TrajFlow: nationwide Pseudo GPS Trajectory Generation with Flow Matching Models

Anonymous authors
Paper under double-blind review

Abstract

The importance of mobile phone GPS trajectory data is widely recognized across many fields, yet the use of real data is often hindered by privacy concerns, limited accessibility, and high acquisition costs. As a result, generating pseudo–GPS trajectory data has become an active area of research. Recent diffusion-based approaches have achieved strong fidelity but remain limited in spatial scale (small urban areas), transportation-mode diversity, and efficiency (requiring numerous sampling steps). To address these challenges, we introduce ***TrajFlow***, the first flow-matching–based generative model for GPS trajectory generation. TrajFlow leverages the flow-matching paradigm to improve robustness across multiple geospatial scales, and incorporates a trajectory harmonization & reconstruction strategy to jointly address scalability, diversity, and efficiency. Using a nationwide mobile phone GPS dataset with millions of trajectories across Japan, we show that TrajFlow consistently outperforms diffusion-based and deep generative baselines at urban, metropolitan, and nationwide levels. As the first nationwide, multi-scale GPS trajectory generation model, TrajFlow demonstrates strong potential to support inter-region urban planning, traffic management, and disaster response, thereby advancing the resilience and intelligence of future mobility systems.

1 Introduction

Human mobility data—particularly the rapidly growing mobile phone GPS data—has been widely applied across diverse domains, including urban studies (Chen et al., 2023; Jin et al., 2023), epidemiological prediction and control (Bao et al., 2020; Zhang et al., 2022), and transportation and travel planning (Torre-Bastida et al., 2018). However, the use of real personal mobility data poses several challenges, such as privacy concerns, limited accessibility, and substantial financial or time costs. Privacy is especially critical, as data collection and utilization may reveal sensitive personal details, which in turn makes human mobility datasets difficult to access and share.

Consequently, recent years have witnessed a growing number of studies on GPS trajectory generation (Zhu et al., 2023b; Jiang et al., 2023; Wei et al., 2024a; Zhu et al., 2023a). Notably, several research gaps remain even in state-of-the-art (SOTA) models for this task: **Multi-scale capability:** Existing models primarily focus on urban-level data and struggle to generalize to regional or nationwide scales. This limitation significantly constrains the practical applicability of generated trajectories, as real-world deployment often requires modeling across multiple spatial levels. In particular, maintaining a stable signal-to-noise ratio (SNR) becomes increasingly challenging when extending trajectory generation from block-level movements to citywide or nationwide patterns, underscoring the need for models with robust multi-scale capability. **Transportation-mode diversity:** Current approaches are largely confined to taxi trajectory data. While taxi GPS traces are valuable, real human mobility involves a much broader range of transportation modes (e.g., train, car, bike and walk), which are not adequately represented in existing models. **Training and inference efficiency:** Most SOTA methods rely on diffusion-based frameworks. Although methods such as DDIM (Denosing Diffusion Implicit Model) can accelerate sampling, they remain computationally expensive. Furthermore, the reliance on evidence lower bound (ELBO)-based optimization introduces additional complexity during training (Kingma & Gao, 2023), making large-scale trajectory generation inefficient and difficult to scale in practice.

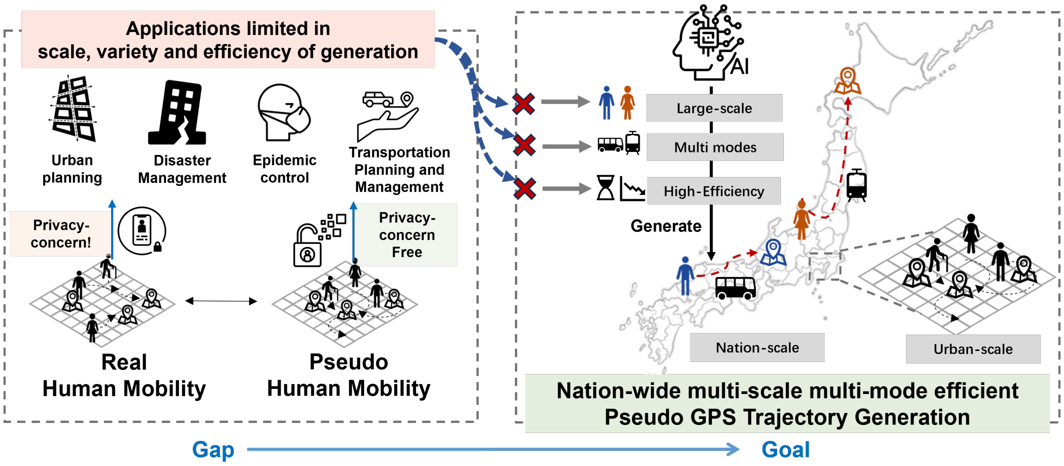


Figure 1: In pseudo-human mobility generation, three key challenges remain to be addressed: multi-scale capability, transportation-mode diversity, and training & inference efficiency.

To address these limitations, we propose **TrajFlow**, a flow-matching–based GPS trajectory generation model designed to produce nationwide, multi-scale pseudo-GPS trajectories. The main contributions of this study are summarized as follows:

- **Novel paradigm:** We present the first flow-matching–based generative framework for GPS trajectory modeling, and show that the flow-matching paradigm improves robustness and stability across multi-scale scenarios.
- **Methodological design:** We integrate trajectory harmonization, OD-conditioned normalization, and flow-based training into a unified framework that jointly addresses scalability, diversity, and efficiency in mobility generation.
- **Empirical validation:** Using a nationwide mobile phone GPS dataset with millions of users, we demonstrate that **TrajFlow** achieves state-of-the-art performance across urban, metropolitan, and nationwide settings, highlighting its value for large-scale human mobility modeling.

2 RELATED WORKS

Human Mobility Generation. Human mobility generation has attracted growing attention across both computer science and social science domains (Feng et al., 2020; Simini et al., 2021; JIAWEI et al., 2024). In the early stages, before the widespread availability of mobile phone positioning data, research primarily relied on mechanism-based approaches. Travel survey data—often referred to as travel diaries—were commonly used. Such surveys typically record an individual's daily travel activities, providing detailed information including the sequence of mobility activities and the modes of transportation. These rich datasets enabled the development of activity-based models for human mobility sequence generation (Karamshuk et al., 2011; Hess et al., 2015; Barbosa et al., 2018). With the growing demand for large-scale mobility data, researchers began to explore alternative sources such as GPS trajectories and call detail records (CDR). For example, TimeGeo (Jiang et al., 2016) extended activity-based modeling by leveraging GPS and CDR data. Different from TimeGeo, studies leveraging mobile phone GPS and CDR data have investigated the generation of more realistic activity patterns by combining deep learning models. For instance, Act2Loc (Liu et al., 2024) focuses on activity-to-location generation, while GeoAvatar (Li et al., 2022; 2025) achieves individualized trajectory generation, reflecting both temporal regularity and personal heterogeneity. More recently, advances in large language models (LLMs) have allowed for interpretable mobility generation at a relatively higher computation cost (JIAWEI et al., 2024; Feng et al., 2025; Beneduce et al., 2025).

GPS trajectory generation. As one of the most widely used data sources for representing human mobility, GPS trajectories have become a central focus in generative modeling. A variety of approaches have been proposed for GPS trajectory generation, ranging from model-based models to learning based frameworks(Yin et al., 2017; Hsu et al., 2024; Zhu et al., 2023b; Wang et al., 2021). Early work often adopted sequence modeling techniques such as Hidden Markov Models (HMMs) and Recurrent Neural Networks (RNNs), including their variants like LSTMs and GRUs (Yu et al., 2019). For example, Song et al. (Liu et al., 2018) applied an HMM-based model to predict and simulate large-scale human mobility patterns following natural disasters. Beyond Markovian models, deep generative model such as Variational Autoencoders (VAEs) or generative adversarial networks (GANs) have been introduced to capture latent mobility representations (Huang et al., 2019; Chen et al., 2021b). More recently, the rapid proliferation of denoising diffusion probabilistic models (DDPM) have further advanced GPS trajectory generation. DiffTraj (Zhu et al., 2023b) and its extensions (Zhu et al., 2024; Wei et al., 2024b) have shown strong performance in generating urban-scale taxi trajectories.

3 PRELIMINARY

3.1 PROBLEM DEFINITION

Definition 1 (Human Mobility (GPS Trajectory))**.** A human mobility record is represented as a tuple (lat, lon, t), indicating that user u visits a geographic location l (specified by latitude and longitude) at time t. A mobility trajectory is then defined as an ordered sequence of such records, $\text{traj} = \{(lat_0, lon_0, t_0), (lat_1, lon_1, t_1), \dots, (lat_n, lon_n, t_n)\}$. In our setting, anonymized data are used, and user identifiers u are replaced with random tokens to ensure privacy.

Definition 2 (GPS Trajectory Generation)**.** Given a ground-truth dataset of GPS trajectories $X = \{\text{traj}_x^1, \text{traj}_x^2, \dots, \text{traj}_x^m\}$, the goal of GPS trajectory generation is to synthesize a pseudo-dataset $Y = \{\text{traj}_y^1, \text{traj}_y^2, \dots, \text{traj}_y^m\}$, such that Y closely matches the distributional characteristics of X while preserving user privacy.

3.2 FLOW MATCHING MODEL

Flow Matching is a powerful class of generative models designed to learn continuous transformations between probability distributions. Unlike diffusion models that rely on a fixed noising process, flow matching models learn a vector field $v_t(x)$ that directly models the ”flow” of particles from a simple base distribution p_0 (e.g., a standard Gaussian) to a complex target data distribution p_1 (e.g., the distribution of real GPS trajectories).

The core idea is to define a probability path $p_t(x)$ and a corresponding time-dependent vector field $v_t(x)$ such that a sample $x_0 \sim p_0$ can be transformed into a sample $x_1 \sim p_1$ by solving the ordinary differential equation (ODE)(Lipman et al., 2022), $\frac{dx_t}{dt} = v_t(x_t)$, where x_t is the state of a particle at time $t \in [0, 1]$. The goal is to train a neural network $v_\theta(x, t)$ to approximate this ground-truth vector field $v_t(x)$.

A key innovation in flow matching is the objective function. Instead of directly regressing on the often-intractable vector field of the marginal probability path p_t, Conditional Flow Matching (CFM) defines a conditional probability path $p_t(x \mid x_1)$ and a corresponding conditional vector field $u_t(x \mid x_1)$ that are much simpler to compute. A common choice for the conditional path is a simple linear interpolation between a noise sample x_0 and a data sample x_1: $p_t(x \mid x_1) = \mathcal{N}\big(x \mid (1-t)x_0 + tx_1,\ \sigma^2 I\big)$. However, the simplest and most common formulation, which we adopt, regresses the model directly on the vector field defined by a straight path between x_0 and x_1: $u_t(x \mid x_1) = x_1 - x_0$. The model $v_\theta(x, t)$ is then trained to predict this vector field u_t given the point x_t on the path. The flow matching objective is a simple regression loss:

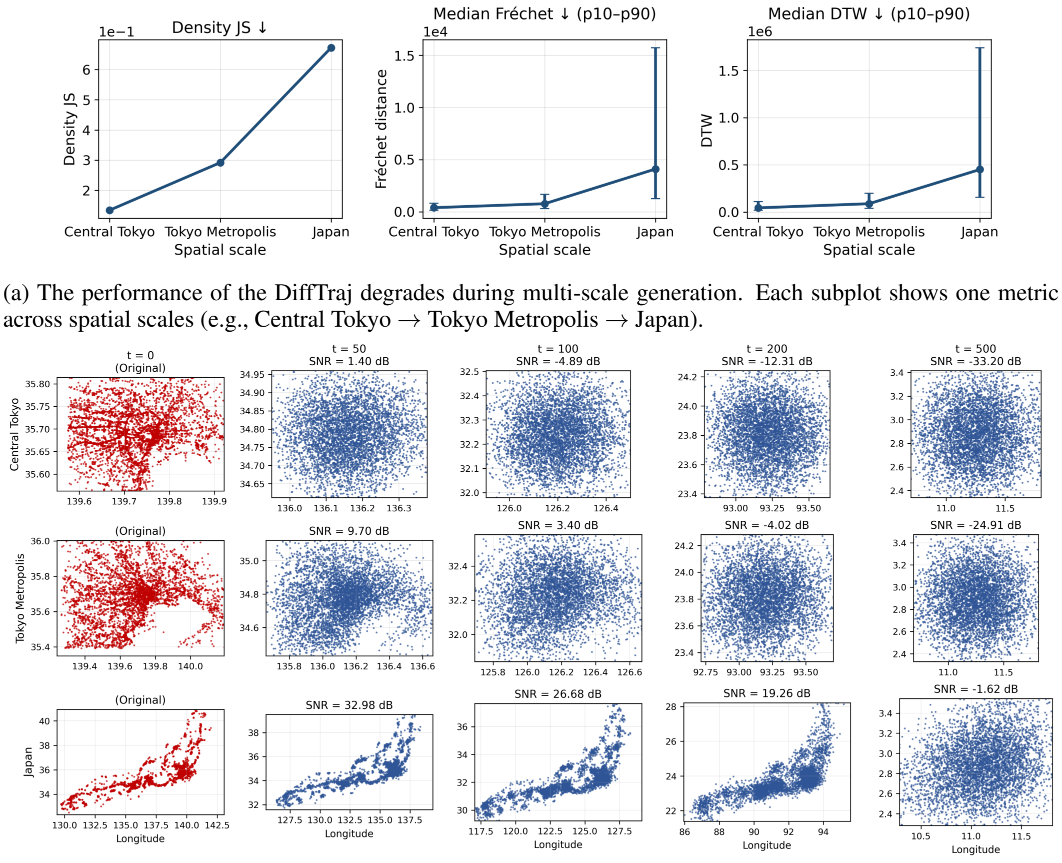
$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,\, p(x_1),\, p(x_0)}\Big[\big\|v_\theta\big((1-t)x_0 + tx_1,\ t\big) - (x_1 - x_0)\big\|^2\Big], \tag{1}$$

where t is sampled uniformly from $[0, 1]$, x_1 is sampled from the real data distribution, and x_0 is sampled from the prior noise distribution. This objective allows the model to learn the complex data distribution stably and efficiently. Once trained, we can generate new data by sampling a point $x_0 \sim p_0$ and solving the learned ODE $\frac{dx}{dt} = v_\theta(x, t)$ from $t = 0$ to $t = 1$ using an ODE solver.

4 TRAJFLOW

4.1 MOTIVATION

Diffusion-based models (e.g., DiffTraj) have achieved high-fidelity GPS trajectory generation at the *urban* scale, but when the spatial grow up, e.g., from urban scale, metropolis scale, and to nationwide scale, we observe a sharp degradation of accuracy across multiple metrics (see Fig. 2a).



(a) The performance of the DiffTraj degrades during multi-scale generation. Each subplot shows one metric across spatial scales (e.g., Central Tokyo → Tokyo Metropolis → Japan).

(b) SNR collapses during the diffusion noising procedure. A noise parameter suitable for a nationwide scale is not suitable for smaller regions (e.g., Tokyo Metropolis).

Figure 2: **(a)** shows the accuracy degradation of DiffTraj at increasing spatial scales. **(b)** shows the SNR collapses when applying a fixed noising parameter across scales.

We suggest that there are two key reasons why diffusion-based SOTA models fail to generalize across scales. First, when expanding to larger geographical regions, the subsets of fine-grained local trajectories become small in magnitude, resulting in an extremely low SNR. In these cases, the weak signal is easily overwhelmed, forcing the reverse process to reconstruct fine-grained structures from highly noisy inputs. Standard mean squared error (MSE) objectives further intensify this challenge by prioritizing absolute rather than relative error, thereby aggravating the imbalance across scales. To address this, *data harmonization becomes essential*: by curating and rescaling data distributions across different spatial scales, we can explicitly compensate for this imbalance and provide a more balanced learning signal (see Sec. 4.2). Second, diffusion models rely on a stepwise denoising process that gradually transforms Gaussian noise into samples. In conditional settings where subsets of data span vastly different numerical ranges—for example, micro-scale local trips versus macro-scale long-distance journeys—the forward noising process injects noise of nearly fixed magnitude regardless of scale. This scale-invariant noise injection leads to a fundamental mismatch that reduces model robustness. To overcome this limitation, *flow matching is necessary*: by directly learning the continuous probability flow between a simple prior and the target distribution, it sidesteps the fixed-step denoising chain and provides a more flexible and scale-adaptive generative mechanism (see Sec . 4.3 & 4.4). In the following, we detail how our proposed framework, TrajFlow, addresses the critical challenges of urban mobility modeling and generates reliable pseudo-GPS trajectories.

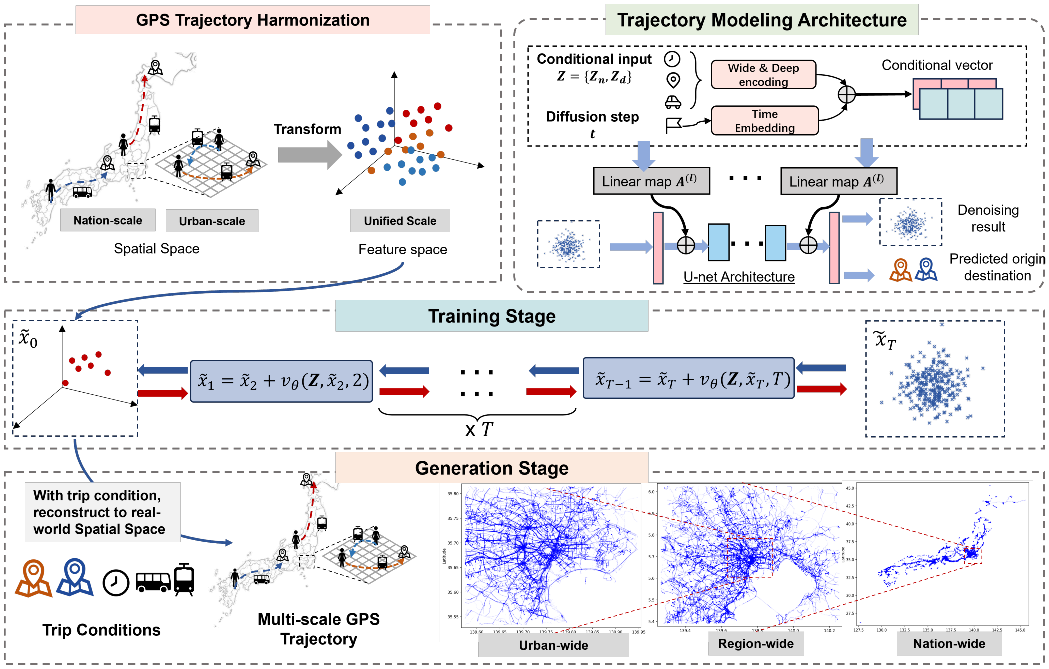


Figure 3: The overview of the proposed TrajFlow.

4.2 Multi-scale GPS Trajectory Harmonization and Reconstruction

To address the SNR imbalance across scales, we adopt a trajectory harmonization & reconstruction strategy in conjunction with the flow matching framework (see the upper-left panel of Fig. 3). Rather than working directly on raw GPS coordinates, which vary across orders of magnitude, we normalize each trajectory individually, rescaling all points to lie in a common bounded range (approximately $\mathcal{N}(0, 1)$). The model thus operates in this normalized space, predicting OD and waypoints under flow matching; afterward, we invert the normalization back to real geographic coordinates. This prevents tiny local displacements from being overwhelmed by large-scale variations, stabilizes gradient magnitudes during optimization, and accelerates convergence (Ioffe & Szegedy, 2015).

In addition, to enhance the efficiency and stability of model learning, we apply a trajectory *trajectory-feature transformation* step that simplifies each trajectory while preserving its essential geometric structure. This kind of trajectory-feature transformation reduces the length of the raw trajectories from L to D, where $D \ll L$, thereby lowering computational overhead and improving training stability. In practice, the process involves recursively removing points that fall within a tolerance ϵ of the line segment formed by their neighbors, retaining only those points that contribute to the overall geometric shape. As a result, long straight segments are compressed into a few representative points, while turning points or areas of high curvature are preserved. In this study, we experiment with multiple harmonization methods and find that the *Ramer–Douglas–Peucker* (RDP) algorithm (Ramer, 1972) provides the best trade-off between compression and fidelity. Details of this algorithm are presented in the Appendix. A.2.

The proposed strategy can be interpreted as analogous to the normalization–denormalization procedure in feature preprocessing: trajectories are first compressed into a compact representation to facilitate efficient model training, and subsequently expanded back to their original scale for deployment. To the best of our knowledge, we are the first to explore the integration of RDP-based processing with a flow matching framework for trajectory generation, and we will demonstrate its advantages in terms of computational efficiency and practical fidelity through experimental studies.

4.3 GPS Trajectory modeling architecture

As shown in the upper-right panel of Fig. 3), we adopt the Wide & Deep module (Zhu et al., 2023b) to embed the input condition of a trajectory, including departure time, OD zone, and transportation mode. Concretely, the conditional vector e_c is formed by fusing a linear wide projection of numeric motion/context characteristics with a non-linear deep projection of discrete context characteristics, and the result e_c is injected into every block of the vector field network $v_\theta(x_t, t, e_c)$ that parameterizes our probability flow:

$$\frac{dx_t}{dt} = v_\theta(x_t, t, e_c). \tag{2}$$

Inputs and feature partition. The first part of the input is numeric features $\boldsymbol{Z}_n$: segment-level or trajectory-level scalars such as instantaneous speed, inter-point distance, elapsed time, cumulative distance/steps (all standardized). The other part is the discrete features $\boldsymbol{Z}_d$, including departure time of a day, transportation mode.

Wide&Deep encoding. Firstly, an MLP layer is employed to map numeric features $\boldsymbol{Z}_n$ to n-dimension vector. Then, each categorical field is embedded into n-dimension (n is defined by hyper-parameter) as e_{wide}; the embeddings are concatenated and passed through two MLP layers with nonlinearity as e_{deep}. Finally, we fuse the e_{wide} and e_{deep} as the conditional vector:

$$e_c = \text{LayerNorm}(e_{wide} + e_{deep}). \tag{3}$$

Time embedding. The continuous flow time $t \in [0, 1]$ is encoded by a sinusoidal/Fourier mapping followed by a small MLP to yield a time vector $e_t \in \mathbb{R}^{d_e}$. We combine it with the condition as a single control signal $\tilde{e} = e_c + e_t$.

Conditional control into the backbone. Let $h^{(\ell)}$ denote the hidden state of block ℓ. We broadcast $\tilde{e}$ to all blocks and inject it as an additive, learnable bias as:

$$h^{(\ell)} \leftarrow f^{(\ell)}\big(h^{(\ell)}\big) + A^{(\ell)}\tilde{e} \qquad (\ell = 1, \ldots, L), \tag{4}$$

where $f^{(\ell)}$ is the block's transformation and $A^{(\ell)}$ is a learned linear map. This keeps the conditioning pathway lightweight and consistent with the ODE parameterization of the vector field.

4.4 Flow-matching Training and inference

At the training stage (i.e., the middle panel of Fig. 3), we use mini-batches of ground-truth trajectories that are first harmonized with RDP and then padded to a common length using a validity mask. For each trajectory, we compute the numeric and discrete conditions, obtain the condition embedding e_c through the Wide&Deep module (Eq. 3), and sample a flow time t with a noise endpoint x_0. The straight-path point x_t and the corresponding target conditional vector field are then constructed according to the flow-matching objective (Eq. 1). In addition, we introduce an auxiliary supervised loss between the predicted and ground-truth OD pairs to enhance spatial and semantic awareness. Model parameters are optimized with a masked regression loss over valid tokens, optionally augmented by smoothness and bounded-support regularizers, followed by standard optimizer updates.

At the inference stage (i.e., the last panel of Fig. 3), given a trajectory condition (e.g., departure time, OD pair, transportation mode), we first compute e_c using the Wide&Deep encoder and prepare the time-embedding schedule. We then initialize x_0 from the base distribution (e.g., standard Gaussian noise). The synthetic trajectory is obtained by numerically integrating the learned flow ODE on $[0, 1]$ (Eq. 2), injecting e_c at each solver step as in Eq. 4. The final state x_1 is subsequently post-processed (e.g., uniform resampling, mapping back to geographic coordinates, and trimming or interpolating to satisfy length priors), yielding the generated trajectory under the specified condition.

5 EXPERIMENTS

In this section, we comprehensively evaluate **TrajFlow** to answer the following research questions: **Q1:** Does TrajFlow outperform baseline methods on key evaluation metrics? **Q2:** How well does TrajFlow perform in multi-scale trajectory generation? **Q3:** To what extent can TrajFlow reproduce transportation-mode diversity? **Q4:** How efficient is TrajFlow in terms of training and inference?

5.1 Settings

Dataset We employed the Japan nationwide Blogwatcher Dataset in 2023. This private dataset contains fine-grained GPS records from users in Japan, collected from Blogwatcher Inc. This data is obtained with the use of multiple applications on mobile devices, comprising millions of trajectories, which include anonymized user IDs, latitude, longitude, timestamps, and transportation modes.

Baselines Diffusion methods represent the current SOTA in urban mobility generation - we adopt *DiffTraj* (Zhu et al., 2023b) as a representative diffusion-based baseline. In addition, we include *TrajVAE* (Chen et al., 2021a) and *TrajGAN* (Rao et al., 2020) as two representative baselines to evaluate the effectiveness of deep learning and adversarial learning approaches for urban mobility generation tasks. To disentangle the contributions of each key component in *TrajFlow*, we evaluate three ablated variants. *w/o-FM:* The CFM objective and ODE solver are replaced with the standard DDPM denoising procedure. *w/o-OD:* The OD prediction head and the per-trajectory processing pipeline are removed, which means harmonization/reconstruction is not conducted - the trajectory will be directly generated in geographic space. *w/o-RDP:* The RDP trajectory-harmonization step is omitted, and raw trajectories (up to 120 points) are fed into the model. These ablations isolate the effects of the flow-matching objective, harmonization/reconstruction method, and trajectory-compression strategy on model fidelity and scalability.

Evaluation Metrics We evaluate generation quality from two perspectives. **Aggregated–level:** The *Jensen–Shannon divergence of spatial density (density JS)* measures how well synthetic data reproduces the population-level geo–distribution of movements. **Trajectory–level:** We assess path similarity using *Dynamic Time Warping (DTW)* and the continuous *Fréchet distance (Fr)*, which is sensitive to overall shape rather than temporal alignment. For both metrics, we report the median, mean, and 10th/90th percentiles (P10/P90) to capture central accuracy and dispersion. Details are provided in Appendix A.1.

5.2 Evaluation

Overall Performance (Q1) Across spatial scales—Center Tokyo (urban), Tokyo Metropolis (metro), and Japan (nationwide)—*TrajFlow* and its variants consistently outperform *DiffTraj* on all metrics. The advantage is modest at the urban and metro levels but becomes pronounced nationwide, highlighting stronger cross-scale generalization. At the urban-wide scale, while *TrajFlow* achieves satisfactory performance, *TrajFlow*-w/o RDP & OD performs better as it directly generates raw coordinates. This indicates that smoothing details at a small scale is not always necessary. At the metropolitan scale, part of the best indicators transferred from TrajFlow-w/o-RDP&OD to TrajFlow-w/o-RDP, showing that the OD prediction-based harmonization and reconstruction component starts to show the benefit as the range increases at the urban scale. And the TrajFlow successfully maintains similar performance as the urban scale. At the nationwide scale, the advantage is substantial: $\text{DTW}_{\text{med}} = 10.977$, $\text{Fr}_{\text{med}} = 0.192$, tight IQR/P90, and the lowest density JS, while alternatives yield larger medians and heavier tails. Although *TrajFlow* is not strictly best on every metric at smaller scales, it is the only method that balances shape fidelity, stability (IQR/P90), and spatial distribution (density JS) across all scales, and it dominates in the nationwide setting. Geographic visualizations are provided in Fig. 4. This is consistent with the analysis in the Sec. 4.1 - while trajectories keep a smaller scale (urban), the SNR issue of DDPM is not serious, but when it comes to nation-wide - mixing of urban-scale, metropolis-scale, and nation-wide scale - SNR issue hinders the performance of DDPM across scales, a limitation that flow matching addresses effectively.

Multiscale-scale robustness (Q2) From city-level to metropolitan-level to nationwide, TrajFlow maintains low median DTW/Fréchet and narrow dispersion, while keeping the JS divergence of density controlled. In conditional settings where subsets of data span vastly different numerical ranges—for example, micro-scale local trips versus macro- scale long-distance trips — the forward noising process injects noise of nearly fixed magnitude regardless of scale. This scale-invariant noise injection leads to a fundamental mismatch that reduces model robustness. This advantage over multi-scale is consistent owing to the conditional flow matching: a deterministic vector field transports probability mass in a single flow, avoiding accumulation of error along long sampling chains and generalizing better when regions and transport modes are diverse.

Transportation-mode diversity (Q3) Preserving the diversity of transportation modes is as important as maintaining spatial fidelity in generative mobility models.

Table 1: Evaluation grouped by region (unit = km). Best metric in **bold**.

Method	Density JS ↓	DTW_{med} ↓	Fr_{med} ↓	DTW_{IQR} ↓	DTW_{P10} ↓	DTW_{P90} ↓	Fr_{IQR} ↓	Fr_{P10} ↓	Fr_{P90} ↓
Central Tokyo									
TrajFlow (ours)	0.0674	20.350	0.304	13.392	10.574	39.119	0.174	0.200	0.674
TrajFlow-w/o-OD	0.3560	916.436	13.862	708.183	432.567	1,740.890	7.313	7.064	20.858
TrajFlow-w/o-RDP	0.0642	22.088	0.340	16.491	11.149	47.959	0.238	0.209	0.873
TrajFlow-w/o RDP & OD	**0.0323**	**8.179**	**0.184**	**4.586**	**4.994**	**14.159**	**0.118**	**0.110**	**0.363**
DiffTraj	0.1340	44.321	0.651	40.713	19.399	109.349	0.544	0.341	1.774
TrajGAN	0.3087	292.430	4.442	448.839	119.230	1,288.929	7.660	1.606	21.477
TrajVAE	0.1041	32.874	0.469	36.387	14.000	103.232	0.679	0.258	1.842
Tokyo Metropolis									
TrajFlow (ours)	0.1239	18.167	0.335	16.892	7.678	44.316	0.333	0.130	0.933
TrajFlow-w/o-OD	0.1064	16.466	0.307	10.189	9.307	32.126	0.192	0.180	0.683
TrajFlow-w/o-RDP	0.1197	18.417	**0.298**	24.152	**6.637**	67.674	0.473	**0.121**	1.339
TrajFlow-w/o RDP & OD	**0.0800**	**14.416**	0.303	**7.684**	8.659	**23.978**	**0.188**	0.176	**0.592**
DiffTraj	0.2918	88.559	1.220	78.339	38.663	199.501	0.982	0.586	3.035
TrajGAN	0.3821	604.399	10.224	1,184.077	155.401	2,854.060	20.627	2.290	51.389
TrajVAE	0.1930	46.363	0.765	54.484	16.234	122.556	1.006	0.250	2.299
Japan nationwide									
TrajFlow (ours)	**0.2270**	**10.977**	**0.192**	**18.221**	**3.984**	**55.964**	**0.361**	**0.072**	**1.119**
TrajFlow-w/o-OD	0.4888	100.271	1.522	75.774	50.877	216.168	1.177	0.774	3.145
TrajFlow-w/o-RDP	0.2734	24.690	0.400	27.928	9.047	92.641	0.511	0.156	1.699
TrajFlow-w/o RDP & OD	0.4865	105.011	1.662	89.509	53.549	280.092	1.380	0.870	3.802
DiffTraj	0.6727	451.042	5.329	635.120	157.379	1,741.025	6.973	1.915	18.924
TrajGAN	0.5278	403.282	6.653	999.210	79.556	2,853.557	17.703	1.134	48.838
TrajVAE	0.5228	135.377	2.216	236.143	28.394	577.139	3.884	0.435	9.919

We evaluate this by comparing per–mode trip distance distributions of ground-truth and generated trajectories in Tokyo. As shown in Fig. 5, the generated data matches the characteristic profiles of four representative modes, maintaining realistic differences in trip lengths. This indicates that *TrajFlow* not only captures spatial fidelity but also preserves transportation-mode diversity (see additional results in Appendix C).

Training and inference efficiency (Q4) *TrajFlow* is optimized with the CFM objective, which requires only a single time step per sample, and generates trajectories by integrating the learned ODE with a small fixed budget of about 10 steps. In contrast, DDPM performance is highly sensitive to the number of denoising steps (see Appendix B): among the tested settings (10, 50, 100, 200, 300), the best trade-off between Jensen–Shannon divergence and trajectory-shape fidelity typically appears near 200 steps, while 300 steps provide only marginal or inconsistent improvements at significantly higher cost. Notably, even with 300 steps, DDPM fails to reach the performance of *TrajFlow*. These results highlight that competitive fidelity can be achieved by *TrajFlow* with far fewer steps (10 steps only) in both training and inference.

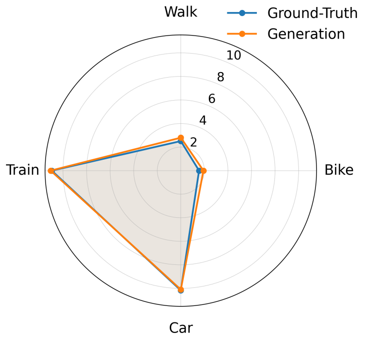


Figure 5: Per-mode average trip distance in Tokyo: ground truth vs. generated.

5.3 Ablation studies

w/o-FM: Replacing the CFM objective with a DDPM-style denoising procedure yields monotonic but limited gains as steps increase, and consistently underperforms *TrajFlow* in DTW, Fréchet, and density JS—even with large step budgets. This confirms flow matching as the main driver of fidelity and stability. **w/o-RDP:** Removing trajectory harmonization retains micro-jitter and redundant points, increasing Fréchet/DTW medians and tail metrics; on the Japan split, it also raises density JS. Thus, RDP improves not only efficiency but also accuracy by suppressing noise and harmonizing sampling. **w/o-OD:** Dropping the OD predictor sometimes lowers density JS in small regions but consistently worsens DTW/Fréchet and dispersion, and degrades both shape and density metrics nationwide. **w/o-RDP & OD:** Combining both removals produces the greatest deterioration, underscoring their critical role, particularly on a nationwide scale.

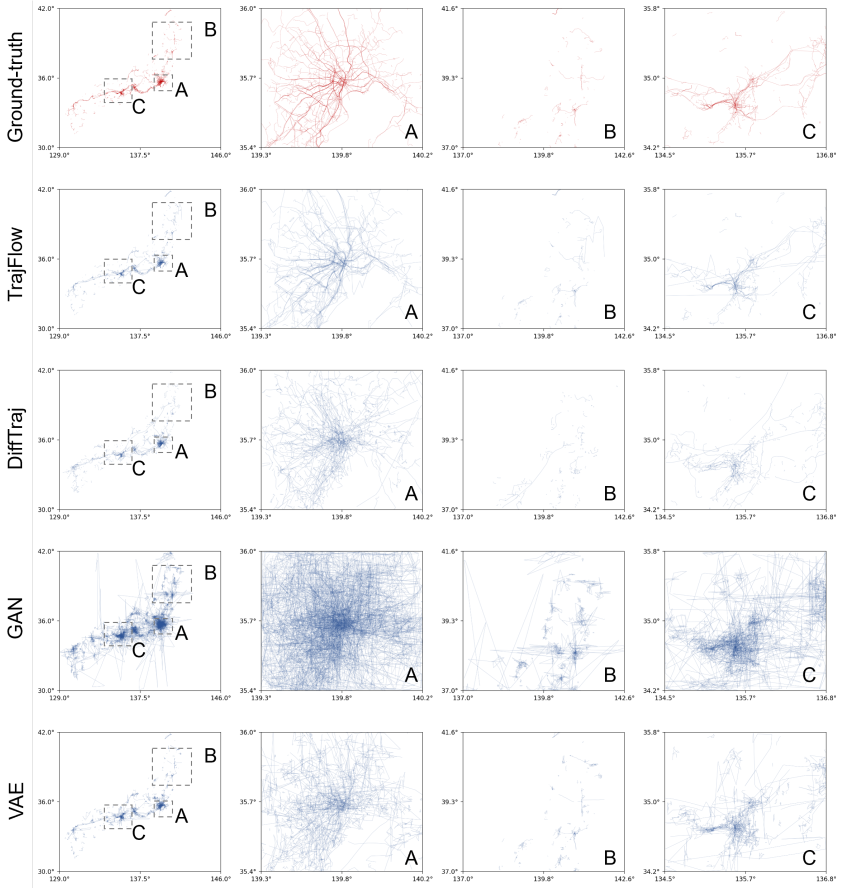


Figure 4: Visualization of trajectory samples. Ground-truth and generated nationwide trajectories are shown with zoomed views highlighting three representative regions: (A) Tokyo Metropolis, (B) Tohoku Area, and (C) Kansai Area, across all generative models.

6 CONCLUSION REMARKS

We introduce **TrajFlow**, a flow–matching–based framework for generating pseudo-GPS trajectories. By integrating trajectory harmonization and reconstruction into a conditional generative framework, TrajFlow addresses the longstanding challenges of multi-scale and multi-modal trajectory generation. Evaluated on a nationwide mobile-phone GPS dataset from Japan, TrajFlow outperforms various baselines. Particularly, it balances trajectory-shape fidelity, stability, and population-level spatial consistency across scales, while maintaining accuracy under fewer ODE steps.

Due to privacy rules and data access, we focus on GPS trajectory generation and do not use any per-user attributes (e.g., age, gender, home–work identifiers) or persistent pseudonymous IDs, and thus the current model does not capture individual preferences. As future work, we aim to extend TrajFlow from GPS trajectory generation to broader human mobility modeling.

LLM USAGE STATEMENT

In accordance with the ICLR 2026 policy on LLM usage, we disclose that LLMs (specifically OpenAI's ChatGPT) were employed as a general-purpose writing assistant. The usage was limited to improving grammar, clarity, and LaTeX formatting of the manuscript.

REFERENCES

H. Bao, X. Zhou, Y. Zhang, Y. Li, and Y. Xie. Covid-gan: Estimating human mobility responses to covid-19 pandemic through spatio-temporal conditional generative adversarial networks. In *Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2020. doi: 10.1145/3397536.3422261.

Hugo Barbosa, Marc Barthélemy, Gourab Ghoshal, C. R. James, Maxime Lenormand, Tristan Louail, Ronaldo Menezes, José J. Ramasco, Filippo Simini, and Marc Tomasini. Human mobility: Models and applications. *Physics Reports*, 734:1–74, 2018. doi: 10.1016/j.physrep.2018.01.001.

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

Xinyu Chen, Jiajie Xu, Rui Zhou, Wei Chen, Junhua Fang, and Chengfei Liu. Trajvae: A variational autoencoder model for trajectory generation. *Neurocomputing*, 428:332–339, 2021a.

Xinyu Chen, Jing Xu, Rui Zhou, Wei Chen, Jian Fang, and Chao Liu. Trajvae: A variational autoencoder model for trajectory generation. *Neurocomputing*, 428:332–339, 2021b. doi: 10.1016/j.neucom.2020.03.120.

Z. Chen et al. Using mobile phone big data to identify inequity of aging groups in transit-oriented development station usage: A case of tokyo. *Transport Policy*, 132:65–75, 2023. doi: 10.1016/j.tranpol.2022.12.010.

Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. Learning to simulate human mobility. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3426–3433, 2020.

Jie Feng, Yuwei Du, Jie Zhao, and Yong Li. Agentmove: A large language model based agentic framework for zero-shot next location prediction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1322–1338, 2025.

Andrea Hess, Karin Anna Hummel, Wilfried N. Gansterer, and Günter Haring. Data-driven human mobility modeling: A survey and engineering guidance for mobile networking. *ACM Computing Surveys*, 48(3):38:1–38:39, 2015. doi: 10.1145/2840722.

Shang-Ling Hsu, Emmanuel Tung, John Krumm, Cyrus Shahabi, and Khurram Shafique. Trajgpt: Controlled synthetic trajectory generation using a multitask transformer-based spatiotemporal model. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, pp. 362–371, 2024.

Dong Huang et al. A variational autoencoder based generative model of urban human mobility. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 425–430, 2019. doi: 10.1109/MIPR.2019.00086.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Shan Jiang, Yingxiang Yang, Shounak Gupta, Daniele Veneziano, Siddharth Athavale, and Marta C. González. The timegeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences*, 113(37):E5370–E5378, 2016. doi: 10.1073/pnas.1524261113.

Wenjun Jiang, Wayne Xin Zhao, Jingyuan Wang, and Jiawei Jiang. Continuous trajectory generation based on two-stage gan. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 4374–4382, 2023.

WANG JIAWEI, Renhe Jiang, Chuang Yang, Zengqing Wu, Ryosuke Shibasaki, Noboru Koshizuka, Chuan Xiao, et al. Large language models as urban residents: An llm agent framework for personal mobility generation. *Advances in Neural Information Processing Systems*, 37:124547–124574, 2024.

Y. Jin et al. Understanding railway usage behavior with ten million gps records. *Cities*, 133:104117, 2023. doi: 10.1016/j.cities.2022.104117.

Dmytro Karamshuk, Chiara Boldrini, Marco Conti, and Andrea Passarella. Human mobility models for opportunistic networks. *IEEE Communications Magazine*, 49(12):157–165, 2011. doi: 10.1109/MCOM.2011.6094021.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

Peiran Li et al. Iiot based trustworthy demographic dynamics tracking with advanced bayesian learning. *IEEE Transactions on Network Science and Engineering*, 10(5):2745–2754, 2022.

Peiran Li et al. Geoavatar: A big mobile phone positioning data-driven method for individualized pseudo personal mobility data generation. *Computers, Environment and Urban Systems*, 119: 102252, 2025. doi: 10.1016/j.compenvurbsys.2025.102252.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Tal Le. Flow matching for generative modeling. *arXiv*, 2022.

Kun Liu, Xin Jin, Shihong Cheng, Song Gao, Lu Yin, and Feng Lu. Act2loc: a synthetic trajectory generation method by combining machine learning and mechanistic models. *International Journal of Geographical Information Science*, 38(3):407–431, 2024. doi: 10.1080/13658816.2023.2292570.

Xia Liu, Hexuan Chen, and Clio Andris. trajgans: Using generative adversarial networks for geo-privacy protection of trajectory data (vision paper). In *Proceedings of the 2018 Location Privacy and Security Workshop (LPS'18), co-located with ACM SIGSPATIAL*, pp. 1–7, 2018.

Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256, 1972.

Jinmeng Rao, Song Gao, Yuhao Kang, and Qunying Huang. Lstm-trajgan: A deep learning approach to trajectory privacy protection. *arXiv preprint arXiv:2006.10521*, 2020.

Filippo Simini, Gianni Barlacchi, Massimilano Luca, and Luca Pappalardo. A deep gravity model for mobility flows generation. *Nature communications*, 12(1):6576, 2021.

Ana I. Torre-Bastida, Javier Del Ser, Ibai Laña, Maitena Ilardia, Miren Nekane Bilbao, and Sergio Campos-Cordobés. Big data for transportation and mobility: recent advances, trends and challenges. *IET Intelligent Transport Systems*, 12(8):742–755, 2018. doi: 10.1049/iet-its.2018.5188.

Xingrui Wang, Xinyu Liu, Ziteng Lu, and Hanfang Yang. Large scale gps trajectory generation using map based on two stage gan. *Journal of Data Science*, 19(1):126–141, 2021.

Tonglong Wei, Youfang Lin, Shengnan Guo, Yan Lin, Yiheng Huang, Chenyang Xiang, Yuqing Bai, and Huaiyu Wan. Diff-rntraj: A structure-aware diffusion model for road network-constrained trajectory generation. *IEEE Transactions on Knowledge and Data Engineering*, 2024a.

Tonglong Wei, Youfang Lin, Shengnan Guo, Yan Lin, Yiheng Huang, Chenyang Xiang, Yuqing Bai, and Huaiyu Wan. Diff-rntraj: A structure-aware diffusion model for road network-constrained trajectory generation. *arXiv*, 2024b. Accepted to IEEE TKDE (per arXiv note).

Mogeng Yin, Madeleine Sheehan, Sidney Feygin, Jean-François Paiement, and Alexei Pozdnoukhov. A generative model of urban activities from cellular data. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1682–1696, 2017.

Yong Yu, Xishi Si, Chunyan Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270, 2019. doi: 10.1162/neco_a_01199.

H. Zhang et al. Epidemic versus economic performances of the covid-19 lockdown: A big data driven analysis. *Cities*, 120:103502, 2022. doi: 10.1016/j.cities.2021.103502.

Y. Zhu et al. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. *arXiv*, 2024.

Yuanshao Zhu, Yongchao Ye, Ying Wu, Xiangyu Zhao, and James Yu. Synmob: Creating high-fidelity synthetic gps trajectory dataset for urban mobility analysis. *Advances in Neural Information Processing Systems*, 36:22961–22977, 2023a.

Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James J. Q. Yu. Difftraj: Generating gps trajectory with diffusion probabilistic model. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023b. URL `https://proceedings.neurips.cc/paper_files/paper/2023/cd9b4a28fb9eebe0430c3312a4898a41-Paper-Conference.pdf`.

A Appendix

A.1 Evaluation Metrics

To quantify the fidelity of the generated trajectories, besides the use of density JS-divergence, we also employed two trajectory(line)-level evaluation metrics: DTW and Fréchet distance. DTW aligns trajectories in time and is robust to local temporal shifts, highlighting point-wise spatiotemporal fidelity. Fréchet distance, in contrast, evaluates the closest continuous matching of two curves and is more sensitive to global path geometry. Using both metrics provides a comprehensive view: DTW captures fine-grained temporal accuracy, while Fréchet complements it by emphasizing overall spatial shape consistency. The definitions are shown as follows:

Aggregated–level: Jensen–Shannon (JS) Divergence of Spatial Density. Let $p(\mathbf{s})$ and $q(\mathbf{s})$ be the normalized 2-D spatial density maps (kernel–smoothed or mesh–count histograms) of the real and generated trajectories, respectively. The JS divergence is the symmetrized and smoothed version of the Kullback–Leibler divergence:

$$\mathrm{JS}(p \parallel q) = \tfrac{1}{2}\,\mathrm{KL}\big(p \parallel \tfrac{p+q}{2}\big) + \tfrac{1}{2}\,\mathrm{KL}\big(q \parallel \tfrac{p+q}{2}\big)\,, \tag{5}$$

where $\mathrm{KL}(p\|m) = \sum_{\mathbf{s}} p(\mathbf{s}) \log \frac{p(\mathbf{s})}{m(\mathbf{s})}$. This metric measures how well the synthetic data reproduces the population-level geographical distribution of movements; lower values indicate closer global density.

Trajectory–level: Dynamic Time Warping (DTW). Given two trajectories $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$, where $a_i, b_j \in \mathbb{R}^2$ are latitude–longitude points, DTW finds an alignment path $\pi = \{(i_k, j_k)\}_{k=1}^{K}$ with $i_1 = j_1 = 1$ and $i_K = m$, $j_K = n$, that minimizes the cumulative Euclidean distance

$$\mathrm{DTW}(A, B) = \min_{\pi} \left\{ \sum_{k=1}^{K} \|a_{i_k} - b_{j_k}\|_2 \right\}. \tag{6}$$

The alignment allows non-linear warping in the time dimension, making DTW robust to local speed or sampling differences and highlighting fine-grained spatiotemporal fidelity.

Trajectory–level: Continuous Fréchet Distance. For the same curves A and B interpreted as continuous functions $\alpha, \beta : [0, 1] \to \mathbb{R}^2$, The Fréchet distance is

$$\mathrm{Fr}(A, B) = \inf_{\phi,\psi} \max_{t\in[0,1]} \big\|\alpha\big(\phi(t)\big) - \beta\big(\psi(t)\big)\big\|_2\,, \tag{7}$$

where ϕ and ψ range over all continuous, non-decreasing re-parameterizations of $[0, 1]$. Intuitively, it is the minimum leash length required for a person and a dog to walk along the two curves without backtracking. Unlike DTW, Fréchet does not explicitly align discrete time stamps; it captures overall geometric similarity and is sensitive to global path shape.

Summary of Usage. For every generated trajectory we compute DTW and Fréchet distances to its nearest ground-truth neighbor and report the **median**, **mean**, and **10th/90th percentiles (P10/P90)** to capture both typical accuracy and dispersion of errors across the distribution.

Fréchet distances capture geometric deviations, whereas DTW captures temporal misalignment. DTW explicitly aligns sequences in the time dimension - it penalizes trajectories that are spatially similar but temporally mismatched. Computing these metrics enables us to capture both geometry and temporal mismatching.

In addition, we report P10 and P90 alongside the median to capture the model's robustness across the long-tailed complexity of human mobility: human mobility is usually long-tail, reporting only the Median (Central Accuracy) may mask model failures on complex outliers. By reporting P10/P90 (extreme/long-tail case), we demonstrate that TrajFlow remains stable and accurate even for the long-tail part trajectories, rather than just fitting the easy majority.

Together, JS divergence evaluates aggregated-level spatial consistency, while DTW and Fréchet jointly measure individual-path fidelity from complementary temporal and geometric perspectives.

A.2 Data Compression Algorithms

A.2.1 Candidate Algorithms

We evaluate seven representative parameterization (data–harmonization) methods implemented in our codebase:

a) **direct_k**: Arc–length reparameterization followed by uniform sampling of K points along $[0, 1]$.

b) **dct**: Discrete Cosine Transform (DCT-II) of the arc–length–uniform curve for $x(t)$ and $y(t)$; keep the first DCT_M coefficients.

c) **rdp_k**: Ramer–Douglas–Peucker harmonization with a binary search on the tolerance so that the simplified polyline contains approximately K vertices; if necessary, linearly interpolate to exactly K points.

d) **anchor**: Least–squares spline fitting with automatically detected "anchors" (including endpoints) that receive a large weight w_{anchor} in the fitting.

e) **spline_lsq**: Least–squares spline fitting with uniform weights (no anchors).

f) **dct_deviation**: Take the straight line connecting start and end points as a baseline; apply DCT to the perpendicular deviation sequence and retain $2K - 4$ coefficients together with the endpoints.

g) **fft_complex**: Represent the curve as $z = x + iy$, apply FFT, and keep the first K complex coefficients.

The RDP algorithm simplifies a trajectory by recursively removing points that lie within a distance threshold ϵ of the line segment connecting their neighboring points, while preserving the essential geometric shape of the trajectory. Formally, given the ground-truth GPS trajectory dataset $\mathcal{X} = \{\mathrm{traj}_x^1, \mathrm{traj}_x^2, \ldots, \mathrm{traj}_x^m\}$, each trajectory $\mathrm{traj} = (l_0, l_1, \ldots, l_n)$ is processed as follows:

Algorithm 1 Ramer–Douglas–Peucker (RDP) Algorithm

```
function RDP($traj, \epsilon$)
    $d_{max} \leftarrow 0$
    $index \leftarrow 0$
    $end \leftarrow$ length($traj$) $- 1$
    for $i = 1$ to $end - 1$ do
        $d \leftarrow$ perpendicularDistance($traj[i]$, line($traj[0], traj[end]$))
        if $d > d_{max}$ then
            $index \leftarrow i$
            $d_{max} \leftarrow d$
        end if
    end for
    if $d_{max} > \epsilon$ then
        $results1 \leftarrow$ RDP($traj[0 \ldots index], \epsilon$)
        $results2 \leftarrow$ RDP($traj[index \ldots end], \epsilon$)
        return concatenate($results1[0 \cdots -1], results2$)
    else
        return $[traj[0], traj[end]]$
    end if
end function
```

A.2.2 Experiments with Different Parameters

To evaluate the influence of trajectory–harmonization granularity, we compared seven parameterization methods—`direct_k`, `dct`, `rdp_k`, `anchor`, `spline_lsq`, `dct_deviation`, and `fft_complex`—under four target point budgets ($K \in \{5, 10, 20, 30\}$) — while the original length is 120, representing compression ratios of approximately $\{4.2\%, 8.3\%, 16.7\%, 25.0\%\}$ respectively.

For each configuration we report the mean Dynamic Time Warping (DTW) and Fréchet distance (Table 2), and visualize qualitative differences in Figures. 6 & 7 & 8 & 9.

Overall, RDP-based simplification (`rdp_k`) consistently achieves the lowest trajectory-level errors, with average DTW decreasing from $\approx$ 2.77 at K=5 to $\approx$ 0.59 at K=30, and Fréchet distance dropping from $\approx$ 0.058 to $\approx$ 0.015. The closely related spline least–squares (`spline_lsq`) and DCT methods follow as the next best performers. Fourier–based `fft_complex` and deviation–only DCT are markedly worse, reflecting their sensitivity to high–frequency noise.

Increasing the point budget unsurprisingly improves accuracy for all algorithms, but the gains taper beyond K=20: DTW and Fréchet for `rdp_k` improve only marginally from K=20 to K=30. Visual inspection (Figures. 6 & 7 & 8 & 9) confirms that K=10, i.e., 16.7% compression rate, already captures the salient geometry of typical trajectories while preserving the strong compression benefits required for efficient training. Based on this analysis, we adopt RDP with K=10 as the default harmonization setting for TrajFlow, striking a practical balance between trajectory fidelity and computational cost.

We highlight two necessities of RDP in our tasks:

- RDP is used only as a dimensionality-reduction method. We compress each raw trajectory (often around 120 points) into a smaller set of keypoints (around 10). This compact representation greatly improves flow-matching training efficiency and stability, especially across multiple spatial scales.
- RDP provides the best balance between compression ratio and reconstruction accuracy. As shown in our ablation study, RDP achieves lower DTW and Fréchet distances than alternatives such as DCT, spline fitting, or FFT. This ensures that the simplified representation maintains high spatial fidelity before the interpolation step.

Table 2: Reconstruction Performance (Avg DTW/Fréchet, unit=km). Best (lowest) values in **bold**.

Method	Param 5		Param 10		Param 20		Param 30	
	DTW	Fréchet	DTW	Fréchet	DTW	Fréchet	DTW	Fréchet
direct_k	3.78	0.0908	1.54	0.0430	0.79	0.0205	0.66	0.0155
dct	3.23	0.0814	1.24	0.0326	0.69	0.0150	0.61	0.0137
rdp_k	**2.77**	**0.0580**	**0.95**	**0.0219**	**0.61**	0.0143	**0.59**	0.0150
anchor	16.42	0.3333	1.75	0.0405	0.76	0.0191	0.61	0.0138
spline_lsq	3.11	0.0665	1.22	0.0271	0.68	0.0150	0.61	**0.0136**
dct_deviation	4.66	0.0951	4.06	0.0874	4.15	0.0842	5.17	0.1025
fft_complex	28.32	0.4655	24.11	0.5120	25.59	0.6054	25.43	0.6363

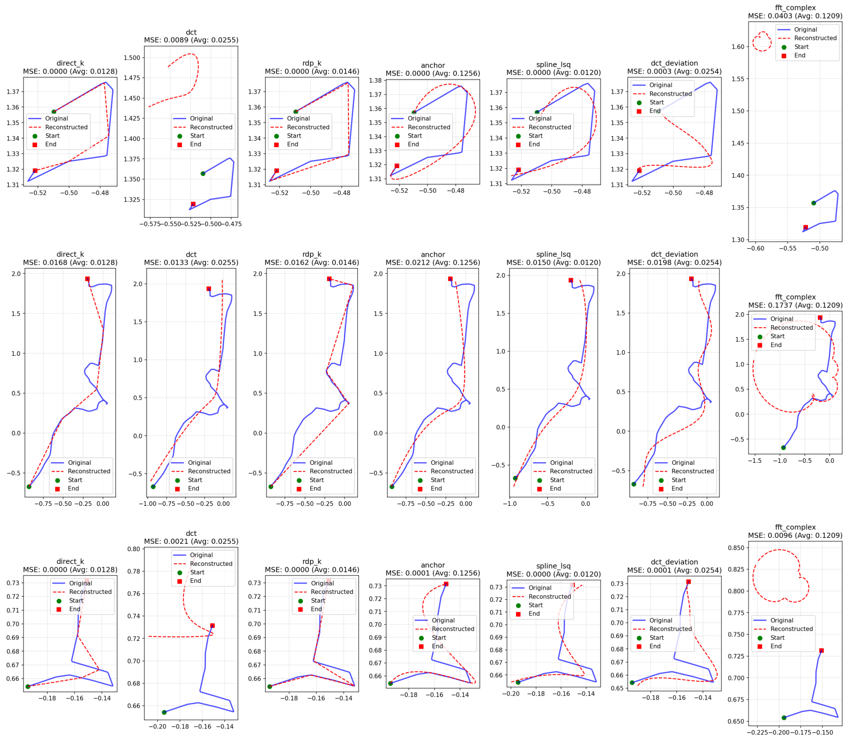


Figure 6: Parameter Number = 5

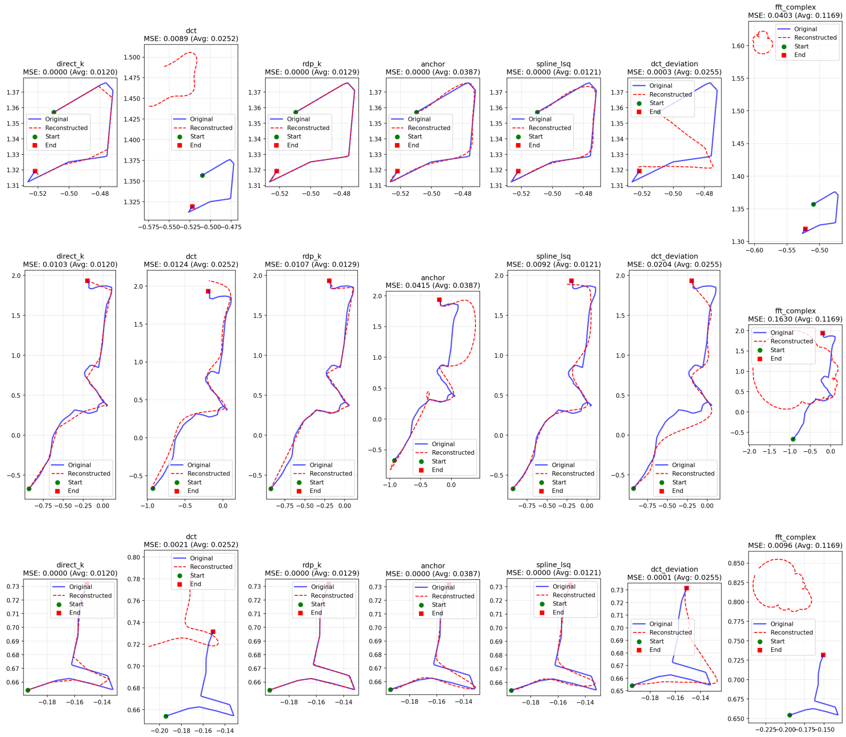


Figure 7: Parameter Number = 10

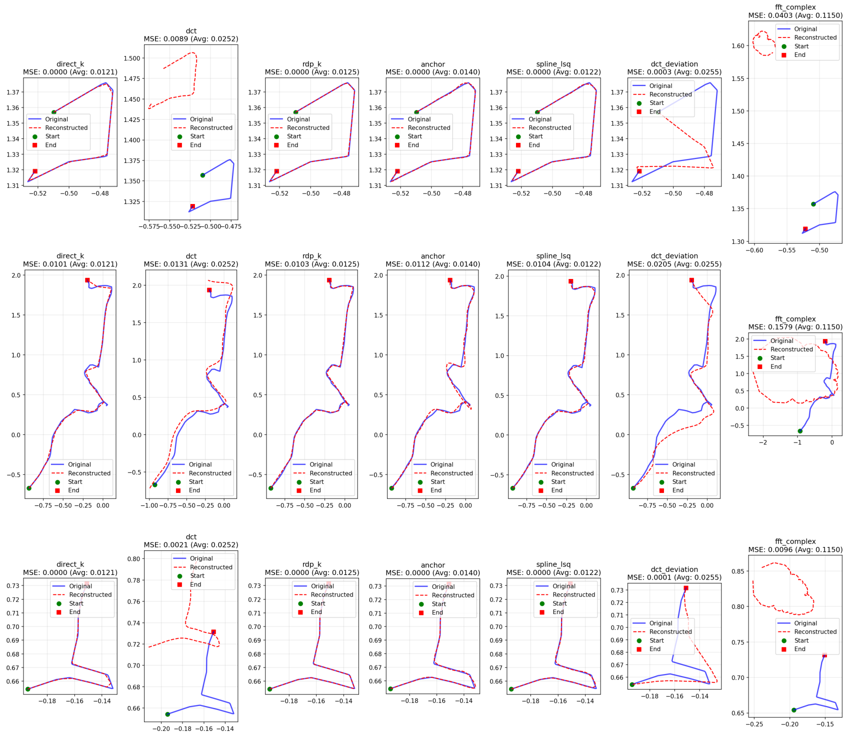


Figure 8: Parameter Number = 20

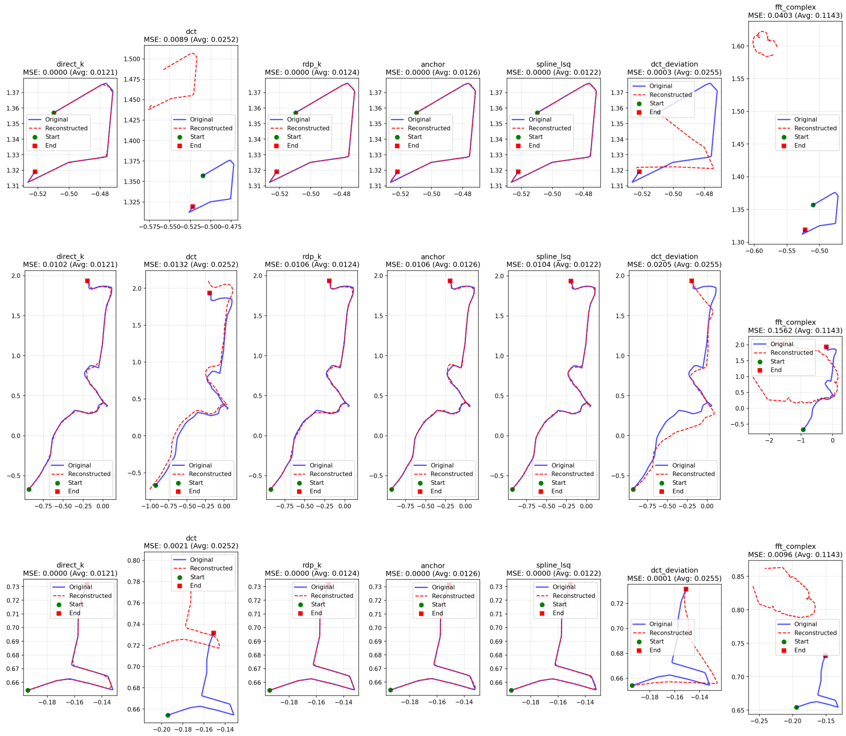


Figure 9: Parameter Number = 30

B Additional comparison with denoising diffusion models

One possible concern is that diffusion models might achieve comparable performance to *TrajFlow* if trained or sampled with a larger number of steps, rather than using the same limited steps as in flow matching. Our experiments confirm that increasing the number of steps indeed improves performance; for instance, with 300 steps, diffusion models approach the performance of the proposed method, although they remain slightly inferior (Table 3). Notably, while further increasing the number of steps might eventually surpass *TrajFlow*, this underscores the key advantage of our flow–matching–based approach: it achieves high fidelity while maintaining superior efficiency.

Table 3: TrajFlow-w/o-FM performance with different training/sampling steps (unit: km). Best metric in **bold**.

Method	Density JS ↓	DTW_{med} ↓	Fr_{med} ↓	DTW_{IQR} ↓	DTW_{P10} ↓	DTW_{P90} ↓	Fr_{IQR} ↓	Fr_{P10} ↓	Fr_{P90} ↓
Central Tokyo									
TrajFlow (ours)	**0.0674**	**20.350**	**0.304**	**13.392**	**10.574**	**39.119**	**0.174**	**0.200**	**0.674**
TrajFlow-w/o-FM (steps=10)	0.3213	362.611	5.844	794.886	86.305	2096.804	14.045	1.149	37.590
TrajFlow-w/o-FM (steps=50)	0.2923	263.889	4.390	694.411	64.512	1827.793	12.156	0.855	32.903
TrajFlow-w/o-FM (steps=100)	0.2269	142.836	2.363	303.280	38.252	749.346	5.244	0.519	12.811
TrajFlow-w/o-FM (steps=200)	0.1208	36.434	0.583	49.489	15.035	130.881	0.975	0.247	2.437
TrajFlow-w/o-FM (steps=300)	0.0807	24.303	0.349	19.773	11.768	57.728	0.331	0.211	1.049
Tokyo Metropolis									
TrajFlow (ours)	**0.1239**	**18.167**	**0.335**	**16.892**	**7.678**	**44.316**	**0.333**	**0.130**	**0.933**
TrajFlow-w/o-FM (steps=10)	0.3833	470.857	7.668	994.283	154.963	3179.940	17.724	2.069	54.807
TrajFlow-w/o-FM (steps=50)	0.3657	429.968	6.866	951.462	115.913	2935.618	16.754	1.555	50.237
TrajFlow-w/o-FM (steps=100)	0.3351	344.806	5.774	781.324	84.420	2006.533	13.825	1.118	36.338
TrajFlow-w/o-FM (steps=200)	0.2600	139.901	2.423	163.029	45.480	416.262	3.040	0.644	7.104
TrajFlow-w/o-FM (steps=300)	0.1690	33.035	0.546	34.456	11.806	84.754	0.653	0.186	1.527
Japan									
TrajFlow (ours)	**0.2270**	**10.977**	**0.192**	**18.221**	**3.984**	**55.964**	**0.361**	**0.072**	**1.119**
TrajFlow-w/o-FM (steps=10)	0.5808	495.961	7.901	969.032	166.495	3267.659	17.900	2.276	56.966
TrajFlow-w/o-FM (steps=50)	0.5632	367.991	5.821	738.327	117.033	2585.896	13.551	1.512	45.142
TrajFlow-w/o-FM (steps=100)	0.5220	210.646	3.344	407.453	70.677	1332.320	7.386	0.996	23.968
TrajFlow-w/o-FM (steps=200)	0.3856	55.417	0.847	88.626	19.797	263.442	1.619	0.311	4.622
TrajFlow-w/o-FM (steps=300)	0.3047	26.125	0.412	36.912	9.376	113.981	0.700	0.151	2.094

Beyond the accuracy performance, we also compared the inference efficiency (time cost) between the proposed method and diffusion-based version (TrajFlow-w/o-FM). As Figure. 10 and Table. 4 shows, the flow matching-based approach has a much higher efficiency than the diffusion-based approach. Specifically, the diffusion-based version requires over 30× more inference time-cost while achieving accuracy levels that are comparable yet still lower than TrajFlow.

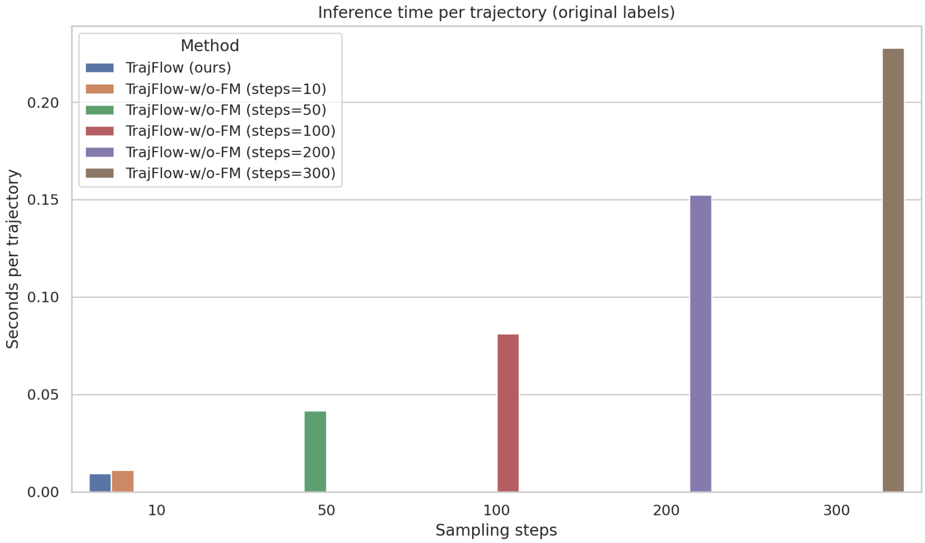


Figure 10: Bar plot for time cost comparison in inference stage.

Table 4: Comparison table for time cost in inference stage.

Region	Method	Steps	Sec/Sample	Sampling Sec	Total Sec
Central Tokyo	TrajFlow	10	0.0097	0.9696	4.8912
Central Tokyo	TrajFlow-w/o-FM (steps=10)	10	0.0100	1.0039	4.9646
Central Tokyo	TrajFlow-w/o-FM (steps=50)	50	0.0382	3.8235	7.7105
Central Tokyo	TrajFlow-w/o-FM (steps=100)	100	0.0761	7.6105	11.5326
Central Tokyo	TrajFlow-w/o-FM (steps=200)	200	0.1585	15.8546	19.8442
Central Tokyo	TrajFlow-w/o-FM (steps=300)	300	0.2237	22.3682	26.3010
Tokyo Metropolis	TrajFlow	10	0.0097	0.9735	4.8695
Tokyo Metropolis	TrajFlow-w/o-FM (steps=10)	10	0.0135	1.3465	5.7994
Tokyo Metropolis	TrajFlow-w/o-FM (steps=50)	50	0.0474	4.7377	8.6324
Tokyo Metropolis	TrajFlow-w/o-FM (steps=100)	100	0.0816	8.1608	12.1447
Tokyo Metropolis	TrajFlow-w/o-FM (steps=200)	200	0.1474	14.7445	18.6668
Tokyo Metropolis	TrajFlow-w/o-FM (steps=300)	300	0.2356	23.5593	27.5961
Japan	TrajFlow	10	0.0086	0.8575	4.7650
Japan	TrajFlow-w/o-FM (steps=10)	10	0.0095	0.9466	4.8438
Japan	TrajFlow-w/o-FM (steps=50)	50	0.0391	3.9072	7.8493
Japan	TrajFlow-w/o-FM (steps=100)	100	0.0860	8.6031	12.5301
Japan	TrajFlow-w/o-FM (steps=200)	200	0.1515	15.1521	19.0516
Japan	TrajFlow-w/o-FM (steps=300)	300	0.2240	22.4007	26.6460

C TRANSPORTATION-MODE DIVERSITY

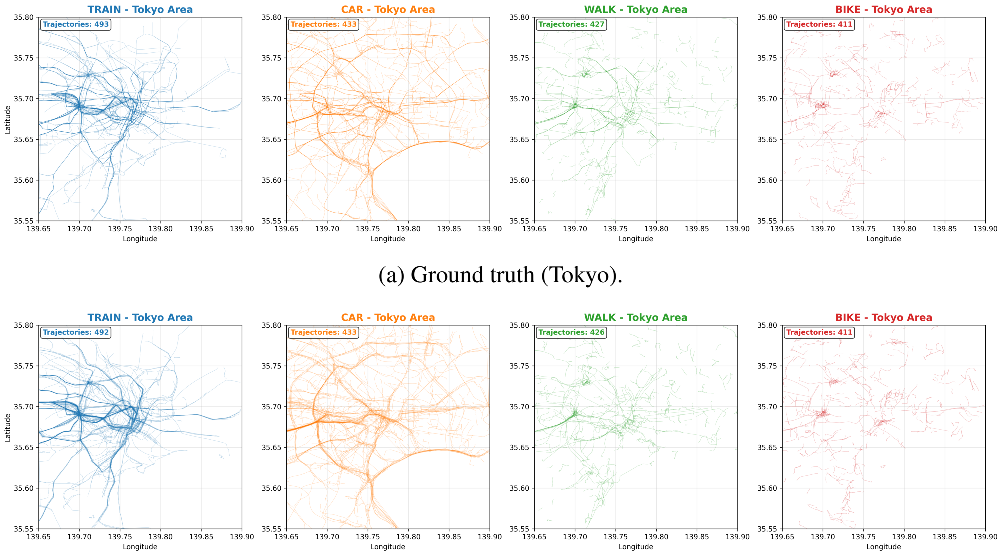
Figures 11 and Figure 12 compare the generated trajectories of our model with the ground truth in four modes of transport. Mode diversity is much more pronounced within Tokyo, where travelers frequently switch among walking, cycling, buses, subways, private cars, and multiple rail systems. In contrast, on nationwide trips, mobility patterns are dominated by a much smaller set of modes (primarily long-distance trains), while local segments are typically sparse or not captured at the same granularity.

C.1 URBAN-WIDE(TOKYO)

In the ground-truth data, distinct patterns emerge: train trips typically span larger spatial scales and extend to the outskirts of Tokyo; car trips are also long but more concentrated in central areas; bike and walk trips both cluster within local communities, though with different levels of continuity. These diverse modal behaviors are well reproduced by our model, demonstrating its ability to capture transportation-mode heterogeneity.

C.2 NATION-WIDE (JAPAN)

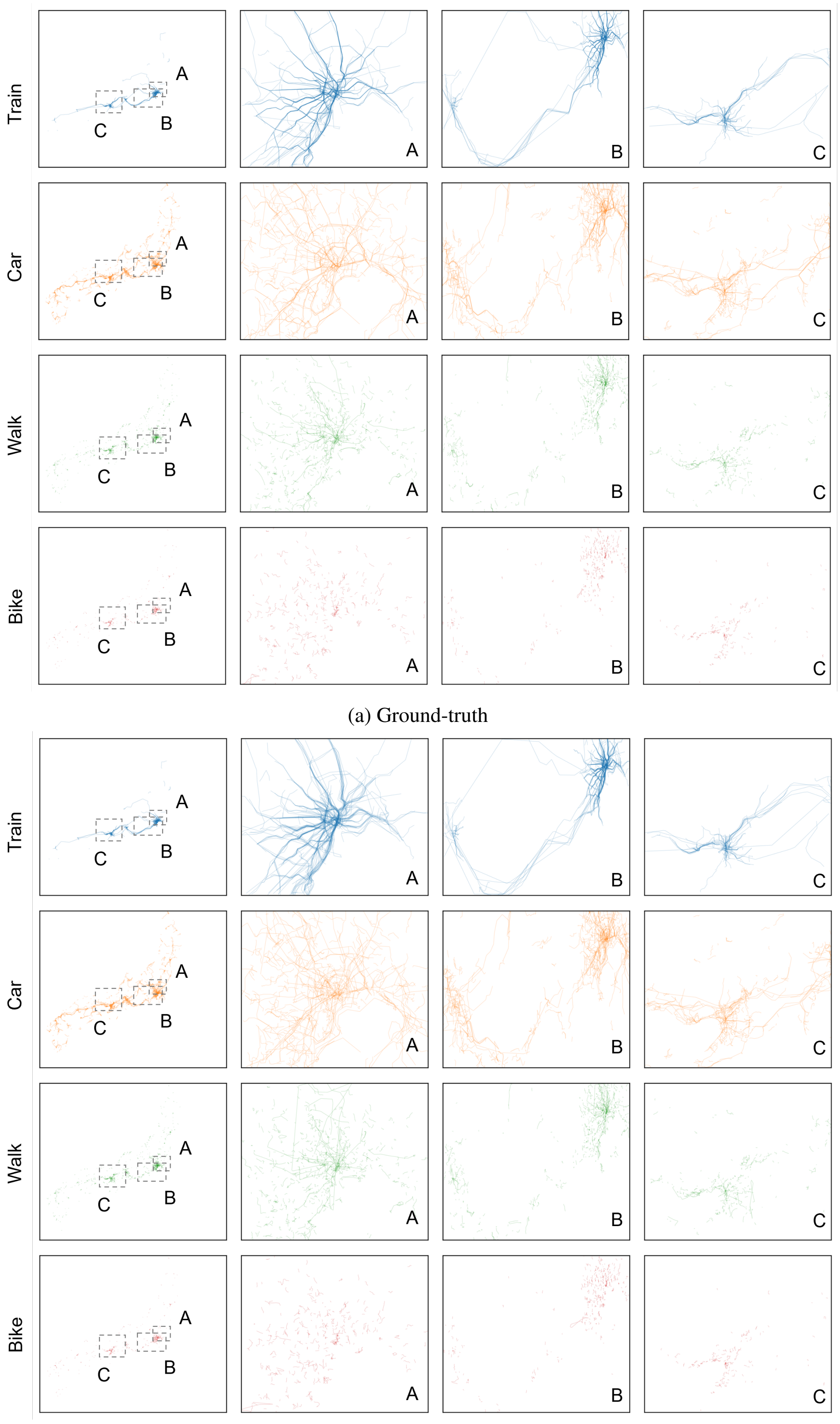
In the ground-truth data, we find that the train trajectories show more concentrate which is reasonable - the choices for long-distance railway network is limited, and car trajectories show more diverse routes. Most walk and bike trajectories are limited in local area.



(a) Ground truth (Tokyo).

(b) Generated (Tokyo).

Figure 11: Transportation-mode diversity visualization: GT vs. Gen in Tokyo.



(a) Ground-truth

(b) Generated

Figure 12: The capability to generate GPS trajectories under given transportation mode conditions. Ground-truth nation-wide trajectories shown with zoomed views highlighting three representative regions: (A) Tokyo Metropolis, (B) Tokaido Area, and (C) Kansai Area, across various transportation modes: Train, Car, Walk, Bike **(a)** shows the ground-truth trajectories. **(b)** shows the generated.

D Reproducibility

D.1 Dataset Information

In accordance with the data policy of Blogwatcher Inc., the complete ground-truth BlogWatcher dataset can only be accessed through a strict application process. Although the dataset used in our study originates from a single data provider, it is sourced from over 140 independent mobile applications (as noted on https://www.blogwatcher.co.jp/), ensuring substantial diversity in data collection. Because comparable multi-scale datasets are not publicly available, we are unable to benchmark TrajFlow on alternative nationwide or multi-level datasets. The dataset we used spans the period from January 2023 to December 2023 and contains approximately 3,000,000 (3 million) trajectories across Japan. For our experiments, we derive three subsets of different spatial scales: (1) Tokyo City, (2) Tokyo Metropolis, and (3) Nationwide.

uid	segment_id	trans_mode1	trans_mode2	time	lat	lon
…	1	STAY	STAY	2023/2/6 0:00	35.9497	139.5576
…	1	STAY	STAY	2023/2/6 0:15	35.9497	139.5576
…	…	…	…	…	…	…
…	2	MOVE	TRAIN	2023/2/6 0:36	36.2269	114.1721
…	…	…	…	…	…	…
…	3	MOVE	WALK	2023/2/6 0:59	36.7236	114.4598

Table 5: Example rows of the dataset we used. (Note: this example data is only for showing data format without real lat/lon values.)

Each trajectory is pre-segmented according to transportation mode. The data fields relevant to this study are summarized in Table 5. Here, `uid` denotes an anonymized user identifier; however, we do not utilize any user-level identifiers in this work to avoid potential privacy concerns. The field `segment_id` indexes individual trajectory segments; `trans_mode1` distinguishes between `STAY` and `MOVE` states, and we use only the `MOVE` segments. The field `trans_mode2` provides fine-grained transportation modes such as `TRAIN`, `CAR`, `WALK`, and `BIKE`. Within movement segments, GPS points are recorded at an average interval of at least one minute.

To validate our method in open source dataset, we additionally evaluate on two open-source datasets in Chengdu and Xi'an, see Section E in the Appendix. In addition, a portion of the processed demo data, together with the implementation and generated results, will be made publicly available upon acceptance of this work.

D.2 Model Architecture

Backbone: We employ a U-Net Architecture as the vector field estimator.

- **Dimensions:** Base channels = 128. Channel multipliers = [1, 2, 2, 2].
- **Structure:** Two residual blocks per stage with SiLU activation. Attention mechanisms are applied at resolution 16.
- **Embeddings:** Timestep embedding dimension is 512 (4× base channels). Hidden dimension for condition embeddings is 512.

Conditioning (Wide & Deep): We utilize a ”Wide & Deep” architecture to encode heterogeneous inputs:

- **Wide Component:** Linear projection of continuous features (e.g., speed, distance).
- **Deep Component:** Categorical embeddings for Departure Time (288 bins), Origin/Destination IDs, and Transportation Modes (5 classes).
- **Fusion:** The outputs are fused into a 128-dimensional condition vector, which is injected into the U-Net residual blocks via cross-attention or additive projection.

D.3 Training Configuration

- **Optimizer:** Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) with no weight decay.

- **Learning Rate:** Initial learning rate set to 1×10^{-4}. We use a `ReduceLROnPlateau` scheduler (factor=0.5, patience=200) monitoring validation loss.
- **Batch Size:** 256 trajectories per GPU.
- **Training Budget:** Up to 500 epochs with Early Stopping (patience=5000 steps, delta=$1e^{-6}$).
- **Regularization:** We apply random condition dropout (Classifier-Free Guidance) during training to prevent overfitting and enable guidance during inference.

D.4 Algorithm Settings

- **Trajectory Harmonization (RDP):** We use the Ramer-Douglas-Peucker (RDP) algorithm for compression.
  - Target keypoints $M = 10$.
  - Epsilon tolerance determined via binary search ($\epsilon_{tol} = 1e^{-5}$).
  - Max sequence length $L_{max} = 120$.
- **Flow Matching Solver:**
  - **Method:** Euler ODE integrator.
  - **Inference Steps:** 10 steps (step size $= 0.1$). This setting was chosen to balance fidelity and efficiency (approx. 0.01s per sample).
- **Diffusion Baseline (for comparison):** Linear β schedule ($1e^{-6}$ to $5e^{-2}$). DDIM sampler used with steps ranging from 10 to 300 for efficiency benchmarking.

D.5 Computing Environment

All experiments were implemented in **PyTorch** and executed on a cluster of four **NVIDIA RTX 6000 Ada GPUs** (40GB memory each). The software environment includes CUDA 12.6 and Python 3.9.

E Validation on open-source data

We report the results based on 1,000,000 trajectory samples for each city. For reference, the previous *DiffTraj* paper reported using 3,493,918 trajectories from Chengdu and 2,180,348 trajectories from Xi'an. Across both cities, *TrajFlow* maintains strong performance and exhibits trends consistent with our main results (see Table. 6).

Table 6: Evaluation grouped by region (unit = km). Best metric in **bold**.

Method	Density JS ↓	DTW_{med} ↓	Fr_{med} ↓	DTW_{IQR} ↓	DTW_{P10} ↓	DTW_{P90} ↓	Fr_{IQR} ↓	Fr_{P10} ↓	Fr_{P90} ↓
Chengdu									
TrajFlow (ours)	0.211	35.911	0.547	29.418	15.839	72.048	0.392	0.279	0.959
TrajFlow-w/o-OD	0.0809	3.696	0.123	2.847	2.180	8.904	0.117	0.0614	0.362
TrajFlow-w/o RDP	0.206	34.687	0.540	30.020	14.650	69.999	0.383	0.266	0.957
TrajFlow-w/o RDP & OD	**0.0140**	**2.158**	**0.0525**	**0.896**	**1.519**	**3.278**	**0.0243**	**0.0354**	**0.0880**
TrajFlow-w/o Flow	0.208	36.457	0.561	30.017	15.948	73.523	0.397	0.278	0.975
TrajFlow-w/o OD & Flow	0.0811	3.724	0.123	2.800	2.142	8.844	0.119	0.0618	0.365
TrajFlow-w/o RDP & Flow	0.203	36.166	0.561	30.311	15.213	73.719	0.388	0.276	0.985
DiffTraj (baseline)	0.0224	3.524	0.0944	1.563	2.312	5.460	0.0482	0.0606	0.161
XiAn									
TrajFlow (ours)	0.284	43.457	0.564	35.390	16.948	79.586	0.379	0.277	0.897
TrajFlow-w/o-OD	0.0820	2.565	0.0842	2.009	1.530	8.488	0.116	0.0453	0.395
TrajFlow-w/o RDP	0.278	40.859	0.549	34.013	15.547	75.445	0.380	0.259	0.875
TrajFlow-w/o RDP & OD	**0.0056**	**1.111**	**0.0279**	**0.401**	0.805	**1.611**	**0.0120**	**0.0192**	**0.0443**
TrajFlow-w/o Flow	0.284	43.572	0.567	35.941	17.259	79.985	0.379	0.276	0.895
TrajFlow-w/o OD & Flow	0.0819	2.354	0.0833	2.115	1.387	8.162	0.119	0.0448	0.395
TrajFlow-w/o RDP & Flow	0.278	40.328	0.555	33.247	15.592	75.524	0.382	0.257	0.872
DiffTraj (baseline)	0.0070	1.154	0.0284	0.494	**0.773**	1.750	0.0132	0.0193	0.0466

F Distinction from existing approaches

Our work introduces several key contributions that differentiate TrajFlow from existing trajectory-generation approaches:

- **First flow-matching model for GPS trajectory generation (methodological novelty).** Prior trajectory-generation methods—including recent diffusion-based models—rely on stochastic reverse-time SDE sampling and require tens to hundreds of denoising steps. In contrast, TrajFlow is the first model to apply the flow-matching paradigm to GPS trajectory generation. Moreover, we introduce a trajectory normalization scheme and a specialized architecture tailored for flow matching, enabling the model to better handle spatial heterogeneity and further improve generation performance.
- **First nationwide-scale, multi-geospatial-level GPS generator (problem-level novelty).** Existing models are restricted to small urban or single-city settings due to spatial heterogeneity and instability. TrajFlow is the first model evaluated at urban, metropolitan, and nationwide scales, trained on millions of mobile-phone GPS trajectories across Japan, demonstrating robust generalization across heterogeneous regions.

G Practical application

TrajFlow handles variable-length trajectories through a conditioning–reconstruction strategy rather than asking the generative model to infer sequence length implicitly.

- Fixed-length representation during training (Section 4.4). For stable batch training under flow matching, all trajectories are represented using a fixed maximum length with padding and validity masks. This allows the model to learn continuous spatial dynamics without being affected by sequence-length variability.
- Explicit duration conditioning (Section 4.3). The model is conditioned on Travel Time and Departure Time, which provide explicit temporal context. Because trajectory duration is given as a conditioning variable, the model does not need to guess or estimate the temporal length.
- Length-consistent reconstruction at inference. At generation time, the model predicts the spatial shape of the trajectory (via harmonized RDP points). The final trajectory is then reconstructed to the target duration using the provided Travel Time condition, ensuring that generated samples match the desired temporal length—whether 10 minutes or 2 hours.

This strategy enables TrajFlow to robustly generate trajectories with diverse and accurate temporal lengths while retaining a stable training process.

H Privacy issue

The inputs to our model are limited to departure time, origin–destination (OD) zones, and transportation mode, all of which are high-level, aggregated attributes that do not include or reveal any user-level identifiers (e.g., user IDs, device IDs, or fine-grained personal metadata). Therefore, the model is not exposed to information that could directly compromise individual privacy.

I Memorization risks

TrajFlow is designed to learn underlying mobility patterns (e.g., road network constraints, route choice) rather than memorizing specific coordinate sequences. Similar to diffusion/flow-matching models in computer vision—which learn the concept of an object rather than copying specific training images—our model generates trajectories by transforming Gaussian noise under OD and time conditions. The flow-matching process injects noise and learns a continuous probability flow from noise to data, making the generation inherently stochastic. As a result, each sample starts from a different noise seed, and even with identical OD/time conditions, the model produces diverse and novel trajectories rather than retrieving or replicating any stored instance. This stochasticity, as in models like Stable Diffusion, ensures diversity and prevents deterministic copying of training trajectories.

In addition, to explicitly prevent the model from ”memorizing” specific conditional mappings (overfitting), we apply Classifier-Free Guidance (CFG) and dropout trick, which consists of a training regularization and an inference mechanism: Training (Condition Dropout): During training, we randomly mask the input conditions (OD/Time/Mode/etc.,) with a probability. This acts as a strong regularizer, forcing the model to learn the general, unconditional distribution of human mobility rather than relying on specific conditions to retrieve stored instances. Inference (Guidance Formula): During generation/inference, we compute a guided update using a weighted linear combination of the conditional and unconditional vector fields. This is the standard classifier-free guidance formula, which increases the model’s sensitivity to the conditioning signal without causing the model to memorize specific condition–trajectory pairs. By adjusting the guidance weight, we can strengthen or relax condition adherence, enabling higher-quality and non-deterministic trajectory samples under the same conditions.

Third, empirically, we observed that the minimum DTW distances between generated samples and the training set are consistently non-zero. And even with the same condition, various routes could be generated.

J Ethics Statement

This work adheres to the ICLR Code of Ethics. All experiments were conducted on anonymized, large-scale mobile phone GPS trajectory data, which was used strictly in aggregated form and under strict privacy rules. No personally identifiable information (PII) or user-level attributes (e.g., age, gender, home–work identifiers, or persistent pseudonymous IDs) were accessed or utilized. Consequently, the model does not attempt to capture or infer individual preferences.

The proposed methods are designed to generate pseudo-GPS trajectories for research on mobility modeling and transportation systems, not to reconstruct or deanonymize individual user behavior. The use of RDP-based harmonization and OD-prediction modules is focused on improving computational efficiency and trajectory-level fidelity, without compromising privacy.

We acknowledge that trajectory data can be sensitive, and inappropriate applications may raise concerns around surveillance or discriminatory use. To mitigate this, we limit our study to methodological contributions and evaluation on aggregated data. Our scope is restricted to GPS trajectory generation (sequences of locations over time), rather than full human mobility modeling that would include activity semantics, trip-chain structure, or purpose-specific constraints.

We believe this research can benefit society by enabling scalable simulation tools for urban planning, transportation analysis, and disaster response, while respecting user privacy. All legal, ethical, and research integrity requirements have been followed in the preparation of this work.