

Multi-Scale Diffusion Transformer for Jointly Simulating User Mobility and Mobile Traffic Pattern

Ziyi Liu, Qingyue Long, Zhiwen Xue, Huandong Wang, *Member, IEEE*, Yong Li, *Senior Member, IEEE*

Abstract—User mobility trajectory and mobile traffic data are essential for a wide spectrum of applications including urban planning, network optimization, and emergency management. However, large-scale and fine-grained mobility data remains difficult to obtain due to privacy concerns and collection costs, making it essential to simulate realistic mobility and traffic patterns. User trajectories and mobile traffic are fundamentally coupled, reflecting both physical mobility and cyber behavior in urban environments. Despite this strong interdependence, existing studies often model them separately, limiting the ability to capture cross-modal dynamics. Therefore, a unified framework is crucial. In this paper, we propose MSTDiff, a Multi-Scale Diffusion Transformer for joint simulation of mobile traffic and user trajectories. First, MSTDiff applies discrete wavelet transforms for multi-resolution traffic decomposition. Second, it uses a hybrid denoising network to process continuous traffic volumes and discrete location sequences. A transition mechanism based on urban knowledge graph embedding similarity is designed to guide semantically informed trajectory generation. Finally, a multi-scale Transformer with cross-attention captures dependencies between trajectories and traffic. Experiments show that MSTDiff surpasses state-of-the-art baselines in traffic and trajectory generation tasks, reducing Jensen-Shannon divergence (JSD) across key statistical metrics by up to 17.38% for traffic generation, and by an average of 39.53% for trajectory generation. The source code is available at: <https://github.com/tsinghua-fib-lab/MSTDiff>.

Index Terms—Cellular traffic, Mobility trajectory, Diffusion models.

I. INTRODUCTION

The widespread deployment of cellular networks and the proliferation of smartphones and IoT devices have greatly increased the scale of mobility data. This growth has accelerated mobile data mining, yielding valuable applications in intelligent transportation, urban planning, and network optimization [1], [2].

However, obtaining large-scale and fine-grained mobility data remains a significant challenge due to privacy concerns, data collection costs, and regulatory constraints. This limitation hinders the progress of many downstream tasks such as trajectory prediction, traffic forecasting, and network resource management. In this context, realistic and controllable data generation methods become increasingly important. Notably,

Ziyi Liu, Qingyue Long, Huandong Wang, and Yong Li are with the Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China. (e-mail: liuziyi24@mails.tsinghua.edu.cn; longqy21@mails.tsinghua.edu.cn; wanghuandong@tsinghua.edu.cn; liyong07@tsinghua.edu.cn)

Zhiwen Xue is with the International School, Beijing University of Posts and Telecommunications, Beijing 100876, China. (e-mail: xzwinner@bupt.edu.cn)

user trajectories and mobile traffic are inherently coupled, capturing both the physical mobility and cyber behavior of individuals. Modeling their joint distribution is critical to support realistic simulation, forecasting, and robust decision-making in mobility-aware systems. However, existing studies often treat trajectory and traffic generation as separate tasks, lacking a unified framework that captures their interdependence.

Recently, diffusion models have achieved state-of-the-art performance across various generative tasks, including image and video generation [3]–[5] and time series modeling [6], owing to their stability and sample quality. In particular, integrating diffusion models with Transformer architectures [7] enables efficient modeling of long-range temporal dependencies, making them well-suited for complex spatiotemporal data generation tasks.

Despite the potential of generative modeling, joint simulation of user trajectories and traffic remains challenging in the following three aspects:

- **Mobile user traffic exhibits periodic and aperiodic characteristics.** Mobile user traffic is highly stochastic, with substantial variation across individuals. While users often follow daily routines that result in periodic data usage patterns, their mobile traffic is also shaped by individual-level randomness and external events. These irregularities differ widely in both their duration and intensity, leading to bursts of traffic usage at varying temporal scales. This makes it particularly challenging to capture the coexisting periodic and aperiodic dynamics.
- **Heterogeneous modeling for mobile user traffic and trajectory.** Previous work on diffusion-based trajectory generation has primarily focused on continuous GPS coordinates [8], [9], thereby overlooking the semantic information of locations, which can be categorized into functional types, with transitions between them reflecting meaningful user behavioral patterns. Applying Gaussian noise in continuous space risks distorting these semantic categories. In contrast, traffic data is inherently a continuous temporal sequence. This discrepancy raises challenges in designing a unified generative model to align and jointly sample across discrete and continuous modalities.
- **Complex interactions between mobile user traffic and trajectory.** Mobile user traffic and trajectories are closely related, with both static and dynamic dependencies. On the one hand, the volume of data usage is influenced by geographic context, as the types of apps used often vary depending on the user’s physical location. As users move across different regions, their data consumption

changes accordingly. On the other hand, even when users remain at the same location, their traffic patterns can fluctuate significantly due to switching between different apps within short time intervals. This occurs at a finer temporal resolution than user movement, making it challenging to capture dependencies between traffic data and trajectories.

In this work, we propose **MSTDiff**, a Multi- Scale Transformer-based Diffusion model for jointly generating user mobile traffic and user trajectories. To address the first challenge, we apply discrete wavelet transforms to the traffic data to enable multi-resolution modeling and representing, allowing the model to capture bursty patterns at varying temporal scales. Building on this decomposition, we introduce a multi-scale Transformer based on the cross-attention mechanism to capture the temporal correlations between traffic and trajectory across different resolutions. Finally, we develop a hybrid denoising framework to jointly model the denoising processes for both continuous traffic and discrete trajectory data. Specifically, we incorporate urban knowledge graph embeddings to encode the semantic information of locations and design a similarity-based transition matrix that guides the discrete diffusion process.

In conclusion, the main contributions of our work are summarized as follows:

- We propose a unified diffusion-based framework that jointly generates continuous traffic data and discrete trajectory sequences.
- We employ wavelet transforms and multi-scale attention mechanisms to model interactions between traffic and trajectories at different resolutions, and design a knowledge-guided discrete diffusion process for trajectory generation.
- We conduct experiments on a large scale real world mobility dataset and show that our model achieves up to 17.38% improvement in Jensen–Shannon divergence on traffic data and an average improvement of 39.53% on the trajectory task.

The conference version of this work is accepted as a short paper at SIGSPATIAL 2025 [10]. In this paper we add trajectory generation experiments with results and visualizations of evaluation statistics, update the loss by adding a cross entropy prediction term for the trajectory branch, and introduce the training and sampling algorithms for the co-denoising process.

II. PRELIMINARIES AND PROBLEM DEFINITION

A. Continuous Diffusion Models

Continuous diffusion models are particularly well-suited for modeling real-valued data, such as the temporal sequences. Based on the forward and reverse processes, diffusion models [11] simulate data distribution by first perturbing clean data with noise and then learning to reverse this corruption to recover the original distribution. For an input space $X \subseteq \mathbb{R}^D$, we consider a data point $x_0 \in X$ sampled from a distribution $q(x_0)$. The forward process produces a sequence of noisy variables $\{x_1, x_2, \dots, x_S\}$ by gradually perturbing x_0 with

noise. The model is trained to learn a distribution $p_\theta(x_0)$ that closely resembles $q(x_0)$.

Forward process. In the forward process, noise is gradually added to the data step by step. This process follows a Markov chain and is defined as:

$$q(x_s | x_{s-1}) = \mathcal{N}(\sqrt{1 - \beta_s} x_{s-1}, \beta_s \mathbf{I}), \quad (1)$$

where β_s is a small number that slowly increases with each step s , meaning the noise becomes stronger over time and less initial information is retained. Define $\alpha_s = 1 - \beta_s$ and $\bar{\alpha}_s = \prod_{i=1}^s \alpha_i$. Then, the noisy sample x_s can be written as: $x_s = \sqrt{\bar{\alpha}_s} x_0 + \sqrt{1 - \bar{\alpha}_s} \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

Reverse process. The reverse process begins from a fully noisy input $x_S \sim \mathcal{N}(0, \mathbf{I})$ and aims to progressively recover the clean data. The denoising step is modeled as:

$$p_\theta(x_{s-1} | x_s) = \mathcal{N}(x_{s-1}; \mu_\theta(x_s, s), \sigma_\theta(x_s, s)\mathbf{I}). \quad (2)$$

The model parameters θ are learned by minimizing a variational upper bound on the negative log-likelihood. Ho et al. [11] reformulated this optimization into a simpler form, where the objective is to minimize the Mean Squared Error (MSE) between the predicted noise $\epsilon_\theta(x_s, s)$ and the true noise ϵ :

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\theta} \mathbb{E}_{x_0 \sim q(x_0), \epsilon \sim \mathcal{N}(0, \mathbf{I}), s} \|\epsilon - \epsilon_\theta(x_s, s)\|_2^2. \quad (3)$$

B. Discrete Diffusion Models

Discrete diffusion models [12], [13] are designed to handle data with inherently discrete structures, such as categorical variables and text tokens. Both continuous and discrete diffusion models involve a forward and a reverse process, but differ in their implementation details. Unlike continuous diffusion models which add Gaussian noise to real-valued data, discrete diffusion models corrupt the data through stochastic transitions between discrete states, which gradually transforms the data toward a uniform distribution over the state space. As discussed by Austin et al. [13], several discrete corruption schemes have been proposed, including discrete Gaussian noise, uniform transition probabilities, and similarity-based transition matrices constructed from embedding spaces. In this work, we focus on transition-matrix-based corruption, motivated by the observation that locations and words share similar characteristics. Both are discrete variables that can be represented in embedding spaces that preserve their semantic meanings.

Forward process. In a discrete diffusion process with a total of S steps, the one-step noise addition at step s is modeled as a single-step Markov transition governed by a transition matrix $Q_s \in \mathbb{R}^{N \times N}$, where N is the number of discrete categories. Each transition probability $q(x_s | x_{s-1})$ is a categorical distribution whose parameters are specified by the corresponding row of Q_s . Assuming x_{s-1} is represented as a one-hot vector, the forward process can be defined as:

$$q(x_s | x_{s-1}) = \text{Cat}(x_s; \mathbf{p} = x_{s-1} Q_s), \quad (4)$$

$$q(x_s | x_0) = \text{Cat}(x_s; \mathbf{p} = x_0 \bar{Q}_s), \quad (5)$$

where $\bar{Q}_s = Q_1 Q_2 \cdots Q_s$.

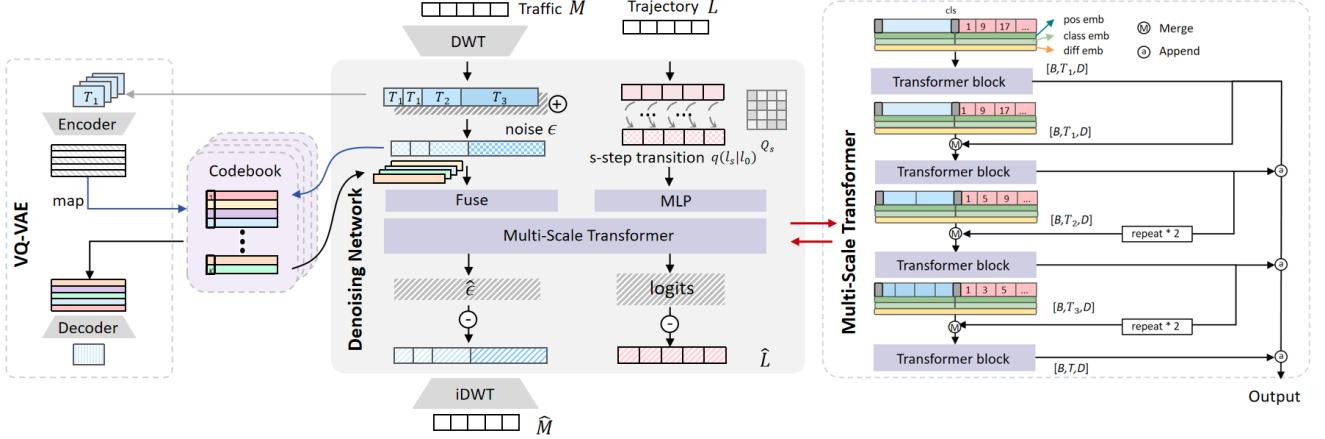


Fig. 1. The Framework of MSTDiff.

Reverse process. The reverse process aims to recover the original data by gradually denoising the corrupted input through a sequence of learned transitions. Instead of directly predicting the distribution $p_\theta(x_{s-1} | x_s)$, the model is trained to approximate the posterior distribution over the original state x_0 , given a noisy input x_s . The neural network in the denoising network outputs a parameterized distribution $p_\theta(x_0 | x_s)$, typically in the form of logits over all discrete categories. The reverse transition $p_\theta(x_{s-1} | x_s)$ can then be computed as follows:

$$p_\theta(x_{s-1} | x_s) = \sum_{\hat{x}_0} q(x_{s-1} | x_s, \hat{x}_0) p_\theta(\hat{x}_0 | x_s). \quad (6)$$

To train the denoising model, discrete diffusion models typically minimize the Kullback-Leibler (KL) divergence between the true reverse posterior $q(x_{s-1} | x_s, x_0)$ and the model's predicted reverse transition $p_\theta(x_{s-1} | x_s)$. The loss function is defined as:

$$\text{D}_{KL}(q(x_{s-1} | x_s, x_0) || p_\theta(x_{s-1} | x_s)). \quad (7)$$

C. Problem Definition

Given a set of mobile user traffic sequences $M \in \mathbb{R}^{U \times T}$ and corresponding trajectory sequences $L \in \{1, \dots, N\}^{U \times T}$, where U denotes the number of users, T is the total number of time steps, and N is the number of possible spatial locations, our goal is to train a unified model F_θ that jointly simulates both traffic data \hat{M} and discrete trajectories \hat{L} . Therefore, the joint generative process is defined as:

$$\hat{M}, \hat{L} = F_\theta(\epsilon_S, l_S). \quad (8)$$

III. METHODS

The framework of MSTDiff is illustrated in Figure 1, which consists of three main modules: a traffic representation module based on discrete wavelet transform and a pre-trained Vector Quantized Variational Autoencoder (VQ-VAE), a trajectory modeling module that constructs a transition mechanism using an urban knowledge graph, and a multi-scale Transformer denoising network for the joint generation of continuous traffic data and discrete trajectories.

A. Wavelet-VQ Module for Traffic Representation

1) Wavelet Transform: The Discrete Wavelet Transform (DWT) enables multi-resolution analysis through adaptable lengths of basis functions, making it well-suited for traffic sequence that exhibit strong non-stationary and multi-scale temporal dynamics.

Given the original traffic sequence m_0 , we apply a three-level DWT to obtain the coefficient sets $\{CA_3, CD_3, CD_2, CD_1\}$, where CA_j and CD_j denote approximation and detail coefficients, respectively. DWT applies low- and high-pass filters at each level with downsampling by a factor of 2. With the Daubechies 1 (db1) wavelet, each level's coefficients are exactly half the length of the previous, and their total length equals the original sequence length T . We then concatenate these coefficients along the last dimension to form a unified multi-scale representation w_0 .

2) Pre-training VQ-VAE: The VQ-VAE is applied to each coefficient set separately to discretize the wavelet coefficients, enabling the capture of high-level traffic patterns with greater stability than continuous latent representations. The encoder transforms the input traffic sequence w into a continuous latent representation z_e . The vector quantization module maps each vector in z_e to its nearest codebook entry from $\mathcal{Z} = \{e_1, e_2, \dots, e_K\}$ to obtain z_q . The decoder reconstructs the input sequence. The loss function is defined as:

$$L = \underbrace{\|w - \hat{w}\|_2^2}_{\text{Reconstruction loss}} + \underbrace{\|\text{sg}[z_e] - z_q\|_2^2}_{\text{VQ loss}} + \underbrace{\beta \|z_e - \text{sg}[z_q]\|_2^2}_{\text{Commitment loss}}, \quad (9)$$

where $\text{sg}[\cdot]$ represents the stop-gradient operator. The first term \mathcal{L}_{rec} ensures that the reconstructed traffic closely matches the original input. The second term \mathcal{L}_{VQ} updates the codebook vectors to better align with the encoder outputs by pulling codewords toward the latent representations. The third loss \mathcal{L}_{com} encourages the encoder outputs to stay close to their assigned codebook entries, preventing the encoder from drifting too far from the discrete latent space.

The pretrained encoder and codebook are then used to extract discrete latent representations of traffic data as inputs to the denoising process.

B. Discrete Diffusion for Trajectory Modeling

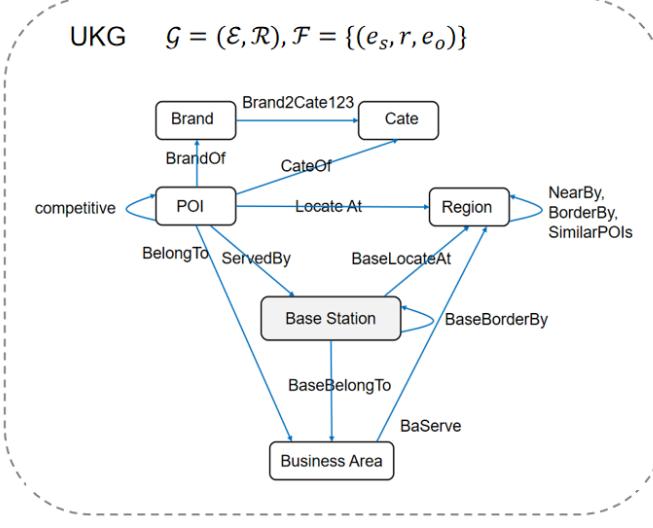


Fig. 2. Illustration of urban knowledge graph

1) *Urban Knowledge Graph Embedding*: Prior studies have demonstrated that the semantic characteristics of places can lead to distinct traffic patterns at base stations [14]–[16]. These traffic patterns influence the data usage behaviors of mobile users. To effectively capture the semantic context of the urban environment, we leverage an urban knowledge graph, which represents relationships between various urban entities in the form of triplets. The urban knowledge graph \mathcal{G} consists of a set of triplets denoted by $\mathcal{F} = (e_s, r, e_o)$, where e_s and e_o are entities in \mathcal{E} . Each relation $r \in \mathcal{R}$ encodes a semantic connection between the corresponding entities. The entities and relations are shown in Fig. 2. Specifically, since base station IDs represent the locations, we focus on relations that are directly associated with base stations: (1) **BaseLocateAt**: a base station is located within a region; (2) **BaseBelongTo**: a base station is associated with a business area; (3) **BaseBorderBy**: a base station shares a border with another base station; (4) **ServedBy**: a base station is served by nearby points of interest (POIs). To learn knowledge graph embeddings for these location entities, we adopt the TuckER model [17], which is an effective tensor factorization approach for knowledge graph representation learning.

2) *Construction of Transition Matrix*: The forward process of a discrete diffusion model, defined in Equation 4, requires the design of a corruption scheme that gradually transforms the data toward a uniform distribution over the state space. Here, we adopt the transition-matrix approach, motivated by the analogy between locations and words, as both are discrete variables whose embeddings preserve semantics.

After obtaining the output of TuckER, we select the top- K nearest neighbors within a distance threshold for each location embedding to form a binary adjacency matrix $A \in \mathbb{R}^{N \times N}$, where

$$A_{ij} = \begin{cases} 1, & \text{if } j \in \text{Nei}(i) \\ 0, & \text{otherwise} \end{cases}. \quad (10)$$

Next, we apply normalization to matrix A as $B = \frac{1}{2K}(A + A^\top)$. The transition rate matrix R has rows summing to zero,

Algorithm 1: Training

```

for  $i = 1$  to  $ITER$  do
     $m_0, l_0 \sim q(M_0), q(L_0);$ 
     $w_0 = \text{DWT}(m_0);$ 
     $h_0 = \text{OneHot}(l_0);$ 
     $s \sim \text{Uniform}(\{1, \dots, S\});$ 
     $\epsilon \sim \mathcal{N}(0, 1);$ 
1.Add noise for traffic data:
     $w_s = \sqrt{\bar{\alpha}_{tr,s}}w_0 + \sqrt{1 - \bar{\alpha}_{tr,s}}\epsilon;$ 
2.Add noise for trajectory:
     $\bar{Q}_s = \exp(\bar{\alpha}_{tj,s}R);$ 
     $q(l_s | l_0) = h_0 \cdot \bar{Q}_s;$ 
3.Compute loss:
     $\hat{\epsilon}_s = f_{\theta,tr}(w_s, q(l_s | l_0), s);$ 
     $\mathbf{z}_s = f_{\theta,tj}(w_s, q(l_s | l_0), s);$ 
     $\mathcal{L} = \text{MSE}(\epsilon, \hat{\epsilon}_s) + D_{\text{KL}}(q(l_{s-1} | l_s, l_0) || p_\theta(l_{s-1} | l_s)) + \text{CE}(l_0, \mathbf{z}_s).$ 

```

with R_{ij} denoting the transition rate from location i to j :

$$R_{ij} = \begin{cases} B_{ij}, & \text{if } i \neq j \\ -\sum_{k \neq i} B_{ik}, & \text{if } i = j \end{cases}. \quad (11)$$

The one-step transition probability matrix Q_s at diffusion step s and the cumulative transition matrix \bar{Q}_s are given by:

$$Q_s = \exp(\alpha_{tj,s}R), \quad \bar{Q}_s = \prod_{i=1}^s Q_i = \exp(\bar{\alpha}_{tj,s}R), \quad (12)$$

where $\bar{\alpha}_{tj,s} = \sum_{i \leq s} \alpha_i$. The transition schedule $\alpha_{tj,s}$ over diffusion steps ensures that at the final step the transition probability approximates a uniform distribution, implying that location information becomes fully randomized.

C. Co-Denoising Process

1) *Multi-Scale Transformer Denoising Network*: This module captures temporal dependencies between traffic and trajectory data. We obtain noisy DWT coefficients $w_s \in \mathbb{R}^{B \times T \times 1}$ and the transition probability $q(l_s | l_0) \in \mathbb{R}^{B \times T \times N}$, where B denotes the batch size. They are concatenated along the last dimension and fed into the multi-scale Transformer $f_\theta(w_s, q(l_s | l_0), s)$. Starting from the coarsest wavelet scale, traffic is encoded by a pre-trained VQ-VAE and then fused with the original traffic features via a gating mechanism, while the corresponding trajectory representation is processed by an MLP. Positional, type and diffusion step embeddings are added before entering Transformer blocks. To preserve coarse-scale context, each Transformer output is merged with the finer-scale representation as input to the next layer, enabling hierarchical refinement across scales. The final output is formed by concatenating the outputs of all layers from coarse to fine.

2) *Loss Function*: The model simultaneously takes continuous traffic data and discrete trajectories as input. Although they share the same diffusion step s , we adopt separate noise schedules to reflect their distinct data characteristics. The multi-scale Transformer outputs predictions separately: (1) the predicted noise $\hat{\epsilon}_s = f_{\theta,tr}(w_s, q(l_s | l_0), s)$ for the traffic data,

Algorithm 2: Sampling

$w_S \sim \mathcal{N}(0, 1)$;
 $l_S \sim \text{Uniform}(\{0, 1, \dots, N-1\})$;
 $p_L = \frac{1}{N} \cdot \mathbf{1}_N \in \mathbb{R}^N$;
for $s = S$ **to** 1 **do**
 1. **Impute samples for traffic data:**
 $z \sim \mathcal{N}(0, 1)$;
 $w_{s-1} = \frac{1}{\sqrt{\alpha_{tr,s}}}(w_s - \frac{1-\alpha_{tr,s}}{\sqrt{1-\bar{\alpha}_{tr,s}}} f_{\theta,tr}(w_s, p_L, s)) + \sigma_s z$;
 2. **Impute samples for trajectory:**
 $\mathbf{z}_s = f_{\theta,tj}(w_s, p_L, s)$;
 $p_\theta(\hat{l}_0 | l_s) = \text{softmax}(\mathbf{z}_s)$;
 $\hat{l}_0 \sim p_\theta(\hat{l}_0 | l_s)$;
 2.1 **Compute posterior:**
 $\hat{h}_0, h_s = \text{OneHot}(\hat{l}_0), \text{OneHot}(l_s)$;
 Let $\tilde{q}_s = h_s Q_s^\top$, $\tilde{q}_{s-1} = \hat{h}_0 \bar{Q}_{s-1}$;
 $q(l_{s-1} | l_s, \hat{l}_0) = \text{Cat}(l_{s-1}; p = \text{Normalize}(\tilde{q}_s \odot \tilde{q}_{s-1}))$;
 2.2 **Compute one-step probability:**
 $p_\theta(l_{s-1} | l_s) = \sum_{\hat{l}_0} q(l_{s-1} | l_s, \hat{l}_0) p_\theta(\hat{l}_0 | l_s)$;
 $p_L \leftarrow p_\theta(l_{s-1} | l_s)$;
 $l_{s-1} \sim p_\theta(l_{s-1} | l_s)$;
 $m_0 = \text{IDWT}(w_0)$;
Return $\{m_0, l_0\}$.

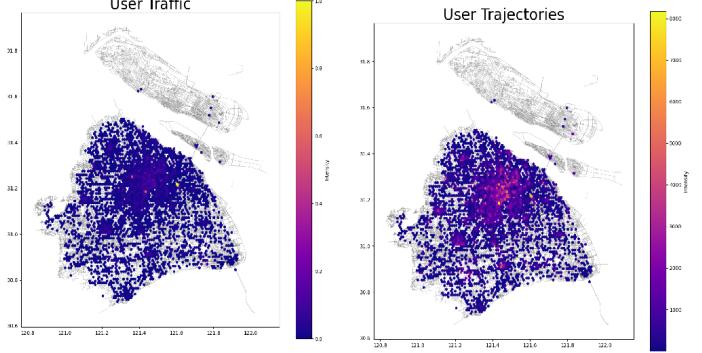
and (2) the predicted logits $\mathbf{z}_s = f_{\theta, t_j}(w_s, q(l_s | l_0), s)$ for the trajectory data.

After obtaining \hat{w}_0 via denoising, we apply the inverse Discrete Wavelet Transform (iDWT) to reconstruct the original input. The DWT–iDWT process acts as an encoder–decoder structure, mapping the time series to a latent domain for denoising.

For trajectory data, the multi-scale Transformer outputs logits \mathbf{z}_s , and applying softmax yields $p_\theta(\hat{l}_0 | l_s) = \text{softmax}(\mathbf{z}_s)$. Given this, the reverse transition probability $p_\theta(l_{s-1} | l_s)$ is computed by marginalizing over possible original states. The total training objective consists of an MSE loss for the traffic branch introduced in Equation 3, and a KL divergence loss for the trajectory branch. In addition, we introduce an auxiliary prediction loss for the trajectory branch that directly supervises the conditional distribution $p_\theta(\hat{l}_0 | l_s)$, leading to more stable training and improved generative quality. Specifically, since l_0 is discrete and the Transformer outputs logits, we instantiate this objective as a cross-entropy loss. Given three hyperparameters λ_1 , λ_2 and λ_3 , the final loss is given by:

$$\begin{aligned}\mathcal{L} &= \lambda_1 \cdot \mathcal{L}_{tr} + \lambda_2 \cdot \mathcal{L}_{tj} + \lambda_3 \cdot \mathcal{L}_{pred}, \\ \mathcal{L}_{tr} &= \text{MSE}(\epsilon, \hat{\epsilon}_s), \\ \mathcal{L}_{tj} &= \text{D}_{\text{KL}}(q(l_{s-1} \mid l_s, l_0) \parallel p_\theta(l_{s-1} \mid l_s)), \\ \mathcal{L}_{pred} &= \text{CE}(l_0, \mathbf{z}_s).\end{aligned}\tag{13}$$

In conclusion, the co-denoising module allows the simultaneous reconstruction of both modalities while preserving their temporal alignment and cross-modal dependencies. Detailed training and sampling algorithms are shown in Algorithm 1 and Algorithm 2.



(a) User normalized tra

(b) User trajectory

IV. EXPERIMENTS

A. Experiment Settings

1) Dataset and Preprocessing: The large-scale user-level cellular traffic dataset was collected in Shanghai in 2016 and spans one week of data. It captures individual users' mobile network usage over time and space, with each record containing a timestamp, the associated base station ID and its GPS coordinates, and the corresponding traffic volume. Visualizations of the dataset are shown in Figure 3. After retaining only the city center region, the dataset includes over 2,000 users and approximately 4,000 unique base station locations, with a temporal resolution of 30 minutes, yielding 336 time points per user for the week.

2) *Baselines*: MSTDiff is compared with the following four baselines for the traffic generation task:

- **CSDI** [6]: CSDI is a diffusion-based model originally developed for probabilistic time series imputation. It utilizes a residual network with a two-dimensional Transformer encoder to capture temporal and feature-wise dependencies separately. This architectural design also enables CSDI to be adapted for time series generation tasks.
 - **DiT** [7]: DiT integrates the Transformer architecture into the diffusion process to effectively capture long-range dependencies. The original paper proposes three variants of the Diffusion Transformer, which have shown strong performance in image generation. In our experiment, we adapt DiT to handle time series modeling.
 - **ADAPTIVE** [18]: ADAPTIVE is a city-scale cellular traffic generation framework that leverages transfer learning to synthesize traffic in cities lacking historical data. It aligns base station representations between the target and source cities, and generates target-city traffic using a feature-enhanced GAN.
 - **MSH-GAN** [19]: MSH-GAN is a GAN-based framework for user-level mobile traffic generation. It captures individual level behaviors using BiLSTM and self-attention mechanisms, while using a Switch Mode Generator to capture group-level behaviors.

Moreover, MSTDiff is compared with the following four baselines for the trajectory generation task:

TABLE I

EVALUATION RESULTS OF TRAFFIC GENERATION. BOLD INDICATES THE BEST PERFORMANCE, UNDERLINED INDICATES THE SECOND-BEST, AND DOUBLE-UNDERLINED INDICATES THE THIRD-BEST.

Methods	Traffic Volume	First-order Difference	Daily Periodic Component
	JSD	JSD	RMSE
CSDI	<u>0.1514</u>	0.1669	<u>0.0160</u>
DiT	0.2272	0.1889	0.0178
ADAPTIVE	<u>0.1289</u>	<u>0.1888</u>	0.0152
MSH-GAN	0.6701	0.5894	<u>0.0169</u>
MSTDiff -w/o Traj	0.1682	0.1675	0.0191
MSTDiff	0.1230	0.1379	0.0263

- **TimeGEO** [20]: It generates urban mobility patterns by modeling temporal behavior with the weekly home-based tour count, dwell rate, and burst rate, and by selecting locations via a rank-based exploration and preferential return model that balances visits to new places and returns to familiar ones.
- **PateGail** [21]: It generates mobility trajectories with a privacy-preserving imitation learning framework built on generative adversarial imitation learning to mimic the process of human decision-making.
- **DiffTraj** [8]: It generates GPS trajectory points with a UNet-based diffusion model conditioned on trip attributes. It uses continuous movement features such as distance, duration and average speed, together with discrete factors including departure time slot and start and end region IDs.
- **VOLUNTEER** [22]: It generates trajectories with a variational framework comprising a user VAE for group-level user distributions and a trajectory VAE for individual mobility patterns. The trajectory VAE decouples travel time and dwell time to generates realistic trajectories.

3) *Metrics*: We evaluate the performance of the traffic generation task using three metrics. The Jensen–Shannon divergence (JSD) of traffic volume distribution, the JSD of first-order differences to assess short-term variation patterns, and the root-mean-square error (RMSE) of daily periodic components extracted via Fourier analysis to measure the alignment of daily cycles. Lower values across these metrics indicate closer alignment between generated and real-world traffic data.

We evaluate the performance of trajectory generation using six metrics. All metrics are computed as the JSD between the generated and real distributions. **Distance** is the distance between adjacent visited locations for each user. **Radius** is the radius of gyration of a user’s trajectory. **DistinctLoc** is the fraction of distinct locations per user. **Duration** is the contiguous stay duration per visit. **G-rank** is the distribution of visit frequency over the locations with the top 10% overall visits. **I-rank** is the distribution of visit frequency over each user’s own visited locations.

B. Traffic Generation Task

The results of traffic generation task are shown in Table I. MSTDiff achieves superior performance in both the JSD of traffic volume and the JSD of first-order differences. Specifically, it reduces the JSD of traffic volume by 4.58% compared to ADAPTIVE, and the JSD of first-order differences by 17.38% compared to the best-performing baseline, CSDI. However, the RMSE is relatively higher, possibly due to the use of wavelet transforms, which are well-suited for capturing sudden bursts but less effective in modeling stable daily periodic patterns.

We also conduct an ablation study where the model uses only traffic data as input, excluding trajectory information. The performance degrades with an increase in JSD. Compared to the ablated version, our full model incorporating both traffic and trajectory inputs achieves a 26.87% reduction in the JSD of traffic volume, and a 17.67% reduction in the JSD of first-order differences. These results highlight the effectiveness of incorporating trajectory information in improving traffic generation performance.

C. Trajectory Generation Task

The results of the trajectory generation task are shown in Table II. MSTDiff achieves the best scores on five of six JSD metrics. Compared with the best performing baselines, it reduces JSD by more than 15% on Distance and Radius, 65.09% on DistinctLoc, and 25.51% on Duration. These gains indicate that the model more faithfully captures users’ location switches, realistic spatial movement, and dwell time distributions, reflecting strong spatiotemporal modeling of trajectories. The distributions of the evaluation statistics are shown in Figure 4, and in most cases MSTDiff is closest to the real data. In most cases the distribution of MSTDiff is closest to the real data. In contrast, its G-rank performance is weaker than other baseline models, suggesting an overall preference for specific locations, yet it models user-level visiting tendencies effectively.

V. RELATED WORK

A. Trajectory Generation

Trajectory generation plays a critical role in urban planning, emergency management, and traffic scheduling. Early studies

TABLE II
EVALUATION RESULTS OF TRAJECTORY GENERATION.

Methods	Distance	Radius	DistinctLoc	Duration	G-rank	I-rank
TimeGEO	<u>0.0840</u>	0.5266	<u>0.4906</u>	<u>0.0639</u>	0.0051	<u>0.1163</u>
PateGail	<u>0.0469</u>	<u>0.4070</u>	<u>0.3990</u>	0.5944	<u>0.0097</u>	<u>0.1138</u>
DiffTraj	0.2525	<u>0.2060</u>	0.5643	0.1746	0.0152	0.2329
VOLUNTEER	0.2782	0.6617	0.6671	<u>0.1356</u>	<u>0.0116</u>	0.3237
MSTDiff	0.0387	0.1737	0.1393	0.0476	0.3533	0.0297

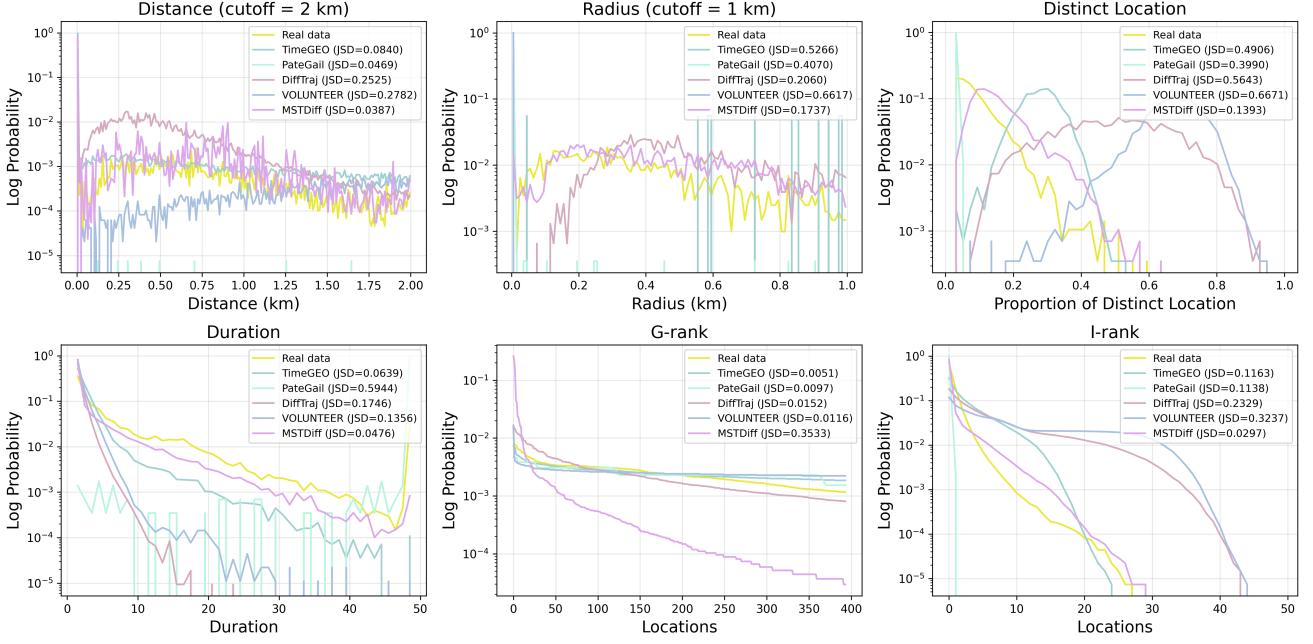


Fig. 4. Visualization of metrics of trajectory generation.

primarily relied on statistical and physical modeling, aiming to uncover universal patterns in human mobility behavior. For example, the Lévy flight model captures the proportion of long- and short-distance movements through a power-law distribution, successfully reproducing the jumpy and irregular characteristics commonly observed in human travel [23]. The Exploration and Preferential Return (EPR) model further introduces location preferences and return mechanisms, offering a more realistic representation of human mobility patterns [24]. The TimeGeo [20] variant of EPR models the relationships between home and work inferred from data and flexible activities' spatial and temporal patterns (e.g., other activities). It sets parameters to capture individuals' circadian rhythms and generates a wide range of empirically observed movement behaviors.

In recent years, the rapid development of artificial intelligence (AI), especially the rise of deep generative models, has opened up new possibilities for trajectory generation. For example, PateGail leverages powerful generative adversarial imitation learning to simulate human decision-making processes and incorporates privacy-preserving mechanisms to protect user data [21]. Long et al. proposed the VOLUNTEER framework, which employs a dual VAE architecture to model both user attribute distributions and trajectory behavior dis-

tributions, significantly improving the realism and diversity of the generated trajectories [22]. Additionally, Zhu et al. introduced a spatiotemporal diffusion probabilistic model for trajectory modeling, effectively combining the diffusion process with the spatiotemporal characteristics of trajectories [8]. Notably, with the widespread adoption of large language models (LLMs), recent studies have begun to explore their potential in modeling human mobility behavior. Wang et al. proposed an LLM-based agent framework for trajectory generation, enabling the model to understand and generate complex movement trajectories that incorporate individual motivations, preferences, and semantic context [25].

However, existing studies focus on the single task of trajectory generation, often overlooking the intrinsic relationship between individual trajectories and user mobile traffic. To address this limitation, we propose a novel framework that jointly generates human mobility trajectories and user mobile traffic.

B. Cellular Traffic Generation

With the rapid growth of mobile networks, simulating cellular traffic at both the city scale and the individual level has become increasingly important for evaluating network

performance and guiding deployment strategies. This section reviews recent studies on cellular traffic generation.

The task of traffic generation has evolved over time. In its early stages, researchers primarily relied on mathematical approaches [26]–[28] to simulate network traffic. For example, Aceto et al. [29] used Markov Chains and Hidden Markov Models to model traffic behavior. With the rise of artificial intelligence, more advanced generative models have emerged. Zhang et al. [30] introduced a densely connected convolutional network to simulate citywide traffic. LSTM-based models were explored by Dalgkitsis et al. [31] and Trinh et al. [32] for time series prediction. More recently, Su et al. [33] presented a lightweight attention-based deep learning framework to generate weekly cell traffic patterns.

Another line of neural network-based models involves GANs, which have been widely applied to generate both flow-level [34], [35] and packet-level traffic [36]–[38] at individual base stations, and typically rely on detailed network configuration data. As a result, they are not well-suited for large-scale generation tasks. To address this, newer models have adapted GANs for city-level applications [18], [39]. For instance, Hui et al. [39] designed a knowledge-enhanced GAN that incorporates multi-periodic urban features and base station locations. Similarly, Zhang et al. [18] proposed ADAPTIVE, a transfer learning framework that aligns traffic patterns from a source city to a target city lacking historical records.

Recently, diffusion models have emerged as strong alternatives in generative tasks, offering greater stability and mitigating issues such as mode collapse that are common in GAN-based approaches. They have been successfully applied in diverse domains, including image generation [3], [4], time-series data imputation [6], as well as traffic prediction [40]. In the field of network traffic generation, Qi et al. [41] introduced CANDLE, a conditional diffusion model that generates 5G traffic in unobserved regions by learning cross-modal relations between 4G and 5G traffic through attention mechanisms, while using GCNs to model spatial correlations. Another diffusion-based approach, STK-Diff by Chai et al. [42], integrates modules for urban context, temporal patterns, and spatial relationships to enable realistic and controllable traffic synthesis at city scale. Moreover, a recent model called NetDiffus, proposed by Sivaroopan et al. [43], addresses network traffic generation by transforming one-dimensional trace sequences into two-dimensional Gramian Angular Summation Field representations. These images capture features such as packet size and inter-arrival time, which are then used to train a diffusion model for synthesizing realistic traffic patterns.

Unlike their work, our model mainly focuses on generating user-level network traffic, as well as incorporating the trajectory data.

VI. CONCLUSION

In this work, we present a unified diffusion Transformer framework for joint simulation of mobile traffic and user trajectories. The model integrates wavelet-based multi-resolution decomposition, a multi-scale Transformer with cross-attention for capturing temporal dependencies, and a hybrid denoising

mechanism that handles both continuous and discrete data. Semantic information is further incorporated via knowledge graph embeddings and a similarity-based transition mechanism for trajectory generation.

REFERENCES

- [1] K. Ramamohanarao, J. Qi, E. Tanin, and S. Motallebi, “From how to where: Traffic optimization in the era of automated vehicles,” in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2017, pp. 1–4.
- [2] H. Chai, T. Jiang, and L. Yu, “Diffusion model-based mobile traffic generation with open data for network planning and optimization,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 4828–4838.
- [3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10684–10695.
- [4] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, “Palette: Image-to-image diffusion models,” in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022.
- [5] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video diffusion models,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 8633–8646. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/39235c56ae13fb05a6adc95eb9d8d66-Paper-Conference.pdf
- [6] Y. Tashiro, J. Song, Y. Song, and S. Ermon, “CSDI: Conditional score-based diffusion models for probabilistic time series imputation,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 804–24 816, 2021.
- [7] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 4195–4205.
- [8] Y. Zhu, Y. Ye, S. Zhang, X. Zhao, and J. Yu, “Diffraj: Generating gps trajectory with diffusion probabilistic model,” in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 65 168–65 188.
- [9] Y. Zhu, J. J. Yu, X. Zhao, Q. Liu, Y. Ye, W. Chen, Z. Zhang, X. Wei, and Y. Liang, “Controll traj: Controllable trajectory generation with topology-constrained diffusion model,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 4676–4687.
- [10] Z. Liu, Q. Long, H. Wang, and Y. Li, “Multi-scale diffusion transformer for jointly simulating user mobility and mobile traffic pattern,” in *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, 2025, pp. 1–4.
- [11] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *CoRR*, vol. abs/2006.11239, 2020.
- [12] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, and M. Welling, “Argmax flows and multinomial diffusion: Learning categorical distributions,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 12 454–12 465.
- [13] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg, “Structured denoising diffusion models in discrete state-spaces,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 17 981–17 993.
- [14] H. Wang, F. Xu, Y. Li, P. Zhang, and D. Jin, “Understanding mobile traffic patterns of large scale cellular towers in urban environment,” in *Proceedings of the 2015 Internet Measurement Conference*, 2015, p. 225–238.
- [15] T. Li, T. Xia, H. Wang, Z. Tu, S. Tarkoma, Z. Han, and P. Hui, “Smartphone app usage analysis: Datasets, methods, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 937–966, 2022.
- [16] B. Cici, M. Gjoka, A. Markopoulou, and C. T. Butts, “On the decomposition of cell phone activity patterns and their connection with urban ecology,” in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc ’15, 2015, p. 317–326.

- [17] I. Balazevic, C. Allen, and T. M. Hospedales, “Tucker: Tensor factorization for knowledge graph completion,” *CoRR*, vol. abs/1901.09590, 2019.
- [18] S. Zhang, T. Li, S. Hui, G. Li, Y. Liang, L. Yu, D. Jin, and Y. Li, “Deep transfer learning for city-scale cellular traffic generation through urban knowledge graph,” 2023, p. 4842–4851.
- [19] T. Li, S. Hui, S. Zhang, H. Wang, Y. Zhang, P. Hui, D. Jin, and Y. Li, “Mobile user traffic generation via multi-scale hierarchical gan,” *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 8, Jul. 2024.
- [20] S. Jiang, Y. Yang, S. Gupta, D. Veneziano, S. Athavale, and M. C. González, “The timegeo modeling framework for urban mobility without travel surveys,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 37, pp. E5370–E5378, 2016.
- [21] H. Wang, C. Gao, Y. Wu, D. Jin, L. Yao, and Y. Li, “Pategail: A privacy-preserving mobility trajectory generator with imitation learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 12, 2023, pp. 14 539–14 547.
- [22] Q. Long, H. Wang, T. Li, L. Huang, K. Wang, Q. Wu, G. Li, Y. Liang, L. Yu, and Y. Li, “Practical synthetic human trajectories generation based on variational point processes,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4561–4571.
- [23] D. Brockmann, L. Hufnagel, and T. Geisel, “The scaling laws of human travel,” *Nature*, vol. 439, no. 7075, pp. 462–465, 2006.
- [24] C. Song, T. Koren, P. Wang, and A.-L. Barabási, “Modelling the scaling properties of human mobility,” *Nature Physics*, vol. 6, no. 10, pp. 818–823, 2010.
- [25] W. JIAWEI, R. Jiang, C. Yang, Z. Wu, R. Shibasaki, N. Koshizuka, C. Xiao *et al.*, “Large language models as urban residents: An ilm agent framework for personal mobility generation,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 124 547–124 574, 2024.
- [26] J. Sommers and P. Barford, “Self-configuring network traffic generation,” in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, 2004, p. 68–81.
- [27] M. C. Weigle, P. Adurthi, F. Hernández-Campos, K. Jeffay, and F. D. Smith, “Tmix: a tool for generating realistic tcp application workloads in ns-2,” vol. 36, no. 3, p. 65–76, jul 2006.
- [28] K. V. Vishwanath and A. Vahdat, “Swing: Realistic and responsive network traffic generation,” *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 712–725, 2009.
- [29] G. Aceto, G. Bovenzi, D. Ciouzon, A. Montieri, V. Persico, and A. Pescapé, “Characterization and prediction of mobile-app traffic using markov modeling,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 907–925, 2021.
- [30] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, “Citywide cellular traffic prediction based on densely connected convolutional neural networks,” *IEEE Communications Letters*, vol. 22, no. 8, pp. 1656–1659, 2018.
- [31] A. Dalgkitsis, M. Louta, and G. T. Karetos, “Traffic forecasting in cellular networks using the lstm rnn,” in *Proceedings of the 22nd Pan-Hellenic Conference on Informatics*, 2018, p. 28–33.
- [32] H. D. Trinh, L. Giupponi, and P. Dini, “Mobile traffic prediction from raw data using lstm networks,” in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 1827–1832.
- [33] J. Su, H. Cai, Z. Sheng, A. Liu, and A. Baz, “Traffic prediction for 5g: A deep learning approach based on lightweight hybrid attention networks,” *Digital Signal Processing*, vol. 146, p. 104359, 2024.
- [34] S. Iannucci, H. A. Kholidy, A. D. Ghimire, R. Jia, S. Abdelwahed, and I. Banicescu, “A comparison of graph-based synthetic data generators for benchmarking next-generation intrusion detection systems,” in *2017 IEEE International Conference on Cluster Computing (CLUSTER)*, 2017, pp. 278–289.
- [35] M. Ring, D. Schlör, D. Landes, and A. Hotho, “Flow-based network traffic generation using generative adversarial networks,” *Computers & Security*, vol. 82, pp. 156–172, 2019.
- [36] A. Cheng, “Pac-gan: Packet generation of network traffic using generative adversarial networks,” in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2019, pp. 0728–0734.
- [37] Y. Yin, Z. Lin, M. Jin, G. Fanti, and V. Sekar, “Practical gan-based synthetic ip header trace generation using netshare,” in *Proceedings of the ACM SIGCOMM 2022 Conference*. New York, NY, USA: Association for Computing Machinery, 2022, p. 458–472.
- [38] B. Dowoo, Y. Jung, and C. Choi, “Pcapgan: Packet capture file generator by style-based generative adversarial networks,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 1149–1154.
- [39] S. Hui, H. Wang, T. Li, X. Yang, X. Wang, J. Feng, L. Zhu, C. Deng, P. Hui, D. Jin, and Y. Li, “Large-scale urban cellular traffic generation via knowledge-enhanced gans with multi-periodic patterns,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, p. 4195–4206.
- [40] S.-S. Kim, M. Chung, and Y.-K. Kim, “Urban traffic prediction using congestion diffusion model,” in *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, 2020, pp. 1–4.
- [41] X. Qi, H. Chai, L. Yu, Y. Li, and Z. Wang, “Regional features conditioned diffusion models for 5g network traffic generation,” in *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 396–409.
- [42] H. Chai, X. Qi, and Y. Li, “Spatio-temporal knowledge driven diffusion model for mobile traffic generation,” *IEEE Transactions on Mobile Computing*, pp. 1–18, 2025.
- [43] N. Sivaroopan, D. Bandara, C. Madarasingha, G. Jourjon, A. P. Jayasumana, and K. Thilakarathna, “Netdiffus: Network traffic generation by diffusion models through time-series imaging,” *Computer Networks*, vol. 251, p. 110616, 2024.