GeoFlow: Geo-Aware Modeling of Inter-Area Relationships in Origin-Destination Flow Prediction and Generation

Zherui Huang [1] **Guanjie Zheng** [1] **Hao Xue** [2 3] **Linghe Kong** [1]

Abstract

Origin–destination (OD) flow modeling underpins urban planning and mobility analysis, but prevailing graph-based methods often neglect salient geographic attributes, limiting their ability to model long-range and multi-area dependencies. In this paper, we introduce GeoFlow, a novel framework that (i) augments area representations with geospatial attributes, including relative positions, k-hop and geodesic distances, (ii) employs a specialized geometric-intrinsic fusion encoder design that combines graph attention for intrinsic area signals with coordinate-aware encoders for global structure, and (iii) adopts an axial-global attention decoder to capture OD-specific competitive dependencies. For OD flow generation, GeoFlow is paired with flow matching models to produce more authentic and diverse mobility samples. Empirically, GeoFlow achieves superior performance in predictive accuracy, while substantially improving generative fidelity and diversity. Ablation and analytical studies confirm the contribution of each component. Code is available at https://github.com/ZheruiHuang/GeoFlow.

1. Introduction

Origin–destination (OD) flow describes movements of people or goods between areas, reflecting social and economic activity at a macro level and supporting applications such as urban planning and behavioral analysis (Batty, 2007; Zhang et al., 2021; Wu et al., 2024). However, large-scale data collection faces statistical and privacy challenges (Simini et al., 2021; Long et al., 2023), and generalizable models are needed to inform urban development, especially in data-scarce or early-planning regions (Rong et al., 2025). Consequently, OD flow prediction and generation have become central research topics (Luca et al., 2021). Prediction methods estimate flows from area attributes (e.g., sociodemographics), while generation methods further employ random seeds and generative models to capture diverse mobility patterns (Liu et al., 2020; Rong et al., 2024). Despite their conceptual differences, both tasks share common modeling principles, and methods are often evaluated jointly (Rong et al., 2023; 2025). Early approaches, including gravity (Zipf, 1946) and radiation models (Simini et al., 2012), were followed by machine learning techniques such as Random Forest (Breiman, 2001; Pourebrahim et al., 2019) and SVR (Drucker et al., 1996; Rodríguez-Rueda et al., 2021). With the rise of deep learning (Dong et al., 2021), models such as DGM (Simini et al., 2021) and GMEL (Liu et al., 2020) achieved notable progress, and recent graph-based methods further advanced prediction and generation by modeling areas as attributed nodes and OD flows as directed weighted edges (Bojchevski et al., 2018; Liu et al., 2020; Rong et al., 2025).

Although existing state-of-the-art methods represent area networks as graph structures to exploit properties such as translational and rotational equivalent (Liu et al., 2020; Satorras et al., 2021), this representation often overlooks easily accessible geographic information, thereby complicating the learning process (Klemmer et al., 2023). In practice, relationships between areas are encoded in matrix form through pairwise adjacency and straight-line distances (Luo et al., 2024; Rong et al., 2025). While this formulation is theoretically lossless and supports network reconstruction, it fails to directly preserve fundamental geometric properties such as collinearity, which can be readily obtained from coordinate systems, and geodesic distances, which are essential but difficult to infer from pairwise matrices (Whiteley et al., 2021). Recovering such properties is challenging, and their absence undermines the model's ability to capture long-range and multi-area dependencies that are crucial for OD flow modeling (Alon & Yahav, 2020). Consequently, current feature representation and encoding strategies remain limited in representing global area-to-area relationships.

This limitation highlights a central challenge in OD flow

[1]Shanghai Jiao Tong University [2]The Hong Kong University of Science and Technology (Guangzhou) [3]The University of New South Wales, Sydney. Correspondence to: Guanjie Zheng <gjzheng@sjtu.edu.cn>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

modeling: how to represent and encode complex inter-area relationships. Graph structures offer desirable invariances and computational advantages, yet they often fail to capture long-range and multi-area geographic interactions (Bamberger et al., 2025). Coordinate-based representations encode geometric attributes directly but are sensitive to the choice of reference origin, scale, and orientation (Bronstein et al., 2021). Besides, some geospatial attributes (e.g., geodesic distance) remain nontrivial to recover from matrix representations (Whiteley et al., 2021). Accordingly, integrating complementary representation methods and developing encoding–decoding strategies that preserve geometric attributes while remaining insensitive to reference choices are essential for improving OD flow prediction and generation.

This paper introduces GeoFlow, a novel framework for OD flow prediction and generation that systematically integrates geospatial attributes and encoder–decoder design. We first perform geospatial attribute augmentation by incorporating k-hop distance, geodesic distance, and relative position alongside straight-line distance, allowing the model to access key spatial relationships directly rather than extracting them from matrices. In the encoder, we propose *geometry-intrinsic fusion* encoder, combining Cartesian coordinates with graph-based representations: a graph attention module encodes intrinsic area attributes (e.g., points of interest) and aggregates influences from neighboring areas, while carefully chosen coordinate centers and orientations preserve relative positions, enabling the model to capture long-range and multi-area dependencies. In the decoder, we introduce an *axial-global attention* mechanism that models competitive relationships between areas while reducing computational costs, making joint attention across hundreds of areas feasible. Finally, for flow generation, we adopt flow matching models, which provide higher stability, efficiency, and generalization than diffusion models (Lipman et al., 2022; Huang et al., 2025). To the best of our knowledge, this is the first work to apply flow matching in the OD flow generation task.

Extensive experiments demonstrate that GeoFlow markedly improves both OD flow prediction and generation. For the prediction task, it consistently achieves substantially lower errors than existing baselines, reflecting a clear advantage. In the generation task, GeoFlow attains superior performance in reconstruction accuracy and produces more authentic and diverse samples. Ablation studies on geospatial representations show that even lightweight attribute augmentations provide notable gains, highlighting the practical value of these previously overlooked features, and ablation experiments on the encoder and decoder architectures confirm their effectiveness. Furthermore, analytical experiments demonstrate the rationale behind the overall design and offer deeper insight into the role of geospatial attributes.

Our contributions can be summarized in three main parts:

1. We propose GeoFlow, a novel method for OD flow prediction and generation that systematically enhances feature representation and model architecture. Each area is augmented with crucial geospatial attributes to capture long-range and multi-area relationships. A geometric-intrinsic fusion encoder module encodes area properties, while an axial-global attention mechanism models competitive dependencies among areas.
2. We evaluate GeoFlow on both OD flow prediction and generation tasks. GeoFlow achieves substantial improvements in prediction accuracy and generates more realistic and diverse samples.
3. We conduct ablation experiments on model components, showing that the proposed approaches substantially improve model performance. Additionally, analytical experiments on geospatial features provide deeper insights into the dynamics of OD flow modeling.

2. Related Work

Existing work on origin–destination (OD) flow prediction or generation falls into two broad categories: principle-driven methods and data-driven methods. **Principle-driven methods** derive flows from explicit mechanistic physical models and statistical formulations. Zipf (1946) employs the gravity model to explain intercity human mobility. Tomazinis (1962) introduces the notion of competitive opportunities and incorporates probability theory to estimate OD flows. Simini et al. (2012) propose a parameter-free radiation model and enhance the accuracy of flow modeling. These formulations provide transparent explanations of aggregate flows but lack flexibility for complex empirical patterns. **Data-driven methods**, which learn flow patterns directly from observed mobility data, have emerged and revitalized the field in recent years. Many studies apply classical machine learning algorithms to OD flow generation. Random Forest (RF) demonstrates strong potential in the task (Pourebrahim et al., 2019). Gradient Boosting Regression Trees (GBRT) enhance predictive accuracy through boosting and have been effectively applied to urban OD prediction (Robinson & Dilkina, 2018). Support Vector Regression (SVR) employs kernel methods to estimate flows between regions based on their urban attributes (Rodríguez-Rueda et al., 2021). Despite their demonstrated success, classical machine learning methods have limitations in fully capturing the complex dynamics and high-dimensional interactions inherent in urban OD flows, paving the way for deep learning-based techniques. The Deep Gravity Model (DGM), inspired by traditional gravity models, uses multi-layer perceptrons to estimate flow distributions (Simini et al., 2021).

Graph-based methods, including NetGAN (Bojchevski et al., 2018) and Geographical Multi-task Embedding Learning (GMEL) (Liu et al., 2020), leverage random walks or graph neural networks to capture spatial and structural information, enabling more accurate OD flow generation. More recent approaches, such as DiffODGen (Rong et al., 2023) and WEDAN (Rong et al., 2025), employ diffusion models to model graph topology and edge weights conditioned on urban attributes. Despite their ability to capture complex spatial and structural patterns, these methods typically abstract areas as attributed nodes, overlooking spatial proximity and inter-area interactions and thereby hindering further advances. TransFlower (Luo et al., 2024) acknowledges the importance of relative location in capturing directional relations between areas; however, its reliance on pairwise attention and limited integration of topological structure restricts its capacity to model OD flows.

3. Method

3.1. Problem Definition, Formulation, and Notation

We consider OD flow modeling as a unified paradigm: the core problem is to characterize flows among areas under given attributes. The paradigm can be instantiated in two ways: (1) When the goal is to estimate missing or future entries, the formulation reduces to prediction; (2) When the goal is to capture the full stochastic behavior of flows, it becomes generation. Both tasks share the same underlying setting but differ in the learning objectives.

Formally, let $R = \{A_i\}_{i=1}^n$ denote a region partitioned into n non-overlapping areas; each area A_i is associated with a feature vector $\boldsymbol{x}_i \in \mathbb{R}^{d_\mathrm{x}}$, and the collection is denoted by a feature matrix $\boldsymbol{X} = [\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_n] \in \mathbb{R}^{n \times d_\mathrm{x}}$. OD flows are represented as a non-negative matrix $\boldsymbol{F} \in \mathbb{R}_{\geq 0}^{n \times n}$ where entry $F_{i,j}$ specifies the flow from A_i to A_j, while inter-area geographic or topological information is captured by a tensor $\boldsymbol{G} \in \mathbb{R}^{n \times n \times d_\mathrm{g}}$, where d_g is the dimension of the per-pair geographic descriptor, including relationships such as adjacency and distance. A single data instance is represented as $D = \{R, \boldsymbol{X}, \boldsymbol{G}, \boldsymbol{F}\}$.

Prediction as supervised estimation. Given a training index set $\Omega \subseteq \{1, \ldots, n\}^2$, the prediction task seeks to estimate $\hat{\boldsymbol{F}} = f(\boldsymbol{X}, \boldsymbol{G}; \boldsymbol{\theta})$ by minimizing the discrepancy between observed and predicted flows:

$$\min_{\boldsymbol{\theta}} \ \mathcal{J}_\mathrm{pred}(\boldsymbol{\theta}) = \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \ell\big(F_{i,j}, \hat{F}_{i,j}\big), \tag{1}$$

where $\ell(\cdot, \cdot)$ is a per-entry loss (e.g., squared error).

Generation as distributional modeling. Beyond point estimation, the generation task requires capturing the full distributional structure of OD flows. This can be formulated via a conditional generator $\hat{\boldsymbol{F}} = g(\boldsymbol{X}, \boldsymbol{G}, z; \boldsymbol{\theta})$, with latent seed $z \sim p_z$, where p_z is the seed distribution, optimized under

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \mathcal{J}_\mathrm{gen}(\boldsymbol{\theta}) = \\ &\quad \mathbb{E}_{z \sim p_z} \big[\ell_\mathrm{rec}(\boldsymbol{F}, \hat{\boldsymbol{F}}) + \alpha \ell_\mathrm{dist}(\boldsymbol{F}, \hat{\boldsymbol{F}}) + \beta \ell_\mathrm{div}(\hat{\boldsymbol{F}})\big], \end{aligned} \tag{2}$$

where ℓ_rec enforces reconstruction fidelity, ℓ_dist measures distributional discrepancy between realistic data and generated ones, and ℓ_div promotes generation diversity. Hyperparameters α and β balance reconstruction fidelity, distributional realism, and sample diversity.

3.2. GeoFlow

Unlike trajectory forecasting, which requires reconstructing detailed travel paths, OD flow tasks focus on aggregate movements between areas. A straightforward baseline concatenates origin and destination features to predict or generate flows from pairwise representations. However, this formulation treats areas as isolated nodes and neglects critical geographic and multi-area dependencies.

To address these limitations, we partition area attributes into two categories and design dedicated encoders. The first, *geometric attributes*, includes coordinates, pairwise distances, and spatial descriptors such as geodesic proximity, which support global and long-range interaction modeling. The second, *intrinsic attributes*, covers local covariates such as points of interest (POIs) and socio-demographics, which characterize production and attraction capacity and primarily affect nearby areas. These two categories differ in semantics, scale, and propagation behavior: geometric attributes impose structural and long-range constraints, whereas intrinsic attributes capture localized demand and short-range interactions. Accordingly, we encode them separately and fuse their representations downstream to integrate global structure with local context. For decoding, we introduce an axial attention mechanism that models competitive dependencies in human mobility: OD pairs sharing the same origin or destination exhibit stronger correlations than unrelated pairs. This mechanism is further combined with global attention, resulting in a decoder that enhances training efficiency while achieving superior performance within comparable computational budgets. Fig. 1 illustrates the overall architecture of the GeoFlow framework.

The following sections elaborate on these components: Sec. 3.2.1 describes the augmented geospatial attributes, Sec. 3.2.2 presents the encoding strategies, and Sec. 3.2.3 details the axial-global attention mechanism.

For notational simplicity, we use $A_{i,j}$ (or $A_{i,j,k}$) to denote the (i, j)-th entry of a two-dimensional matrix (or the (i, j, k)-th entry of a three-dimensional tensor) $\boldsymbol{A}$. We write $\boldsymbol{A}_{i,:}$ for the i-th row of $\boldsymbol{A}$ and $\boldsymbol{A}_{:,j}$ for its j-th column. Anal-

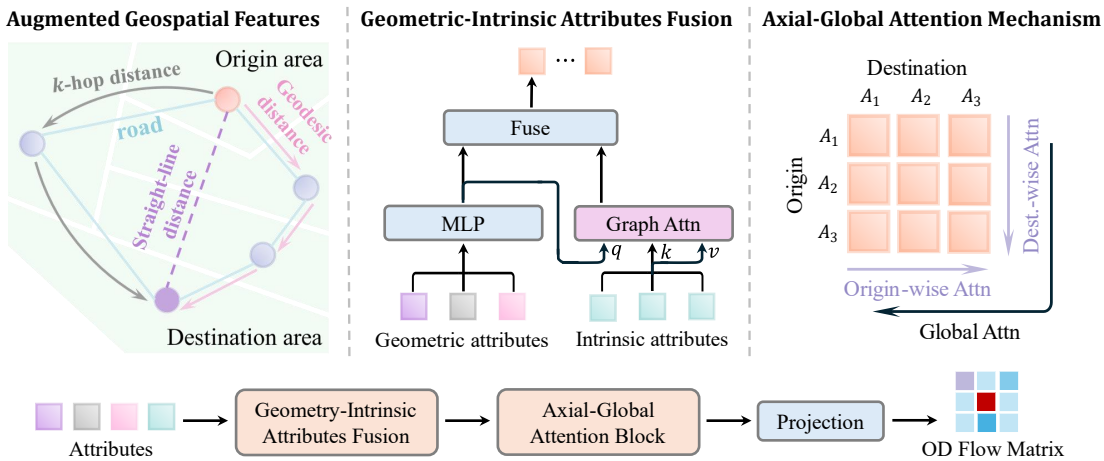

Figure 1. **Framework of GeoFlow.** The framework comprises three main components: the augmented geospatial relation descriptors, the encoder for fusing geometric and intrinsic attributes, and the axial–global attention–based decoder.

ogous notation similarly applies to tensors of order three and higher.

3.2.1. Geospatial Feature Augmentation

Since graph-based approaches typically store geographical relationships as adjacency or distance matrices, recovering relative positions or geodesic distances from such representations is difficult, limiting the model's ability to capture essential factors that shape travel behavior (Whiteley et al., 2021; Klemmer et al., 2023). Rather than relying solely on deeper or wider networks, explicitly providing these salient features offers a direct and cost-effective alternative, while also supplying inductive signals that stabilize and accelerate training.

In this work, we augment each area pair with three additional geographic relation descriptors: **relative position**, **k-hop distance**, and **geodesic distance**. The relative position is defined as the coordinate difference between two areas under a Cartesian system assigned to the region. To ensure translational and rotational invariance while retaining meaningful orientation, we place the origin at the region's geometric center, align the x-axis with the principal orientation (the direction of greatest variance), and normalize coordinates to $[-1, 1]$, thereby mitigating scale differences across regions. The k-hop distance is the length of the shortest unweighted path between two areas, reflecting topological proximity. The geodesic distance, or free-flow distance, is the minimal travel length along the network, capturing realistic accessibility. Fig. 1 upper left illustrates an example of the augmented geospatial features. By incorporating these attributes, GeoFlow provides richer geographic context, improving both predictive accuracy and training stability in OD flow modeling.

3.2.2. Encoder: Geometric–Intrinsic Fusion

To effectively leverage the enriched geospatial features, we employ a dedicated encoder that fuses geometric and intrinsic area attributes. As shown in Fig. 1 upper middle, for the geometric attributes, which include relative position, straight-line distance, k-hop distance, and geodesic distance, we encode each pairwise attribute vector using a multilayer perceptron (MLP). These geometric attributes are normalized to remove scale-induced biases. By adopting a consistent regional coordinate system and applying systematic normalization, the model can more readily infer derived geometric cues (e.g., relative bearing) within a fixed domain. Normalized relative positions also reduce biases introduced by absolute location and orientation, obviating the need for explicit trigonometric encodings. For a region with n areas, let the geometric information be denoted by the tensor $\boldsymbol{G} \in \mathbb{R}^{n \times n \times d_{\mathrm{g}}}$. We employ an MLP $\phi_{\mathrm{g}} : \mathbb{R}^{d_{\mathrm{g}}} \to \mathbb{R}^{d_{\mathrm{e}}}$ elementwise to obtain an encoded geographic embedding

$$\boldsymbol{E}_{\mathrm{g}} = \phi_{\mathrm{g}}(\boldsymbol{G}) \in \mathbb{R}^{n \times n \times d_{\mathrm{e}}}, \tag{3}$$

where d_{e} is the dimension of embedding $\boldsymbol{E}_{\mathrm{g}}$.

For intrinsic attributes, given their dependence on neighboring areas, we adopt a graph attention-based encoding (Veličković et al., 2017). Unlike prior approaches that rely solely on local message passing and are therefore constrained by network depth, we introduce an axial-attention design to enhance information propagation efficiency. An area's influence on others depends on both its intrinsic attributes and geographic relations. Accordingly, GeoFlow conditions attention on the geographic tensor when computing inter-area interactions. Formally, let $\boldsymbol{X} \in \mathbb{R}^{n \times d_{\mathrm{x}}}$ denote the intrinsic attribute matrix. We first project $\boldsymbol{X}$ as intrinsic embedding $\boldsymbol{E}_{\mathrm{x}}$ via an MLP $\phi_{\mathrm{x}} : \mathbb{R}^{d_{\mathrm{x}}} \to \mathbb{R}^{d_{\mathrm{e}}}$:

$$\boldsymbol{E}_{\mathrm{x}} = \phi_{\mathrm{x}}(\boldsymbol{X}), \tag{4}$$

and then update $\boldsymbol{E}_{\mathrm{x}}$ via an attribute aggregation. Specifically, we first use a learnable weighted vector $\boldsymbol{w}_{\mathrm{geo}} \in \mathbb{R}^{d_{\mathrm{g}}}$ to produce a score matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$:

$$\boldsymbol{S} = \mathrm{softmax}(\boldsymbol{G}\boldsymbol{w}_{\mathrm{geo}}). \tag{5}$$

Intuitively, an area's intrinsic attributes are influenced by information that propagates from other areas through the spatial structure. The score matrix $\boldsymbol{S}$ is designed to capture this influence pattern, where $S_{i,j}$ denotes the influence weight from area A_j to area A_i, inferred from their geographic descriptors. Next, we use $\boldsymbol{S}$ to modulate query formation from intrinsic attributes and compute pairwise aggregation weights $\boldsymbol{A}$ via scaled dot-product:

$$\boldsymbol{A} = \mathrm{softmax}\left(\frac{(\boldsymbol{S}\boldsymbol{X}\boldsymbol{W}_{\mathrm{qx}})(\boldsymbol{X}\boldsymbol{W}_{\mathrm{kx}})^{\top}}{\sqrt{d_{\mathrm{h}}}}\right), \tag{6}$$

$$(\widetilde{\boldsymbol{E}}_{\mathrm{x}})_{i,:} = \sum_{k=1}^{n} A_{i,k}\,(\boldsymbol{E}_{\mathrm{x}})_{k,:}. \tag{7}$$

where $\boldsymbol{W}_{\mathrm{qx}}, \boldsymbol{W}_{\mathrm{kx}} \in \mathbb{R}^{d_{\mathrm{x}} \times d_{\mathrm{h}}}$ are learnable projection matrices, and d_{h} represents the hidden dimension.

Then, we employ an MLP to integrate the geographic embedding $\boldsymbol{E}_{\mathrm{g}}$ and the intrinsic embedding $\widetilde{\boldsymbol{E}}_{\mathrm{x}}$, producing the joint representation $\boldsymbol{Z}_0 \in \mathbb{R}^{n \times n \times d_{\mathrm{z}}}$. Formally, let $\phi_{\mathrm{f}} : \mathbb{R}^{3d_{\mathrm{e}}} \to \mathbb{R}^{d_{\mathrm{z}}}$ denote the fusion MLP. For each OD pair (i, j), the fused embedding is obtained as

$$(\boldsymbol{Z}_0)_{i,j,:} = \phi_{\mathrm{f}}\big(\,[(\boldsymbol{E}_{\mathrm{g}})_{i,j,:} \,\|\, (\widetilde{\boldsymbol{E}}_{\mathrm{x}})_{i,:} \,\|\, (\widetilde{\boldsymbol{E}}_{\mathrm{x}})_{j,:}]\,\big), \tag{8}$$

where $\|$ denotes concatenation. The joint embedding $\boldsymbol{Z}_0$ serves as the unified representation for subsequent OD flow prediction and generation.

3.2.3. Decoder: Axial–Global Attention

Naive full attention can capture interactions among OD pairs (Vaswani et al., 2017). Still, it is computationally prohibitive: for a region with n areas, the n^2 OD pairs incur an $O(n^4)$ cost for pairwise attention, limiting the depth of dynamics the model can afford under practical budgets (Ho et al., 2019). Empirically, OD pairs that share an origin or share a destination are more strongly coupled. Fixing an origin induces comparisons among candidate destinations, while a destination's finite capacity differentiates travelers from different origins, underscoring the relevance of origin- and destination-conditioned interactions. Accordingly, we adopt an axial attention scheme that alternately applies origin-wise and destination-wise attention for stacked blocks. This design efficiently captures origin-conditioned and destination-conditioned relational patterns without evaluating full pairwise attention, reducing computational cost to $O(n^3)$. To preserve the rich interactions, we introduce a redesigned *compact* global-attention mechanism to capture long-range and multi-area dependencies with $O(n^2)$ computational complexity. The decoder takes the unified OD representation $\boldsymbol{Z}_0$ from Eq. 8 as input and refines it with a stack of L_{a} axial-attention layers and L_{g} compact global-attention layers.

Axial-attention mechanism. Each axial-attention layer consists of an origin-wise sublayer followed by a destination-wise sublayer, and thus performs two successive updates of the embedding. For the $(k{+}1)$-th axial-attention layer, where $k \in \{0, 1, \ldots, L_{\mathrm{a}} - 1\}$, we use $\boldsymbol{Z}_{2k}$ and $\boldsymbol{Z}_{2k+1}$ to represent the inputs to the corresponding origin-wise and destination-wise attention sublayers, respectively. Specifically, $\boldsymbol{Z}_0$ is fed into the first origin-wise attention layer as input.

We first introduce the standard scaled dot-product attention (Vaswani et al., 2017) as

$$\mathrm{Attn}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{(\boldsymbol{Q}\boldsymbol{W}_{\mathrm{q}})(\boldsymbol{K}\boldsymbol{W}_{\mathrm{k}})^{\top}}{\sqrt{d_{\mathrm{k}}}}\right)\boldsymbol{V}\boldsymbol{W}_{\mathrm{v}}, \tag{9}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$ are query, key, value, respectively. $\boldsymbol{W}_{\mathrm{q}}$, $\boldsymbol{W}_{\mathrm{k}}$, $\boldsymbol{W}_{\mathrm{v}}$ are learnable matrices. d_{k} is the feature dimension. The formulation can be readily extended to the multi-head one; here, we omit it for brevity. The self-attention can be written as

$$\mathrm{SelfAttn}(\boldsymbol{M}) = \mathrm{Attn}(\boldsymbol{M}, \boldsymbol{M}, \boldsymbol{M}). \tag{10}$$

The origin-wise and destination-wise attention for the $(k{+}1)$-th axial-attention layer is then formulated as

$$\left(\mathcal{A}^{\mathrm{orig}}_{k+1}(\boldsymbol{Z}_{2k})\right)_{i,:,:} = \mathrm{SelfAttn}\left((\boldsymbol{Z}_{2k})_{i,:,:}\right) \tag{11}$$

$$\left(\mathcal{A}^{\mathrm{dest}}_{k+1}(\boldsymbol{Z}_{2k+1})\right)_{:,j,:} = \mathrm{SelfAttn}\left((\boldsymbol{Z}_{2k+1})_{:,j,:}\right). \tag{12}$$

We define the axial-attention mechanism for the $(k{+}1)$-th layer as

$$\mathcal{A}^{\mathrm{axial}}_{k+1}(\boldsymbol{Z}_{2k}) = \left(\mathcal{A}^{\mathrm{dest}}_{k+1} \bullet \mathcal{A}^{\mathrm{orig}}_{k+1}\right)(\boldsymbol{Z}_{2k}), \tag{13}$$

where $\bullet$ denotes function composition. After L_{a} axial-attention layers, we obtain $\boldsymbol{Z}_{2L_{\mathrm{a}}}$, the output representation of the axial-attention block. It explicitly models interactions among areas in direct competition (e.g., sharing the same origin or destination context), while implicitly aggregating global information through iterative propagation across layers.

Compact global-attention mechanism. To avoid losing the complex, multi-area interactions among OD pairs, we introduce a computation-efficient compact global-attention pathway that complements the axial-attention blocks. Instead of applying full attention to all n^2 OD pairs, we first aggregate OD information at the area level, then further apply global-attention layers to provide an additional low-resolution channel for information exchange across the entire region. Given $\boldsymbol{Z}_{2L_{\mathrm{a}}} \in \mathbb{R}^{n \times n \times d_{\mathrm{z}}}$, we compute origin-wise and destination-wise pooled representations by average

pooling:

$$(\boldsymbol{H}_{\text{orig}})_{i,:} = \frac{1}{n}\sum_{j=1}^{n}(\boldsymbol{Z}_{2L_{\text{a}}})_{i,j,:}, \tag{14}$$

$$(\boldsymbol{H}_{\text{dest}})_{j,:} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{Z}_{2L_{\text{a}}})_{i,j,:}, \tag{15}$$

where $\boldsymbol{H}_{\text{orig}}, \boldsymbol{H}_{\text{dest}} \in \mathbb{R}^{n\times d_{\text{z}}}$ summarize OD information for each origin and destination, respectively. Then we concatenate them and pass the result through a projection:

$$\boldsymbol{H}_0 = \phi_{\text{h}}\big([\boldsymbol{H}_{\text{orig}} \parallel \boldsymbol{H}_{\text{dest}}]\big), \tag{16}$$

where $\phi_{\text{h}} : \mathbb{R}^{2d_{\text{z}}} \to \mathbb{R}^{d_{\text{z}}}$ is a projection MLP. Here, we interpret $\boldsymbol{H}_0 \in \mathbb{R}^{n\times d_{\text{z}}}$ as an area-level latent state that consolidates OD context from the perspectives of serving as the origin and the destination, which is subsequently refined by global self-attention

$$\boldsymbol{H}_1 = \text{SelfAttn}(\boldsymbol{H}_0). \tag{17}$$

We repeat this operation L_{g} times, yielding the output $\boldsymbol{H}_{L_{\text{g}}}$, which encodes explicit global interactions across all areas via a compact set of area-level embeddings. Through this global-attention pathway, all areas interact explicitly via a compact set of representations at the aggregated level, enabling GeoFlow to capture high-order, long-range, and origin– and destination-crossing dependencies while keeping the computational cost manageable.

Then, we unsqueeze the $\boldsymbol{H}_{L_{\text{g}}}$ along the origin and destination axes to form origin-aligned and destination-aligned views $\widetilde{\boldsymbol{H}}_{\text{orig}}$ and $\widetilde{\boldsymbol{H}}_{\text{dest}} \in \mathbb{R}^{n\times n\times d_{\text{z}}}$ by

$$(\widetilde{\boldsymbol{H}}_{\text{orig}})_{i,j,:} = (\boldsymbol{H}_{L_{\text{g}}})_{i,:}, \quad \forall\, j \tag{18}$$

$$(\widetilde{\boldsymbol{H}}_{\text{dest}})_{i,j,:} = (\boldsymbol{H}_{L_{\text{g}}})_{j,:}, \quad \forall\, i, \tag{19}$$

and expand them back to the $n \times n$ OD grid via a Hadamard product so that each OD pair receives the corresponding origin- and destination-conditioned context,

$$\widetilde{\boldsymbol{H}}_{\text{OD}} = \widetilde{\boldsymbol{H}}_{\text{orig}} \circ \widetilde{\boldsymbol{H}}_{\text{dest}}. \tag{20}$$

Finally, we pass the resulting OD-level embedding through an MLP to obtain the final decoder output,

$$\boldsymbol{Z}_{\text{dec}} = \phi_{\text{out}}(\widetilde{\boldsymbol{H}}_{\text{OD}}). \tag{21}$$

3.3. OD Flow Prediction and Generation

GeoFlow provides an expressive, geometry-aware representation of intra-region area relationships that preserves multi-area spatial and relational cues necessary for accurate OD modeling. We apply GeoFlow to both OD flow prediction and generation tasks.

Prediction. For the prediction task, we append an MLP to the decoder output to produce the predicted OD flow matrix. Let $\boldsymbol{Z}_{\text{dec}} \in \mathbb{R}^{n\times n\times d_{\text{z}}}$ denote the decoder output; we project it via MLP $\phi_{\text{p}} : \mathbb{R}^{d_{\text{z}}} \to \mathbb{R}$, as

$$\hat{\boldsymbol{F}} = \phi_{\text{p}}(\boldsymbol{Z}_{\text{dec}}) \tag{22}$$

where $\hat{\boldsymbol{F}} \in \mathbb{R}^{n\times n}$ is the predicted OD flow matrix. The prediction loss is the average element-wise squared error

$$\mathcal{L}_{\text{pred}} = \frac{1}{n^2}\left\|\hat{\boldsymbol{F}} - \boldsymbol{F}\right\|_F^2, \tag{23}$$

where $\boldsymbol{F}$ is the ground-truth OD flow matrix and $\|\cdot\|_F$ denotes the Frobenius norm.

Generation. For the generation task, we adopt continuous-time flow matching (Lipman et al., 2022) conditioned on GeoFlow's output embedding. We embed the intermediate instantaneous state at timestamp t, denoted by $\boldsymbol{F}_t$, via an extra MLP encoder ψ and fuse it with geometry and intrinsic embeddings. Formally, in Eq. 8, the fused embedding is turned to obtain as

$$(\boldsymbol{Z}_t)_{i,j,:} = \phi_{\text{f}}^{\text{gen}}\big(\,[(\boldsymbol{E}_{\text{g}})_{i,j,:} \| (\widetilde{\boldsymbol{E}}_{\text{x}})_{i,:} \| (\widetilde{\boldsymbol{E}}_{\text{x}})_{j,:} \| \psi(\boldsymbol{F}_t)_{i,j,:}]\,\big). \tag{24}$$

The decoder then maps $\boldsymbol{Z}_t$ to a time-conditioned OD representation $\boldsymbol{Z}_{\text{dec},t}$, which is followed by an MLP to predict the velocity field $\hat{\boldsymbol{V}}(\boldsymbol{F}_t, t, \boldsymbol{Z}_{\text{dec},t}; \boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$. The model is trained by mean squared error against the analytic target velocity induced by a perturbation schedule. The loss function can be written as

$$\mathcal{L}_{\text{gen}} = \mathbb{E}_{D,t,\xi}\left\|\hat{\boldsymbol{V}} - \boldsymbol{V}_{\text{tgt}}(\boldsymbol{F}_t \mid \boldsymbol{F})\right\|_F^2. \tag{25}$$

For a common linear-Gaussian schedule $\boldsymbol{F}_t = \alpha(t)\boldsymbol{\xi} + \sigma(t)\boldsymbol{F}$, where $\alpha(t)$ and $\sigma(t)$ are predefined scalar schedule functions that control the interpolation from Gaussian noise $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ to data $\boldsymbol{F}$. In particular, $t = 0$ corresponds to the noise distribution and $t = 1$ corresponds to the data distribution. $\boldsymbol{V}_{\text{tgt}}$ represents the target velocity. At inference, we integrate the learned ordinary differential equation $\mathrm{d}\hat{\boldsymbol{F}}_t/\mathrm{d}t = \hat{\boldsymbol{V}}$ from $\hat{\boldsymbol{F}}_0 = \xi$ to obtain $\hat{\boldsymbol{F}}_1$. This formulation keeps training stable, conditions sampling on geometry and area identity, and yields faster inference than diffusion methods. For more details, please refer to Appendix A.1.

4. Experiments

4.1. Setup

Objectives. We aim to address three core questions through our experiments: (1) *How does GeoFlow perform compared to existing methods?* To this end, we design a suite of comprehensive metrics to systematically evaluate GeoFlow against competitive baselines in Sec. 4.2. (2) *Are the individual components of GeoFlow effective?* We conduct targeted

Table 1. **Quantitative comparison on OD flow prediction and generation tasks.** Results are averaged over three independent splits. ↑ indicates higher is better, and ↓ indicates lower is better.

Method	Commuting			Freight			Air Travel			Div. ↑
	CPC ↑	NRMSE ↓	JSD ↓	CPC ↑	NRMSE ↓	JSD ↓	CPC ↑	NRMSE ↓	JSD ↓	
Prediction										
Gravity Model (Zipf, 1946)	0.287	5.631	0.575	0.314	2.968	0.380	0.396	1.505	0.044	–
RF (Pourebrahim et al., 2019)	0.403	3.274	0.300	0.440	1.674	0.191	0.584	0.904	0.034	–
GBRT (Robinson & Dilkina, 2018)	0.401	2.981	0.316	0.430	2.234	0.209	0.563	0.832	0.026	–
SVR (Rodríguez-Rueda et al., 2021)	0.369	3.048	0.353	0.409	1.577	0.162	0.542	0.785	0.026	–
DGM (Simini et al., 2021)	0.383	2.943	0.312	0.451	1.529	0.153	0.538	0.766	0.025	–
GMEL (Liu et al., 2020)	0.380	3.155	0.297	0.431	1.504	0.127	0.531	0.686	0.023	–
TransFlower (Luo et al., 2024)	0.486	2.146	0.281	0.593	1.042	0.098	0.773	0.463	0.016	–
$\text{GeoFlow}_{\text{prediction}}$ (Ours)	**0.604**	**1.727**	**0.171**	**0.645**	**0.606**	**0.048**	**0.808**	**0.360**	**0.013**	–
Generation										
NetGAN (Bojchevski et al., 2018)	0.405	6.404	0.252	0.346	1.500	0.199	0.370	1.745	0.046	0.065
DiffODGen (Rong et al., 2023)	0.464	7.852	0.204	0.408	1.481	0.135	0.439	1.913	0.036	0.048
WEDAN (Rong et al., 2025)	0.527	4.609	0.196	0.480	1.183	0.128	**0.486**	1.485	0.036	0.053
$\text{GeoFlow}_{\text{generation}}$ (Ours)	**0.566**	**4.330**	**0.174**	**0.485**	**0.883**	**0.111**	0.457	**1.213**	**0.031**	**0.069**

ablation studies to verify the contribution of each component in Sec. 4.3. (3) *Is the design motivation of GeoFlow well-founded?* We perform a series of analysis experiments to demonstrate that our observations and motivations are not merely intuitive but also supported by empirical evidence in Sec. 4.4.

Dataset. We evaluate three OD flow regimes, covering commuting, freight logistics, and air-travel mobility. For **commuting flows**, we use the CommutingODGen dataset, a large-scale open-source benchmark for OD flow-related tasks that contains commuting flows across diverse urban environments in the United States (Rong et al., 2025). For **freight logistics**, we use the Freight Analysis Framework (FAF) database[1] released by the Bureau of Transportation Statistics in the U.S. Department of Transportation, which provides U.S. freight flows with annual estimates from 2018 to 2024. For **air-travel mobility**, we use the Tourism Volumes and Flows series[2] published by New Zealand's Ministry of Business, Innovation and Employment, which reports destination-level visitor volumes and origin-specific visitor flows. More details on dataset construction are provided in Appendix B.1.

Metrics. Since GeoFlow is evaluated on both OD flow prediction and generation, we adopt a set of metrics from multiple perspectives. (1) **Numerical accuracy.** For the prediction task, we employ Common Part of Commuting (CPC) and Normalized Root Mean Square Error (NRMSE) to quantify the alignment between predicted flows and ground truth values. (2) **Distributional similarity.** Following prior work (Rong et al., 2023; 2025), we use Jensen–Shannon Divergence (JSD) to measure the discrepancy between the generated and true distributions of inflow, outflow, and OD flows. (3) **Diversity.** For the generation task, in addition to reconstruction accuracy and distributional similarity, we adopt the sample-level dissimilarity to assess diversity (Div.) across generated samples, which helps prevent potential mode collapse. Please refer to Appendix B.2 for more detailed information on metrics.

[1] www.bts.gov/faf

[2] www.mbie.govt.nz/immigration-and-tourism/tourism-research-and-data/tourism-data-releases

4.2. Comparative Evaluation

Baselines. We compare GeoFlow against a broad and representative set of baselines: the classical gravity model; machine-learning regressors including RF, GBRT, and SVR; deep-learning approaches DGM, GMEL, and TransFlower; and recent graph-based methods NetGAN, DiffODGen, and WEDAN. More details and discussion about the baselines are provided in Appendix G.

Results and analysis. Quantitative results are summarized in Tab. 1. Overall, GeoFlow achieves the best overall performance across both prediction and generation tasks on the three OD regimes. For prediction, deep models generally outperform classical baselines, and GeoFlow further improves over the strongest transformer baseline TransFlower, with notable gains in CPC and reductions in NRMSE and JSD, e.g., on Commuting (CPC 0.604 vs. 0.486, NRMSE 1.727 vs. 2.146, JSD 0.171 vs. 0.281), Freight (CPC 0.645 vs. 0.593, NRMSE 0.606 vs. 1.042, JSD 0.048 vs. 0.098), and Air Travel (CPC 0.808 vs. 0.773, NRMSE 0.360 vs. 0.463, JSD 0.013 vs. 0.016). For generation, recent graph-

Table 2. **Ablation studies.** SL, Rel., k-hop, and FF denote straight-line distance, relative coordinates, k-hop distance, and free-flow (geodesic) distance. Enc. and Dec. denote the proposed geo-conditioned encoder and axial-global attention decoder. Unchecked Enc./Dec. entries indicate that the corresponding component is replaced by a baseline alternative.

ID	Input features				Architecture		Prediction			Generation		
	SL	Rel.	k-hop	FF	Enc.	Dec.	CPC ↑	NRMSE ↓	JSD ↓	CPC ↑	NRMSE ↓	JSD ↓
Input features												
1	✓				✓	✓	0.594	1.994	0.172	0.462	6.162	0.252
2	✓		✓	✓	✓	✓	0.591	1.817	0.179	0.515	**4.313**	0.230
3	✓	✓		✓	✓	✓	0.581	2.382	0.186	0.514	4.851	0.224
4	✓	✓	✓		✓	✓	0.588	2.078	0.188	0.510	4.611	0.222
5	✓	✓	✓	✓	✓	✓	**0.597**	**1.730**	**0.171**	**0.527**	4.330	**0.194**
Architecture												
6	✓	✓	✓	✓			0.567	2.134	0.201	0.510	6.780	0.209
7	✓	✓	✓	✓	✓		0.588	1.800	0.187	0.514	6.625	0.217
8	✓	✓	✓	✓		✓	0.580	2.092	0.199	0.521	5.100	0.220
9	✓	✓	✓	✓	✓	✓	**0.597**	**1.730**	**0.171**	**0.527**	**4.330**	**0.194**

based generative baselines DiffODGen and WEDAN improve upon NetGAN, and GeoFlow achieves the best overall quality–diversity trade-off, yielding the highest diversity (Div. 0.069) and consistently lower NRMSE and JSD across three datasets. In particular, GeoFlow attains the best CPC on Commuting and Freight (CPC 0.566 and 0.485), while on Air Travel it remains competitive in CPC (0.457 vs. 0.486) and substantially improves reconstruction and distribution matching (NRMSE 1.213 vs. 1.485, JSD 0.031 vs. 0.036). These results suggest that explicitly incorporating geospatial structure with OD-aware decoding improves both predictive accuracy and generative fidelity. Further analyses on region size effects and generalizability are provided in Appendix C.2.

4.3. Ablation Studies

We conduct ablation studies on the Commuting dataset under both prediction and generation settings, using CPC, NRMSE, and JSD for evaluation. Additional ablation results on the Freight dataset are provided in Appendix C.1.

Ablation on geospatial feature augmentation. Tab. 2 shows that augmenting straight-line distance with relative coordinates, k-hop distance, and free-flow distance consistently improves generation quality over the SL-only baseline, and the full feature set achieves the best overall performance (ID 5), increasing generation CPC from 0.462 to 0.527 and reducing generation NRMSE/JSD from 6.162/0.252 to 4.330/0.194. These features complement each other by encoding orientation-aware location cues and network-based accessibility, leading to the strongest overall feature combination.

Ablation on encoder and decoder architecture. Replacing the proposed encoder with a conventional graph convolution module or replacing the axial-global attention decoder with global-only attention degrades performance, as shown in Tab. 2. The full encoder-decoder architecture achieves the best results (ID 9). For prediction, it improves CPC/NRMSE/JSD from 0.567/2.134/0.201 to 0.597/1.730/0.171 over ID 6; for generation, it improves the corresponding metrics from 0.510/6.780/0.209 to 0.527/4.330/0.194.

Table 3. **Similarity analysis among geographic information.** F. norm represents the Frobenius norm, and MAD represents the maximal absolute difference. Each matrix is normalized to row-sum to one.

	SL vs. k-hop	SL vs. FF	k-hop vs. FF
F. norm	8.07	0.741	7.98
MAE	0.149	0.021	0.146
MAD	0.424	0.128	0.407

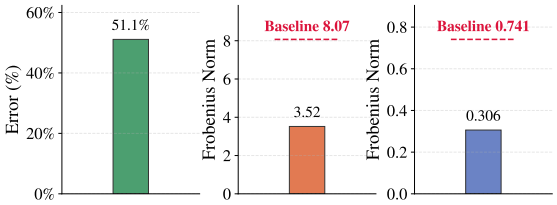

Figure 2. Prediction errors when inferring higher-order geospatial attributes from adjacency and straight-line distance matrices.

4.4. Analysis Experiments

Complementarity of straight-line and network-based distances. We compare straight-line (SL) distance with k-hop and free-flow (FF) distances using row-normalized matrices. Tab. 3 shows that SL and FF are relatively close in aggregate (Frobenius norm 0.741, MAE 0.021), yet still exhibit noticeable worst-case deviations (MAD 0.128). In contrast, k-hop differs substantially from both SL and FF (Frobenius norm $\approx$ 8.0, MAE $\approx$ 0.15). This indicates that SL alone cannot capture network-induced distortions, motivating the inclusion of network-based descriptors in modeling geospatial dependencies.

Difficulty of inferring higher-order geospatial attributes. We test whether higher-order cues can be recovered from adjacency and SL distance by training an MLP to predict relative angles, k-hop, and FF distances, which shows large reconstruction errors, including a 51.1% average relative error for angles and substantial Frobenius-norm discrepancies for k-hop and FF. This suggests that shortest-path and angular structure are hard to infer implicitly, supporting our choice to provide these features explicitly via preprocessing.

5. Conclusion

In this work, we introduce GeoFlow, a novel framework for modeling origin–destination (OD) flows. GeoFlow augments critical yet often overlooked geospatial relationships as representations and employs a geometric–intrinsic fusion encoder coupled with an axial–global attention decoder, achieving state-of-the-art performance on both prediction and generation tasks. Extensive experiments validate the effectiveness of the proposed approach. Ablation and analytical studies further highlight the significance of geospatial attributes and the encoder–decoder design, offering new insights for OD flow modeling and related applications.

Acknowledgements

This work was sponsored by 2025 Shanghai Key Technology Research and Development Program, Next- Generation Information Technology Project under Grant No. 25511103800.

Impact Statement

This work aims to advance machine learning methods for origin-destination flow modeling. Such models may support urban planning, mobility analysis, and infrastructure assessment, but their outputs should not be used as the sole basis for high-stakes planning or policy decisions without domain validation. Like other black-box neural network models, GeoFlow may produce plausible but unreliable outputs under distribution shifts or unexpected input conditions, and its predictions cannot be fully explained by explicit physical mobility mechanisms. The attention-based case studies in this paper provide only partial qualitative interpretability and should not be interpreted as complete causal or physical explanations of mobility behavior. In addition, GeoFlow assumes access to reliable geographic signals, such as network connectivity and geodesic distances. When these signals are unavailable, outdated, or biased, the predicted or generated flows may inherit corresponding errors. Care should therefore be taken when applying the model in data-sparse regions or in settings involving sensitive mobility patterns.

References

Abdi, H. and Williams, L. J. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Alon, U. and Yahav, E. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

Bamberger, J., Gutteridge, B., Roux, S. l., Bronstein, M. M., and Dong, X. On measuring long-range interactions in graph neural networks. *arXiv preprint arXiv:2506.05971*, 2025.

Batty, M. *Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals*. The MIT press, 2007.

Bojchevski, A., Shchur, O., Zügner, D., and Günnemann, S. Netgan: Generating graphs via random walks. In *International conference on machine learning*, pp. 610–619. PMLR, 2018.

Breiman, L. Random forests. *Machine learning*, 45(1): 5–32, 2001.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Dong, S., Wang, P., and Abbas, K. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., and Vapnik, V. Support vector regression machines. *Advances in neural information processing systems*, 9, 1996.

Ho, J., Kalchbrenner, N., Weissenborn, D., and Salimans, T. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Huang, Y., Liang, K., Dong, Z., Qu, X., Tianxiang, W., Han, Y., Xu, J., Zhou, B., and Wang, Y. Flow matching for denoised social recommendation. In *Forty-second International Conference on Machine Learning*, 2025.

Klemmer, K., Safir, N. S., and Neill, D. B. Positional encoder graph neural networks for geographic data. In *International conference on artificial intelligence and statistics*, pp. 1379–1389. PMLR, 2023.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R. T., Lopez-Paz, D., Ben-Hamu, H., and Gat, I. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Liu, Z., Miranda, F., Xiong, W., Yang, J., Wang, Q., and Silva, C. Learning geo-contextual embeddings for commuting flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 808–816, 2020.

Long, Q., Wang, H., Li, T., Huang, L., Wang, K., Wu, Q., Li, G., Liang, Y., Yu, L., and Li, Y. Practical synthetic human trajectories generation based on variational point processes. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4561–4571, 2023.

Luca, M., Barlacchi, G., Lepri, B., and Pappalardo, L. A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)*, 55(1):1–44, 2021.

Luo, Y., Wan, Z., Chen, Y., Mai, G., Chung, F.-l., and Larson, K. Transflower: An explainable transformer-based model with flow-to-flow attention for commuting flow prediction. *arXiv preprint arXiv:2402.15398*, 2024.

Pourebrahim, N., Sultana, S., Niakanlahiji, A., and Thill, J.-C. Trip distribution modeling with twitter data. *Computers, Environment and Urban Systems*, 77:101354, 2019.

Robinson, C. and Dilkina, B. A machine learning approach to modeling human migration. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pp. 1–8, 2018.

Rodríguez-Rueda, P., Ruiz-Aguilar, J., González-Enrique, J., and Turias, I. Origin–destination matrix estimation and prediction from socioeconomic variables using automatic feature selection procedure-based machine learning model. *Journal of urban planning and development*, 147 (4):04021056, 2021.

Rong, C., Ding, J., Liu, Z., and Li, Y. Complexity-aware large scale origin-destination network generation via diffusion model. *arXiv preprint arXiv:2306.04873*, 2023.

Rong, C., Ding, J., and Li, Y. An interdisciplinary survey on origin-destination flows modeling: Theory and techniques. *ACM Computing Surveys*, 57(1):1–49, 2024.

Rong, C., Ding, J., Liu, Y., and Li, Y. A large-scale dataset and benchmark for commuting origin-destination flow generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Simini, F., González, M. C., Maritan, A., and Barabási, A.-L. A universal model for mobility and migration patterns. *Nature*, 484(7392):96–100, 2012.

Simini, F., Barlacchi, G., Luca, M., and Pappalardo, L. A deep gravity model for mobility flows generation. *Nature communications*, 12(1):6576, 2021.

Tomazinis, A. R. A new method of trip distribution in an urban area. *Highway Research Board Bulletin*, 1962.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Whiteley, N., Gray, A., and Rubin-Delanchy, P. Matrix factorisation and the interpretation of geodesic distance. *Advances in Neural Information Processing Systems*, 34: 24–38, 2021.

Wu, Y., Xie, X., Yu, C., Xiao, Z., Zhang, J., Xu, Z., and Mai, Z. A survey on origin-destination flow prediction. In *2024 11th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pp. 48–52. IEEE, 2024.

Zhang, J., Che, H., Chen, F., Ma, W., and He, Z. Short-term origin-destination demand prediction in urban rail transit systems: A channel-wise attentive split-convolutional neural network method. *Transportation Research Part C: Emerging Technologies*, 124:102928, 2021.

Zipf, G. K. The P_1 P_2 / D hypothesis: On the intercity movement of persons. *American sociological review*, 11 (6):677–686, 1946.

Appendix

A. Method Details

A.1. Flow Matching Models for Origin-Destination Flow Generation

To generate high-fidelity origin-destination (OD) flow matrices, we adopt Flow Matching (FM), an efficient and expressive framework for generative modeling (Lipman et al., 2022; 2024). In contrast to diffusion models, which typically require simulating stochastic or ordinary differential equations (ODEs) during training, FM directly learns the vector field that transports a simple prior distribution to the complex data distribution. This simulation-free paradigm substantially improves training efficiency. In the following, we use $\boldsymbol{x}$ to denote the vectorized form of the OD flow matrix $\boldsymbol{F}$, following the conventional notation in the flow matching literature.

A.1.1. Formulation

The core principle of FM is to define a continuous-time flow from a simple prior distribution p_0, typically a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, to the target data distribution p_1, which in our case corresponds to OD matrices. This flow is governed by an ODE:

$$\frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{v}_t(\boldsymbol{x}_t), \tag{26}$$

where $t \in [0, 1]$ is a pseudo-time variable, $\boldsymbol{x}_t$ denotes the state at time t, and $\boldsymbol{v}_t$ is a time-dependent vector field. The goal is to train a neural network $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ to approximate this vector field.

To derive a tractable training objective, we first define a conditional probability path $p_t(\boldsymbol{x}|\boldsymbol{x}_1)$ that connects a prior sample $\boldsymbol{x}_0 \sim p_0$ to a data sample $vx_1 \sim p_1$. Following Lipman et al. (2022), we adopt a simple linear interpolation path:

$$\boldsymbol{x}_t = (1 - t)\boldsymbol{x}_0 + t\boldsymbol{x}_1, \tag{27}$$

with the corresponding conditional vector field:

$$\boldsymbol{u}_t(\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_1) = \boldsymbol{x}_1 - \boldsymbol{x}_0. \tag{28}$$

For this linear path, the target vector field is constant and independent of both $\boldsymbol{x}_t$ and t.

The FM objective is to minimize the expected mean squared error between the predicted vector field $\boldsymbol{v}_{\boldsymbol{\theta}}$ and the target vector field $\boldsymbol{u}_t$:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t\sim\mathcal{U}(0,1),\boldsymbol{x}_0\sim p_0,\boldsymbol{x}_1\sim p_1}\left[\|\boldsymbol{v}_{\boldsymbol{\theta}}((1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1, t) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\|_2^2\right]. \tag{29}$$

A.1.2. Training

In practice, optimization proceeds by sampling mini-batches. For each training step, a data point $\boldsymbol{x}_1$ (an OD matrix), a noise sample $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and a time step $t \sim \mathcal{U}(0, 1)$ are sampled. The interpolated state $\boldsymbol{x}_t$ and target velocity $\boldsymbol{v}_{\mathrm{tgt}} = \boldsymbol{x}_1 - \boldsymbol{x}_0$ are then computed. The model $\boldsymbol{v}_{\boldsymbol{\theta}}$ takes $\boldsymbol{x}_t$, t, and associated context as input to predict a velocity $\boldsymbol{v}_{\mathrm{pred}}$, and the mean squared error between $\boldsymbol{v}_{\mathrm{pred}}$ and $\boldsymbol{v}_{\mathrm{tgt}}$ is minimized via backpropagation.

A.1.3. Sampling

Once trained, $\boldsymbol{v}_{\boldsymbol{\theta}}$ enables the generation of new OD matrices by solving the initial value problem defined by the learned ODE. Starting from a prior sample $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we integrate the ODE from $t = 0$ to $t = 1$ using a numerical solver. In our implementation, the sampling process employs the forward Euler scheme:

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t + \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\Delta t, \tag{30}$$

where Δt is the integration step size. After $n_{\mathrm{steps}} = 1/\Delta t$ iterations, the resulting state $\boldsymbol{x}_1$ constitutes a novel sample drawn from the learned distribution. This procedure is executed independently for each graph in a batch, enabling efficient generation of multiple OD matrices.

A.2. Preprocessing

In this work, GeoFlow is trained and evaluated on the three different OD datasets, in which each region is represented as a graph in matrix form. The nodes correspond to geographic areas (e.g., census tracts), and the edges represent road network adjacency and Euclidean distance. For each region, the dataset provides node attributes, spatial coordinates, a road network adjacency matrix, a Euclidean distance matrix, and the ground-truth commuting OD flow matrix. A standardized preprocessing pipeline is applied to harmonize these inputs and derive multi-view relational structures.

The preprocessing for each urban region consists of the following steps:

1. **Area Feature Scaling.** For each region, we construct a raw feature vector by concatenating demographic attributes with counts of points of interest (POIs). To ensure comparability across features, we apply column-wise max normalization, dividing each feature value by the maximum observed within the same region. This rescales all features to the range $[0, 1]$.

2. **Coordinate Normalization.** The raw geographic coordinates (zone centroids) are standardized to obtain a canonical representation invariant to translation and rotation. Specifically, we first subtract the mean to center the coordinates. We then apply Principal Component Analysis (PCA) (Abdi & Williams, 2010) to align the coordinate axes with directions of maximal variance. Finally, the transformed coordinates are rescaled to the range $[-1, 1]$ by dividing by the maximum absolute value along each axis.

3. **OD Flow Normalization.** The raw OD matrix, containing integer-valued flows, is normalized to form a probability distribution by dividing by the total commuting volume of the region. This total volume is stored and later used during inference to rescale generated probability matrices back to absolute flows.

4. **Multi-view Affinity Construction.** To capture diverse spatial relationships between zones, we construct three affinity matrices from different distance measures. Given a distance matrix $\boldsymbol{D}$, we compute an affinity matrix $\boldsymbol{A}$ using an exponential kernel:

$$A_{i,j} = \exp(-D_{i,j}/\tau), \tag{31}$$

   where τ is set to the median of all positive, finite distances in $\mathbf{D}$. This produces a dense, weighted graph where affinity decreases with distance. The three distance notions are:

   - **Euclidean Affinity:** Based on straight-line distance between zone centroids, capturing pure spatial proximity.
   - **Topological Affinity:** Based on unweighted shortest-path distance (number of hops) on the road network, reflecting connectivity in the transport infrastructure.
   - **Network-Geodesic Affinity:** Based on shortest-path distance on the road network weighted by edge lengths, representing the most efficient travel distance along the network.

   For each affinity matrix, we also compute its symmetrically normalized form,

$$\boldsymbol{A}_{\text{norm}} = \boldsymbol{D}_{\text{deg}}^{-1/2} \boldsymbol{A} \boldsymbol{D}_{\text{deg}}^{-1/2}, \tag{32}$$

   where $\boldsymbol{D}_{\text{deg}}$ is the weighted generalized degree matrix with diagonal entries $(\boldsymbol{D}_{\text{deg}})_{i,i} = \sum_j A_{i,j}$, i.e., the total connection strength of node i. Symmetric normalization mitigates scale differences introduced by heterogeneous node degrees and yields affinity matrices that are numerically stable and comparable across regions. We include all normalized affinity matrices in the geometric relation descriptor $\boldsymbol{G}$, which supplies multi-relational contexts to downstream modules.

B. Experiment Details

B.1. Dataset Construction

The datasets cover distinct OD regimes, including commuting, freight logistics, and air travel. Tab. 4 summarizes the scale and sparsity of the three datasets.

For the Commuting dataset, we use the CommutingODGen benchmark (Rong et al., 2025), where each sample corresponds to a region-level OD matrix with provided area attributes, geographic relationships, and observed commuting flows.

Table 4. **Dataset statistics.** The number of areas per region is reported as min/mean/max, and OD matrix sparsity is the fraction of zero entries averaged over regions.

	Commuting	Freight	Air Travel
# Region samples	2252	5000	1980
# Areas per region	4 / 30.1 / 1318	10 / 56.9 / 132	43 / 43.9 / 62
OD matrix sparsity	9.19%	0.02%	50.66%

The Freight dataset is built from FAF5 2022 freight flows. We aggregate OD freight tonnage between FAF zones and derive distances from FAF distance-band midpoints. We then generate 5,000 freight subregions, with region sizes sampled from a $\text{Beta}(2, 3.21)$ distribution scaled to $[10, 132]$, resulting in an average of approximately 56.9 areas per region. Each subregion starts from a FAF area sampled according to its total inbound and outbound freight volume, and is expanded through freight-flow neighbors weighted by tonnage. This procedure produces localized freight-flow subnetworks used as region-level OD samples.

The Air Travel dataset is constructed from the raw visitor-days OD table by grouping records according to temporal resolution, population segment, and date. For each group, we create one full origin–destination bipartite network and 21 additional random subnetworks by sampling 70% of origin markets and 70% of destinations, yielding 1,980 region-level samples in total. Since the data are intrinsically bipartite, only origin-to-destination entries are valid observations. Therefore, the observation mask is set to valid for the origin-destination entries and invalid for origin-origin, destination-destination, and destination-to-origin entries. OD flows are computed only on valid entries accordingly.

B.2. Metric Details

We provide the formal definitions and additional details of the evaluation metrics as follows.

Numerical Accuracy. To evaluate OD flow prediction and reconstruction, we employ three standard metrics:

- **Common Part of Commuting (CPC).** CPC measures the similarity between two OD matrices by computing the fraction of overlapping flow volume:

$$\text{CPC}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{2\sum_{i,j} \min(X_{i,j}, Y_{i,j})}{\sum_{i,j} X_{i,j} + \sum_{i,j} Y_{i,j}}, \tag{33}$$

  where $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the predicted and ground-truth OD matrices, respectively. Higher values indicate better alignment.

- **Root Mean Square Error (RMSE).** RMSE quantifies the average magnitude of error in flow counts:

$$\text{RMSE}(\boldsymbol{X}, \boldsymbol{Y}) = \sqrt{\frac{1}{n^2} \sum_{i,j} (X_{i,j} - Y_{i,j})^2}. \tag{34}$$

- **Normalized Root Mean Square Error (NRMSE).** NRMSE normalizes RMSE by the standard deviation of the ground-truth matrix, enabling comparison across different scales:

$$\text{NRMSE}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\text{RMSE}(\boldsymbol{X}, \boldsymbol{Y})}{\sigma_{\boldsymbol{Y}}}, \quad \sigma_{\boldsymbol{Y}} = \sqrt{\frac{1}{n^2} \sum_{i,j} (Y_{i,j} - \mu_{\boldsymbol{Y}})^2}, \quad \mu_{\boldsymbol{Y}} = \frac{1}{n^2} \sum_{i,j} Y_{i,j}. \tag{35}$$

- **Mean Absolute Error (MAE).** MAE provides a scale-independent measure of average deviation:

$$\text{MAE}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{n^2} \sum_{i,j} |X_{i,j} - Y_{i,j}|. \tag{36}$$

Distributional Similarity. To assess whether generated flows capture realistic distributional patterns, we compute the Jensen–Shannon divergence (JSD) between the distributions of inflows, outflows, and OD entries:

$$\text{JSD}(P \| Q) = \frac{1}{2}\text{KL}(P \| M) + \frac{1}{2}\text{KL}(Q \| M), \tag{37}$$

where $M = \frac{1}{2}(P + Q)$ and $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence. JSD is symmetric and bounded between 0 and 1, with lower values indicating closer distributions.

Diversity. For the generation task, we evaluate sample diversity using the metric defined as

$$\mathrm{Diversity}(\{\tilde{\boldsymbol{F}}^k\}_{k=1}^K) = \frac{2}{K(K-1)} \sum_{k<\ell} d(\tilde{\boldsymbol{F}}^k, \tilde{\boldsymbol{F}}^\ell), \tag{38}$$

where $\{\tilde{\boldsymbol{F}}^k\}_{k=1}^K$ are independently generated OD matrices under the same conditioning, and $d(\cdot,\cdot)$ is a distance measure (e.g., in this work, RMSE). This metric computes the average pairwise dissimilarity among generated samples, with higher values indicating greater diversity. Intuitively, a model affected by mode collapse produces more similar or even identical samples with uniformly sampled random seeds, leading to high similarity (low dissimilarity and diversity), whereas a well-generalized model generates sufficiently distinct OD matrices, resulting in lower similarity and higher diversity.

B.3. Training Details

GeoFlow is trained end-to-end using the AdamW optimizer with an initial learning rate of 1×10^{-3} and a batch size of 64. The training objective is the Mean Squared Error (MSE) between the predicted and ground-truth normalized OD matrices. A ReduceLROnPlateau learning rate scheduler is employed, which decreases the learning rate by a factor of 0.5 if the validation metric does not improve for 10 consecutive epochs. To mitigate overfitting, early stopping is applied with a patience of 100 epochs. Model performance is evaluated on the validation set after each epoch, and the checkpoint achieving the best validation performance is retained. All experiments are implemented in PyTorch and conducted on a single NVIDIA GeForce RTX 4090 GPU.

B.4. Hyperparameters

The model architecture and training configuration are controlled by a set of specified hyperparameters. The hidden dimension for geometric and intrinsic embeddings is set to 32, which we found to provide a favorable balance between computational efficiency and predictive accuracy. The encoder consists of two multilayer perceptron (MLP) layers for encoding geometric attributes and fusing geometric and intrinsic attributes, followed by two graph attention layers. The decoder is composed of two layers of axial self-attention and two layers of global self-attention. All attention modules employ four heads, and dropout is disabled throughout training.

To ensure robustness, we conducted sensitivity analyses over different hidden dimensions, numbers of encoder/decoder layers, and attention heads. The reported configuration consistently achieved the best trade-off between performance and efficiency across validation runs, and is therefore adopted in the main experiments.

C. Additional Experiments

C.1. Additional Ablation Studies

Tab. 5 reports ablation results on the Freight dataset. For geospatial feature augmentation, the full feature set achieves the strongest overall balance: it obtains the best prediction CPC and JSD, the second-best prediction NRMSE, and the best generation JSD. Although the best generation CPC and NRMSE are achieved by partially ablated feature variants, the full feature set remains competitive on both metrics, suggesting that different geographic cues contribute complementary benefits under different evaluation criteria. For architecture ablation, the full encoder-decoder model achieves the best prediction performance across all metrics and the best generation JSD. The decoder-only variant obtains the best generation CPC and NRMSE, indicating that the axial-global attention decoder is particularly important for reconstruction quality, while combining the proposed encoder and decoder gives the strongest distributional fidelity. Overall, the Freight results support the usefulness of both geospatial feature augmentation and the proposed architecture, with some metric-specific trade-offs in the generation setting on the Freight dataset.

C.2. Generalizability

Beyond the main-text evaluations, we further probe the generalizability of GeoFlow along three complementary axes on the Commuting dataset (Rong et al., 2025). First, in the prediction setting, we vary the spatial scale of training and evaluation regions to quantify how well the model transfers across omitted area scales. Second, in the generation setting, we construct

Table 5. **Ablation studies on the Freight dataset.** Bold and underlined values denote the best and second-best results within each ablation block, respectively.

ID	Input features				Architecture		Prediction			Generation		
	SL	Rel.	k-hop	FF	Enc.	Dec.	CPC ↑	NRMSE ↓	JSD ↓	CPC ↑	NRMSE ↓	JSD ↓
Input features												
1	✓				✓	✓	0.578	0.865	0.091	0.456	1.070	0.196
2	✓		✓	✓	✓	✓	0.621	0.672	0.068	0.432	1.038	0.166
3	✓	✓		✓	✓	✓	0.617	0.702	<u>0.063</u>	**0.523**	0.887	<u>0.142</u>
4	✓	✓	✓		✓	✓	<u>0.633</u>	**0.608**	0.064	0.412	**0.803**	0.148
5	✓	✓	✓	✓	✓	✓	**0.640**	<u>0.612</u>	**0.048**	<u>0.487</u>	<u>0.878</u>	**0.111**
Architecture												
6	✓	✓	✓	✓			0.616	0.686	0.084	0.401	0.900	0.203
7	✓	✓	✓	✓	✓		<u>0.635</u>	<u>0.618</u>	<u>0.053</u>	<u>0.488</u>	<u>0.817</u>	<u>0.126</u>
8	✓	✓	✓	✓		✓	0.634	0.630	<u>0.053</u>	**0.525**	**0.774**	0.134
9	✓	✓	✓	✓	✓	✓	**0.640**	**0.612**	**0.048**	0.487	0.878	**0.111**

Table 6. Partition of regions by size.

Region Size	# Areas	# Regions
Small	$[0, 10)$	1080
Medium	$[10, 100)$	1023
Large	$[100, \infty)$	149
All	$[0, \infty)$	2252

longitudinal partitions of the study area and test whether a model trained on one part of the space can faithfully reproduce patterns in another. Finally, we perform a fine-grained performance analysis by conditioning on OD-flow magnitude and OD distance, which reveals how error is distributed across practically relevant regimes.

C.2.1. Generalizability Across Area Scales

To better understand how predictive performance and transferability vary with the spatial scale of regions, we adopt a scale-aware evaluation protocol that explicitly varies both the region sizes present during training and the region sizes on which models are evaluated.

Setup. For the experiments, regions are partitioned by the number of constituent areas into three size classes: small, medium, and large, as summarized in Tab. 6. We train distinct models under a set of controlled training compositions formed by varying the inclusion and exclusion of these size classes (for example, ID 4 is trained on the union of small- and medium-sized regions). Each trained model is evaluated separately on small, medium, large, and on the union of all regions (All). This design isolates how particular training mixtures influence performance both within and across scales and enables a principled comparison of generalizability under differing class imbalances and distributional shifts.

Analysis. When the evaluation scale is absent from the training pool (IDs 6, 11, and 16), performance on the held-out class deteriorates consistently. RMSE and MAE increase, and CPC declines, which confirms that excluding a scale impairs direct transfer. Crucially, however, the magnitude of this degradation is modest relative to a model trained on the full corpus (IDs 5, 9, and 13). In other words, although removing the target scale produces a measurable loss, the model nevertheless retains a substantial portion of its predictive ability across scales, indicating that it learns components of OD structure that generalize beyond the observed size class.

Table 7. **Results across training–evaluation scale pairs on the Commuting dataset.** IDs indicate the different configurations of training and evaluation set composition.

ID	Training Region Size			Evaluation Region Size	Metrics					
	Small	Medium	Large		CPC ↑	RMSE ↓	MAE ↓	JSD_{in} ↓	JSD_{out} ↓	JSD_{all} ↓
1	✓	✓	✓	All	0.630	72.95	47.09	**0.231**	0.126	0.158
2		✓	✓		0.587	83.29	51.89	0.279	0.210	0.166
3	✓		✓		0.575	93.46	57.99	0.270	0.160	0.226
4	✓	✓			**0.647**	**64.91**	**41.63**	0.241	**0.096**	**0.141**
5	✓	✓	✓	Small	0.769	72.62	49.68	0.244	0.099	0.136
6		✓	✓		0.638	108.4	67.31	0.349	0.292	0.182
7	✓		✓		0.729	85.68	60.44	0.286	0.104	0.173
8	✓	✓			**0.785**	**66.26**	**46.21**	**0.238**	**0.087**	**0.134**
9	✓	✓	✓	Medium	0.559	62.39	40.12	0.202	0.119	0.154
10		✓	✓		**0.597**	**55.30**	**35.36**	**0.201**	0.095	0.130
11	✓		✓		0.484	96.23	58.55	0.240	0.196	0.261
12	✓	✓			0.579	58.00	36.15	0.205	**0.086**	**0.124**
13	✓	✓	✓	Large	0.139	102.62	46.06	0.504	**0.234**	0.305
14		✓	✓		**0.150**	**93.32**	53.70	**0.310**	0.401	**0.301**
15	✓		✓		0.081	130.90	**36.33**	0.360	0.315	0.372
16	✓	✓			0.115	147.9	76.15	0.334	0.362	0.343

Table 8. **Generalizability across longitudinal partitions.** "All – West" (resp. "All – East") denotes that the model is trained on all regions except those in the held-out western (resp. eastern) partition. "All – Random" uses a randomly sampled held-out set with the same cardinality.

Train on	Evaluate on	CPC ↑	RMSE ↓	JSD ↓
All - West	West	0.572	111.1	0.176
All - East	East	0.561	116.7	0.177
All - Random	Random	0.566	101.2	0.174

Table 9. **Cross-morphology transfer on the Commuting dataset.** Each row denotes the morphology group used for training, and each column denotes the group used for evaluation. The four groups correspond to Rural-Compact, Elongated-Sparse, Suburban, and Metropolitan regions. Results are reported in CPC.

Train \ Test	Rural	Elong.	Suburb.	Metro.	Overall
Rural	0.722	0.644	0.380	0.033	0.507
Elong.	0.681	0.671	0.441	0.077	0.527
Suburb.	0.686	0.636	**0.580**	0.134	0.594
Metro.	0.283	0.230	0.393	**0.135**	0.319
Full-data	**0.730**	**0.699**	0.541	0.107	**0.597**

C.2.2. Generalizability Across Longitudinal Partitions

To complement the above scale-based study, we evaluate generalizability under a different kind of spatial shift in the generation setting. We partition the study areas into eastern and western subsets based on longitude and additionally construct a random partition with the same cardinality as a baseline.

Setup. For the "All - West" configuration, the generative model is trained on all regions except those in the western subset and then evaluated exclusively on the held-out West partition. "All - East" and "All - Random" are defined analogously. We report CPC, RMSE, and JSD between the generated and ground-truth OD distributions on the corresponding held-out partitions in Tab. 8.

Analysis. Across all three configurations, performance remains stable. CPC stays at a similar level across splits, RMSE differences are moderate, and JSD exhibits minor fluctuations. The random partition does not confer a clear advantage over structured East/West splits, suggesting that the model does not overfit to idiosyncratic geographic patterns of any particular longitudinal band. Instead, it captures generative regularities that transfer across distinct spatial subdomains.

C.3. Generalization Across City Morphologies

Setup. To examine whether GeoFlow generalizes across heterogeneous urban layouts, we conduct a morphology-based analysis on the Commuting dataset. We first characterize each region using a set of morphology-related descriptors that summarize its spatial scale, geometric configuration, network connectivity, distance distribution, and OD-flow structure. These descriptors capture broad differences in regional size, spatial anisotropy, compactness, graph density, and flow sparsity or concentration. After standardization, we apply K-means clustering to partition the regions into four morphology groups. Based on the aggregate profiles of the resulting clusters, we assign them interpretable names: Rural-Compact, Elongated-Sparse, Suburban, and Metropolitan. The morphology labels are used only to construct evaluation partitions and are not provided as additional supervision to the model.

Analysis. Tab. 9 reports cross-morphology transfer results, where each model is trained on a single morphology group and evaluated across all groups. Within the single-morphology models, the morphology-matched setting usually achieves the best or near-best performance on its corresponding target group, suggesting that shared spatial layout, graph structure, and OD sparsity patterns facilitate transfer. However, models trained on a single morphology often generalize poorly to structurally different groups, especially to Metropolitan regions. In contrast, training on the full dataset yields the best overall performance, indicating that exposure to diverse urban forms improves aggregate robustness. These results suggest that, by using region-centered, PCA-aligned, normalized relative coordinates together with network-based geographic descriptors, GeoFlow relies on transferable relative spatial relations rather than a fixed city-specific coordinate system.

C.3.1. Performance Analysis

Scale-dependent predictive difficulty. This partitioning makes explicit two salient characteristics of the dataset: (1) small and medium regions dominate numerically, whereas large regions are comparatively scarce; and (2) the intrinsic complexity of OD structure grows with region size. Consistent with these observations, as shown in Tab. 7, predictive difficulty exhibits a stable ordering: models attain the best results on small regions, intermediate results on medium regions, and the worst results on large regions. Large-region evaluations display substantially higher RMSE and lower CPC, indicating both greater pointwise error and poorer fidelity of the learned OD distributions. This pattern is interpretable as the consequence of increased intra-region heterogeneity and multi-modal travel structure in larger spatial extents.

Table 10. **Performance breakdown by OD flow magnitude.** Ratio denotes the proportion of OD pairs falling into each flow bin, and Inv. Areas is the fraction of areas that participate in at least one interaction within the bin. NRMSE is the RMSE normalized within absolute flow magnitude.

Flow Scale	Proportion	Inv. Areas	CPC ↑	RMSE ↓	NRMSE ↓
$[0, 5)$	83.18%	85.84%	0.105	146.1	219.1
$[5, 20)$	11.62%	96.46%	0.269	185.7	19.56
$[20, 100)$	4.49%	100.0%	0.493	256.1	6.404
$[100, 500)$	0.69%	99.71%	0.620	373.9	2.063
$[500, 1000)$	0.02%	48.08%	0.640	665.9	1.036
$[1000, \infty)$	0.01%	8.55%	0.513	1330	1.010

Table 11. **Performance breakdown by normalized OD distance.** Ratio and Inv. Areas are defined as in Tab. 10.

Distance Scale	Proportion	Inv. Areas	CPC ↑	RMSE ↓	NRMSE ↓
$[0, 0.1)$	4.67%	100.0%	0.556	447.9	22.56
$[0.1, 0.2)$	8.66%	49.85%	0.418	389.2	46.27
$[0.2, 0.5)$	31.13%	73.16%	0.471	244.5	47.34
$[0.5, 1)$	40.33%	95.28%	0.483	163.4	55.51
$[1, 2)$	15.16%	100.0%	0.408	122.6	46.37
$[2, \infty)$	0.05%	39.82%	0.251	64.62	8.626

Performance profile across flow scales. Tab. 10 conditions performance on OD-flow magnitude. The distribution is highly skewed. More than 83% of OD pairs have flows below 5, whereas flows above 100 account for less than 1% of the pairs. As flow increases, absolute RMSE naturally grows, but normalized error (NRMSE) drops and CPC improves from 0.105 in the smallest bin to around 0.62–0.64 for flows in $[100, 1000)$. This indicates that, in relative terms, the model is substantially more reliable on high-volume OD pairs that dominate aggregate mobility patterns, while low-flow interactions, which are both noisier and less constrained by data, remain the most challenging. Moreover, when baseline flows are very small, CPC becomes highly sensitive to minor absolute fluctuations: a handful of misallocated trips can induce large apparent drops in correlation, making the low-flow bins look disproportionately poorly modeled. This is consistent with the interpretation that OD pairs with very low counts contain a substantial random component, and such randomness, combined with a small base volume, translates into amplified variability of distributional metrics. The slight CPC decrease in the ≥ 1000 bin is plausibly attributable to extreme data sparsity (only 0.01% of pairs, involving 8.55% of areas).

Performance profile across distance scales. We further examine performance as a function of normalized OD distance in Tab. 11. Most OD pairs correspond to short- to medium-range trips: distances in $[0.2, 1)$ already account for over 70% of the mass. CPC is highest for the shortest-range bin $[0, 0.1)$ and remains relatively stable for distances up to 1, indicating that the model captures local and intra-region mobility patterns well. For very long-range interactions (≥ 2), CPC drops markedly to ≈ 0.25 despite a lower NRMSE, reflecting that these flows are extremely sparse (only 0.05% of OD pairs, involving 39.82% of areas) and concentrated along a small number of long-distance corridors. Overall, the distance-conditioned analysis shows that the model is most accurate in the distance regimes that dominate real-world demand, while rare long-distance trips and very short but noisy interactions remain more difficult to predict perfectly.

C.4. Model Capacity Scaling

To assess the scalability of our architecture, we systematically increase model capacity along two axes: width (hidden dimension) and depth (number of encoder/decoder attention layers and attention heads) with all other optimization hyperparameters and training schedules kept fixed.

Tab. 12 reports the results. Increasing the hidden dimension from 16 to 64 (IDs 1–3) yields consistent gains. CPC improves from 0.576 to 0.593, RMSE drops from 107.5 to 92.43, and JSD decreases from 0.212 to 0.194. This indicates that the model can effectively leverage additional representational capacity, and that the architecture behaves in a stable, roughly monotone way under width scaling. Adding moderate depth and more heads on top of a 32-dimensional backbone (ID 4) further improves all metrics, reaching higher CPC and lower RMSE, and lower JSD. The incremental improvements

Table 12. **Effect of model capacity on performance.** We vary the hidden dimension and the number of encoder/decoder attention layers and attention heads, while keeping all other hyperparameters fixed. The number of attention layers is applied to both axial and global attention blocks.

Hidden Dimension	# Encoder Attn Layers	# Decoder Attn Layers	# Attn Heads	CPC ↑	RMSE ↓	JSD ↓
16	1	1	2	0.576	107.5	0.212
32	1	1	2	0.582	98.59	0.200
64	1	1	2	0.593	92.43	0.194
32	2	2	4	0.601	86.98	0.189

Table 13. **Scalability profiling of GeoFlow across different structural scales.** We report average per-step training and inference latency for prediction and generation.

# Areas	Prediction		Generation	
	Train	Infer	Train	Infer
$[0, 10)$	18.6	4.08	17.5	105.1
$[10, 20)$	18.1	4.13	17.7	107.2
$[20, 50)$	18.2	4.26	17.9	110.3
$[50, 100)$	18.4	4.32	18.1	110.8
$[100, 200)$	19.1	4.45	17.6	109.2
$[200, 500)$	25.1	7.26	23.7	169.9
$[500, \infty)$	157.2	37.0	158.8	918.5

suggest diminishing returns at higher capacities, consistent with a regime where performance is increasingly data-limited rather than model-limited. Overall, these results show that our model scales gracefully. Larger configurations provide better accuracy, but even the smallest variant (ID 1) attains a substantial fraction of the performance of the largest one. This makes it possible to trade off compute and memory for accuracy depending on deployment constraints, while retaining the same architectural design and training procedure.

C.5. Scalability Profiling

We further profile the computational scalability of GeoFlow with respect to the structural scale of a region. Here, structural scale refers to the number of areas in a region, and therefore to the number of OD pairs to be modeled. This is distinct from the numerical scale of OD flows, which is mitigated by the OD-flow normalization step in preprocessing. For each scale group, we report the average per-step wall-clock latency for both prediction and generation, including training and inference.

As shown in Tab. 13, the latency remains nearly flat for most regions below 200 areas. Prediction training stays around 18–19 ms per step, while prediction inference remains around 4 ms. Generation training has a similar cost to prediction training, since flow matching uses a single sampled time step during each training update. In contrast, generation inference is substantially slower because sampling requires multiple numerical integration steps rather than a single forward pass. The latency increases moderately for regions with 200–500 areas and rises sharply for very large regions with more than 500 areas. This pattern indicates that GeoFlow is computationally efficient for the majority of regions in the benchmark, while very large regional graphs remain the main computational bottleneck due to the rapid growth of OD-pair interactions with the number of areas.

C.6. Comparison of Diffusion Models and Flow Matching Models

To further assess the effectiveness of our framework, we conduct a comparative study between flow-matching and diffusion models. In this setting, only the flow-matching components are replaced with their diffusion counterparts, while all other configurations remain unchanged. Since diffusion models follow a different generative paradigm and training procedure, we perform extensive hyperparameter tuning to ensure comparable model size and computational cost. As shown in Fig. 3, although the diffusion model initially exhibits normal metric progression, it undergoes severe oscillations and sustained degradation. This instability arises from the inherently slower and noisier training dynamics of diffusion, which hinder

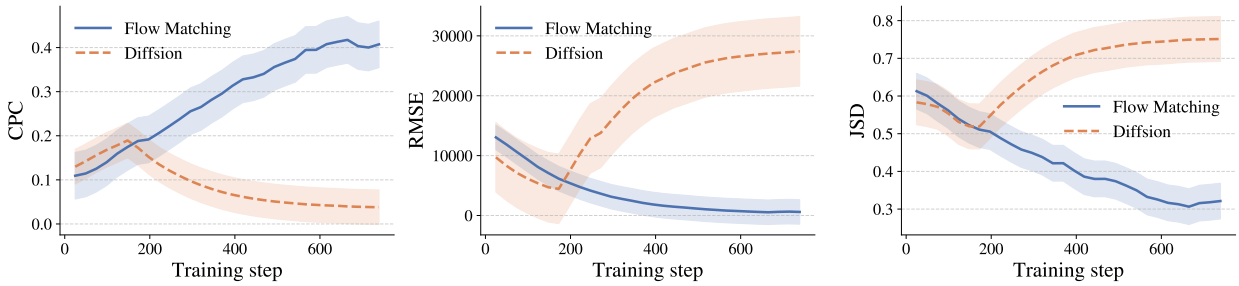

Figure 3. **Training dynamics comparison between flow-matching and diffusion models.** Flow-matching demonstrates steady improvements across metrics, while diffusion exhibits instability and performance degradation.

Table 14. **Computational cost of GeoFlow architecture compared to the TransFlower.** We report average training and inference wall-clock time, average and peak floating-point operations (FLOPs), and peak GPU memory usage.

	Training Time (s)	Inference Time (s)	Avg. FLOPs	Peak FLOPs	Peak Mem. (MiB)
TransFlower	0.0580	0.4468	1.49×10^8	8.82×10^{10}	2.31×10^4
GeoFlow	0.0160	0.1104	6.12×10^8	3.58×10^{11}	2.23×10^4

efficient convergence. In contrast, the flow-matching model exhibits consistent and monotonic improvements throughout training, indicating its stability and efficiency in capturing the underlying OD flow structure.

D. Computational Complexity and Resource Usage

We profile the computational footprint of our proposed GeoFlow architecture against the transformer-based baseline TransFlower. Table 14 reports average training and inference wall-clock time, average and peak FLOPs, and peak GPU memory usage, computed using the same configuration as that used to obtain its main performance results reported in the main experiments. All measurements are taken from steady-state runs after warm-up, so the numbers reflect typical rather than best-case performance.

GeoFlow achieves noticeably shorter wall-clock times than TransFlower: training is roughly $3.6\times$ faster (0.0160 s vs. 0.0580 s) and inference is about $4\times$ faster (0.1104 s vs. 0.4468 s). At the same time, GeoFlow requires around $4\times$ higher average FLOPs (6.12×10^8 vs. 1.49×10^8) and a corresponding increase in peak FLOPs (3.58×10^{11} vs. 8.82×10^{10}), while peak memory consumption remains comparable between the two models. This contrast highlights that GeoFlow adopts a more compute-intensive yet highly parallelizable design that leads to better hardware utilization and throughput, whereas TransFlower offers lower FLOP counts at the cost of higher latency. Taken together with the accuracy improvements of GeoFlow, these results indicate that GeoFlow provides a favorable accuracy–efficiency trade-off, while remaining tractable at city scale on a single general-purpose GPU.

E. Recommendations for training-set composition

The generalizability experiments in Appendix C.2.1 suggest several practical guidelines for choosing which size classes to include when constructing the training set. The selection of region sizes has systematic and interpretable consequences for both within-scale accuracy and cross-scale transfer.

Models trained on the combination of small and medium areas tend to produce the strongest results for small-scale and aggregate evaluations (IDs 4, 8, 12 and 16 in Tab. 7) and, in aggregate metrics, can even outperform the model trained on the full corpus (ID 4 vs. ID 1). This apparent advantage is primarily attributable to the numerical predominance of small and medium areas: their abundance supplies stronger and more stable gradient signals during optimization, which reduces average error on the dominant scales and therefore improves aggregate statistics. However, the same training regimen is weaker on large-area evaluation than the model trained on all scales (ID 16 vs. ID 13). Including the relatively scarce but distributionally distinct large-area examples in the training set increases the model's ability to capture large-area specific structure and yields better performance on large areas and, importantly, stronger overall generalizability across all scales.

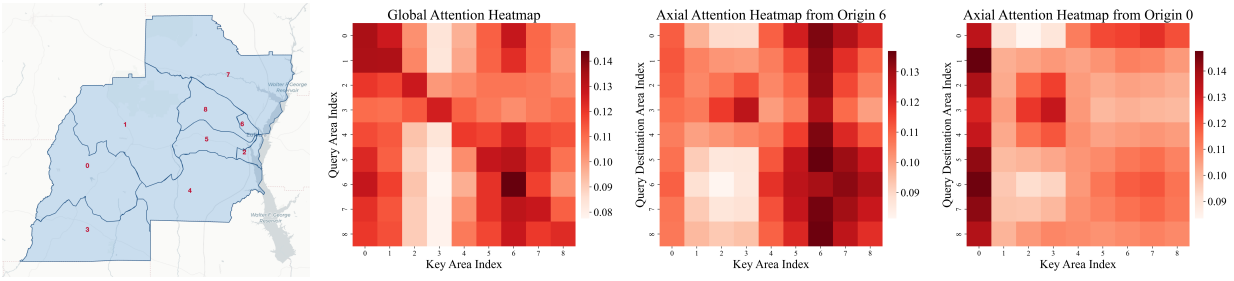

Figure 4. **Qualitative case study of learned attention in Barbour County, Alabama, United States.** Panels from left to right show the geographic layout of areas, the global attention heatmap, axial attention over destinations for origin area 6, and axial attention over destinations for origin area 0.

By contrast, training on medium and large areas is most effective when the evaluation target is medium or large (IDs 10 and 14), which suggests that medium areas provide transitional patterns that facilitate transfer toward large regimes. Naively pairing small and large areas without an intermediate representation often degrades performance (IDs 7, 11 and 15), likely because the omission of medium examples amplifies the domain shift between the two extremes.

In summary, the results point to a practical trade-off. Emphasizing small and medium areas during training tends to improve aggregate metrics and accuracy on the majority of areas in our corpus, whereas incorporating the relatively scarce large-area examples improves fidelity on large areas and enhances cross-scale robustness. This pattern appears consistently in our dataset but should be interpreted with caution rather than as a universal prescription. The balance between aggregate performance and coverage of data-scarce large areas is inherently dataset dependent and therefore calls for pragmatic, case-by-case choices when assembling training data in practical applications.

F. Qualitative Evaluation

F.1. Case Study

To validate that GeoFlow captures meaningful spatial dependencies rather than merely fitting statistical correlations, we visualize the learned attention weights for a representative region, Barbour County, Alabama, United States. We examine the spatial interaction of both global and axial attention mechanisms.

In the second panel of Fig. 4, the global attention heatmap shows, for each pair of areas, how much the area in column j contributes to the representation of the area in row i. A prominent vertical band of high intensity appears at area 6, which corresponds to Eufaula, the largest city and economic center of Barbour County. This indicates that many other areas place relatively high attention on area 6, consistent with a hub role in the local mobility system. In contrast, area 3, which is located near the county boundary, attracts noticeably less attention from the rest of the county. The weaker column associated with area 3 reflects its more peripheral role and suggests that changes in this edge area have a smaller impact on the representations of other areas.

At a finer scale, the axial attention heatmaps in the third and fourth panels of Fig. 4 show the distribution of attention over destinations for two different origins. The third panel corresponds to an urban center, origin area 6, and exhibits a relatively diffuse pattern with a maximum attention value of about 0.137 spread across several neighboring destinations such as indices 5, 7 and 8. This suggests that flows originating from the center are influenced by a combination of multiple nearby areas, and the model distributes attention over a broader neighborhood that matches the multifaceted role of the city core. The fourth panel corresponds to a peripheral origin, area 0, and shows a sharper and more self focused pattern, with a maximum attention value of about 0.148 concentrated on index 0. For this edge area, outflow is therefore governed mainly by the intrinsic characteristics of the origin rather than by strong interactions with alternative destinations. The contrast between diffuse urban attention and focused peripheral attention indicates that GeoFlow reflects the spatial heterogeneity of commuting flows and adjusts its effective receptive field to the functional role of each area, rather than relying only on distance as in classical gravity models.

F.2. Visualization

To qualitatively assess the performance of GeoFlow, we randomly select several regions for visualization. As illustrated in Fig. 5, for each region we provide its geographic map, the ground truth OD flow, the model prediction, and three independently generated samples. The prediction results are designed to closely approximate the most likely OD flows, while the generated samples reflect the overall flow patterns while preserving key structural characteristics. This comparison demonstrates that GeoFlow not only accurately predicts dominant flows but also effectively captures the variability and essential features of the underlying OD distribution.

G. Discussion of Baseline Methods

To provide a comprehensive evaluation of GeoFlow, we compare against a representative set of baselines spanning principle-driven, machine learning, and deep learning paradigms. In this section, we summarize their main characteristics, strengths, and limitations.

Gravity Model. The gravity model is among the earliest and most widely used approaches to OD flow modeling. It posits that flows are positively associated with the ”mass” (e.g., population or economic activity) of the origin and destination and inversely associated with the distance between them, typically through a power-law or exponential decay. Its transparent mathematical form provides strong interpretability and has long served as a benchmark in mobility studies. However, its reliance on simple parametric functions limits flexibility in capturing heterogeneous and nonlinear mobility patterns observed in real data.

Classical Machine Learning Regressors. Tree-based models such as Random Forest (RF) and Gradient Boosted Regression Trees (GBRT), as well as kernel-based methods like Support Vector Regression (SVR), represent the next stage in OD flow modeling. These approaches directly learn mappings from urban attributes to flow values. RF and GBRT benefit from their ability to capture nonlinear feature interactions and to provide feature importance measures, while SVR leverages kernel functions to model similarity in high-dimensional feature spaces. Although they improve predictive accuracy over principle-driven models, their reliance on hand-crafted features and limited ability to capture spatial dependencies constrain their performance on large, complex urban systems.

Deep Learning Models. The Deep Gravity Model (DGM) builds on the intuition of gravity models but employs neural networks to learn flexible production, attraction, and impedance functions. This extension enables DGM to capture nonlinearities beyond traditional formulations while maintaining interpretability to some degree. The Geographical Multi-task Embedding Learning model (GMEL) incorporates graph attention networks to integrate spatial and structural signals, offering improved representation learning. However, these models generally treat areas as independent entities with limited consideration of long-range spatial interactions, which can restrict their expressiveness.

Graph-based Generative Models. Recent advances have explored graph neural networks and generative architectures to model OD flows as attributed graphs. NetGAN employs random walks to learn latent graph structures and can synthesize plausible OD matrices, but struggles with capturing fine-grained attributes. DiffODGen introduces diffusion processes to jointly model topology and edge weights, while WEDAN applies diffusion-based denoising to learn conditional flow generation. These methods capture complex structural dependencies and achieve strong generative performance, yet they typically abstract areas as nodes without explicit treatment of spatial proximity, limiting their ability to represent geographic relations that influence mobility.

Transformer-based Methods. TransFlower adapts transformer architectures to OD flow modeling by introducing relative positional embeddings, highlighting the importance of spatial relations between origins and destinations. While it demonstrates strong performance, its reliance on pairwise attention incurs high computational cost and can be difficult to scale to large urban systems. Furthermore, the integration of topological information remains limited, leaving room for improvement in modeling multi-relational geographic contexts.

In summary, principle-driven models emphasize interpretability but are constrained in flexibility; machine learning regressors improve predictive power but fail to capture structural dependencies; deep learning models introduce nonlinear representation learning but often overlook geographic context; and graph generative and transformer-based methods achieve strong expressiveness at the expense of interpretability or efficiency. GeoFlow is designed to address these gaps by explicitly incorporating multi-relational geographic structures while maintaining scalability, enabling more faithful modeling of OD flows across diverse urban contexts.

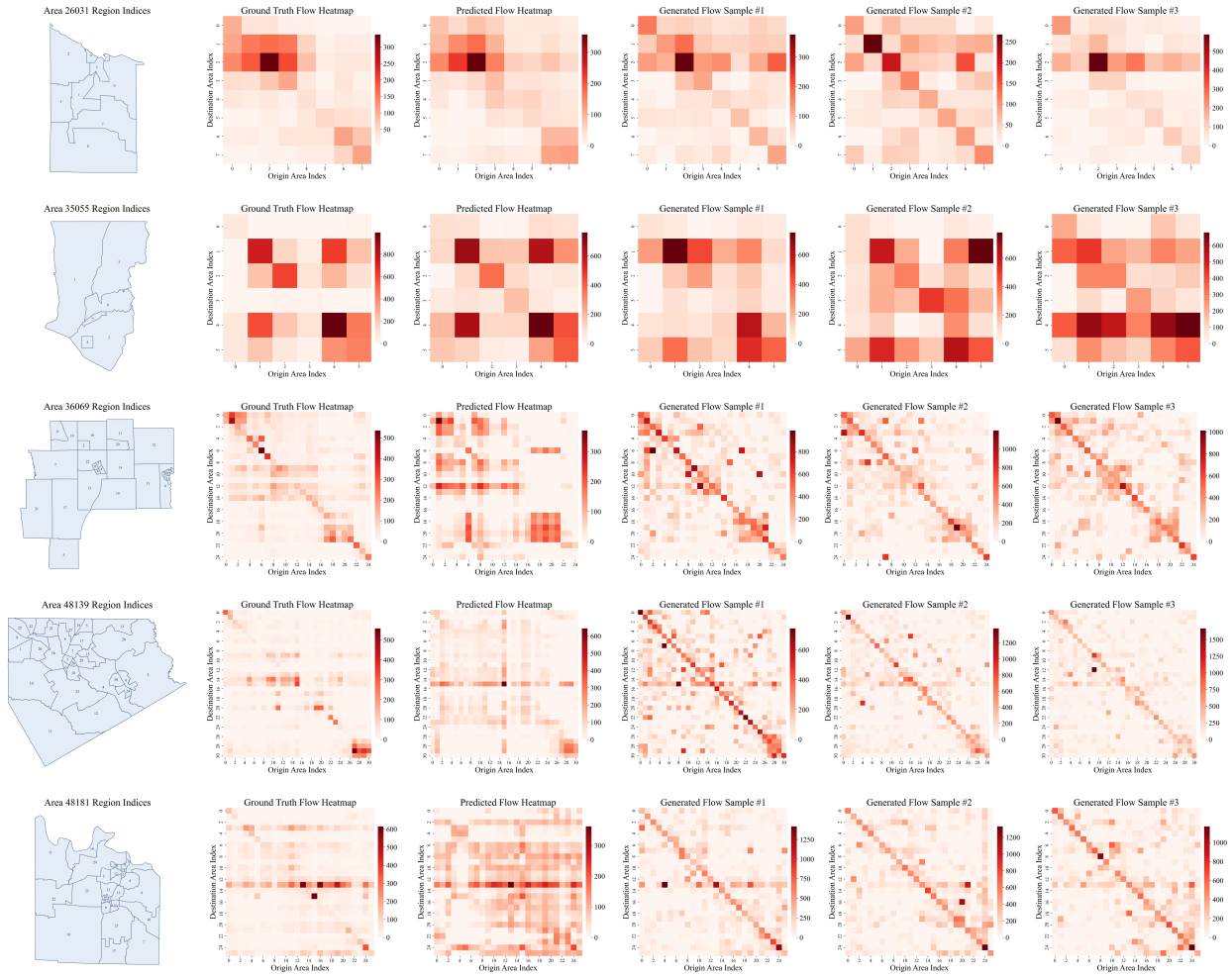

Figure 5. **Qualitative comparison of OD flows.** For each selected region, we show the geographic map, the ground truth, the model prediction, and three independently generated samples.