The final sample is the predicted location, *i.e.* $\hat{\mathbf{y}}_x = \Pi(\mathbf{y}_{t_L})$ for query image x.

Velocity network architecture. We use the AdaLN-MLP velocity network of Dufour et al. (2025). The network receives the sample $\mathbf{y}_t$ as input, together with the conditioning vector $\mathbf{c}_t$. Internally, it lifts $\mathbf{y}_t$ to a feature in $\mathbb{R}^d$ and refines it with a sequence of residual MLP blocks, each modulated by $\mathbf{c}_t$ through adaptive layer normalization (Peebles & Xie, 2023). A final adaptive normalization followed by a linear layer maps the refined feature to the velocity in $\mathbb{R}^3$. Further details are provided in Appendix D.1 and in Dufour et al. (2025).

3.3 Generative Geolocation with In-Context Learning

Herein, we adopt the same training objective, inference-time sampling, and velocity network as in Section 3.2, but we modify its conditioning vector to allow capturing context.

Context by retrieval. We complement the velocity network with an external memory, namely the database $\mathcal{D}$. For each query image, we retrieve the K most similar database images according to the cosine similarity of their descriptors, forming the ranked context $\mathcal{C}_x = \left\{(\mathbf{x}^{(j)}, \mathbf{y}^{(j)}, s^{(j)})\right\}_{j=1}^{K}$, where $(\mathbf{x}^{(j)}, \mathbf{y}^{(j)})$ denotes the j-th most similar database image, $s^{(j)} \in [-1, 1]$ its similarity to query descriptor $\mathbf{x}$, and $s^{(j)} \geq s^{(j')}$ for $j < j'$. Pairing each retrieved image with its known location $\mathbf{y}^{(j)}$ gives the context the form of the demonstrations used in in-context learning (Brown et al., 2020).

Context-aware conditioning. We now represent $p_\theta(\mathbf{y}_x \mid \mathbf{x}, \mathcal{C}_x)$ through the velocity field $\mathbf{v}_\theta(\mathbf{y}_t \mid \mathbf{c}_t)$, where a context network g_θ produces the conditioning vector $\mathbf{c}_t = g_\theta(\mathbf{x}, t, \mathcal{C}_x) \in \mathbb{R}^d$. Compared to the conditioning vector of the context-free model, which depends only on the query image and time, ours depends also on the context. Therefore, it allows the context to influence the generated location.

Context representations. Each context element is mapped to a visual token $\mathbf{z}_j \in \mathbb{R}^d$ and a geographic token $\mathbf{p}_j \in \mathbb{R}^d$. We compute $\mathbf{z}_j$ by summing the learned projections of the image descriptor and its similarity to the query, and a learned embedding of its rank in the context. More formally, $\mathbf{z}_j = h_v(\mathbf{x}^{(j)}) + h_s(s^{(j)}) + \mathbf{e}_j$, where $h_v : \mathbb{R}^D \to \mathbb{R}^d$ and $h_s : \mathbb{R} \to \mathbb{R}^d$ are linear layers, and the rank embedding $\mathbf{e}_j \in \mathbb{R}^d$ is one of K learned vectors. Similarly, $\mathbf{p}_j = h_p(\mathbf{y}^{(j)})$, where $h_p : \mathbb{R}^3 \to \mathbb{R}^d$ maps the location with a linear layer followed by layer normalization. We additionally insert the query image into the context at index $j = 0$, which we call the *anchor token*, with visual token $\mathbf{z}_0 = h_v(\mathbf{x}) + h_s(1) + \mathbf{e}_0$, where the learned embedding $\mathbf{e}_0$ distinguishes it from the retrieved images. Since the location of the query image is unknown, its geographic token $\mathbf{p}_0 \in \mathbb{R}^d$ is a learned vector independent of the query image.

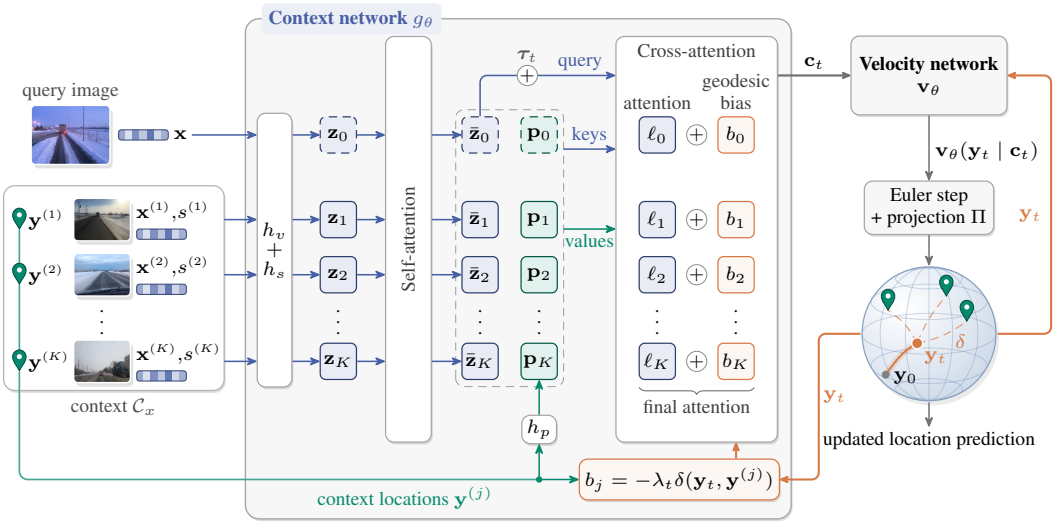


Figure 2: **Overview of RAFMa.** Retrieval is used to obtain the most similar images to the query image and create context $\mathcal{C}_x$ with tuples of image descriptor $\mathbf{x}^{(j)}$, similarity to the query $s^{(j)}$, and location $\mathbf{y}^{(j)}$. Inside the context network g_θ, context examples are mapped to visual tokens $\mathbf{z}_j$ and geographic tokens $\mathbf{p}_j$, together with an anchor token $\mathbf{z}_0$, derived from the query image, whose $\mathbf{p}_0$ is a learned token that carries no location. Self-attention processes all the visual tokens, then cross-attention is used with the anchor token as query, all visual tokens as keys to aggregate the geographic tokens as values. Each attention logit ℓ_j is added to the geodesic bias b_j, derived from the geodesic distance $\delta(\mathbf{y}_t, \mathbf{y}^{(j)})$ between the current sample and the location of context image j. The output at the anchor token is the conditioning vector $\mathbf{c}_t$ which drives the velocity network, and an Euler step with projection Π on the sphere moves the sample to the updated location.

Visual context interaction via self-attention. To exchange information within the context, we process the visual tokens jointly via self-attention, allowing them to influence each other's representations. The process and output are denoted by

$$[\bar{\mathbf{z}}_0; \ldots; \bar{\mathbf{z}}_K] = \mathrm{SA}([\mathbf{z}_0; \ldots; \mathbf{z}_K]), \tag{4}$$

where $\bar{\mathbf{z}}_j$ are the updated visual tokens.

Geographic context aggregation via cross-attention. We aggregate geographic evidence through cross-attention, where the updated anchor token $\bar{\mathbf{z}}_0$ acts as the query, the updated visual tokens $\bar{\mathbf{z}}_j$ act as keys, and the geographic tokens $\mathbf{p}_j$ act as values. We use standard multi-head attention (Vaswani et al., 2017) but omit the head index for clarity. With query $\mathbf{q} = \boldsymbol{W}_q(\bar{\mathbf{z}}_0 + \boldsymbol{\tau}_t)$ and keys $\mathbf{k}_j = \boldsymbol{W}_k \bar{\mathbf{z}}_j$, the output is

$$\ell_j = \frac{\mathbf{q}^\top \mathbf{k}_j}{\sqrt{d_k}}, \qquad \alpha_j = \frac{\exp(\ell_j)}{\sum_{j'=0}^{K} \exp(\ell_{j'})}, \qquad \mathbf{r}_t = \sum_{j=0}^{K} \alpha_j \, \boldsymbol{W}_v \mathbf{p}_j, \tag{5}$$

where $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, and $\boldsymbol{W}_v$ are learned projections and d_k is the key dimension. As in a standard pre-norm transformer, the query is layer-normalized before $\boldsymbol{W}_q$ and the aggregate $\mathbf{r}_t$ is projected and added to $\bar{\mathbf{z}}_0 + \boldsymbol{\tau}_t$, followed by a feed-forward layer, with the usual residual connections.

Since the weights sum to one over the retrieved images and the anchor token, assigning mass to the anchor token reduces the total weight on the retrieved locations. As $\mathbf{p}_0$ carries no location, this allows the module to attenuate the geographic influence of the context while anchoring the conditioning more on the query image. Conversely, putting mass on the context allows the module to gather geographic information from it, grounding the prediction in known locations. The output of this block is the refined anchor token, which we now term $\mathbf{c}_t$ and use as the conditioning vector. The context thus modulates every block of the velocity network.

Geographic distance bias in cross-attention. The attention weights of Equation (5) depend on appearance, rank, similarity, and time, but not on the locations of the retrieved images. They therefore cannot account for the geographic compatibility between the context and the location being generated. We inject this information directly into the attention logits through an additive bias b_j, which replaces the weights of Equation (5) by,

$$\alpha_j = \frac{\exp(\ell_j + b_j)}{\sum_{j'=0}^{K} \exp(\ell_{j'} + b_{j'})}, \qquad b_j = -\lambda_t \, \delta(\mathbf{y}_t, \mathbf{y}^{(j)}), \tag{6}$$

where $\lambda_t = \boldsymbol{\omega}^\top \boldsymbol{\tau}_t + \beta$ is a learned, time-dependent coefficient, and $\delta(\mathbf{y}_t, \mathbf{y}^{(j)}) = 2\,\mathrm{atan2}\big(\|\mathbf{y}_t - \mathbf{y}^{(j)}\|_2,\, \|\mathbf{y}_t + \mathbf{y}^{(j)}\|_2\big) \in [0, \pi]$ is the geodesic distance between the current sample $\mathbf{y}_t$ and the retrieved location $\mathbf{y}^{(j)}$. Since the anchor token has no location, we set $b_0 = 0$. This geodesic distance increases or decreases the attention on the corresponding context tokens depending on the sign of λ_t, while its magnitude may vary along the trajectory. We initialize $\boldsymbol{\omega}$ and β to zero, so the bias b_j initially leaves the logits unchanged. The construction extends ALiBi (Press et al., 2022), which adds a distance-based bias to the attention logits. With this term, the refined conditioning vector also depends on the current sample, $\mathbf{c}_t = g_\theta(\mathbf{x}, t, \mathcal{C}_x, \mathbf{y}_t)$, while the velocity network $\mathbf{v}_\theta(\mathbf{y}_t \mid \mathbf{c}_t)$ is unchanged.

4 Experiments

4.1 Experimental Setup

Training and inference details are in Appendix A.

Datasets. Street-view images: OSV-5M (Astruc et al., 2024) provides 4.9 M training images and 210K test queries; its split separates capture sequences and excludes training images within 1 km of a test query. Flickr images: following the standard practice, we train on the 4.7 M photographs of MP-16 (Müller-Budack et al., 2018) and evaluate on the YFCC4k and Im2GPS3k queries (Vo et al., 2017). In all datasets, the entire training set also forms our database unless otherwise stated.

Models. We use frozen StreetCLIP (Haas et al., 2023) for image descriptors on OSV-5M as a specialist encoder, and DINOv3-L (Siméoni et al., 2025), SigLIP2-so400m (Tschannen et al., 2025) and Perception Encoder PE-Core-G (Bolya et al., 2025) on both OSV-5M and MP-16 as general vision foundation models. The velocity network is of width $d = 768$ and depth 18, resulting in 119 M parameters, while the context module has 16 M additional parameters. Dimensionality D varies per encoder. Unless stated otherwise, the context holds the $K = 64$ most similar database images, retrieved once per query and the context images are fixed during generation.

Prediction. A prediction is a sample generated with classifier-free guidance; details are given in Appendix A. We further apply the post-processing of Vo et al. (2017): if any context image lies within a fixed radius of the prediction, the prediction is replaced by the location of the most similar such image. The radius value is given in Appendix A, and Tables 7 and 8 report results without this step.

Metrics. We report country, region, subregion and city accuracy on OSV-5M (mapping predicted coordinates to geographic entities according to the standard protocol), and accuracy within 1, 25, 200, 750 and 2500 km on the other datasets. GeoScore and the mean great-circle error (distance in km) summarize the whole error distribution. Definitions are in Appendix A.1.

Compared methods. Among predictors built on a vision encoder, we compare with GeoCLIP (Vivanco Cepeda et al., 2023), LocDiff (Wang et al., 2025), HierLoc (Gadi et al., 2026), GeoSURGE (Daruna et al., 2026), and GeoToken (Ghasemi et al., 2025). Among systems that use an MLLM at inference, we compare with GeoRanker (Jia et al., 2025) and the MLLM variant of GeoToken. For all of them, we report their published numbers; older methods are omitted from the main tables, but Appendix C lists every method with additional metrics. RFM $\mathcal{S}_2$ denotes our reproduction of Dufour et al. (2025) at its larger size, using the official code, *i.e.* the *baseline model*

Table 1: Geolocation performance on OSV-5M, grouped by network used for descriptors. RFM $\mathcal{S}_2$ is the context-free baseline. Best result in bold, second best underlined. HierLoc does not state which DINOv3 variant is used.

				Administrative accuracy (%) ↑			
Method	Backbone	GeoScore ↑	Dist. (km) ↓	Country	Region	Subreg	City
1-NN	StreetCLIP	3645	1337	74.6	47.1	29.2	20.8
RFM $\mathcal{S}_2$	StreetCLIP	3985	936	81.0	54.2	31.1	17.5
HierLoc	StreetCLIP	3862	1051	80.3	53.1	39.2	22.5
RAFMa	StreetCLIP	4062	858	82.8	58.4	37.6	26.1
RFM $\mathcal{S}_2$	DINOv3-L	4012	926	80.9	55.7	33.0	19.1
HierLoc	DINOv3	3963	861	82.9	55.0	**40.7**	23.3
RAFMa	DINOv3-L	4085	852	82.5	59.5	38.9	**26.9**
RFM $\mathcal{S}_2$	SigLIP2-so400m	4021	862	81.8	54.5	31.2	17.8
RAFMa	SigLIP2-so400m	4073	807	83.1	57.5	36.3	24.5
1-NN	PE-Core-G	3895	924	80.9	50.7	31.1	21.7
RFM $\mathcal{S}_2$	PE-Core-G	4285	490	87.4	59.6	34.8	19.7
RAFMa	PE-Core-G	**4342**	**442**	**88.8**	**62.8**	39.8	26.8

Table 2: Geolocation performance on the two Flickr benchmarks, grouped by network used for descriptors. RFM $\mathcal{S}_2$ is the context-free baseline. Best result in bold, second best underlined. Systems that use an MLLM at inference are listed separately, listing the model in the second column. HierLoc does not state which DINOv3 variant is used, nor LocDiff which CLIP variant.

Method	Backbone	YFCC4k					Im2GPS3k				
accuracy (%) within km ↑		1	25	200	750	2500	1	25	200	750	2500
GeoCLIP	CLIP-L	9.6	19.3	32.6	55.0	74.7	14.1	34.5	50.7	69.7	83.8
LocDiff	CLIP	–	33.3	46.7	65.2	79.9	10.9	34.0	53.3	72.5	85.2
GeoSURGE	CLIP-L	19.9	33.6	48.7	67.4	82.0	17.2	42.5	58.1	74.6	87.6
GeoToken	CLIP-L	24.3	35.3	46.6	64.2	78.6	16.8	39.6	53.8	70.8	85.0
RFM $\mathcal{S}_2$	DINOv3-L	2.5	23.7	40.6	60.2	78.2	3.5	36.0	51.8	68.9	83.9
HierLoc	DINOv3	8.4	30.2	43.3	61.7	75.8	11.3	43.8	58.4	74.1	85.1
RAFMa	DINOv3-L	28.9	39.5	48.3	63.6	79.8	17.5	42.1	54.4	70.2	84.4
RFM $\mathcal{S}_2$	SigLIP2-so400m	3.1	28.7	47.3	66.9	81.9	5.0	40.1	57.5	75.5	88.9
RAFMa	SigLIP2-so400m	32.3	44.7	55.8	70.2	83.8	15.1	42.8	58.8	75.6	87.8
1-NN	PE-Core-G	34.5	45.7	55.3	67.6	80.1	17.8	44.1	58.5	73.1	85.8
RFM $\mathcal{S}_2$	PE-Core-G	5.2	37.7	59.7	77.6	88.4	7.4	48.7	70.2	84.9	**93.6**
RAFMa	PE-Core-G	**39.8**	**56.1**	**68.4**	**80.4**	**89.1**	18.8	**52.0**	**70.5**	**85.1**	**93.6**
Multimodal Large Language Model systems											
GeoRanker[‡]	Qwen2-VL-7B	32.9	43.5	54.3	69.8	82.5	18.8	45.1	61.5	76.3	89.3
GeoToken	Gemini 2.0 Flash	25.4	38.5	51.4	68.0	81.0	**19.0**	46.0	60.1	76.6	88.8

[‡]GeoRanker additionally uses GPT-4V to generate candidates.

that uses no context. It is trained on the same descriptors, with the same recipe and the same velocity network architecture as RAFMa, so that every block of Tables 1 and 2 pairs RAFMa with its context-free counterpart. 1-NN predicts the location of the most similar database image.

4.2 Results

Tables 1 and 2 compare RAFMa with its context-free baseline and with published methods. On OSV-5M, the context improves every metric over the baseline for each of the four encoders, and RAFMa sets a new state of the art on every metric except subregion accuracy, where HierLoc leads. Note that HierLoc supervises for and predicts within the same set of discrete geographic entities used in the evaluation protocol, whereas we map coordinates to the nearest entity. On the Flickr benchmarks, the

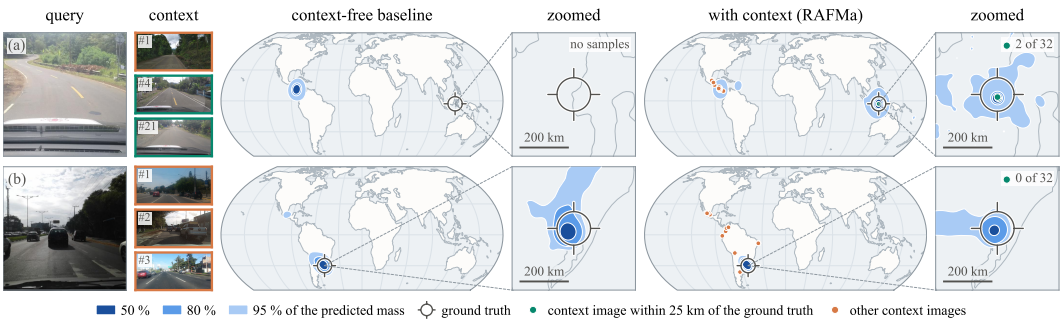


Figure 3: **Predicted distributions with RAFMa and with the context-free baseline model**, for two OSV-5M test queries. On the left we depict the query and three example images specifying their rank in the sorted context. Green border depicts examples that are within 25 km of the ground truth. Both models use the same training set, which is also used as database for RAFMa. Distributions obtained by generating 2,048 samples and performing KDE.

Table 3: **Ablation study** on OSV-5M with StreetCLIP. Models are trained using the same recipe. W/o geographic tokens: use visual tokens as values.

model	GeoScore ↑	Dist. (km) ↓	Administrative accuracy (%) ↑			
			Country	Region	Subregion	City
RAFMa	**4062**	**858**	**82.8**	**58.4**	**37.6**	**26.1**
w/o geographic tokens	4026	861	81.8	55.2	33.2	19.9
w/o geographic distance bias	4023	886	81.9	56.3	35.1	23.3
w/o geographic tokens & distance bias	4011	885	81.5	54.8	32.7	19.0
w/o context (Dufour et al., 2025)	3985	936	81.0	54.2	31.1	17.5

gains over the baseline are largest at fine scales and diminish at the coarsest level, where performance is close to saturation. With SigLIP2-so400m descriptors, RAFMa is competitive on YFCC4k with MLLM systems of several billion parameters, while training only 135M parameters on top of a 400M frozen encoder. With PE-Core-G descriptors, RAFMa outperforms the MLLM systems at every radius on YFCC4k, and from 25 km upward on Im2GPS3k. This also shows that RAFMa benefits from stronger vision encoders without any increase in the size of its trainable part, *i.e.* the context module and the velocity network. Given the same retrieval, RAFMa outperforms GeoRanker at 25 km and beyond in every setting; see Appendix B.2.

Predicted distributions. Figure 3 shows how the context affects the predicted distribution. In (a), the baseline is confidently wrong, placing its mass on another continent; two context images near the true location, at ranks 4 and 21, suffice to move most of the mass there, even though the most similar one lies thousands of kilometers away. In (b), no context image lies near the true location, and the model keeps the distribution where the baseline places it, ignoring the misleading context. In neither case does the model copy the most similar image or average the context. Appendix B.3 shows more examples and reports how often each situation occurs.

Component contributions. Table 3 removes some of the components of the context module. Both the use of geographic representation as values in the cross attention and the geographic distance bias in the attention logits play an essential role for high performance. Further analysis of how the trained model uses its context is shown in Appendix B.

Training data and database size. Figure 4 (left) varies the training data while RAFMa retrieves from the complete database. Trained on 5% of the images, RAFMa already surpasses the baseline trained on all of them. Figure 4 (right) varies the database of three RAFMa models. The baseline trained on all images is outperformed by RAFMa with just 25% of the training images when its database is restricted to the same subset. Accuracy keeps rising with the database, so new labeled images improve the model without retraining. For the models trained on 5% and 25%, they recover

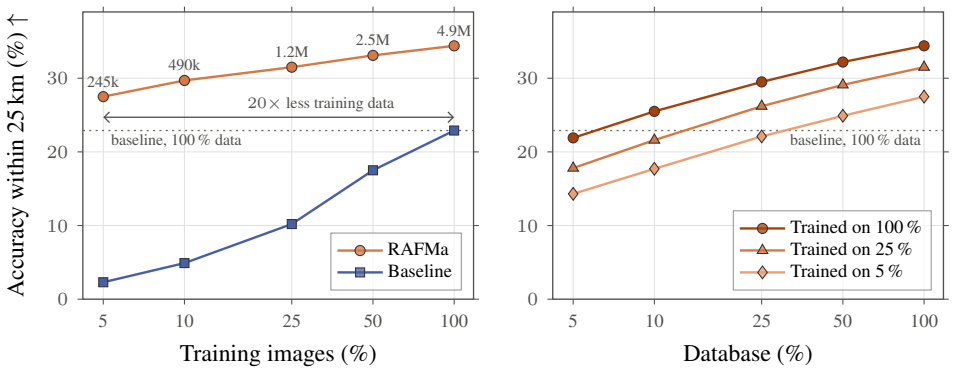


Figure 4: **Varying training set and database size** on OSV-5M with StreetCLIP descriptors. (left) Both models trained on the same nested subsets of the training set. The baseline model is context-free, while RAFMa retrieves from the full database, which is equal to the complete training set. (right) Three RAFMa models, trained on 5 %, 25 % and 100 % of the training set, each evaluated using nested subsets of the database of increasing size. The 5 % and 25 % models increase the database size with images never used in training. Dotted line used for the context-free baseline trained on the full training set.

about two thirds of the gain that training on them would bring. Moreover, with a fixed database and a frozen RAFMa model, improving retrieval alone boosts performance significantly. This is achieved by merging the contexts from multiple encoders; see Appendix B.1.

Progressively increasing the context. Figure 5 progressively reveals the fixed set of 32 context images on OSV-5M queries. We perform this in two ways, from first to last, starting from the most visually similar images to the query, and from last to first. In the former case, which reflects the way our model operates in practice, we observe a rapid performance increase and then a slower one, which nevertheless manages to take advantage of the increase in the context quality. In the latter case, which is synthetically created, we observe a slow but steady increase; the model benefits even with context of lower quality. Overall, gains in accuracy are consistent with the context quality.

5 Conclusion

We introduce RAFMa, a generative geolocation model that conditions spherical flow matching on retrieved geotagged images, treated as in-context demonstrations. Retrieval augmentation proves highly effective; it consistently improves over the same generator without context, at all scales, and allows the model to exploit geotagged images it never saw during training, reaching the accuracy of its context-free counterpart with a fraction of the training data. RAFMa sets a new state of the art on OSV-5M, and on the Flickr benchmarks it is competitive with systems built on MLLMs with billions of parameters, while training only 135M parameters. Finally, RAFMa benefits significantly from stronger and larger vision encoders, without any change to the architecture or the size of its trainable part. This makes retrieval-augmented generation a simple and scalable recipe for geolocation, whose knowledge grows with the database and whose accuracy grows with the representation quality.

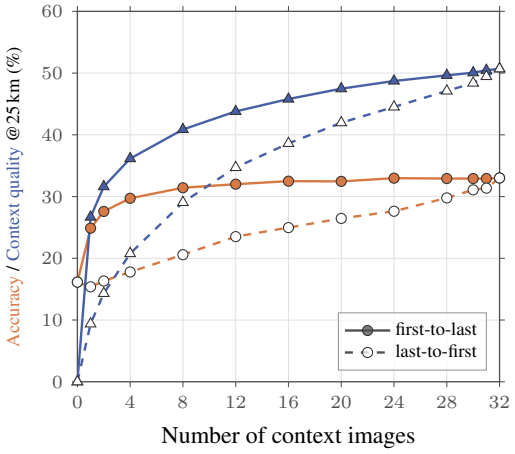


Figure 5: **Progressive context reveal on OSV-5M.** Context quality: the fraction of queries with at least one context image within 25 km of the true location.

AI USE STATEMENT

We used large language models as coding assistants, to write analysis scripts, and to edit the wording of the manuscript. They were not used to generate research ideas or results. The authors reviewed all generated code and text and take full responsibility for the content.

ETHICS STATEMENT

Visual geolocation is dual-use. When a photograph can be linked to a person, inferring where it was taken can reveal where that person has been, and Liu et al. (2025) show that off-the-shelf vision–language models already make this practical for non-experts. Retrieval moves part of this risk into the database, since a query is localised through other geotagged photographs of the same place, on Flickr including the same user's own uploads. We collect no new data and use only the public OSV-5M (Astruc et al., 2024), MP-16 (Müller-Budack et al., 2018), YFCC4k, and Im2GPS3k (Vo et al., 2017) benchmarks under their released licences. We do not intend the method for locating individuals without their consent.

REFERENCES

Guillaume Astruc, Nicolas Dufour, Ioannis Siglidis, Constantin Aronssohn, Nacim Bouia, Stephanie Fu, Romain Loiseau, Van Nguyen Nguyen, Charles Raude, Elliot Vincent, et al. OpenStreetView-5M: The many roads to global visual geolocation. In *CVPR*, 2024. 1, 2, 6, 10, 20

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. In *NeurIPS*, 2025. 6

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *ICML*, 2022. 2

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 2, 4

Ron Campos, Ashmal Vayani, Parth Parag Kulkarni, Rohit Gupta, Aizan Zafar, Aritra Dutta, and Mubarak Shah. GAEA: A geolocation aware conversational assistant. In *WACV*, 2026. 2

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *ICLR*, 2024. 2, 3

Brandon Clark, Alec Kerrigan, Parth Parag Kulkarni, Vicente Vivanco Cepeda, and Mubarak Shah. Where we are and what we're looking at: Query based worldwide image geo-localization using hierarchies and scenes. In *CVPR*, 2023. 2, 20

Angel Daruna, Nicholas Meegan, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Geo-SURGE: Geo-localization using semantic fusion with hierarchy of geographic embeddings. In *CVPR*, 2026. 2, 6

Nicolas Dufour, David Picard, Vicky Kalogeiton, and Loic Landrieu. Around the world in 80 timesteps: A generative approach to global visual geolocation. In *CVPR*, 2025. 1, 2, 3, 4, 6, 8, 20, 22

Hari Krishna Gadi, Daniel Matos, Hongyi Luo, Lu Liu, Yongliang Wang, Yanfeng Zhang, and Liqiu Meng. HierLoc: Hyperbolic entity embeddings for hierarchical visual geolocation. In *ICLR*, 2026. 2, 6

Narges Ghasemi, Amir Ziashahabi, Salman Avestimehr, and Cyrus Shahabi. GeoToken: Hierarchical geolocalization of images via next token prediction. In *ICDM*, 2025. 3, 6

Lukas Haas, Silas Alberti, and Michal Skreta. Learning generalized zero-shot learners for open-domain image geolocalization. In *arXiv*, 2023. 2, 6

Lukas Haas, Michal Skreta, Silas Alberti, and Chelsea Finn. PIGEON: Predicting image geolocations. In *CVPR*, 2024. 3, 20

James Hays and Alexei A. Efros. IM2GPS: Estimating geographic information from a single image. In *CVPR*, 2008. 2, 3

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *arXiv*, 2022. 13

Mike Izbicki, Evangelos E. Papalexakis, and Vassilis J. Tsotras. Exploiting the Earth's spherical geometry to geolocate images. In *ECML PKDD*, 2019. 2

Pengyue Jia, Yiding Liu, Xiaopeng Li, Yuhao Wang, Yantong Du, Xiao Han, Xuetao Wei, Shuaiqiang Wang, Dawei Yin, and Xiangyu Zhao. G3: An effective and adaptive framework for worldwide geolocalization using large multi-modality models. In *NeurIPS*, 2024. 3, 15, 20

Pengyue Jia, Seongheon Park, Song Gao, Xiangyu Zhao, and Sharon Li. GeoRanker: Distance-aware ranking for worldwide image geolocalization. In *NeurIPS*, 2025. 3, 6, 14

Giorgos Kordopatis-Zilos, Panagiotis Galopoulos, Symeon Papadopoulos, and Ioannis Kompatsiaris. Leveraging EfficientNet and contrastive learning for accurate global-scale location estimation. In *ICMR*, 2021. 3

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020. 2

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. 2, 3

Yi Liu, Gelei Deng, Junchen Ding, Yuekang Li, Tianwei Zhang, Weisong Sun, Yaowen Zheng, and Jingquan Ge. Mission: Impossible – image-based geolocation with large vision language models. *Proceedings on Privacy Enhancing Technologies*, 2025. 2, 10

Eric Müller-Budack, Kader Pustu-Iren, and Ralph Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In *ECCV*, 2018. 1, 2, 6, 10, 20

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 4

Shraman Pramanick, Ewa M. Nowara, Joshua Gleason, Carlos D. Castillo, and Rama Chellappa. Where in the world is this image? Transformer-based geo-localization in the wild. In *ECCV*, 2022. 2, 20

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *ICLR*, 2022. 6

Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. CPlaNet: Enhancing image geolocalization by combinatorial partitioning of maps. In *ECCV*, 2018. 2, 20, 21

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. In *arXiv*, 2025. 6

Jonas Theiner, Eric Müller-Budack, and Ralph Ewerth. Interpretable semantic photo geolocation. In *WACV*, 2022. 2

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. In *arXiv*, 2025. 6

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

Vicente Vivanco Cepeda, Gaurav Kumar Nayak, and Mubarak Shah. GeoCLIP: Clip-inspired alignment between locations and images for effective worldwide geo-localization. In *NeurIPS*, 2023. 2, 6

Nam Vo, Nathan Jacobs, and James Hays. Revisiting IM2GPS in the deep learning era. In *ICCV*, 2017. 2, 3, 6, 10, 13

Zhangyu Wang, Zeping Liu, Jielu Zhang, Zhongliang Zhou, Qian Cao, Nemin Wu, Lan Mu, Yang Song, Yiqun Xie, et al. LocDiff: Identifying locations on Earth by diffusing in the Hilbert space. In *NeurIPS*, 2025. 2, 3, 6, 20

Tobias Weyand, Ilya Kostrikov, and James Philbin. PlaNet – photo geolocation with convolutional neural networks. In *ECCV*, 2016. 1, 2, 20

Biao Wu, Meng Fang, Ling Chen, Ke Xu, Tao Cheng, and Jun Wang. Vision-language reasoning for geolocalization: A reinforcement learning approach. In *AAAI*, 2026. 2, 20

Zhongliang Zhou, Jielu Zhang, Zihan Guan, Mengxuan Hu, Ni Lao, Lan Mu, Sheng Li, and Gengchen Mai. Img2Loc: Revisiting image geolocalization using multi-modality foundation models and image-based retrieval-augmented generation. In *SIGIR*, 2024. 3, 20

A Training and Inference

Objective. We train the context network g_θ and the velocity network $\mathbf{v}_\theta$ jointly from scratch by minimizing Equation (2) with $\mathbf{c}_t = g_\theta(\mathbf{x}, t, \mathcal{C}_x, \mathbf{y}_t)$ as the conditioning vector. Time is sampled as $t = \psi(u)$ with $u \sim \mathcal{U}[0,1]$, where ψ is a monotone sigmoid warp of $[0,1]$ onto itself, and the regression target is the velocity with respect to t. During training, the training set also serves as the database, and the entry of the query itself is removed from its context. The image encoder stays frozen throughout training; the projections on top of it, ϕ in the context-free baseline and h_v, h_s, h_p in ours, are learned.

Guided sampling. At inference we encode the query, retrieve its context once, and sample with classifier-free guidance (Ho & Salimans, 2022). The velocity in the update of Equation (3) is replaced by the guided field

$$\mathbf{v}^w_\theta(\mathbf{y}_t \mid \mathbf{c}_t, \mathbf{c}^\emptyset_t) = (1+w)\,\mathbf{v}_\theta(\mathbf{y}_t \mid \mathbf{c}_t) - w\,\mathbf{v}_\theta(\mathbf{y}_t \mid \mathbf{c}^\emptyset_t), \tag{7}$$

where $\mathbf{c}_t = g_\theta(\mathbf{x}, t, \mathcal{C}_x, \mathbf{y}_t)$ and $\mathbf{c}^\emptyset_t = g_\theta(\emptyset, t, \emptyset, \mathbf{y}_t)$. Here $\emptyset$ marks a dropped input, so the second term is the unconditional field, evaluated without query or context content. A single scale $w \geq 0$ acts on the query and the context jointly, and $w = 0$ recovers the conditional field. The time grid is warped as in training, $t_n = \psi(n/L)$. Guidance changes the distribution that is sampled: $w = 0$ samples the learned conditional model, while positive guidance substantially reduces sample diversity in our experiments (Appendix B.3). Each raw point prediction uses one guided draw per query with $w = 2$ and $L = 250$ steps.

Conditioning dropout. Query dropout zeros the query and its context, which trains the unconditional field of Equation (7). Context dropout zeros the whole context, and context-image and location dropout zero individual context images or only their locations, so the model does not rely on any single context image. Dropped inputs keep their slots and rank embeddings, and a zeroed location has no geodesic bias. Rates are listed in Table 9.

Prediction post-processing. The generated coordinate is continuous, so at the finest scale it can land beside a real place rather than on it, in the water off a coastal city for instance. Following Vo et al. (2017), we therefore snap it to a context image. We scan the context in retrieval order and output the location of the first image within a fixed radius of the prediction. If no image is that close, the prediction is kept. The radius is selected on training queries of each domain and fixed before benchmark evaluation. It maximizes accuracy at 1 km, averaged over all encoders, while no encoder loses more than 0.1 percentage points at 25 km. This gives 2 km for OSV-5M and 10 km for YFCC4k and Im2GPS3k. Tables 7 and 8 report our models with and without this step.

A.1 Evaluation

Coordinate metrics. Let $\delta_x = 6371\,\delta(\hat{\mathbf{y}}_x, \mathbf{y}_x)$ be the great-circle error in kilometres of the prediction $\hat{\mathbf{y}}_x$ for query image x, with δ the geodesic distance of Section 3.3. Over the set $\mathcal{Q}$ of test queries, we report the mean error and

$$\text{GeoScore} = \frac{1}{|\mathcal{Q}|} \sum_{x \in \mathcal{Q}} 5000 \exp(-\delta_x / 1492.7).$$

GeoScore is bounded between 0 and 5000, with sensitivity to distance decreasing exponentially. Accuracy at radius r is the percentage of queries with $\delta_x < r$.

Administrative accuracy. For our coordinate predictions, country, region, subregion, and city accuracy compare reverse-geocoded labels with the query labels. Each level uses only queries with a ground-truth label at that level.

B Additional Context Analysis

B.1 RAFMa with Multiple Retrieval Encoders

The context of RAFMa is retrieved from a database, so it can be improved after training. Table 4 keeps a trained StreetCLIP model fixed and only changes the encoders that select its context images, while the model still reads their StreetCLIP descriptors. Selecting the context with additional encoders improves every metric, and each added encoder helps further. Retrieving with the concatenated descriptors of StreetCLIP, DINOv3-L and PE-Core-G raises GeoScore from 4062 to 4152, lowers the mean error from 858 to 770 km and raises city accuracy from 26.1 to 28.6 %. The performance of a trained model can thus be increased substantially without any retraining. This is practical: when a stronger encoder appears, it suffices to extract its descriptors for the queries and the database images and use them for retrieval.

Table 4: **RAFMa with Multiple Retrieval Encoders on OSV-5M.** The same frozen StreetCLIP model receives $K = 64$ context images retrieved with the listed encoders. The model always reads StreetCLIP descriptors, and cosine similarities are recomputed with StreetCLIP to match training. Union merges the context lists of the encoders round-robin, skipping duplicates (32 per encoder for a pair, 22/21/21 for all three). Concatenation retrieves with the L2-normalized descriptors of all encoders, concatenated and PCA-whitened to 768 dimensions on the database images.

Retrieval encoders	Combination	GeoScore ↑	Dist. (km) ↓	Administrative accuracy (%) ↑			
				Country	Region	Subreg	City
StreetCLIP	–	4062	858	82.8	58.4	37.6	26.1
StreetCLIP + DINOv3-L	Union	4101	825	83.5	60.1	39.4	27.7
StreetCLIP + PE-Core-G	Union	4107	809	83.8	59.8	39.1	27.3
StreetCLIP + DINOv3-L + PE-Core-G	Union	4121	802	84.0	60.7	40.0	28.2
StreetCLIP + DINOv3-L + PE-Core-G	Concatenation	**4152**	**770**	**84.7**	**61.5**	**40.7**	**28.6**

B.2 RAFMa and GeoRanker Comparison with Shared Retrieval

GeoRanker (Jia et al., 2025) re-ranks retrieved candidates with an MLLM and returns the location of the top one. In Table 5, GeoRanker and RAFMa start from the same 20 retrieved images, in the same order, but use them in different roles. GeoRanker, as released, returns the location of one of the first 14 (YFCC4k) or 12 (Im2GPS3k) images, sees ranks 16–20 only as textual negatives, and ignores the ranks in between. RAFMa receives all 20 images as context and predicts a location anywhere on the globe, which the post-processing may replace by the location of any of the 20. The comparison thus fixes the retrieval, not the candidate sets. Regime (a) retrieves from the full database, while regime (b) removes every database image within 1 km of the query, as the OSV-5M split does by design; it thus separates generalizing from the context from selecting a near-duplicate of the query. The removal uses the ground truth, so it is a stress test rather than an inference procedure.

From 25 km upward, RAFMa outperforms GeoRanker in both regimes, on both benchmarks and with either retrieval, and the margin grows once the nearby images are removed. At 1 km, both methods depend on selecting a context image taken at the query location. With the full database, RAFMa is ahead on YFCC4k and 2–4 points behind on Im2GPS3k. Without nearby images, nothing is left to select: GeoRanker drops to zero and our post-processing below 1 %, while our prediction itself still reaches 3.1–6.0 %. RAFMa thus generalizes from the context rather than only choosing among it.

B.3 Predicted Distributions: Qualitative Examples and Analysis

Figures 6 to 7 compare the distributions predicted by the context-free baseline and by RAFMa on ten OSV-5M test queries. They are drawn without guidance, since guidance concentrates the samples around a single point. On OSV-5M, the median distance between draws of the same query falls from 93 km without guidance to about 1 m with $w = 2$. The examples show three behaviours. A few correct context images move a confidently wrong prediction to the ground truth, even when the most similar context image is far away (Figure 6). Context images select and sharpen the correct

Table 5: **RAFMa and GeoRanker comparison with shared retrieval.** 1-NN, retrieval-only GeoRanker and RAFMa retrieve from the MP16-Pro database (GeoRanker's 4.12 M-image MP-16 subset) with PE-Core-G or with G3 (Jia et al., 2024), the retrieval GeoRanker uses to collect its candidates. RAFMa receives all 20 of GeoRanker's retrieved candidates as its context ($K = 20$), in the same order. GeoRanker is used as released. It selects among the first 14 (YFCC4k) or 12 (Im2GPS3k) candidates, uses ranks 16–20 only as textual negatives, and ignores the ranks in between. The frozen models are evaluated with (a) the full database and (b) every database image within 1 km of the query removed and replaced by the next retrieved one. Best result per column and panel in bold, second best underlined.

Method	Retrieval	YFCC4k					Im2GPS3k				
accuracy (%) within km ↑		1	25	200	750	2500	1	25	200	750	2500
(a) Full database											
1-NN	PE-Core-G	32.5	43.5	53.3	66.2	79.6	17.2	43.5	57.7	72.6	86.0
GeoRanker	PE-Core-G	35.6	46.8	57.5	72.0	83.7	**21.6**	49.3	65.3	80.4	91.1
RAFMa w/o post-processing	PE-Core-G	20.2	**52.7**	**66.4**	**79.6**	**88.8**	8.1	**51.6**	**70.4**	85.1	93.6
RAFMa	PE-Core-G	**36.5**	52.5	**66.4**	**79.6**	**88.8**	18.0	51.4	**70.4**	85.2	93.6
1-NN	G3	27.6	36.2	44.8	60.1	75.0	12.5	33.9	45.0	61.3	78.5
GeoRanker	G3	32.3	42.9	53.5	68.5	81.4	18.6	43.6	59.8	75.4	88.2
RAFMa w/o post-processing	G3	10.7	49.3	65.6	78.9	88.6	6.9	50.1	70.1	**85.3**	**94.0**
RAFMa	G3	32.5	49.3	65.5	78.9	88.6	16.2	50.3	69.9	**85.3**	**94.0**
(b) Database images within 1 km of the query excluded											
1-NN	PE-Core-G	0.0	20.5	35.2	54.3	72.8	0.0	37.6	53.7	70.6	84.9
GeoRanker	PE-Core-G	0.0	23.7	40.3	62.0	78.8	0.0	44.1	62.4	79.1	90.6
RAFMa w/o post-processing	PE-Core-G	3.1	33.1	55.4	**75.7**	**87.7**	4.2	**49.0**	69.6	84.8	93.4
RAFMa	PE-Core-G	0.3	32.9	55.3	**75.7**	**87.7**	0.2	48.5	69.6	84.8	93.4
1-NN	G3	0.0	14.9	26.9	48.0	67.5	0.0	28.3	41.2	59.3	77.5
GeoRanker	G3	0.0	21.1	37.1	59.1	76.4	0.0	38.7	57.1	74.3	87.7
RAFMa w/o post-processing	G3	**4.4**	**33.2**	**56.2**	75.6	87.6	**6.0**	**49.0**	**69.7**	**85.0**	**93.9**
RAFMa	G3	0.9	33.1	**56.2**	75.6	87.6	0.7	48.9	69.6	**85.0**	**93.9**

mode of an ambiguous prediction, and when every context image is far away, the distribution stays where the image alone places it (Figure 7).

Table 6 shows how often these situations occur. The first two behaviours need a correct context image that is not the most similar one, which holds for a quarter of the queries; there, the context raises accuracy at 25 km from 27.7 to 40.3 %. When the most similar context image is correct, accuracy rises from 52.2 to 89.4 %. When no context image is within 200 km, both models are almost always wrong. The context hurts noticeably only when its nearest image lies 25–200 km from the ground truth, lowering accuracy from 6.4 to 1.8 %.

Table 6: **Context quality analysis.** The 210,122 OSV-5M test queries are grouped by where their 32 context images lie relative to the ground truth. The last three columns give, within each group, the percentage of queries whose prediction is within 25 km of the ground truth, for RAFMa, the context-free baseline, and the location of the most similar context image. Predictions are single guided draws without post-processing.

			accuracy at 25 km (%)		
context images	queries	share	RAFMa	baseline	top-1 image
top-1 within 25 km	55,778	26.5 %	89.4	52.2	100
top-1 beyond 25 km, another within 25 km	51,758	24.6 %	40.3	27.7	0
none within 25 km, some within 200 km	63,132	30.0 %	1.8	6.4	0
none within 200 km	39,454	18.8 %	1.0	1.4	0
all queries	210,122	100 %	34.4	22.9	26.5

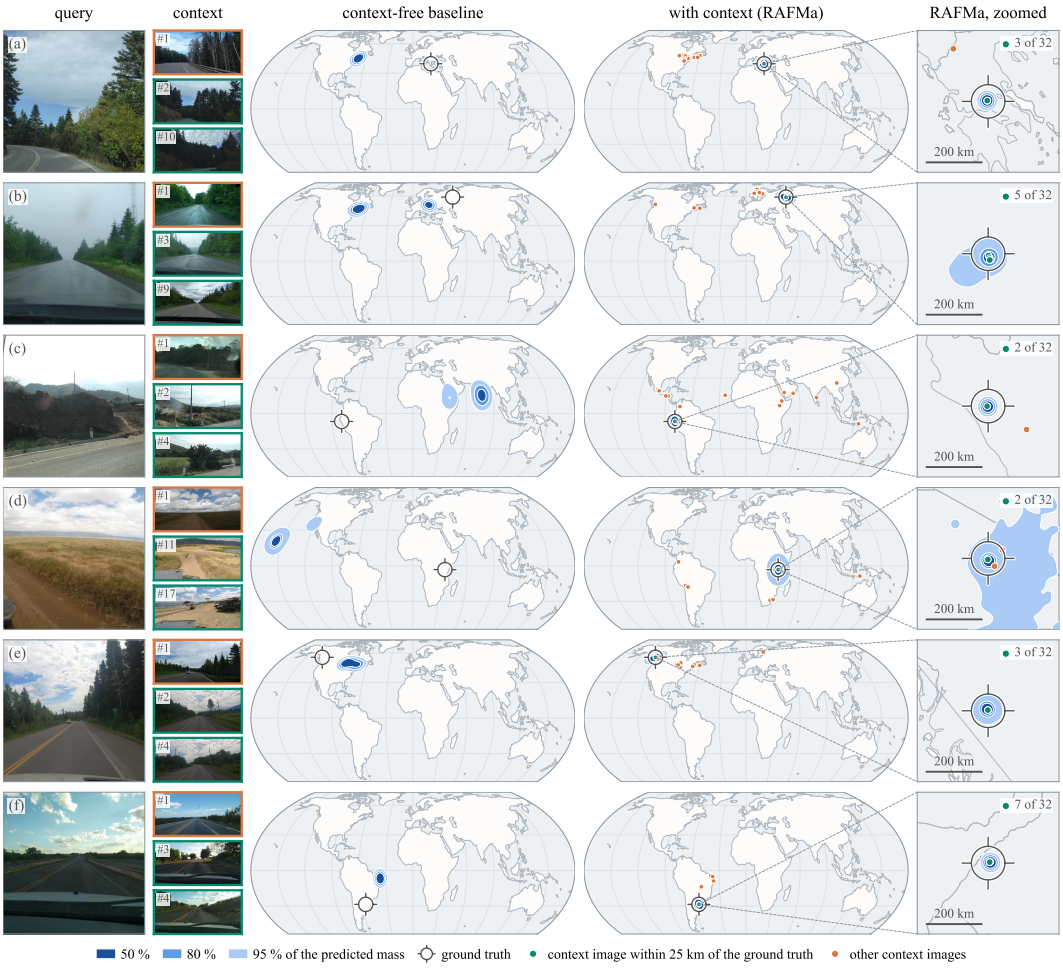


Figure 6: **A few correct context images override a confident but wrong baseline prediction.** The context-free baseline places its mass on another continent. With 32 context images, of which the most similar is wrong and only a few are within 25 km of the ground truth (green, with their ranks, count in the zoom panel), RAFMa moves most of its mass to the ground truth. Both models use StreetCLIP descriptors. Shaded regions hold 50, 80 and 95 % of 2,048 unguided samples under a von Mises–Fisher kernel, 500 km wide on the world maps and 20 km in the zoom panels. Images from OSV-5M (Mapillary, CC-BY-SA).

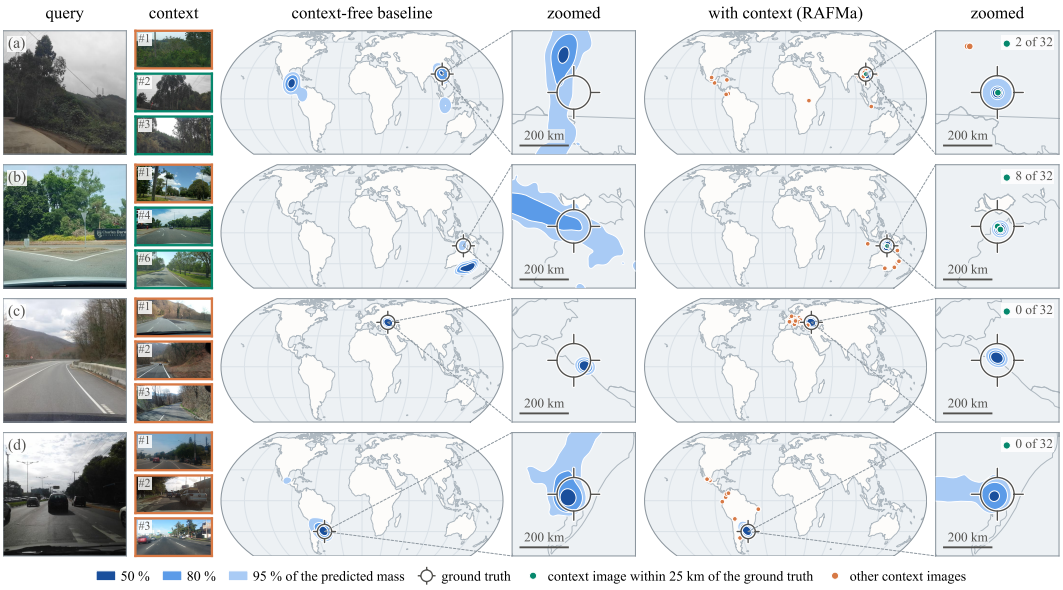


Figure 7: **Context images may resolve an ambiguous image, and far context images can be ignored.** In rows (a) and (b), the context-free baseline is split between regions with a diffuse minor mode near the ground truth, which the context images select and sharpen. In rows (c) and (d), no context image is within 200 km of the ground truth and our distribution stays where the baseline places it. Drawn as Figure 6, with a zoom panel for the baseline as well.

B.4 Correlation between Prediction Error and Nearest Context-Image Distance

Figure 8 relates the error of each prediction to how close its context comes to the true location. On all three benchmarks, the error grows with the distance to the nearest context image (Spearman $\rho = 0.70$–0.78), so the model exploits a nearby context image when there is one. Eleven context images contain a nearby one more often than three and lower the median error on every benchmark, *e.g.* from 161 to 112 km on YFCC4k. The wide scatter at short distances shows, however, that a nearby context image does not guarantee an accurate prediction.

B.5 Attention and Context-Image Pull

Figure 9 tests whether the cross-attention weight of a context image indicates how much it influences the prediction with respect to its location. We mask one context image at a time and measure how much the velocity changes toward its location. Attention is correlated with the pull magnitude only for large values of attention, not for smaller ones. We observe that large attention weights occur either to strongly pull the prediction towards the image's location or push away from it.

B.6 The Learned Bias Schedule

Figure 10 shows how the learned geodesic bias of Equation (6) changes along the sampling trajectory. Over the first third of the steps, λ_t stays close to zero and even changes sign, so the attention is driven by appearance, while the locations still enter through the attention values. It then rises steeply and saturates near 650 over the last thirty steps, where a context image loses a factor e of weight at about 10 km from the current sample. Over the middle of the trajectory, the rise roughly follows $1/(1-t)$. This is the scale a distance bias would need to track the remaining distance: on the geodesic path of Equation (1), the sample $\mathbf{y}_t$ lies at $(1-t)\,\delta(\mathbf{y}_0, \mathbf{y}_1)$ from its target, so a bias whose width shrinks in proportion to this distance has $\lambda_t \propto 1/(1-t)$. The model is not given this form; it learns a similar schedule from data. Geographic weighting thus becomes more local as the sample approaches its final location. The bias only shifts the attention logits and imposes no hard geographic cutoff.

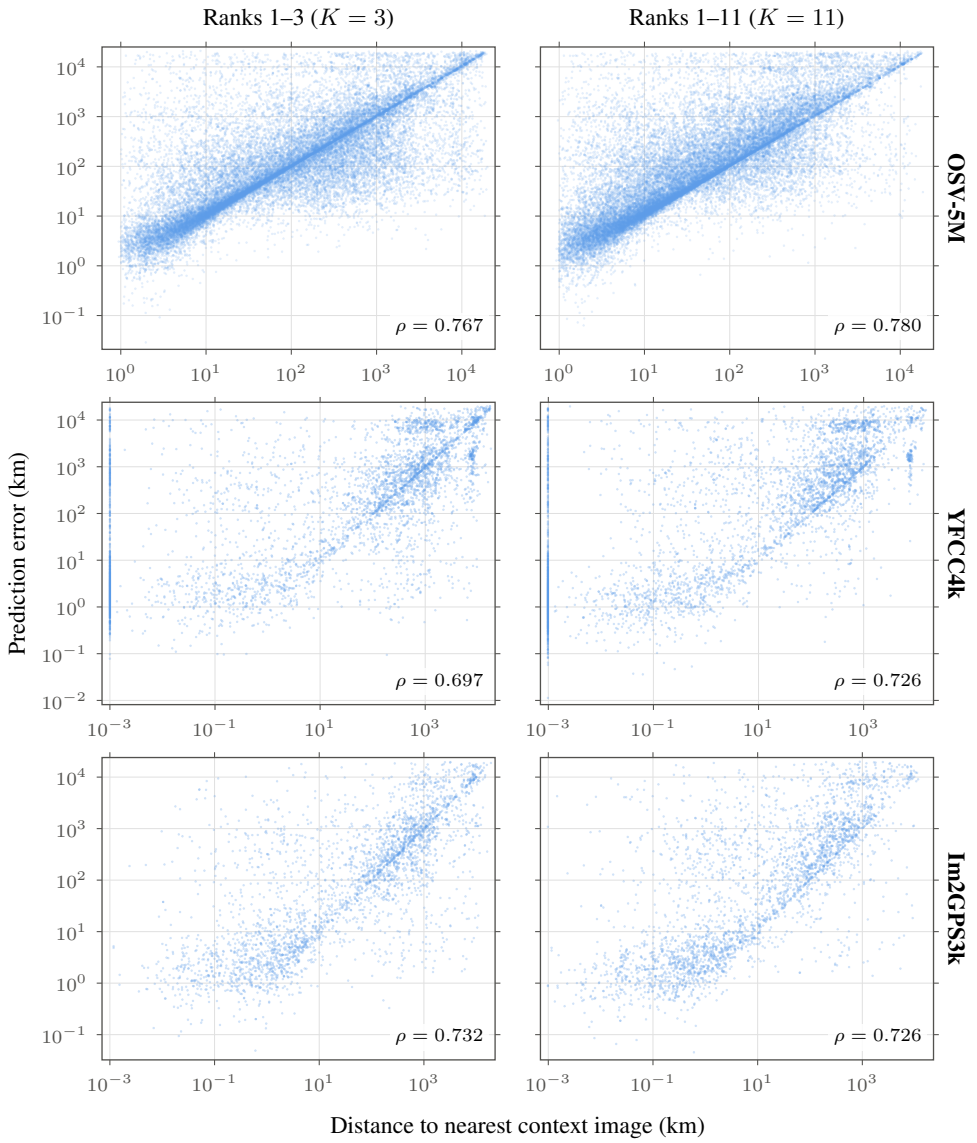


Figure 8: **Nearest context-image distance and prediction error.** Frozen models receive the context images at retrieval ranks 1–3 ($K = 3$, left) or 1–11 ($K = 11$, right). Each point is a query. The horizontal axis is the minimum distance from its true location across those context images, and the vertical axis is the error of the prediction generated using exactly the same context. The frozen model and sampler's initial noise are held fixed across settings. ρ is Spearman's rank correlation. OSV-5M uses every eighth test query. Both Flickr benchmarks use their full test sets. Distances below 1 m are shown at 1 m only for display; statistics use the original values.

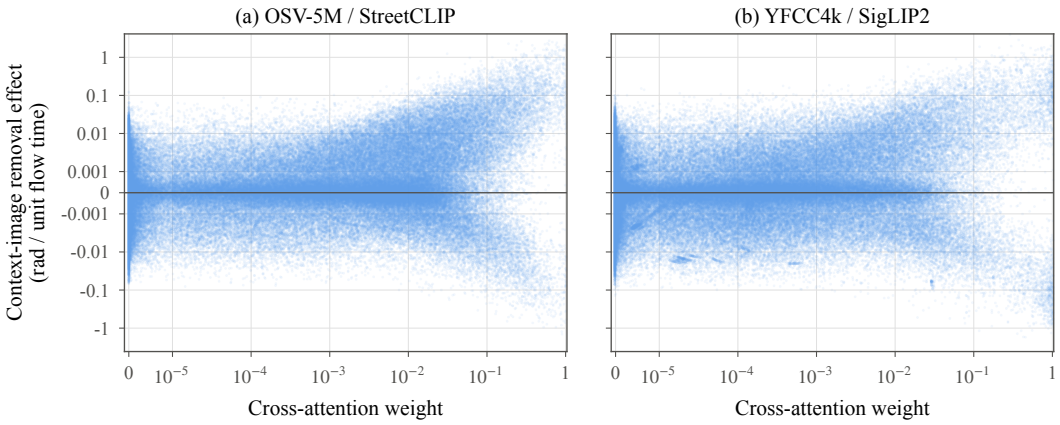


Figure 9: **Cross-attention and context-image removal effects** on 1,024 random test queries each of OSV-5M (StreetCLIP) and YFCC4k (SigLIP2-so400m). Each point is one context image of one query. The horizontal axis is its cross-attention weight, averaged across heads. The vertical axis is its removal effect: the difference between the conditional velocities with the full context and with this context image masked, projected onto the direction toward its location; positive effects point toward the context image. Both axes use logarithmic spacing away from zero, with linear regions for attention below 10^{-5} and effects within $\pm 10^{-3}$; all observations, including exact zeros, are shown.

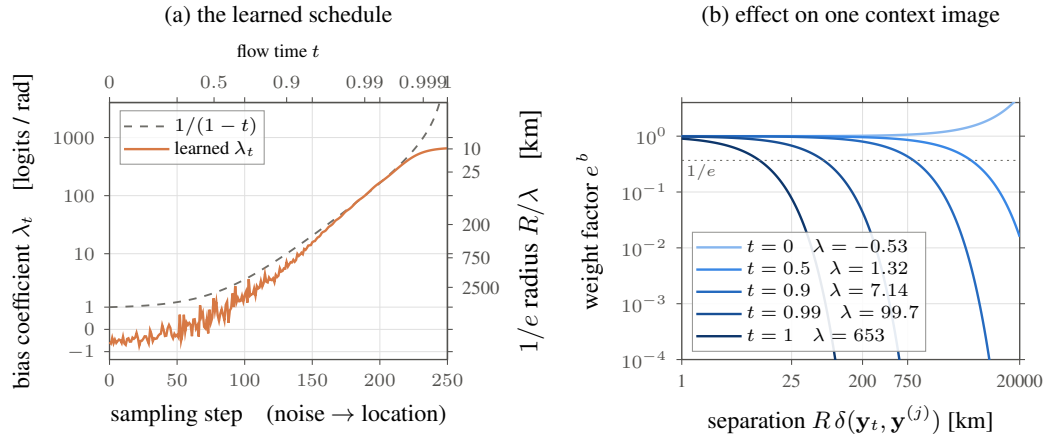


Figure 10: **The geodesic bias over the trajectory**, for the StreetCLIP model on OSV-5M. *(a)* The learned coefficient λ_t of Equation (6), evaluated at the 251 times of the $L = 250$ sampling grid; the lower axis is the step index and the upper axis the corresponding flow time, which the sigmoid time warp spaces unevenly. For $\lambda_t > 0$, the right-hand axis reads the same curve as R/λ_t, the separation in kilometres at which a context image loses one logit, with R the Earth's radius. *(b)* The resulting weight factor $e^{\,b}$ of a context image as a function of its separation from the current sample, at five times. Both panels are functions of the trained weights alone and involve no data.

C Complete Benchmark Results

Tables 7 and 8 extend Tables 1 and 2 with the methods omitted from the main text and with the metrics the main tables leave out: accuracy within radius on OSV-5M, and GeoScore and mean error on the Flickr benchmarks, for our models and their context-free baselines. The omitted methods, all with their published numbers, are ISNs (Müller-Budack et al., 2018), LocDiff (Wang et al., 2025), the hybrid, zero-shot StreetCLIP and regression models of Astruc et al. (2024) and the published RFM $\mathcal{S}_2$ result (Dufour et al., 2025) on OSV-5M, and PlaNet (Weyand et al., 2016), CPlaNet (Seo et al., 2018), ISNs, TransLocator (Pramanick et al., 2022), GeoDecoder (Clark et al., 2023), PIGEOTTO (Haas et al., 2024), Img2Loc (Zhou et al., 2024), G3 (Jia et al., 2024) and Geo-R (Wu et al., 2026) on the Flickr benchmarks.

Table 7: **Complete OSV-5M results.** Table 1 extended with the methods omitted there and with accuracy within radius. Published rows are as reported, with a dash where a metric is not reported. RFM $\mathcal{S}_2$ rows without a qualifier are our reproduction. LocDiff labels these three values as accuracy within 25, 200 and 750 km, but the values it reports for its RFM $\mathcal{S}_2$ baseline match the city, region and country accuracies of Dufour et al. (2025); we therefore show them in the administrative columns. Rows w/o post-processing omit the snapping to a context image (Section 4.1). Best result per column in bold.

				Administrative accuracy (%) ↑				Accuracy (%) within km ↑				
Method	Backbone	GeoScore ↑	Dist. (km) ↓	Country	Region	Subreg	City	1	25	200	750	2500
ISNs	ResNet101	3331	2308	66.8	39.4	18.4	4.2	–	–	–	–	–
Hybrid	CLIP-L, fine-tuned	3361	1814	68.0	39.4	10.3	5.9	–	–	–	–	–
LocDiff	CLIP	–	–	77.0	46.3	–	11.0	–	–	–	–	–
SC 0-shot	StreetCLIP	2273	2854	38.4	20.8	9.9	14.8	–	–	–	–	–
Regression	StreetCLIP	3028	1481	56.5	16.3	1.5	0.7	–	–	–	–	–
1-NN	StreetCLIP	3645	1337	74.6	47.1	29.2	20.8	0.0	26.5	53.6	72.6	87.2
RFM $\mathcal{S}_2$, published	StreetCLIP	3767	1069	76.2	44.2	–	5.4	–	–	–	–	–
RFM $\mathcal{S}_2$	StreetCLIP	3985	936	81.0	54.2	31.1	17.5	0.09	22.9	65.2	81.4	91.2
HierLoc	StreetCLIP	3862	1051	80.3	53.1	39.2	22.5	–	–	–	–	–
RAFMa	StreetCLIP	4062	858	82.8	58.4	37.6	26.1	0.16	34.4	66.7	82.7	92.0
w/o post-processing	StreetCLIP	4062	858	82.8	58.4	37.6	26.1	0.55	34.4	66.7	82.7	92.0
1-NN	DINOv3-L	3483	1765	70.8	48.3	31.7	22.7	0.0	29.4	54.8	68.6	81.8
RFM $\mathcal{S}_2$	DINOv3-L	4012	926	80.9	55.7	33.0	19.1	0.13	25.3	67.0	81.9	91.1
HierLoc	DINOv3	3963	861	82.9	55.0	**40.7**	23.3	–	–	–	–	–
RAFMa	DINOv3-L	4085	852	82.5	59.5	38.9	**26.9**	0.16	**35.2**	68.5	83.1	91.9
w/o post-processing	DINOv3-L	4085	852	82.5	59.5	39.0	**26.9**	**0.58**	**35.2**	68.5	83.1	91.9
1-NN	SigLIP2-so400m	3667	1269	75.6	46.9	28.6	19.7	0.0	25.4	53.9	73.2	87.7
RFM $\mathcal{S}_2$	SigLIP2-so400m	4021	862	81.8	54.5	31.2	17.8	0.11	23.5	65.8	82.2	91.9
RAFMa	SigLIP2-so400m	4073	807	83.1	57.5	36.3	24.5	0.13	32.2	66.7	83.1	92.6
w/o post-processing	SigLIP2-so400m	4073	807	83.1	57.6	36.4	24.6	0.49	32.3	66.7	83.1	92.6
1-NN	PE-Core-G	3895	924	80.9	50.7	31.1	21.7	0.0	27.6	58.5	79.0	91.6
RFM $\mathcal{S}_2$	PE-Core-G	4285	490	87.4	59.6	34.8	19.7	0.13	26.0	72.2	88.9	96.0
RAFMa	PE-Core-G	**4342**	**442**	**88.8**	**62.8**	39.8	26.8	0.17	35.0	**73.3**	**89.9**	**96.5**
w/o post-processing	PE-Core-G	**4342**	**442**	**88.8**	**62.8**	39.8	26.8	0.56	35.0	**73.3**	**89.9**	**96.5**

Table 8: **Complete Flickr results.** *Top:* Table 2 extended with the methods omitted there and with 1-NN, which uses the location of the most similar database image as prediction; published rows are as reported, and the PlaNet row is the reproduction of Seo et al. (2018). Best result in bold, second best underlined. *Bottom:* GeoScore and mean great-circle error of our models, their context-free baselines and 1-NN, which the compared methods do not report; best result in bold. Rows w/o post-processing omit the snapping to a context image (Section 4.1).

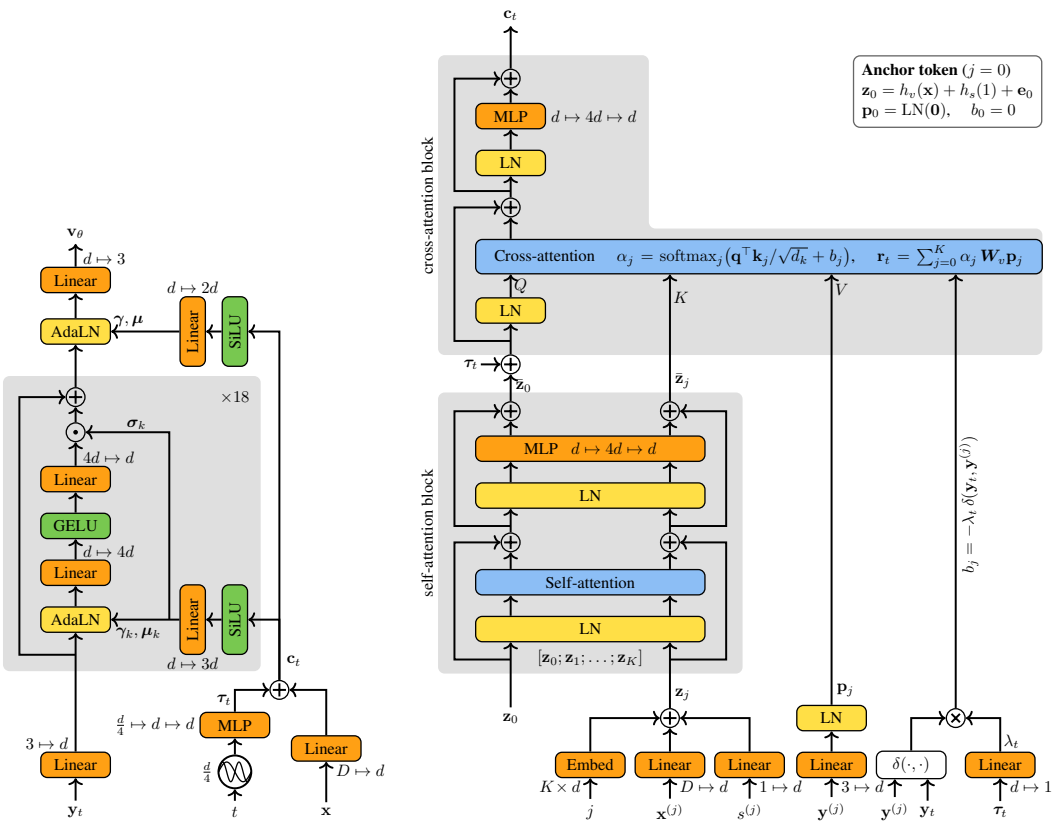
Method	Backbone	YFCC4k					Im2GPS3k				
	accuracy (%) within km ↑	1	25	200	750	2500	1	25	200	750	2500
PlaNet	InceptionV3	5.6	14.3	22.2	36.4	55.8	8.5	24.8	34.3	48.4	64.6
CPlaNet	InceptionV3	7.9	14.8	21.9	36.4	55.5	10.2	26.5	34.6	48.6	64.6
ISNs	ResNet101	6.7	16.5	24.2	37.5	54.9	10.5	28.0	36.6	49.7	66.0
TransLocator	ViT-B/16	8.4	18.6	27.0	41.1	60.4	11.8	31.1	46.7	58.9	80.1
GeoDecoder	Swin-B	10.3	24.4	33.9	50.0	68.7	12.8	33.5	45.9	61.0	76.1
GeoCLIP	CLIP-L	9.6	19.3	32.6	55.0	74.7	14.1	34.5	50.7	69.7	83.8
PIGEOTTO	CLIP	10.4	23.7	40.6	62.2	77.7	11.3	36.7	53.8	72.4	85.3
LocDiff	CLIP	–	33.3	46.7	65.2	79.9	10.9	34.0	53.3	72.5	85.2
GeoSURGE	CLIP-L	19.9	33.6	48.7	67.4	82.0	17.2	42.5	58.1	74.6	87.6
GeoToken	CLIP-L	24.3	35.3	46.6	64.2	78.6	16.8	39.6	53.8	70.8	85.0
1-NN	DINOv3-L	29.6	36.8	42.8	53.5	67.4	17.0	38.8	47.2	59.9	75.5
RFM $\mathcal{S}_2$	DINOv3-L	2.5	23.7	40.6	60.2	78.2	3.5	36.0	51.8	68.9	83.9
HierLoc	DINOv3	8.4	30.2	43.3	61.7	75.8	11.3	43.8	58.4	74.1	85.1
RAFMa	DINOv3-L	28.9	39.5	48.3	63.6	79.8	17.5	42.1	54.4	70.2	84.4
w/o post-processing	DINOv3-L	11.9	39.6	48.3	63.6	79.8	8.1	42.1	54.4	70.2	84.4
1-NN	SigLIP2-so400m	31.3	39.8	48.0	61.9	76.0	13.7	36.2	48.4	65.4	81.1
RFM $\mathcal{S}_2$	SigLIP2-so400m	3.1	28.7	47.3	66.9	81.9	5.0	40.1	57.5	75.5	88.9
RAFMa	SigLIP2-so400m	32.3	44.7	55.8	70.2	83.8	15.1	42.8	58.8	75.6	87.8
w/o post-processing	SigLIP2-so400m	14.2	44.8	55.8	70.1	83.8	8.3	42.9	58.9	75.6	87.8
1-NN	PE-Core-G	34.5	45.7	55.3	67.6	80.1	17.8	44.1	58.5	73.1	85.8
RFM $\mathcal{S}_2$	PE-Core-G	5.2	37.7	59.7	77.6	88.4	7.4	48.7	70.2	84.9	**93.6**
RAFMa	PE-Core-G	**39.8**	56.1	**68.4**	**80.4**	**89.1**	18.8	52.0	**70.5**	**85.1**	**93.6**
w/o post-processing	PE-Core-G	21.6	**56.2**	**68.4**	**80.4**	**89.1**	8.8	**52.2**	**70.5**	85.0	**93.6**
Multimodal Large Language Model systems											
Img2Loc	GPT-4V	14.1	29.6	41.4	59.3	76.9	17.1	45.1	57.9	72.9	84.7
G3	GPT-4V	24.0	36.9	47.0	64.3	78.2	16.7	40.9	55.6	71.2	84.7
GeoRanker	Qwen2-VL-7B	32.9	43.5	54.3	69.8	82.5	18.8	45.1	61.5	76.3	89.3
Geo-R	Qwen2.5-VL-7B	10.5	22.7	40.0	60.8	75.8	18.1	41.5	58.3	75.3	86.4
GeoToken	Gemini 2.0 Flash	25.4	38.5	51.4	68.0	81.0	**19.0**	46.0	60.1	76.6	88.8

Method	Backbone	YFCC4k		Im2GPS3k	
		GeoScore ↑	Dist. (km) ↓	GeoScore ↑	Dist. (km) ↓
1-NN	DINOv3-L	2843	2854	3156	2375
RFM $\mathcal{S}_2$	DINOv3-L	3130	2054	3513	1614
RAFMa	DINOv3-L	3320	1858	3594	1541
w/o post-processing	DINOv3-L	3318	1859	3593	1541
1-NN	SigLIP2-so400m	3219	2188	3381	1831
RFM $\mathcal{S}_2$	SigLIP2-so400m	3393	1714	3807	1140
RAFMa	SigLIP2-so400m	3598	1523	3803	1205
w/o post-processing	SigLIP2-so400m	3596	1523	3802	1205
1-NN	PE-Core-G	3486	1793	3715	1418
RFM $\mathcal{S}_2$	PE-Core-G	3846	1095	4210	**687**
RAFMa	PE-Core-G	**4033**	**1004**	**4213**	709
w/o post-processing	PE-Core-G	4032	**1004**	4212	709

D ARCHITECTURE AND CONFIGURATION

D.1 ARCHITECTURE AND HYPERPARAMETERS

Velocity network and context blocks. Figure 11 shows both parts of the model. Inside AdaLN, layer normalization has no learned affine parameters, and the output layer of each block's two-layer GELU MLP is initialized to zero; a final condition-dependent scale and shift precede the linear map to the three-dimensional velocity. The context-free baseline is conditioned on $\mathbf{c}_t = \phi(\mathbf{x}) + \boldsymbol{\tau}_t$; ours on the output $\mathbf{c}_t$ of the context network, which conditions every block and the final layer. The cross-attention query is the self-attended anchor token plus the time embedding, $\bar{\mathbf{z}}_0 + \boldsymbol{\tau}_t$, and $\bar{\mathbf{z}}_0$ also remains a key. The self-attention block of the context network is a standard pre-norm transformer block; the cross-attention block normalizes only its query token, since the keys come from already contextualized visual tokens and the geographic tokens carry their own layer normalization (Section 3.3). The anchor token's location-free geographic token $\mathbf{p}_0$ is this normalization applied to a zero vector, *i.e.* its learned bias, and the shared visual-modality embedding is absorbed into the bias of h_v in the notation of Section 3.3.



(a) Velocity network $\mathbf{v}_\theta$ (b) Context network g_θ

Figure 11: **Architecture.** *(a)* The AdaLN-MLP velocity network $\mathbf{v}_\theta$ of Dufour et al. (2025): each of its 18 blocks modulates its normalized input with the scale $\boldsymbol{\gamma}_k$ and shift $\boldsymbol{\mu}_k$ and gates its MLP output with $\boldsymbol{\sigma}_k$. The conditioning vector is $\phi(\mathbf{x}) + \boldsymbol{\tau}_t$ in the context-free baseline and the output of the context network in RAFMa. *(b)* The context network g_θ, drawn for one context image j and one attention head; all context images and the anchor token share the visual projection. The visual tokens $\mathbf{z}_0, \ldots, \mathbf{z}_K$ pass through self-attention. The updated anchor token plus the time embedding queries cross-attention, all updated visual tokens are keys, and the geographic tokens $\mathbf{p}_j$ bypass self-attention and enter as values. The geodesic bias b_j on the attention logits is the only path by which $\mathbf{y}_t$ reaches the context network. Orange: learned linear maps; yellow: normalizations; green: activations; blue: attention; white: no parameters.

Table 9: **Configuration** of the models in Tables 1 and 2. Within a benchmark, the frozen image encoder is the only part that changes between their rows. The context-free baseline keeps query dropout but has no context module or context dropout.

descriptor dimension D	768–1280
velocity network width / depth	768 / 18
MLP expansion	4
context blocks: self / cross	1 / 1
attention heads	4
context size K	64
query / whole-context dropout	0.1 / 0.3
context-image / location dropout	0.15 / 0.15
time warp ψ	sigmoid, $(\mathrm{start}, \mathrm{end}) = (-7, 3)$
optimizer	LAMB, $\beta = (0.9, 0.999)$, weight decay 0.05
learning rate / schedule	8×10^{-4} / 500-step warmup, cosine decay
batch size / steps	1024 / 1 M
precision	fp32
sampling steps L / guidance w	250 / 2