# TAMING CONTINUOUS LOCATION TOKENS FOR GENERALIZED NEXT POINT-OF-INTEREST RECOMMENDATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Next point-of-interest (POI) prediction is the task of estimating a user's future geographical location, supporting applications from restaurant recommendation to real estate shopping. Many state-of-the-art methods assign discrete embeddings to each POI, ignoring spatial relationships and limiting knowledge transfer to nearby unseen locations. To address this limitation, our framework combines a continuous latitude–longitude encoder using random Fourier features (RFF) with a causal transformer that predicts continuous location representations from historical check-in sequences. On Massive-STEPS, a multi-city check-in dataset, we examine how RFF frequency bandwidth controls representation granularity: lower frequencies enforce smooth spatial priors, whereas higher frequencies encourage precise coordinate memorization. We illustrate this tradeoff by withholding a portion of locations from training and comparing the recall on both seen and unseen POIs: a lower-frequency RFF configuration raises average unseen recall@100 across six city datasets from 36.4% to 42.9%, while seen recall falls from 63.8% to 57.5%. Further lowering the frequencies does not consistently improve unseen recall, indicating an optimal granularity among the tested settings. Additionally, we compare RFF with discrete location encoders and use a national-scale real estate shopping dataset to validate our approach. These findings provide empirical guidance for choosing the spatial scale of RFF embeddings, so that learned geographic patterns generalize to destinations absent from training data.

## 1 INTRODUCTION

Next location prediction is the task of predicting a user's next location according to past movements. This problem has applications across many industries. For example, in real estate recommendation, we want to characterize the distribution over geographical locations in which a customer is interested in purchasing or renting a home, based on past browsing history. In a navigation application, we want high-confidence predictions of where a user might travel next, to automatically recommend destinations. Many existing methods learn embeddings for users and POIs and use graph-based neural networks (Yun et al., 2019; Yan et al., 2023) to learn their relationships. These methods typically assume that a fixed set of users and POIs are known at training time. In industry applications, this assumption often proves too stringent. In fact, stakeholders are often especially interested in the recall of novel items, such as a new restaurant that recently opened or new construction in real estate. To bridge this disconnect, we study *generalized next location prediction*, aiming to predict destinations both observed during training and absent from training.

**Problem Scope** Given a sequence of locations that a user visited, our goal is to rank candidate coordinates for their next location. Traditional methods (e.g. Yan et al. (2023)) model users and POIs as discrete nodes with their own embeddings. These approaches generally achieve state-of-the-art recall on users and POIs seen during training, but are not designed to work on new users and POIs. In our work, we use a slightly different framework that evaluates how well a model can infer future movement based on past check-ins, rather than its ability to build static user and location profiles. At inference time, we input the user's most recent visits to a sequence model, as a list of latitude-longitude tuples, and the model outputs the token corresponding to the predicted next location. This allows prediction for users absent from training without a static embedding for each user. We omit auxiliary POI attributes, such as category labels, to compare location encoders using



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.

the same geographic information. For numerical evaluation, we embed candidate coordinates and rank them according to cosine similarity with the predicted location embedding, and measure recall of the true next visited location. For the experiments on open-source data, we use the Massive-STEPS benchmark (Wongso et al., 2025), a recent large-scale check-in trajectory dataset, but we use our own train/test splits. To illustrate potential for industry impact, we additionally benchmark on a closed-source national real estate shopping dataset.

**Approaches** We empirically compare discrete and continuous location encoders for next-POI prediction. Discrete methods either learn an embedding for each POI or partition the geographic area into regions using zip codes, administrative boundaries, or H3 cells. We find that these methods do not generalize well to unseen locations, since the encoding does not enforce any continuity between geographically close features. Continuous location encoders such as random Fourier features (RFF) (Rahimi & Recht, 2007) and Sphere2Vec (Mai et al., 2023b) encode locations using sinusoidal functions with different frequencies. The canonical motivation is two-fold: (1) Fourier mappings enable neural networks to learn high-frequency patterns in low-dimensional inputs (Tancik et al., 2020); (2) before the multilayer perceptron (MLP), RFF inner products approximate Gaussian-kernel similarities at the sampled spatial scales, connecting the frequency settings to the spatial relationships represented by the encoder.

**Contributions** We make two main contributions to the location prediction literature. Firstly, *our results show that using RFF features for location recommendation models improves over discretization approaches along multiple evaluation axes:* (1) better recall when ranking at both the POI level and at the H3 cell level, (2) better recall of both seen and unseen POIs, and (3) visually smoother preference distribution when plotted as a heatmap. Secondly, *we characterize how adjusting RFF frequency settings changes the balance between seen- and unseen-location accuracy.* We illustrate this tradeoff by withholding a portion of locations from training and comparing the recall on both seen and unseen POIs: a lower-frequency RFF configuration raises average unseen recall@100 across a 6 city subset of Massive-STEPs from 36.4% to 42.9%, while seen recall falls from 63.8% to 57.5%. We additionally demonstrate this tradeoff visually using heatmaps overlaid on the street map.

## 2 Related Work

### 2.1 Location Encoders

A location encoder maps latitude and longitude to an embedding vector. This representation is useful for downstream tasks, such as image geolocation and predicting metrics like forest cover, elevation, and population density. Tile-based approaches like Tile2Vec (Jean et al., 2019) and S2Vec (Choudhury et al., 2026) learn functions from map tiles to embeddings using metric learning or masked auto-encoder frameworks. SatCLIP (Klemmer et al., 2025) and GeoCLIP (Vivanco Cepeda et al., 2023) learn to align location embeddings with satellite image embeddings and embeddings of photos taken at that location, respectively, using a contrastive approach (see also Dhakal et al. (2025); Mai et al. (2023a); Astruc et al. (2026)). MoRA (Wen et al., 2026) extends contrastive learning to multiple modalities, including POI text and tabular demographics, and uses payment interaction data to train aligned encoders. Location representation approaches generally fall into two categories. Studies like Tang et al. (2015) use discretization-based methods, which partition the geographic area and learn an embedding for each region. Studies (Mai et al., 2020; Dhakal et al., 2025; Klemmer et al., 2025; Mac Aodha et al., 2019; Rahimi & Recht, 2007; Tancik et al., 2020; Cole et al., 2023; Mai et al., 2023b; Rußwurm et al., 2024) use continuous encoders. Space2Vec (Mai et al., 2020) encodes continuous coordinates with sine and cosine functions at geometrically spaced scales, and combines the resulting features through a learned multilayer perceptron. The encoding supports both absolute positions and relative displacements between neighboring points, enabling location and spatial-context modeling across multiple scales. These architectures are conveniently implemented by Torch Spatial (Wu et al., 2024).

### 2.2 Next POI Prediction

Traditional baselines use Markov chains (Rendle et al., 2010) or recurrent neural nets (Liu et al., 2016; Feng et al., 2018; Zhao et al., 2020), sometimes with time information injected (Sun et al., 2020; Wu et al., 2020; Luo et al., 2021). Newer methods use transformers (Yang et al., 2022), graph neural networks (Rao et al., 2022; Yan et al., 2023), candidate-conditioned models (Yu et al.,

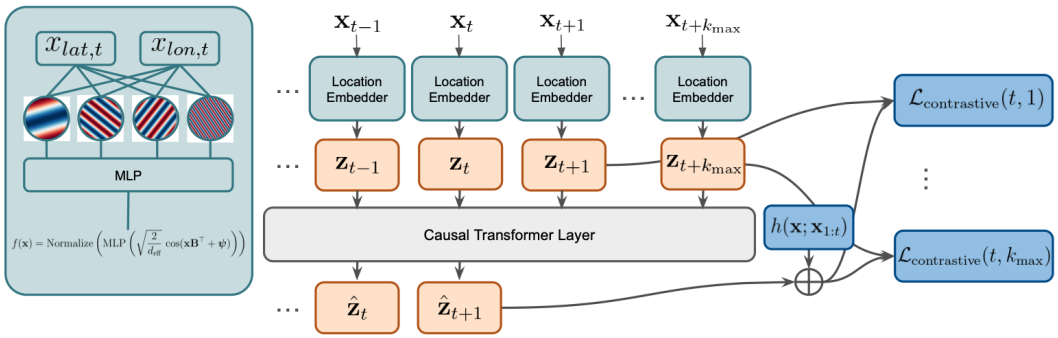



Figure 1: Location prediction architecture. A location encoder consisting of a Fourier mapping followed by an MLP feeds into a sequence model that predicts the next location embedding. The diagram shows trainable frequency vectors and phase offsets. A contrastive loss compares the predicted embedding with the next $k_{\max}$ location embeddings; the heuristic-residual learning variant adds the prior score before applying this loss.

2026), diffusion models (Qin et al., 2023), and unified human mobility models (Xu et al., 2024; Han et al., 2025). Methods such as GETNext (Yang et al., 2022) and STHGCN (Yan et al., 2023) learn relationships between user and POI nodes in a fixed training catalog. Recent work also uses LLMs to predict locations from textual representations of user trajectories, through prompting or task-specific fine-tuning (Feng et al., 2024; Beneduce et al., 2025; Wang et al., 2023; Li et al., 2024; Zhong et al., 2025). Complementary to the above cited literature, we focus on optimizing the recall of locations absent from the training data, and studying the scenario where location metadata is not available.

## 3 Method

**Problem Setting and Notation** Let $m$ be the maximum number of past locations in the user sequence supplied to the model. Let $\mathbf{x}_t := (x_{lat,t}, x_{lon,t})$ denote the coordinates of the user's location at discrete timestep $t$. Let $f(\mathbf{x}) \in \mathbb{R}^d$ denote the embedding of location $\mathbf{x}$. Let $\mathbf{z}$ denote $f(\mathbf{x})$ for convenience. Let $\hat{\mathbf{z}}_{t+1} = g(f(\mathbf{x}_1), \cdots, f(\mathbf{x}_t)) \in \mathbb{R}^d$ denote the predicted location embedding at the next time step, where $g$ is the sequence model. The transformer receives the encoder outputs after L2 normalization. For scoring, we use cosine similarity, $\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b}/(\|\mathbf{a}\|_2 \|\mathbf{b}\|_2)$. Let $d(\mathbf{x}_1, \mathbf{x}_2)$ denote the geographic distance between coordinates $\mathbf{x}_1$ and $\mathbf{x}_2$, approximated by the Haversine formula.

### 3.1 Main Framework

We treat the location prediction problem as a next token prediction task and a contrastive metric learning problem. The transformer sequence model predicts the next location embedding $\hat{\mathbf{z}}_{t+1}$, and we use an NT-Xent style loss to optimize it toward the true $f(\mathbf{x}_{t+1})$. Specifically:

$$\mathcal{L}_{\text{contrastive}}(t, k) = -\log \frac{\exp\left(\text{sim}(\hat{\mathbf{z}}_{t+1}, \mathbf{z}_{t+k})/\tau\right)}{\exp\left(\text{sim}(\hat{\mathbf{z}}_{t+1}, \mathbf{z}_{t+k})/\tau\right) + \sum_{\mathbf{x}^- \in \mathcal{N}(\mathbf{x}_{t+k})} \exp\left(\text{sim}(\hat{\mathbf{z}}_{t+1}, f(\mathbf{x}^-))/\tau\right)} \tag{1}$$

where $t$ denotes the timestep, $k$ the look-ahead horizon, and $\tau > 0$ the temperature. $\mathbf{x}^-$ denotes a negative location sample, and $\mathcal{N}(\mathbf{x}_{t+k})$ is the set of negative locations for target $\mathbf{x}_{t+k}$. Multi-step supervision encourages prediction beyond the immediate next location; its effect is evaluated jointly with heuristic-residual learning in the ablation experiment section. For Massive-STEPS, the number of unique POIs is small, so the negative sampling function simply returns the set of all unique coordinates. For the real estate dataset, we use all city centroids in the dataset as shared negatives across the batch, and sample a number of geographically proximate coordinates per training example as hard negatives (e.g. random points within a 5 km ring around the target location). Let $V_k$ contain the mini-batch positions $(i, t)$ with an available, unmasked target at horizon $k$. For nonempty $V_k$, we

Table 1: POI-level next location recall@$k$ (%) on Massive-STEPS; paired columns give $k = 10, 100$. Results cover the full test set of users absent from training, including both seen and unseen POIs and both first visits and revisits. ($\alpha = 0.6$)

| Method | New York | | Istanbul | | Jakarta | | São Paulo | | Moscow | | Bandung | | Petaling Jaya | | Kuwait City | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | 100 | 10 | 100 | 10 | 100 | 10 | 100 | 10 | 100 | 10 | 100 | 10 | 100 | 10 | 100 | 10 | 100 |
| Spatial | 5.3 | 17.2 | 6.2 | 17.5 | 6.9 | 21.9 | 7.7 | 22.4 | 11.8 | 30.5 | 7.5 | 21.8 | 4.7 | 16.8 | 5.7 | 15.4 | 7.0 | 20.4 |
| Popularity | 3.9 | 10.6 | 10.2 | 31.9 | 9.3 | 23.0 | 5.0 | 21.5 | 8.4 | 25.7 | 12.2 | 27.9 | 11.5 | 27.0 | 20.2 | 64.4 | 10.1 | 29.0 |
| Gravity | 7.4 | 21.8 | 19.3 | 45.7 | 15.8 | 38.6 | 13.1 | 33.1 | 19.4 | 45.8 | 16.8 | 39.1 | 15.7 | 39.7 | 21.1 | 62.4 | 16.1 | 40.8 |
| Gravity+history | 26.1 | 43.4 | 38.8 | 61.7 | 29.6 | 56.0 | 42.0 | 62.0 | 40.4 | 63.4 | 32.8 | 58.7 | 29.9 | 56.2 | 41.7 | 73.3 | 35.2 | 59.3 |
| Discrete | 25.6 | 42.6 | 39.9 | 65.7 | 29.9 | 56.5 | 41.5 | 62.3 | 40.4 | 63.5 | 33.1 | 59.3 | 30.7 | 58.7 | 43.2 | 77.4 | 35.5 | 60.7 |
| Region | 26.1 | 42.2 | 38.3 | 58.6 | 29.5 | 53.9 | 41.9 | 60.3 | 40.0 | 61.3 | 32.4 | 56.2 | 29.7 | 54.3 | 41.3 | 69.4 | 34.9 | 57.0 |
| H3 (res=9) | 25.9 | 42.2 | 38.5 | 59.8 | 29.4 | 54.1 | 41.7 | 61.1 | 39.8 | 60.5 | 32.5 | 56.7 | 29.6 | 53.5 | 41.7 | 71.8 | 34.9 | 57.5 |
| Space2Vec-grid | 26.3 | 44.9 | 40.2 | 66.5 | 30.4 | 58.3 | 42.4 | 64.2 | 40.9 | 65.4 | 33.5 | 60.6 | 31.2 | 59.8 | 43.5 | 77.9 | 36.1 | 62.2 |
| Space2Vec-hex | 26.2 | 45.2 | 40.2 | 66.4 | 30.4 | 58.4 | 42.4 | 64.2 | 40.9 | 65.6 | 33.5 | 60.8 | 31.1 | 59.8 | **43.5** | 77.9 | 36.0 | 62.3 |
| RFF | **26.3** | **45.3** | **40.3** | **66.6** | 30.4 | **58.5** | 42.4 | **64.4** | **41.0** | 65.7 | 33.5 | 61.1 | **31.2** | **60.1** | 43.5 | 78.1 | 36.1 | 62.5 |
| RFF (trainable) | 26.3 | 45.3 | 40.2 | **66.6** | **30.5** | 58.4 | **42.5** | 64.4 | 41.0 | **65.7** | **33.6** | **61.3** | 31.1 | 60.1 | 43.5 | **78.2** | **36.1** | **62.5** |

average over these positions separately for each horizon, with discount $\gamma_{\text{decay}}$:

$$\mathcal{L}_{\text{overall}} = \sum_{k=1}^{k_{\max}} \frac{\gamma_{\text{decay}}^{k-1}}{|V_k|} \sum_{(i,t)\in V_k} \mathcal{L}^{(i)}_{\text{contrastive}}(t,k) \tag{2}$$

### 3.2 Heuristic-Residual Learning

As is common in recommendation applications, the likelihood function of the prediction is strongly influenced by the overall popularity, historical user preference, and geographical proximity to the last known location. Therefore, we find that it is better to train the model on the residual error of the heuristic predictors. We use the strongest prior on the Massive-STEPS dataset, labeled as Gravity+history in Table 1, combining popularity, history and proximity. In the heuristic-residual learning variant, we train the neural score as an additive correction to the weighted prior score using the contrastive loss. The heuristic is defined as:

$$h(\mathbf{x}; \mathbf{x}_{1:t}) = (1-\beta_{\text{heuristic}}) \underbrace{(\log(1+\text{pop}(\mathbf{x})) - \gamma \log(d(\mathbf{x}_t, \mathbf{x}) + \epsilon_d))}_{\text{Gravity Component}} + \beta_{\text{heuristic}} \underbrace{\log(1+c_t(\mathbf{x}))}_{\text{Repeat-visit Prior}} \tag{3}$$

where $\epsilon_d > 0$ keeps the score finite at zero distance, $\text{pop}(\mathbf{x})$ counts occurrences of $\mathbf{x}$ as a training target, $d(\mathbf{x}_t, \mathbf{x})$ is the distance between the current position $\mathbf{x}_t$ and the candidate POI $\mathbf{x}$. $c_t(\mathbf{x})$ is the cumulative count of visits to POI $\mathbf{x}$ by the user up to timestep $t$. $\beta_{\text{heuristic}}$ sets the weight of the repeat-visit term relative to Gravity. We then replace the scoring function in Equation 1 with:

$$s_t(\mathbf{x}) = \text{sim}(\hat{\mathbf{z}}_{t+1}, f(\mathbf{x}))/\tau + \alpha_t h(\mathbf{x}; \mathbf{x}_{1:t}) \tag{4}$$

During heuristic-residual learning, a sigmoid head maps each transformer state to the sequence-dependent heuristic weight $\alpha_t$. After training, we can either use the trained adaptive $\alpha_t$ value or a fixed $\alpha$, determined by a validation set. Appendix B provides a comparison between both scoring rules, but we primarily report using a fixed $\alpha$.

### 3.3 Random Fourier Features

A Fourier transform describes a function through its sinusoidal components. A spatial wave has the form $a\cos(\mathbf{b}^\top \mathbf{x} + \phi)$: $a$ controls the wave's amplitude, the direction and length of $\mathbf{b}$ set its orientation and frequency, and the phase offset $\phi$ shifts its peaks and troughs. Zero phase ($\phi = 0$) leaves the cosine unshifted. Low frequencies represent gradual spatial changes; high frequencies represent finer detail. Random Fourier features (RFF) (Rahimi & Recht, 2007) provide sampled waves that a learned model can combine.

We standardize latitude and longitude separately using training-sequence statistics (Appendix B), giving coordinates $\tilde{\mathbf{x}}$. The encoder with fixed RFF features is:

$$f(\mathbf{x}) = \mathrm{MLP}\left(\sqrt{2/d_{\mathrm{rff}}}\,[\sin(\tilde{\mathbf{x}}\mathbf{B}^\top + \boldsymbol{\psi}), \cos(\tilde{\mathbf{x}}\mathbf{B}^\top + \boldsymbol{\psi})]\right). \tag{5}$$

The rows of $\mathbf{B} \in \mathbb{R}^{(d_{\mathrm{rff}}/2)\times 2}$ are frequency vectors. $\boldsymbol{\psi} \in \mathbb{R}^{d_{\mathrm{rff}}/2}$, initialized to zero, is the phase offset. Each produces a sine and a cosine feature, scaled equally by $\sqrt{2/d_{\mathrm{rff}}}$. The MLP learns how to weigh and combine these features; we normalize its output when computing similarity scores.

**Controlling the RFF frequencies** We divide the vectors equally among $J$ groups, called frequency bands. In band $j$, vector entries are drawn from $\mathcal{N}(0, \omega_j^2)$, with standard deviations $\omega_j$ increasing by a constant ratio from $\omega_{\min}$ to $\omega_{\max}$. Lower values of $\omega_{\min}, \omega_{\max}$ favor broader spatial patterns; higher values favor finer detail. $J$ sets the number of scales, and $d_{\mathrm{rff}}$ sets the feature count, giving $d_{\mathrm{rff}}/(2J)$ vectors per band. At fixed feature count, more bands leave fewer vectors per scale. We standardize latitude and longitude using training-sequence means and standard deviations, so that markets with tightly clustered or widely dispersed locations use a common coordinate scale. This makes the initial RFF frequency settings comparable in relative spatial resolution across markets, while their corresponding physical wavelengths vary with each market's geographic spread.

Along a coordinate axis spanning $D$ km per transformed unit, $(\omega_{\min}, \omega_{\max})$ gives reference wavelengths $[2\pi D/\omega_{\max},\, 2\pi D/\omega_{\min}]$ km, where a wavelength is one full peak-to-peak cycle. These endpoints describe the Gaussian sampling scales; individual sampled waves can have shorter or longer periods (Appendix B).

At initialization, for fixed standardized coordinates, a band's expected Fourier similarity decays as $\exp(-\omega_j^2\|\tilde{\mathbf{x}} - \tilde{\mathbf{y}}\|^2/2)$ (Rahimi & Recht, 2007). This connects frequency to spatial scale before the MLP; the learned coordinate scaling and MLP also affect the final embedding.

For fixed RFF features, $\mathbf{B}$ is sampled once and held constant during training. An unshifted cosine gives identical features at $\tilde{\mathbf{x}}$ and $-\tilde{\mathbf{x}}$; including sine removes this reflection symmetry. Trainable RFF uses $d_{\mathrm{rff}}$ cosine features and learns both frequency vectors and phase offsets, starting from the same frequency sampling scheme and phases uniform on $[0, 2\pi)$.

## 4 Experiments

We compare POI and H3-cell recall on Massive-STEPS, examine how encoder frequencies and size affect seen and unseen recall, and report a supporting real-estate transaction study.

**Baselines** We compare against 4 heuristic ranking baselines. "Popularity" ranks POIs by their training target counts, highest first. "Spatial" ranks POIs by distance from the user's last visited location, nearest first. "Gravity" combines training popularity with a penalty for distance from the user's last visited location (Eq. 3). "Gravity+history" (the full prior) adds the repeat-visit term. Unless otherwise noted, $\beta_{\text{heuristic}} = 0.95$ weighs log-history by 0.95 and gravity by 0.05, emphasizing revisits. We use $\gamma = 1.5$ for the log-distance penalty; larger values more strongly favor nearby POIs. We compare against 3 discrete location representation baselines. "Discrete" refers to learning one embedding per POI; "region" refers to learning one per region; and "H3"[1] refers to one per H3 cell. For the region baseline, we use the publicly available administrative boundaries in OpenStreetMap (OSM). H3 is a geospatial indexing system that partitions the world into hexagonal cells. We default to resolution 9, which has an edge length of about 200 meters, but also experiment with resolutions 10 and 11. Higher numbers indicate finer resolutions. We compare against both the grid and hex versions of Space2Vec Mai et al. (2020), implemented by Wu et al. (2024). We compare fixed RFF features (Eq. 5) with trainable RFF features that learn wave frequencies, directions, and phase offsets. Both train an MLP after the Fourier features.

**Evaluation Metrics** For Massive-STEPS, a standalone model ranks candidate locations using only similarity to the predicted embedding $\hat{\mathbf{z}}_{t+1}$; an ensemble combines this score with a heuristic score. We measure recall@$k$ (R@$k$) and mean reciprocal rank (MRR) of the actual next location. When ranking individual POIs, region/H3 encoders give identical model scores to candidates sharing the same embedding, but adding the heuristic breaks model-score ties.

[1] `https://h3geo.org`

### 4.1 RESULTS ON MASSIVE-STEPS

Massive-STEPS (Wongso et al., 2025) contains location trajectories gathered across 15 cities. We experiment with the 8 cities that contain >100K user check-ins. Within this set, the largest city is Istanbul, with 544K check-ins and 30K unique POIs; the smallest is Moscow, with 106K check-ins and 9K unique POIs.

**POI-level Results** Training and test user sets are disjoint. The test set contains both POIs that appear as training targets (seen) and POIs that do not (unseen). Table 1 reports recall over this full test set, including first visits and revisits. Each encoder is combined with Gravity+history using the same inference-time weight, $\alpha = 0.6$. Gravity+history is a strong baseline, capturing popularity, proximity and repeat visits. Continuous encoders outperform the discrete POI baseline in average recall. RFF slightly exceeds Space2Vec-hex in average R@100 (62.5% versus 62.3%), but this small difference does not establish superiority across training seeds. Fixed and trainable RFF have the same rounded average R@100 here.

Table 2: H3-level next location R@100 ($\alpha = 0.7$). We rank H3 cells that contain at least one POI by the location embedding corresponding to the center of the H3 cell. We report recall of the H3 cell containing the true next POI.

(a) H3 Resolution 9 ($\omega_{min}$ = 0.05, $\omega_{max}$ = 5)

| Method | # params | New York | Istanbul | Jakarta | São Paulo | Moscow | Bandung | Avg |
|---|---|---|---|---|---|---|---|---|
| H3 (res=9) | 3.9M | 76.5 | 84.9 | 80.4 | 82.2 | 88.3 | 83.5 | 82.6 |
| Discrete | 17.0M | 73.1 | 81.1 | 77.2 | 78.8 | 84.7 | 79.9 | 79.1 |
| Region | 3.2M | 76.0 | 84.2 | 79.6 | 81.4 | 87.2 | 83.5 | 82.0 |
| Spatial | | 57.5 | 57.9 | 52.8 | 53.1 | 72.5 | 58.0 | 58.7 |
| Popularity | | 30.8 | 56.8 | 48.3 | 40.8 | 60.8 | 59.9 | 49.6 |
| Gravity | | 62.8 | 75.8 | 68.2 | 65.1 | 80.5 | 72.0 | 70.7 |
| Gravity+history | | 76.0 | 84.2 | 79.4 | 81.7 | 88.1 | 83.3 | 82.1 |
| RFF | 7.9M | **77.1** | **85.1** | **80.7** | **83.0** | **88.6** | **84.4** | **83.1** |

(b) H3 Resolution 10 ($\omega_{min}$ = 0.1, $\omega_{max}$ = 10)

| Method | # params | New York | Istanbul | Jakarta | São Paulo | Moscow | Bandung | Avg |
|---|---|---|---|---|---|---|---|---|
| H3 (res=10) | 6.6M | 58.5 | **75.0** | 70.9 | 72.4 | **75.0** | **70.5** | 70.4 |
| Discrete | 17.0M | 54.8 | 68.7 | 68.4 | 69.4 | 70.7 | 66.8 | 66.5 |
| Region | 3.2M | 58.6 | 72.9 | 69.6 | 71.3 | 73.0 | 68.2 | 68.9 |
| Spatial | | 38.0 | 41.3 | 39.7 | 38.4 | 49.8 | 35.1 | 40.4 |
| Popularity | | 13.9 | 37.4 | 34.7 | 27.3 | 36.1 | 37.5 | 31.2 |
| Gravity | | 40.7 | 60.4 | 55.7 | 48.1 | 60.7 | 51.7 | 52.9 |
| Gravity+history | | 58.1 | 73.1 | 70.2 | 71.9 | 73.9 | 69.3 | 69.4 |
| RFF | 7.9M | **59.6** | 74.3 | **71.0** | **73.4** | 74.9 | 70.3 | **70.6** |

(c) H3 Resolution 11 ($\omega_{min}$ = 0.5, $\omega_{max}$ = 50)

| Method | # params | New York | Istanbul | Jakarta | São Paulo | Moscow | Bandung | Avg |
|---|---|---|---|---|---|---|---|---|
| H3 (res=11) | 10.9M | 47.8 | **68.0** | 64.4 | 66.6 | 65.8 | 63.6 | 62.7 |
| Discrete | 17.0M | 46.4 | 62.3 | 61.8 | 64.7 | 63.6 | 61.2 | 60.0 |
| Region | 3.2M | 48.2 | 64.9 | 62.8 | 65.5 | 64.7 | 60.7 | 61.1 |
| Spatial | | 24.4 | 27.5 | 33.2 | 29.8 | 37.5 | 27.6 | 30.0 |
| Popularity | | 10.1 | 31.0 | 26.4 | 21.8 | 26.3 | 28.2 | 24.0 |
| Gravity | | 28.3 | 49.9 | 47.3 | 38.4 | 48.4 | 41.8 | 42.3 |
| Gravity+history | | 48.3 | 65.6 | 63.6 | 66.1 | 65.2 | 62.0 | 61.8 |
| RFF | 7.9M | **49.7** | 67.0 | **64.6** | **67.6** | **66.9** | **64.0** | **63.3** |

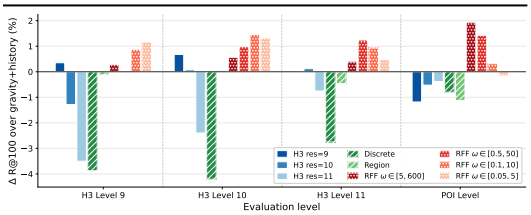
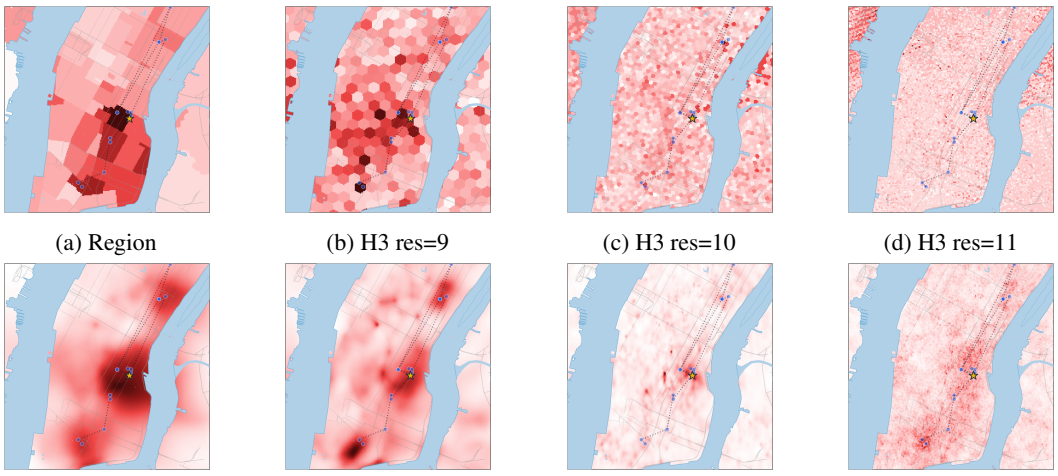



Figure 2: R@100 gain over the gravity+history heuristic for each model across different resolutions. Same data as Table 2.

**H3-level Recall** The POI-level recall results showed that continuous location encoders outperform POI level embeddings w.r.t. POI recall. We show analogous results for different H3 resolutions in Table 2. For these experiments, we rank H3 cells by the location embedding of the center of each H3 cell. For the discrete baseline, we rank by the embedding of the POI closest to the center of each H3 cell; for the region baseline, we rank by the embedding of the region containing the centroid of each H3 cell. For the heuristic baselines, we rank by the maximum score over all POIs within each H3 cell. The most natural baseline for this experiment is the H3 embedding using the same resolution as the evaluation. This is included as the first row in each sub-table of Table 2. Similarly, we must tune the $\omega$ parameters controlling the kernel width for each resolution individually, such that the encoder's output is relatively constant over each H3 cell. We experimentally determined good ballpark numbers for $\omega_{min}$ and $\omega_{max}$. Table 2 shows that, when these two hyperparameters are chosen appropriately, the RFF encoder matches the performance of the H3 embedding at levels 9 and 10, while exceeding the H3 performance at level 11. Figure 2 shows these results visually. Generally speaking, when evaluating at a coarser granularity (smaller H3 resolution), the advantage of using a continuous encoder degrades for two reasons: (1) when there is more data per H3 cell, the advantage of implicit spatial continuity diminishes, since the relationships can be inferred from the data, (2) when there are fewer H3 cells, the H3 encoder requires fewer parameters than the RFF encoder. Figure 3 contains a visualization of the preference distribution over next POI candidates, with varying $\omega$.

**Seen vs. Unseen** To distribute held-out targets throughout each city, Table 3 and Figure 4 withhold 20% of POIs per H3 resolution-9 cell as



(a) Region (b) H3 res=9 (c) H3 res=10 (d) H3 res=11

(e) RFF $\omega \in [0.05, 5]$ (f) RFF $\omega \in [0.1, 10]$ (g) RFF $\omega \in [0.5, 50]$ (h) RFF $\omega \in [5, 600]$

Figure 3: RFF frequency controls the spatial detail of next-location predictions. For this New York City trajectory, increasing $(\omega_{\min}, \omega_{\max})$ shifts RFF predictions from broad patterns to finer local variation (bottom row). Region and H3 encoders set spatial detail through their boundaries (top row). Star = true next location; dots = trajectory.

training targets. Reducing the RFF frequency scales from $(5, 600)$ to $(0.1, 10)$ increases average unseen recall from 36.4% to 42.9%, while reducing seen recall from 63.8% to 57.5%. Unseen recall increases in all six cities, by 5.0–8.3 percentage points. Further reducing the scales to $(0.05, 5)$ yields 42.7% average unseen recall, with mixed effects across cities. Gravity+history remains strong: RFF $(0.1, 10)$ exceeds it by 0.7 points in average unseen recall and in four of six cities. We also observed that RFF encoders with fewer parameters generally favor unseen recall, while larger encoders favor seen recall (Appendix C, Figure 8). Across six cities and three training seeds, the highest-frequency RFF setting yields higher seen recall and lower unseen recall than either lower-frequency setting, when using model scores alone (Appendix C.1, Table 7). The intermediate range, $(0.1, 10)$, improves unseen recall over $(0.05, 5)$ in five cities across all three seeds.

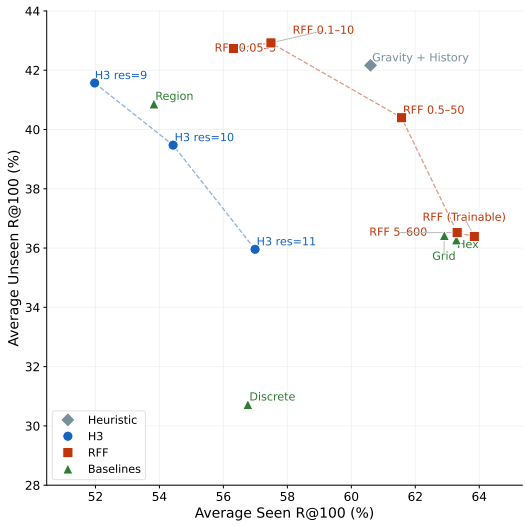



Figure 4: Seen vs. unseen recall tradeoff, averaged across cities using the training-target holdout in Table 3. Lower-frequency RFF favors unseen POIs up to $(0.1, 10)$. Higher frequencies favor seen POIs. Gravity+history remains a strong baseline on both axes.

The tradeoff persists when targets are absent from the available user history: coarse RFF improves first-view unseen recall over fine RFF in both tested cities, whereas fine RFF improves seen recall (Appendix A, Table 5). Combining RFF with Spatial, with the weight chosen on validation data, does not establish a consistent improvement over Spatial alone.

### 4.2 Results on Real Estate Transaction Prediction

Massive-STEPS is limited to disjoint metropolitan regions. To show that the RFF location predictor can work at a national-scale and interpolate to non-urban areas within the supported region, we train the proposed model on a real-estate recommendation dataset. In this dataset, the input is a sequence of locations corresponding to properties that the user clicked on, viewed, impressioned on, or otherwise interacted with. Every trajectory ends in a transaction containing the location of the property purchased. In Table 4, we compare methods based on their ability to predict the location of the property purchased.

Table 3: Seen/unseen POI R@100: lower RFF frequencies favor unseen recall up to a point. A nominal 20% of POIs per H3 cell (res=9) are withheld as training targets. Seen/unseen denotes whether a POI appears as a training target. Results include first views and revisits by the same user. RFF pairs give $(\omega_{min}, \omega_{max})$. Encoder scores are combined with Gravity+history ($\alpha = 0.4$).

| **Method** | **New York** | | **Istanbul** | | **Jakarta** | | **São Paulo** | | **Moscow** | | **Bandung** | | **Average** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Seen | Unseen | Seen | Unseen | Seen | Unseen | Seen | Unseen | Seen | Unseen | Seen | Unseen | Seen | Unseen |
| Spatial | 14.9 | 21.5 | 17.1 | 18.9 | 20.7 | 24.5 | 19.9 | 26.6 | 28.3 | 35.2 | 21.2 | 22.9 | 20.3 | 24.9 |
| Popularity | 13.9 | 0.0 | 37.5 | 0.0 | 30.3 | 0.0 | 30.4 | 0.0 | 32.6 | 0.0 | 32.4 | 0.0 | 29.5 | 0.0 |
| Gravity | 28.5 | 3.3 | 52.1 | 5.5 | 48.2 | 4.4 | 42.8 | 5.2 | 52.9 | 9.0 | 45.2 | 4.4 | 44.9 | 5.3 |
| Gravity+history | 45.0 | 36.4 | 65.2 | **40.2** | 61.3 | 37.3 | 64.6 | 52.6 | 66.3 | 44.1 | 61.2 | 42.4 | 60.6 | 42.2 |
| Discrete | 41.1 | 25.2 | 68.0 | 27.2 | 60.3 | 23.2 | 63.4 | 40.3 | 62.7 | 32.4 | 59.3 | 28.7 | 59.1 | 29.5 |
| Region | 39.9 | 34.4 | 55.8 | 38.9 | 53.6 | 35.2 | 59.3 | 51.3 | 60.3 | 44.6 | 53.7 | 40.9 | 53.8 | 40.9 |
| H3 (res=9) | 36.5 | 35.1 | 57.0 | 38.6 | 52.3 | 36.5 | 57.2 | 52.4 | 55.9 | 46.6 | 53.6 | 40.9 | 52.1 | 41.7 |
| Space2Vec-grid | 45.9 | 31.7 | 70.6 | 33.8 | 64.2 | 33.0 | 66.7 | 47.5 | 67.4 | 38.6 | 62.7 | 33.9 | 62.9 | 36.4 |
| Space2Vec-hex | 46.4 | 31.8 | 70.8 | 33.1 | 64.4 | 32.9 | 67.0 | 47.9 | 68.1 | 38.7 | 63.1 | 33.3 | 63.3 | 36.3 |
| RFF $(5, 600)$ | 46.7 | 32.0 | **71.3** | 33.1 | **64.9** | 31.7 | 67.2 | 48.4 | 68.4 | 39.5 | **64.2** | 33.7 | **63.8** | 36.4 |
| RFF (trainable) | **47.1** | 31.2 | 71.1 | 33.3 | 64.8 | 32.9 | **67.6** | 48.0 | **68.5** | 39.1 | 64.0 | 33.7 | **63.8** | 36.4 |
| H3 (res=10) | 38.8 | 33.9 | 61.5 | 36.0 | 53.8 | 34.5 | 59.2 | 50.1 | 58.2 | 44.1 | 55.0 | 38.3 | 54.4 | 39.5 |
| H3 (res=11) | 40.2 | 30.2 | 65.8 | 33.8 | 56.5 | 30.9 | 61.5 | 46.6 | 60.8 | 38.6 | 57.1 | 35.7 | 57.0 | 36.0 |
| RFF $(0.05, 5)$ | 42.3 | 36.7 | 61.4 | 39.2 | 56.0 | 37.0 | 60.1 | 53.0 | 60.1 | **47.7** | 58.0 | **42.7** | 56.3 | 42.7 |
| RFF $(0.1, 10)$ | 42.3 | **37.0** | 63.2 | 39.0 | 57.0 | **38.2** | 61.1 | **53.5** | 61.6 | **47.7** | 59.8 | 42.0 | 57.5 | **42.9** |
| RFF $(0.5, 50)$ | 45.0 | 35.4 | 68.4 | 36.2 | 61.9 | 37.0 | 65.6 | 50.9 | 65.3 | 44.8 | 63.3 | 38.1 | 61.6 | 40.4 |

Table 4: Continuous encoders improve validation recall and MRR over Gravity+history in this real-estate task. Values are percentage-point differences over 1,026 candidate entries sampled using the true transaction city. Encoders use the full prior at $\alpha = 0.52$, $\gamma = 0.40$, and $\beta = 0.20$.

| Method | ΔR@1 | ΔR@10 | ΔR@100 | ΔMRR |
|---|---|---|---|---|
| Random | -0.91 | -9.81 | -40.81 | -3.85 |
| Spatial | -0.90 | -7.94 | -21.27 | -2.95 |
| Popularity | -0.04 | -2.97 | -5.50 | -0.76 |
| Gravity | +0.24 | -0.15 | -0.13 | +0.12 |
| Gravity+history | +0.00 | +0.00 | +0.00 | +0.00 |
| H3 (res 6) | +0.11 | -1.95 | -7.43 | -0.40 |
| RFF | +0.86 | +1.76 | +1.79 | +1.34 |
| RFF (trainable) | **+1.74** | +3.70 | **+4.11** | **+2.54** |
| Space2Vec-grid | +1.00 | +3.25 | +3.41 | +1.95 |
| Space2Vec-hex | +1.05 | **+3.79** | +3.35 | +2.06 |

For each validation example, we sample 512 points within the true transaction city and 512 outside its boundaries, and include the city centroid and true location. The resulting 1,026 entries may contain duplicate coordinates. "Random" uniformly orders this pool; "Spatial" ranks by distance to the last user event. Heuristic parameters are selected on validation data; the common $\alpha$ maximizes trainable-RFF validation MRR and is used for every encoder. Popularity and history scores use Gaussian kernel density estimates (Appendix B). Trainable RFF has the highest reported validation R@100 and MRR, while Space2Vec-hex leads R@10.

### 4.3 Ablation Results, Qualitative Results and Discussion

Figure 5 shows the effect of the heuristic-residual learning and multi-step prediction. These two features force the model to focus on the samples that are harder to predict, where the heuristic does not already perform well. This lowers the recall of the model by itself ($\alpha = 0$) while improving the peak recall of the model-heuristic ensemble. Figure 7 in the Appendix performs a detailed error analysis. The improvement of RFF is most pronounced on POIs that were visited fewer than 250 times in the training set, and where the user moved >15 km from their previous known position. Finally, we provide visualizations of the approximate probability density of the next POI prediction across all 8 cities in the Massive-STEPS dataset in Figure 6.

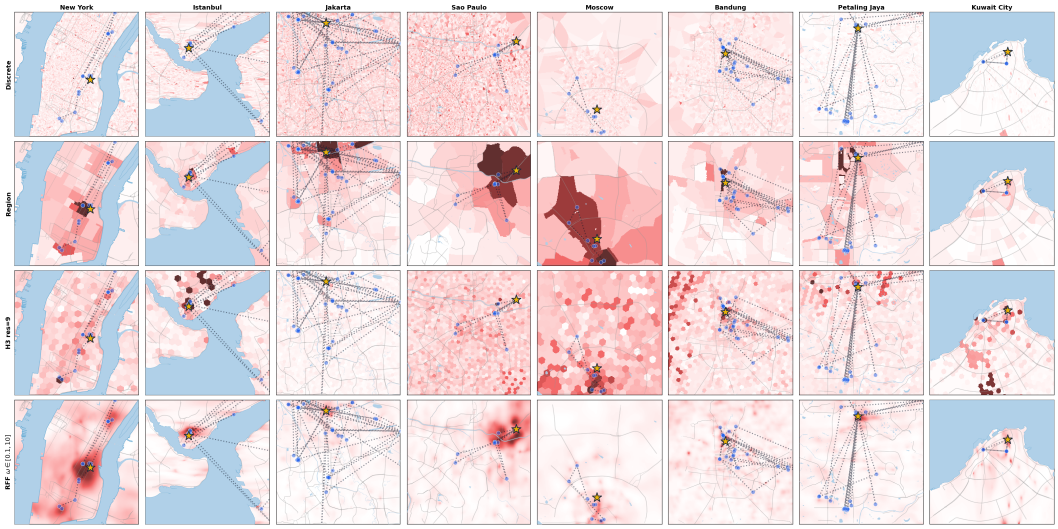



Figure 6: Qualitative comparison of preference heatmaps between RFF and discrete features. The intensity of the heatmap represents the likelihood of the user visiting that region in the next timestep. Blue dots represent the input trajectory, and star represents the actual next check-in. When we use discrete POI embeddings, the model does not learn to map nearby POIs to similar features, resulting in a noisy preference heatmap. When using region embeddings, the heatmap makes more sense visually, but the granularity is limited by the size of the administrative regions. H3 embeddings suffer from both problems. The heatmap drawn using RFF features is visually smoother and a qualitatively more reasonable representation of the true preference distribution.

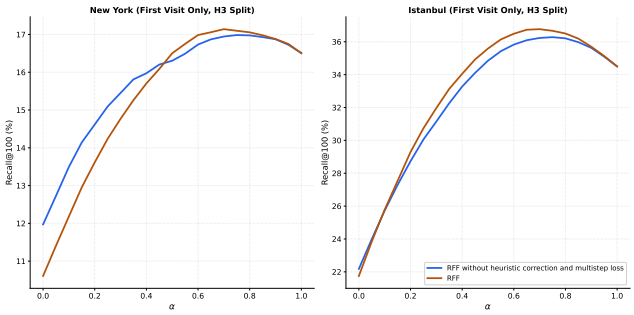



Figure 5: Joint ablation of heuristic-residual learning and multi-step prediction. Increasing $\alpha$ weighs the full distance–popularity–history heuristic more heavily. Adding both changes increases the highest recall on the plotted curves for first visits to seen and unseen POIs.

## 5 Conclusion

In this work, we firstly showed that using RFF features for location recommendation modeling improves over discretization approaches such as H3 and region embeddings. Secondly, our experiments quantify how adjusting RFF frequency scales shifts the balance between seen and unseen next-location recall, with further frequency reductions offering no consistent gain. We shared empirical results on both POI level recall and H3 cell level recall. We visualized how the preference heatmap changes as the frequency parameter is tuned, and compared the predicted preference visualization qualitatively against discrete embedding approaches. Finally, we extended our results to a national-scale real estate location dataset.

## Limitations

We isolate location encoding and omit auxiliary signals such as time intervals, POI categories and road networks. The frequency scales $\omega_{\min}$ and $\omega_{\max}$ require tuning for the application and coordinate standardization. The cross-encoder comparisons report point estimates without variation across training seeds; small differences between encoders therefore remain descriptive. Only RFF receives the full frequency sweep, limiting conclusions about its advantage over equally tuned alternatives. The real-estate results use validation data and transaction-city-conditioned candidates; independent evaluation with candidates available before purchase remains necessary.

### AI USE STATEMENT

In this work, we used generative AI tools for writing experimentation code and polishing the paper. We have not used generative AI tools for writing the initial draft or the core ideation. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Guillaume Astruc, Eduard Trulls, Jan Hosang, Loic Landrieu, and Paul-Edouard Sarlin. Unigeoclip: Unified geospatial contrastive learning. *arXiv preprint arXiv:2604.11668*, 2026.

Ciro Beneduce, Bruno Lepri, and Massimiliano Luca. Large language models are zero-shot next location predictors. *IEEE Access*, 13:77456–77467, 2025.

Shushman Choudhury, Chandrakumari Suvarna, Iveel Tsogsuren, Abdul Rahman Kreidieh, Elad Aharoni, Chun-Ta Lu, and Neha Arora. S2vec: Self-supervised geospatial embeddings for the built environment. *ACM Transactions on Spatial Algorithms and Systems*, 12(3):1–21, 2026.

Elijah Cole, Grant Van Horn, Christian Lange, Alexander Shepard, Patrick Leary, Pietro Perona, Scott Loarie, and Oisin Mac Aodha. Spatial implicit neural representations for global-scale species mapping. In *International conference on machine learning*, pp. 6320–6342. PMLR, 2023.

Aayush Dhakal, Srikumar Sastry, Subash Khanal, Adeel Ahmad, Eric Xing, and Nathan Jacobs. Range: Retrieval augmented neural fields for multi-resolution geo-embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24680–24689. IEEE/CVF, 2025.

Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, pp. 1459–1468, 2018.

Shanshan Feng, Haoming Lyu, Fan Li, Zhu Sun, and Caishun Chen. Where to move next: Zero-shot generalization of llms for next poi recommendation. In *2024 ieee conference on artificial intelligence (cai)*, pp. 1530–1535. IEEE, 2024.

Chonghua Han, Yuan Yuan, Yukun Liu, Jingtao Ding, Jie Feng, and Yong Li. Unimove: A unified model for multi-city human mobility prediction. *arXiv preprint arXiv:2508.06986*, 2025.

Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3967–3974, 2019.

Konstantin Klemmer, Esther Rolf, Caleb Robinson, Lester Mackey, and Marc Rußwurm. Satclip: Global, general-purpose location embeddings with satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 4347–4355, 2025.

Peibo Li, Maarten De Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D Salim. Large language models for next point-of-interest recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 1463–1472, 2024.

Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Yingtao Luo, Qiang Liu, and Zhaocheng Liu. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*, pp. 2177–2185, 2021.

Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. In *2019 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9595–9605. IEEE, 2019.

Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. Multi-scale representation learning for spatial feature distributions using grid cells. *arXiv preprint arXiv:2003.00824*, 2020.

Gengchen Mai, Ni Lao, Yutong He, Jiaming Song, and Stefano Ermon. Csp: Self-supervised contrastive spatial pre-training for geospatial-visual representations. In *International Conference on Machine Learning*, pp. 23498–23515. PMLR, 2023a.

Gengchen Mai, Yao Xuan, Wenyun Zuo, Yutong He, Jiaming Song, Stefano Ermon, Krzysztof Janowicz, and Ni Lao. Sphere2vec: A general-purpose location representation learning over a spherical surface for large-scale geospatial predictions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202:439–462, 2023b.

Yifang Qin, Hongjun Wu, Wei Ju, Xiao Luo, and Ming Zhang. A diffusion model for poi recommendation. *ACM Transactions on Information Systems*, 42(2):1–27, 2023.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Xuan Rao, Lisi Chen, Yong Liu, Shuo Shang, Bin Yao, and Peng Han. Graph-flashback network for next location recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1463–1471, 2022.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 811–820, 2010.

Marc Rußwurm, Konstantin Klemmer, Esther Rolf, Robin Zbinden, and Devis Tuia. Geographic location encoding with spherical harmonics and sinusoidal representation networks. In *International Conference on Learning Representations*, volume 2024, pp. 1746–1759, 2024.

Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 214–221, 2020.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Kevin Tang, Manohar Paluri, Li Fei-Fei, Rob Fergus, and Lubomir Bourdev. Improving image classification with location context. In *Proceedings of the IEEE international conference on computer vision*, pp. 1008–1016, 2015.

Vicente Vivanco Cepeda, Gaurav Kumar Nayak, and Mubarak Shah. Geoclip: Clip-inspired alignment between locations and images for effective worldwide geo-localization. *Advances in Neural Information Processing Systems*, 36:8690–8701, 2023.

Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. Where would i go next? large language models as human mobility predictors. *arXiv preprint arXiv:2308.15197*, 2023.

Ya Wen, Jixuan Cai, Qiyao Ma, Linyan Li, Xinhuan Chen, Chris Webster, and Yulun Zhou. Mora: Mobility as the backbone for geospatial representation learning at scale. In *International Conference on Learning Representations*, volume 2026, pp. 37647–37664, 2026.

Wilson Wongso, Hao Xue, and Flora D Salim. Massive-steps: Massive semantic trajectories for understanding poi check-ins–dataset and benchmarks. *arXiv preprint arXiv:2505.11239*, 2025.

Nemin Wu, Qian Cao, Zhangyu Wang, Zeping Liu, Yanlin Qi, Jielu Zhang, Joshua Ni, Xiaobai Yao, Hongxu Ma, Lan Mu, et al. Torchspatial: A location encoding framework and benchmark for spatial representation learning. *Advances in Neural Information Processing Systems*, 37: 81437–81460, 2024.

Yuxia Wu, Ke Li, Guoshuai Zhao, and Xueming Qian. Personalized long-and short-term preference learning for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(4):1944–1957, 2020.

Xiaohang Xu, Renhe Jiang, Chuang Yang, Zipei Fan, and Kaoru Sezaki. Taming the long tail in human mobility prediction. *Advances in Neural Information Processing Systems*, 37:54748–54771, 2024.

Xiaodong Yan, Tengwei Song, Yifeng Jiao, Jianshan He, Jiaotuan Wang, Ruopeng Li, and Wei Chu. Spatio-temporal hypergraph learning for next poi recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pp. 403–412, 2023.

Song Yang, Jiamou Liu, and Kaiqi Zhao. Getnext: Trajectory flow map enhanced transformer for next poi recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*, pp. 1144–1153, 2022.

Zhenyu Yu, Chunlei Meng, Yangchen Zeng, Mohd Yamani Idna Idris, Jihong Guan, and Shuigeng Zhou. Cast-poi: Candidate-conditioned spatiotemporal modeling for next poi recommendation. *arXiv preprint arXiv:2604.20845*, 2026.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

Pengpeng Zhao, Anjing Luo, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S Sheng, and Xiaofang Zhou. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(5):2512–2524, 2020.

Lin Zhong, Lingzhi Wang, Xu Yang, and Qing Liao. Comapoi: A collaborative multi-agent framework for next poi prediction bridging the gap between trajectory and language. In *Proceedings of the 48th international acm sigir conference on research and development in information retrieval*, pp. 1768–1778, 2025.

# A First View Evaluations

Our main results include revisits to the same location by the same user, which are easier to predict than other samples. In this section, we present results on only the first-view samples. Training, validation and test users are disjoint (80/10/10%). Removing a location splits a sequence into separate segments. A first view means that the target is absent from the user sequence available at inference time, including the current input location. Figure 5 instead checks only the history preceding the current input.

We compare fixed coarse RFF ($(\omega_{\min}, \omega_{\max}) = (0.5, 5)$) and fine RFF ($(5, 600)$), with 32 bands. These runs use the location exclusions defined above. Each run lasts at most 30 epochs with early stopping after four non-improving epochs, and we use the earliest checkpoint attaining the highest validation first-view training-unseen Recall@100 for the model combined with the full prior at $\alpha = 0.6$ for both the combined and model-only rows. Ranking ties receive expected recall under uniform random ordering.

Table 5: First-view recall@100 (%). Seen/unseen denotes whether the target coordinate occurs in training data.

| | New York | | Istanbul | |
|---|---|---|---|---|
| Method | Seen | Unseen | Seen | Unseen |
| Spatial | 15.27 | 19.80 | 18.10 | 21.74 |
| Full prior | 31.37 | 2.23 | 56.18 | 1.33 |
| Coarse RFF | 15.36 | 13.84 | 38.07 | 19.45 |
| Fine RFF | 18.89 | 5.69 | 53.11 | 12.62 |
| Coarse RFF + full prior | 30.06 | 6.10 | 57.60 | 5.66 |
| Fine RFF + full prior | 31.75 | 4.17 | 60.61 | 3.19 |

Fine RFF increases seen first-view recall and decreases unseen first-view recall in both cities. This comparison changes both $\omega_{\min}$ and $\omega_{\max}$. Appendix B reports separate experiments that learn frequency scales within a fixed range.

**Interpolation with spatial scores.** With the selected model parameters held fixed, we choose the inference-time Spatial weight $\alpha$ from $\{0, .1, .2, .3, .4, .5, .6, .7, .8, .9, .95, .98, 1\}$ on validation first-view training-unseen Recall@100; exact ties favor the larger heuristic weight. Selections are recorded before testing. New York selects Spatial alone ($\alpha = 1$) for both coarse and fine RFF. Istanbul selects Spatial weights 0.95 (coarse) and 0.98 (fine), yielding test recall 22.40% and 21.43%, respectively, versus 21.74% for Spatial. The estimated gain of coarse RFF plus Spatial over Spatial alone is +0.67 percentage points with a paired user-bootstrap interval of $[-0.81, 2.26]$; this does not establish an improvement. Both cities select zero full-prior weight for this subgroup when $\alpha$ is separately selected for ensembles using Gravity+history.

**Popularity ablation.** The full prior includes popularity at training and inference time. Held-out candidates have zero training counts, so training-count popularity can disadvantage them relative to candidates observed as training targets. Removing training-count popularity only at inference time, while retaining distance and history terms at heuristic weight $\alpha = 0.6$, yields coarse-RFF ensemble recall 24.11% in Istanbul versus 21.74% for Spatial (difference +2.37 points, interval $[1.38, 3.61]$). New York gives 19.86% versus 19.80%, without a comparable improvement. This ablation isolates inference-time popularity with training unchanged. Popularity accumulated after training could be useful for new locations, but requires a chronological evaluation and is left for future work.

**Scope and uncertainty.** The test candidate sets include both training-seen and training-unseen locations: 39,654 in New York and 30,227 in Istanbul. There are 5,327 and 6,611 first-view training-unseen prediction events, respectively. Some of these locations also occur in validation data. Restricting targets to the test-reserved locations, excluded from both training and validation, leaves 1,682 and 2,574 first-view events; coarse RFF plus Spatial gives 20.55% versus 19.43% for Spatial in Istanbul, with difference interval $[-0.61, 2.97]$. The exploratory 95% intervals use percentiles from 10,000 bootstrap resamples of users, keeping each user's events together and comparing methods on

the same resamples. They measure uncertainty for each comparison with the selected models fixed, excluding variation across training seeds and model selection.

## B SCORING AND SUBGROUP DETAILS

**Coordinate standardization.** Latitude and longitude are standardized separately by a batch-normalization layer. Its means and variances are estimated from training sequences and kept fixed during training and inference; its scale and offset parameters are learned. For the experiments in Appendix A, we remove every location reserved for validation or testing from training sequences before estimating these statistics.

For the wavelength conversion, $D = c\sqrt{v + 10^{-5}}/|g_{\mathrm{BN}}|$, where $v$ and $g_{\mathrm{BN}}$ are the saved coordinate variance and gain, and $c$ is the local distance in kilometers per degree at the mean latitude. For example, the New York $(0.1, 10)$ RFF checkpoint used for H3-cell evaluation gives reference wavelengths of approximately 3.5–348 km north–south and 3.0–295 km east–west. These are local distance equivalents of the Fourier input periods.

**Learned and fixed prior weights.** The training score can be evaluated directly using the saved sequence-dependent weight $\alpha_t$ in Equation 4. The fixed-weight ensembles reported here standardize the neural and prior scores separately over candidates and compute $s = (1-\alpha)(s_m - \mu_m)/\sigma_m + \alpha(h-\mu_h)/\sigma_h$. Here $s_m$ is the neural score, and $\mu_m, \sigma_m, \mu_h, \sigma_h$ are the means and standard deviations of the two score sets. Popularity uses training target counts; history uses the user sequence available at inference time.

**Real-estate training and scoring.** Training uses all US city centroids as shared negatives across the batch and $k_{\text{hard}}$ geographically proximate coordinates per example, sampled within a 1–5 km ring around the target. For evaluation at sampled coordinates, we approximate popularity and user-history densities using Gaussian kernel density estimates (KDEs):

$$h(\mathbf{x}; \mathbf{x}_{1:t}) = (1-\beta_{\text{heuristic}}) \underbrace{(\log(\epsilon + p_{\text{pop}}(\mathbf{x})) - \gamma \log(d(\mathbf{x}_t, \mathbf{x}) + \epsilon_d))}_{\text{Gravity Component}} + \beta_{\text{heuristic}} \underbrace{\log(\epsilon + p_{\text{history}}(\mathbf{x}))}_{\text{History Component}} \tag{6}$$

Here $p_{\text{pop}}$ uses training locations belonging to the true transaction city, and $p_{\text{history}}$ uses locations in the user's available history. The popularity KDE thus also uses knowledge of the transaction city.

**Location exclusions and group counts.** Coordinates are converted to 32-bit floating point before identifying unique locations. We reserve separate validation and test locations within each H3 resolution-9 cell containing POIs. Test-reserved locations are excluded from validation; final testing uses the full candidate set, including validation-reserved locations, after model selection. The available user sequence contains at most 128 locations and restarts after each removed location. Table 6 separates seen from unseen targets and first views from revisits; the smaller revisit groups support more limited conclusions.

**Constrained frequency learning.** We also train coarse and fine RFF with one learned multiplier per frequency band, restricted to $[0.5, 2]$. Each starts from the same RFF features as its fixed-frequency baseline. Neither establishes a first-view unseen improvement over that baseline in New York. The coarse standalone difference is +0.24 percentage points (interval $[-0.53, 1.04]$), and the fine difference +0.08 ($[-0.78, 0.98]$).

## C ADDITIONAL ABLATIONS AND QUALITATIVE RESULTS

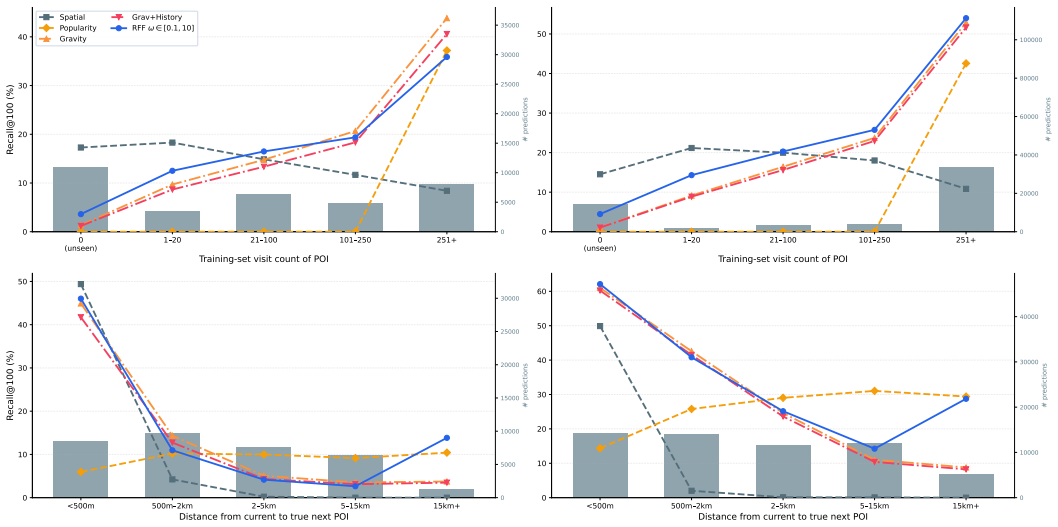



Figure 7: RFF recall relative to the baselines in New York and Istanbul, grouped by training popularity and distance from the last visited location. The first-visit filter follows Figure 5.

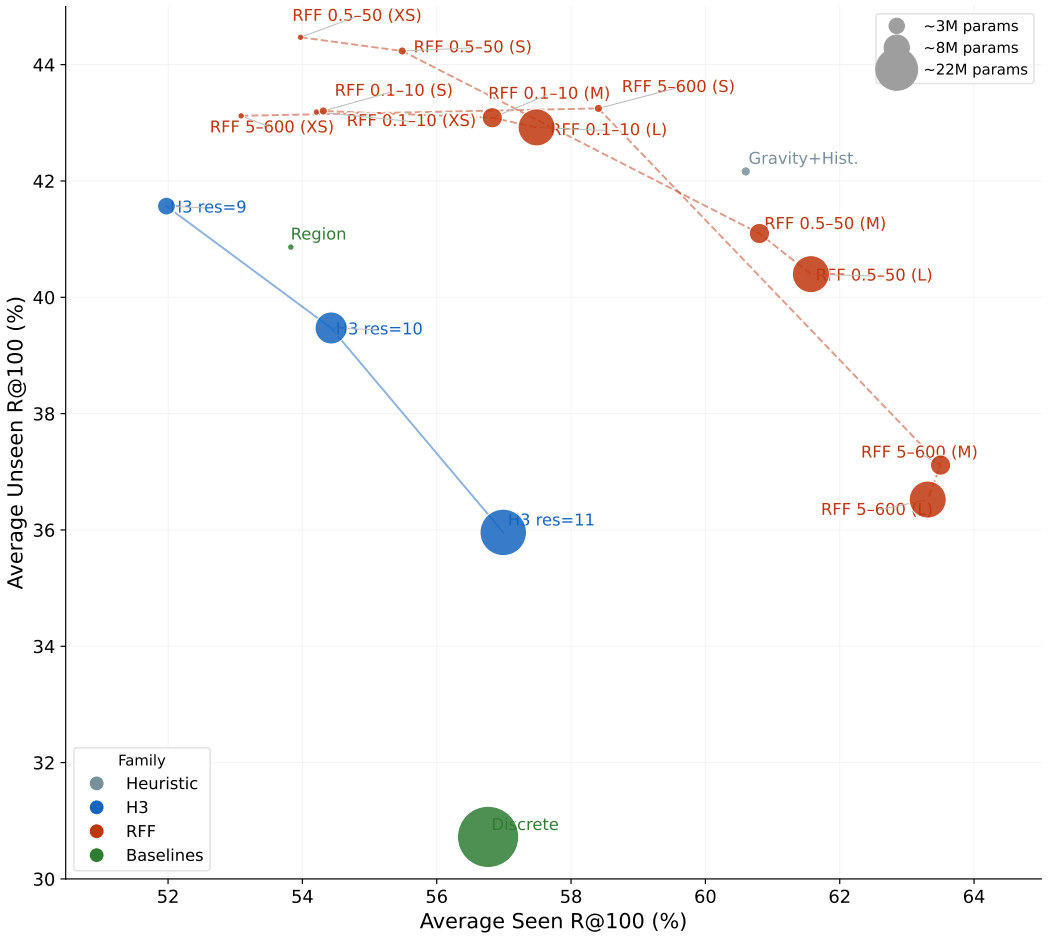



Figure 8: Seen vs. unseen recall tradeoff at varying encoder sizes, using the training-target holdout in Table 3. Among the tested RFF encoders, smaller models favor unseen recall and larger models favor seen recall. The XS variant omits the MLP and achieves the highest unseen recall in this sweep.

Table 6: Recall@100 (%) comparison between first-view samples and revisit samples. FV: target absent from the available user sequence; RV: target already in that sequence. Seen/unseen denotes presence in training data. Counts are prediction events.

| City | Method | FV seen | FV unseen | RV seen | RV unseen |
|---|---|---|---|---|---|
| New York | Events | 10577 | 5327 | 168 | 130 |
| | Spatial | 15.27 | 19.80 | 24.40 | 31.54 |
| | Full prior | 31.37 | 2.23 | 100.00 | 100.00 |
| | Coarse RFF | 15.36 | 13.84 | 31.55 | 31.54 |
| | Fine RFF | 18.89 | 5.69 | 46.43 | 22.31 |
| | Coarse + prior | 30.06 | 6.10 | 100.00 | 96.15 |
| | Fine + prior | 31.75 | 4.17 | 100.00 | 99.23 |
| Istanbul | Events | 19069 | 6611 | 822 | 232 |
| | Spatial | 18.10 | 21.74 | 47.08 | 43.97 |
| | Full prior | 56.18 | 1.33 | 100.00 | 100.00 |
| | Coarse RFF | 38.07 | 19.45 | 70.19 | 36.64 |
| | Fine RFF | 53.11 | 12.62 | 78.10 | 31.47 |
| | Coarse + prior | 57.60 | 5.66 | 100.00 | 98.28 |
| | Fine + prior | 60.61 | 3.19 | 100.00 | 98.71 |

### C.1 RFF FREQUENCY TRADEOFF ACROSS TRAINING SEEDS

We train each of three RFF ranges in six cities with 3 random seeds, giving 54 runs. A new 80/10/10% split separates training, validation and test users, with a nominal 20% of POIs per H3 resolution-9 cell withheld as training targets, following Table 3. Evaluation includes first visits and revisits. Training uses the single-step contrastive objective, and ranking uses model scores alone. We select each checkpoint by validation unseen R@100 and each city's range by its mean validation score across the three seeds. Ranking ties use expected recall under uniformly random ordering.

Each model uses 2,048 RFF features across 64 bands, a $2048 \to 2048 \to 512$ MLP, and one 512-dimensional transformer layer with eight attention heads. We train with AdamW at an initial learning rate of $4 \times 10^{-5}$ and effective batch size 512, for up to 15 epochs with early-stopping patience four.

Table 7: RFF frequency tradeoff across three training seeds. Model-only Recall@100 (%), mean ± standard deviation; † marks the validation-selected range. Every city and seed shows higher seen recall with the highest-frequency range and higher unseen recall with either lower range. Further lowering the range offers no consistent gain.

| City | $(\omega_{\min}, \omega_{\max})$ | Seen | Unseen |
|---|---|---|---|
| New York | $(0.05, 5)$ | $13.43 \pm 0.44$ | $19.73 \pm 0.89$ |
| | $(0.1, 10)^{\dagger}$ | $15.83 \pm 0.89$ | $21.34 \pm 0.19$ |
| | $(5, 600)$ | $32.80 \pm 1.96$ | $9.68 \pm 1.68$ |
| Istanbul | $(0.05, 5)$ | $27.19 \pm 3.66$ | $14.52 \pm 0.59$ |
| | $(0.1, 10)^{\dagger}$ | $34.51 \pm 2.99$ | $16.63 \pm 1.43$ |
| | $(5, 600)$ | $62.20 \pm 1.53$ | $8.47 \pm 0.23$ |
| Jakarta | $(0.05, 5)$ | $23.94 \pm 0.87$ | $18.10 \pm 0.21$ |
| | $(0.1, 10)^{\dagger}$ | $29.21 \pm 0.58$ | $20.66 \pm 0.22$ |
| | $(5, 600)$ | $50.01 \pm 2.48$ | $8.42 \pm 0.55$ |
| São Paulo | $(0.05, 5)$ | $24.33 \pm 1.93$ | $26.66 \pm 1.83$ |
| | $(0.1, 10)^{\dagger}$ | $29.50 \pm 1.18$ | $31.15 \pm 0.96$ |
| | $(5, 600)$ | $51.34 \pm 2.64$ | $20.18 \pm 1.39$ |
| Moscow | $(0.05, 5)$ | $35.65 \pm 1.48$ | $37.49 \pm 1.18$ |
| | $(0.1, 10)^{\dagger}$ | $40.16 \pm 1.33$ | $38.80 \pm 0.61$ |
| | $(5, 600)$ | $56.91 \pm 1.71$ | $17.66 \pm 0.92$ |
| Bandung | $(0.05, 5)^{\dagger}$ | $27.07 \pm 1.02$ | $21.92 \pm 0.78$ |
| | $(0.1, 10)$ | $31.08 \pm 0.93$ | $21.58 \pm 0.49$ |
| | $(5, 600)$ | $58.53 \pm 0.44$ | $5.16 \pm 0.46$ |