Spatio-Temporal Diffusion Model for Cellular Traffic Generation

Xiaosi Liu, Xiaowen Xu, Zhidan Liu, *Senior Member, IEEE*, Zhenjiang Li, *Member, IEEE*, and Kaishun Wu, *Fellow, IEEE*

Abstract—In the digital era, the increasing demand for network traffic necessitates strategic network infrastructure planning. Accurate modeling of traffic demand through cellular traffic generation is crucial for optimizing base station deployment, enhancing network efficiency, and fostering technological innovation. In this paper, we introduce STOUTER, a spatio-temporal diffusion model for cellular traffic generation. STOUTER incorporates noise into traffic data through a forward diffusion process, followed by a reverse reconstruction process to generate realistic cellular traffic. To effectively capture the spatio-temporal patterns inherent in cellular traffic, we pre-train a temporal graph and a base station graph, and design the Spatio-Temporal Feature Fusion Module (STFFM). Leveraging STFFM, we develop STUnet, which estimates noise levels during the reverse denoising process, successfully simulating the spatio-temporal patterns and uncertainty variations in cellular traffic. Extensive experiments conducted on five cellular traffic datasets across two regions demonstrate that STOUTER improves cellular traffic generation by 52.77% in terms of the Jensen-Shannon Divergence (JSD) metric compared to existing models. These results indicate that STOUTER can generate cellular traffic distributions that closely resemble real-world data, providing valuable support for downstream applications.

Index Terms—Cellular traffic, spatio-temporal graph, diffusion model, data generation

1 Introduction

As a fundamental component of mobile wireless communication infrastructure, cellular networks serve as critical enablers for advancing smart cities, Internet of Things (IoT), autonomous driving, and telemedicine. The proliferation of Fifth Generation (5G) technology has further cemented their role as indispensable tools for modern information systems [1], supporting massive device connectivity while delivering high-speed, low-latency communication and intelligent cross-industry services [2]. Amidst accelerating societal digitalization, escalating traffic demands necessitate efficient infrastructure operation and flexible resource allocation, making optimal network planning and strategic resource distribution persistent challenges [3]–[8].

Cellular traffic prediction has emerged as a key strategy for dynamic resource allocation [9]–[12]. However, existing methods face two critical limitations: (1) dependence on extensive historical traffic data for target areas leads to computationally intensive processes and prohibitive prediction latency, and (2) restricted access to real-time operator data due to privacy concerns impedes practical implementation. Adding to these issues, suboptimal base station deployments in many regions create additional complexities for network optimization. Current deployment strategies — including manual site selection [13], drone-assisted placement [14], and shared infrastructure [15] — often prioritize geographical factors over actual traffic demand patterns, underscoring the need for more holistic solutions.

Synthetic cellular traffic generation offers a promising alternative by simulating network behavior using open-source data. While deep learning approaches like autoregressive CNNs [16] and GAN-based methods [17] have demonstrated success in device-level traffic synthesis, their scalability to large-scale base station deployments remains constrained. Recent large-scale GAN variants [18], [19] incorporate urban knowledge graphs and multi-period classification but face practical barriers, including data acquisition challenges [20], [21], mode collapse risks [22], [23], and limited diversity in generated outputs [24]. Moreover, existing methods predominantly model predefined spatio-temporal patterns while neglecting inherent traffic uncertainties within identical contexts.

In this paper, we propose a novel cellular traffic generation method capable of effectively capturing the spatio-temporal characteristics of a region while simulating the uncertainty in traffic fluctuations. This approach aims to provide valuable data support for research on network resource optimization and deployment. However, designing such a method poses significant challenges due to the complex patterns inherent in cellular traffic:

- *This work was supported in part by National Natural Science Foundations of China under Grant 62172284 and the Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things under Grant 2023B1212010007.*
- *Xiaosi Liu is with College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, 518060. (E-mail: liuxiaosi2022@email.szu.edu.cn)*
- *Xiaowen Xu and Zhidan Liu are with INTR Thrust, System Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, 510000. (E-mails: xxu265@connect.hkust-gz.edu.cn, zhidanliu@hkust-gz.edu.cn)*
- *Zhenjiang Li is with Department of Computer Science, City University of Hong Kong, Hong Kong, China, 999077. (e-mail: zhenjiang.li@cityu.edu.hk)*
- *Kaishun Wu is with DSA Thrust and IoT Thrust, Information Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, 510000. (E-mail: wuks@hkust-gz.edu.cn)*

** This study was conducted by Xiaosi Liu during her internship at The Hong Kong University of Science and Technology (Guangzhou). (Corresponding author: Dr. Zhidan Liu.)*

- **Long-term and short-term periodic patterns.** Cellular traffic exhibits intricate temporal patterns, including daily fluctuations that correlate with work and recreational schedules, as well as weekly trends characterized by distinct weekday and weekend usage behaviors.
- **Spatial patterns.** Cellular traffic varies across different base stations due to differing human activity levels. Densely populated areas typically experience higher network traffic than sparsely populated ones, complicating the delineation of work and residential zones for each base station.
- **Uncertainty pattern.** Even within the same base station or region, cellular traffic can exhibit significant irregular fluctuations due to the unpredictable nature of human activities. Variations in user demand can arise from differing bandwidth requirements of various applications and services.

To address these challenges, we present **STOUTER**, a **S**patio-**T**emporal diffusi**O**n model for cell**U**lar **T**raffic gen**ER**ation. First, we design a temporal graph structure to represent the hourly and daily temporal relationships in cellular traffic, enabling the capture of both short-term and long-term periodic variations. Second, to distinguish cellular traffic patterns among base stations in various regions, we construct a base station graph that integrates Point of Interest (POI) information and distance relationships between base stations. Third, to model uncertain fluctuations in cellular traffic, we design a generative diffusion model that incorporates spatio-temporal features into the traffic generation process. During the denoising phase, we use an initial Gaussian distribution to effectively simulate traffic uncertainty. Additionally, we introduce the Spatio-Temporal Feature Fusion Module (STFFM), which preserves traffic periodicity and base station-specific patterns during the generation process. This allows us to reconstruct cellular traffic data with realistic spatio-temporal characteristics from an initial Gaussian distribution characterized by uncertainty. In summary, the contributions of our work are as follows:

- We propose a spatio-temporal diffusion model for large-scale cellular traffic generation that simulates uncertain variations effectively.
- We construct a temporal graph to model both long-term and short-term traffic patterns, and develop a base station graph to extract spatial traffic characteristics, integrating these into the traffic generation process through STFFM within Spatio-Temporal UNet (STUnet).
- Extensive experiments conducted on multiple real cellular traffic datasets from two regions demonstrate that STOUTER improves traffic generation by 52.77% in terms of the Jensen-Shannon Divergence (JSD) metric compared to state-of-the-art methods, indicating its capability to generate long-term data closely resembling real traffic and providing valuable support for downstream applications.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the preliminary definitions and complex patterns of cellular traffic, following with an overview of our STOUTER framework. Sections 4 and 5 elaborate the spatio-temporal graph model and the diffusion-based traffic generation model, respectively. Section 6 evaluates the performance of STOUTER, and Section 7 finally concludes the paper.

2 Related Work

2.1 Cellular traffic generation

Traditional methods primarily relied on mathematical models to generate cellular traffic. The traffic generators were designed to synthesize data that closely resembled real-world network traffic in a closed-loop manner [25], [26], and the generated traffic data is mainly used for testing network equipment, services, and security protocols [27], rather than to assist in the deployment of cellular base stations.

Recently, some researchers have explored machine learning-based approaches for traffic generation, such as employing autoregressive models [16], [28] and GAN models [17], [29], [30] to synthesize cellular traffic data. Although these methods protect data privacy while generating traffic for a single device, they primarily focus on traffic generation for a limited range or even one single network device. In contrast, large-scale cellular traffic generation must consider city-wide base station deployment. In our work, we take into account the topological structure between base stations at a city scale, allowing us to model the spatial relationships among them effectively.

Some studies [18], [19] have proposed GAN methods for generating city-scale cellular traffic, by leveraging urban knowledge graphs to capture the spatial semantics of base stations. For example, ADAPTIVE [18] addresses the issue of limited historical data in 5G base station deployment by designing a deep transfer learning framework for the generation of cellular traffic. This framework transfers the traffic knowledge graph from a source city to a target city, allowing the GAN model to incorporate learned spatial and temporal patterns. Hui et al. [19] developed a GAN model that integrates multi-cycle patterns to simulate daily, weekly, and long-term traffic cycle patterns, with the aim of replicating the long-term performance of cellular traffic. However, GAN-based traffic generation methods often suffer from data instability [24]. Artifacts such as unrealistic or noisy data points may emerge, compromising the practicality of the generated traffic data. In addition, during training, GAN models are prone to mode collapse [22], [23], which limits the diversity of generated data and hinders their ability to fully capture the underlying distribution of real-world cellular traffic.

Furthermore, STK-Diff [31] uses urban knowledge graphs as semantic information and develops a spatio-temporal knowledge-driven diffusion model for mobile traffic generation. Urban knowledge graphs utilize graph structures to organize relationships between entities, such as user behavior, spatio-temporal associations, and functional complementarity. They require the integration of multi-source data, such as trajectory data, socioeconomic indicators, and text descriptions, and depend on domain knowledge to build semantic relationships. Without domain knowledge, these graphs are susceptible to semantic gaps [32]. Furthermore, challenges such as data source limitations, privacy protection, and intellectual property restrictions make

it difficult to obtain urban knowledge graphs [20], [21]. Therefore, OpenDiff [33] proposes a mobile traffic generation method based on publicly available data, including population density, points of interest (POIs), and satellite imagery. However, the dynamic nature of human activities limits the reliability of such data, as statistical indicators like population density may become outdated, failing to accurately reflect real-time mobile traffic trends. POI is a fundamental data used to represent specific locations, and can be easily accessed and downloaded from OpenStreetMap [34] or other platforms. Different from previous works, we integrate the spatial relationships of base stations with surrounding POI information to extract potential spatial patterns in cellular traffic, which provides a more comprehensive representation of real-world traffic dynamics.

2.2 Cellular traffic prediction

Cellular traffic prediction models forecast future traffic volumes of base stations using historical data, aiding network management and supporting various network applications. Effective prediction requires not only temporal modeling of traffic patterns at individual base stations but also the ability to capture spatial dependencies and variations in future traffic distributions within a given area [35].

LSTM-GPR [36] combines Long Short-Term Memory (LSTM) networks with Gaussian Process Regression (GPR) to predict traffic for individual cell base stations. CCSANet [37] employs a convolutional LSTM and a self-attention network based on correlation to predict traffic in complex cellular networks. STA-GCN [9] and STEP [10] utilize graph convolutional networks (GCNs) for spatio-temporal predictions of cellular traffic, while ST-Tran [11] and ST-InducedTrans [12] integrate time and space Transformer modules for spatio-temporal cellular traffic prediction.

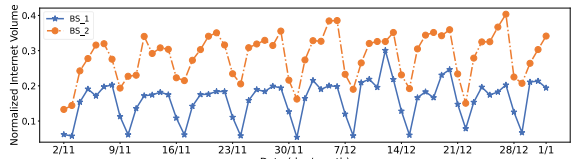
However, traffic prediction relies on a substantial amount of historical data, which limits its applicability to traffic generation task. Additionally, long-term predictions may suffer from cumulative errors, leading to progressive declines in model accuracy over extended time horizons.

2.3 Time series modeling based on diffusion model

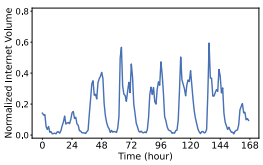
Denoising Diffusion Probabilistic Models (DDPM) [38] are generative models that reconstruct target data samples, such as images or audio, by iteratively removing noise from noisy inputs through a step-by-step denoising process. Diffusion models have been extensively applied to time series prediction [39], interpolation [40], [41], and data generation [42], [43] due to their ability to model complex and high-dimensional data distributions. Compared to GAN models, diffusion models can produce more stable outputs and are less prone to mode collapse. By iteratively refining random noise, these models effectively restore the underlying data distribution, improving both accuracy and quality in the generated samples.

For example, DiffSTG [44] designs UGnet to apply diffusion models to spatio-temporal graph prediction, addressing uncertainty and complex spatio-temporal dependencies in data modeling. DiffTraj [45] employs diffusion models for generating GPS trajectories, which tackles the privacy issues in location-based data. KSTDiff [46] introduces a knowledge-enhanced spatio-temporal diffusion model for urban pedestrian flow prediction, enabling pedestrian flow data generation without reliance on historical records.

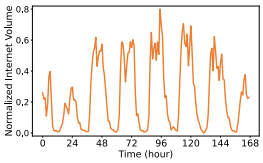
Building on these advancements, our study applies generative diffusion models to large-scale cellular traffic generation in base stations, tackling the challenges of spatio-temporal variability and uncertainty in network traffic.



(a) Daily Internet traffic volume of 2 base stations.



(b) Hourly Internet traffic volume of Base Station 1.



(c) Hourly Internet traffic volume of Base Station 2.

Fig. 1: Statistics on (a) daily Internet traffic and (b, c) hourly Internet traffic for two typical base stations, where 'BS_1' and 'BS_2' represent Base Station 1 and Base Station 2 respectively

.

3 Preliminary

In this section, we first introduce the basic definitions and complex patterns of cellular traffic. Then, we present the overview of STOUTER.

3.1 Problem definition

Definition 1 (Base Station). Given an area where a large-scale base station is deployed, the set of cellular base stations is represented as $\mathbf{B} = \{b_i\}_{i=1}^{N_{bs}}$, where N_{bs} denotes the total number of base stations.

Definition 2 (Cellular Traffic). Given a set of cellular base stations, its corresponding cellular traffic data is denoted by $\mathbf{X} = \{\mathcal{X}_i\}_{i=1}^{N_{bs}}$. The traffic for the i-th base station is represented by $\mathcal{X}_i = \{x^{i,j}\}_{j=1}^{N_t}$, where N_t indicates the length of the time series. The traffic for the i-th base station during time period j is represented as $x^{i,j} = \{v_k^{i,j}\}_{k=1}^{N_{ts}}$, where N_{ts} is the number of timestamps within each period.

For example, if each period corresponds to one day and is divided into 24 time slots, then $v_k^{i,j}$ refers to the network traffic volume at the i-th base station during the k-th hour on the j-th day.

Problem 1 (Cellular Traffic Generation). Based on the definitions above, the cellular traffic generation problem is defined as follows: *given a base station b_i and the target cellular traffic generation period j, the corresponding cellular traffic $x^{i,j}$ is generated.* Our goal is to generate the cellular traffic $\hat{\mathbf{X}}$ for the target cellular base stations $\hat{\mathbf{B}}$ within the specified time period based on certain historical cellular traffic $\mathbf{X}$ and its corresponding cellular base stations $\mathbf{B}$. The objectiveness is to minimize the distribution difference between $\hat{\mathbf{X}}$ and $\mathbf{X}$.

3.2 Complex patterns of cellular traffic

To analyze the complex patterns of cellular traffic, we visualized Internet traffic data from two typical base stations in the Milan dataset (see more details about the dataset in Section 6.1), as shown in Figure 1. Specifically, Figure 1(a) illustrates the average daily traffic at these two base stations from November 2 to January 1, 2014. Figure 1(b) and Figure 1(c) display traffic statistics over a 7-day period for each base station, segmented into hourly intervals. Noting that these traffic data are normalized by Min-Max normalization.

Pattern 1: ***Long-term and short-term periodic patterns.*** Analysis of the two-month traffic data in Figure 1(a) reveals that network traffic is lowest on Sundays (November 3, 10, 17, 24 and December 1, 8, 15, 22, 29), while midweek traffic is consistently higher. This recurring weekly pattern indicates long-term periodicity in cellular traffic. Similarly, Figures 1(b) and 1(c) show consistent daily fluctuations, where traffic peaks and declines follow a regular hourly pattern, highlighting short-term periodicity. These observations suggest that cellular traffic exhibits structured variations on different time scales. Therefore, it is essential to effectively capture both long-term trends and short-term fluctuations. Relying solely on a single periodic pattern to represent time periods may fail to fully capture these periodic dependencies. Instead, a modeling approach that integrates both long-term temporal patterns and short-term correlations is necessary for an accurate representation of traffic.

Pattern 2: ***Spatial pattern.*** As shown in Figure 1(a), the traffic volumes differ significantly between the two base stations, with Base Station 1 experiencing lower network traffic than Base Station 2. However, their overall traffic trends remain similar, with less traffic on weekends compared to weekdays. Additionally, Base Station 1 exhibits a downward trend in traffic, whereas Base Station 2 shows an opposite pattern, further highlighting spatial heterogeneity in cellular traffic. This suggests that while base stations may share global traffic trends, they also exhibit distinct local variations. Effectively modeling both inter-station similarities and local differences remains a key challenge in cellular traffic generation, necessitating an approach that can distinguish unique spatial patterns while preserving overall correlations.

Pattern 3: ***Uncertainty pattern.*** As mentioned in Section 1, network traffic at the same base station within the same time period exhibits inherent uncertainty. For example, in Figure 1(a), while the weekly traffic patterns of a base station remain similar in trend throughout four weeks, the actual traffic volumes vary considerably. If a traffic generation model accounts only for spatio-temporal patterns without incorporating uncertainty, it will produce fixed-volume traffic, failing to capture the natural fluctuations present in real cellular traffic. Therefore, effective modeling of these unpredictable variations is essential to ensure that generated traffic accurately reflects both large-scale patterns and fine-grained volume differences observed in real-world data.

3.3 Overview

Figure 2 illustrates the framework of STOUTER, which employs a denoising network that integrates spatio-temporal information to iteratively refine generated data and produce large-scale cellular traffic data.

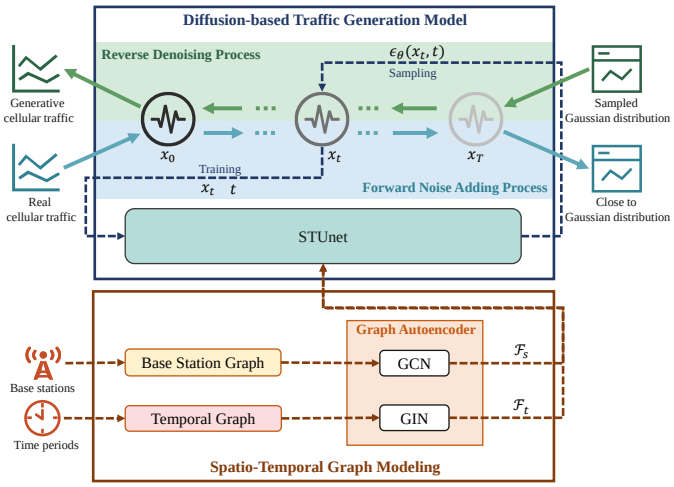


Fig. 2: Framework of STOUTER.

First, we build a temporal graph structure to capture the underlying temporal periodic patterns by embedding temporal graph nodes. Second, to differentiate the spatial characteristics of various base station regions, we construct a graph-based representation of base stations, encoding each base station node using a graph autoencoder. Third, we design a spatio-temporal denoising diffusion model and introduce Spatio-Temporal UNet (STUnet) as the denoising network. During training, STUnet first generates noisy traffic data through the forward diffusion process. Then, in the backward process, it recovers realistic traffic by progressively refining samples drawn from a random Gaussian distribution, effectively simulating uncertainty in cellular traffic. Leveraging temporal periodic patterns and spatial base station representations, STUnet denoises the sampled traffic data, ultimately generating realistic cellular traffic that preserves inherent uncertainty patterns.

4 Spatio-Temporal Graph Modeling

In this section, we present the construction of the temporal graph and the base station graph, which capture the temporal periodicity and spatial dependencies of cellular traffic, respectively. We then utilize graph autoencoders to obtain latent representations of the corresponding graph nodes, preserving essential spatio-temporal features for downstream traffic generation. As shown in Figure 3, the encoder generates corresponding node representations for the temporal graph and base station graph. The decoder

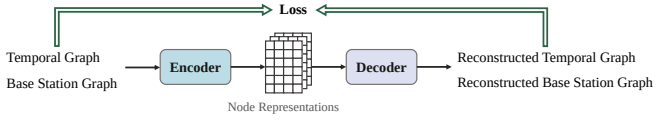


Fig. 3: Spatio-temporal graph autoencoder.

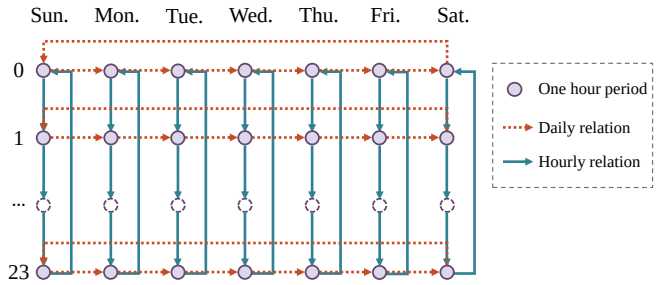


Fig. 4: The temporal graph, where the nodes consist of time periods and the edges consist of two types: daily relation and hourly relation.

then reconstructs the spatio-temporal graph. Finally, the model is optimized by calculating the loss between the reconstructed spatio-temporal graph and the original spatio-temporal graph.

4.1 Temporal graph for time period encoding

To capture the temporal variations in cellular traffic, we should consider its long-term and short-term characteristics. To this end, we construct a temporal graph structure, as illustrated in Figure 4.

We model hourly cellular traffic statistics as graph nodes, where a single day is divided into 24 time periods, each corresponding to one of the 24 graph nodes. Adjacent temporal nodes are connected by directed edges, representing the short-term hourly progression of traffic throughout the day. Additionally, to capture long-term temporal dependencies, we introduce directed edges between the same hourly nodes across different days within a week (Sunday to Saturday, 7 days). These connections encode the chronological relationships between corresponding time periods, effectively modeling the recurring weekly traffic patterns.

Let $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{A}_t, \mathcal{H}_t)$ denote the temporal graph, where $\mathcal{V}_t$ is the set of nodes, each corresponding to a one-hour time period. $\mathcal{A}_t$ consists of the edges that describe connections, which are categorized into two types: the first type represents hour-level temporal relationships, capturing sequential dependencies within a 24-hour cycle; the second type represents daily-level temporal relationships, modeling recurring traffic patterns across the same hour in different days over a week. $\mathcal{H}_t$ is the initial representation of the nodes, encoded using one-hot encoding.

The node representations for the temporal graph are learned through a graph neural network (GNN), which embeds and trains the nodes to capture their structural relationships. To achieve this, we utilize the encoder-decoder framework introduced in GraphMAE [47] to facilitate the learning of these representations. Given its strong capability in capturing graph structural information, we adopt the Graph Isomorphism Network (GIN) [48] as both the graph encoder and decoder in the learning process:

$$\begin{aligned} \mathcal{F}_t &= GIN_E(\mathcal{A}_t, \mathcal{H}_t), \\ \mathcal{Z}_t &= GIN_D(\mathcal{A}_t, \mathcal{F}_t), \end{aligned} \tag{1}$$

where $\mathcal{F}_t$ denotes the temporal node representation produced by the encoder, and the decoder generates the restored node representation $\mathcal{Z}_t$.

4.2 Base station graph for spatial representation

Traffic patterns vary across different regions due to human activity dynamics. To effectively differentiate cellular traffic characteristics across base stations, we perform representation learning on cellular base stations and construct a base station graph, which captures the spatial dependencies and relationships among them.

Let $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{A}_s, \mathcal{H}_s)$ denote the base station graph, where $\mathcal{V}_s$ represents the set of base station nodes and $\mathcal{A}_s$ defines the adjacency relationships between base stations. The duality of wireless network coverage (covered/uncovered) fulfills the essential requirement for connectivity in real-world scenarios. In particular, during cellular network planning, binary edges can reduce the complexity of the model and are more suitable for the topological analysis of large-scale networks [49]. Consequently, an undirected binary edge is established between two base stations if their distance is less than a predefined threshold, denoted as d_{BS}. $\mathcal{H}_s$ represents the initial feature representation of the base station graph. In large-scale urban base station deployments, 99% of the base stations are located within $1\,km$ of the nearest base station [50]. Therefore, we set the threshold $d_{BS} = 1\,km$ in our study.

Figure 5 illustrates the base station graph, where POIs within each base station's coverage area are categorized into eight groups: *Education*, *Medical*, *Public*, *Entertainment*, *Traffic*, *Food*, *Shop*, and *Others*. To generate the initial node representations, we compute the number of POIs in each category covered by a given base station, ensuring that the spatial characteristics of different regions are well captured.

For base station graph representation learning, we employ an encoder-decoder model based on GNNs, following a similar node embedding approach as used for temporal graphs. Considering that GCNs [51] excel in node classification tasks, we thus adopt GCNs as the encoder-decoder architecture to learn the graph node representations of base stations, i.e.:

$$\begin{aligned} \mathcal{F}_s &= GCN_E(\mathcal{A}_s, \mathcal{H}_s), \\ \mathcal{Z}_s &= GCN_D(\mathcal{A}_s, \mathcal{F}_s), \end{aligned} \tag{2}$$

where $\mathcal{F}_s$ denotes the encoded base station node representation, and $\mathcal{Z}_s$ denotes the reconstructed node representations generated by the decoder.

4.3 Optimization function

To effectively pretrain the temporal graph and base station graph, we optimize the encoders GIN_E and GCN_E by minimizing the scaled cosine error (SCE) loss function, which ensures that the learned node representations retain essential structural and feature information:

$$\mathcal{L}_{pre} = \frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}, h_i \in \mathcal{H}, z_i \in \mathcal{Z}} \left(1 - \frac{h_i^T z_i}{\|h_i\| \cdot \|z_i\|}\right)^{\gamma}, \tag{3}$$

where the scaling factor $\gamma > 1$, v_i represents the final node embedding from $\mathcal{V}_t$ or $\mathcal{V}_s$, h_i denotes the original node

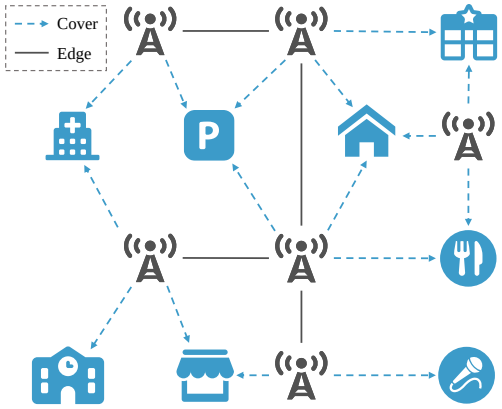


Fig. 5: The base station graph, where the nodes consist of base stations and eight types of POIs (i.e., Education, Medical, Public, Entertainment, Traffic, Food, Shop, and Others). Within the threshold d_{BS} range, edges are established between base stations, and base stations cover POIs.

feature from $\mathcal{H}_t$ or $\mathcal{H}_s$, and z_i is the restored node feature from the decoder. We introduce a scaling factor to the cosine error loss to assign higher weights to samples with larger errors in the reconstructed representation.

5 Diffusion-based Traffic Generation

In this section, we introduce a spatio-temporal diffusion model for cellular traffic generation, incorporating both the spatio-temporal patterns and the uncertainty pattern.

The diffusion model simulates traffic uncertainty through a two-phase process: forward diffusion and reverse denoising. In the forward process, the model progressively injects noise into real traffic data, gradually transforming it into a near-random prior distribution. In the reverse process, the model starts from randomly sampled noise and iteratively removes noise to reconstruct realistic traffic samples. To effectively capture uncertainty variations in traffic data, we propose a cellular traffic generation method based on a spatio-temporal diffusion framework. This method employs two Markov chains: one is a forward Markov chain that adds noise to real traffic data, mapping the real traffic distribution to a predefined prior distribution (e.g., a Gaussian distribution). The other is a reverse Markov chain that reconstructs the true traffic distribution from the prior distribution by iteratively refining generated traffic samples. Given the complex spatio-temporal patterns of cellular traffic, it is crucial to not only learn the noise patterns in traffic generation but also to align the generated data with real-world spatio-temporal structures during the denoising process. To this end, we design STUnet to effectively guide the traffic reconstruction process to preserve realistic patterns.

5.1 Forward cellular traffic noise adding process

In the forward diffusion process, cellular traffic data is incrementally corrupted by adding noise. As noise accumulates, the data distribution gradually approaches a Gaussian distribution.

Given a real cellular traffic data sample $x \sim q(x)$ [1], we generate a sequence of noisy traffic data $x_0 \sim x_T$ through a forward Markov chain, where noise is progressively introduced. The time steps for noise addition are indexed as $1 \sim T$, where x_0 represents the original noise-free traffic sample, and $x_1 \sim x_T$ denotes the traffic data with increasing levels of noise. This process can be interpreted as gradually erasing the spatio-temporal characteristics of the real traffic distribution, such that the final state x_T approximates a Gaussian distribution. The process is described by the following equation:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{4}$$

The transition probability function in this diffusion process follows a Gaussian distribution, where the mean is $\mu = \sqrt{1-\beta_t}x_{t-1}$ and the variance is $\sigma^2 = \beta_t$, i.e.:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{5}$$

where $\beta_t \in (0,1)$ is a hyperparameter controlling the diffusion intensity at time step t and I represents the identity matrix. The sequence $\{\beta_1, \beta_2, ..., \beta_T\}$ is designed to be increasing. As the time step t progresses, β_t gradually increases, resulting in more noise being added at each step. We utilize linear scheduling to compute the intermediate noise levels for the β_t sequence, starting from a value of 10^{-4} and ending at 0.02. Using this Gaussian transition kernel, we can derive the probability function $q(x_t|x_0)$ at any time step $t \in \{0, 1, ..., T\}$. Defining $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{i=0}^{t} a_i$, we have

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)\, I\right). \tag{6}$$

Consequently, the noisy traffic sample at time step t can be computed by combining the original traffic data and Gaussian noise:

$$x_t = \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, \tag{7}$$

where $\epsilon \sim \mathcal{N}(0, I)$ denotes the added Gaussian noise.

5.2 Reverse cellular traffic data denoising process

In the reverse process, we first sample a random Gaussian distribution to serve as the initial state for the generated cellular traffic. Then, we iteratively refine the traffic data through a reverse denoising operation to restore the expected distribution.

Since directly computing $q(x_0|x_{1:T})$ is intractable, we follow Denoising Diffusion Probabilistic Models (DDPM) [38], which define the reverse process as a Markov chain parameterized by a neural network. We thus train a neural network p_θ to model this reverse denoising process.

Given an initial Gaussian-distributed sample x_T, the reverse diffusion process from time step T to 0 follows:

$$\begin{aligned} p_\theta(x_{0:T}) &= p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \\ p_\theta(x_{t-1}|x_t) &= \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right). \end{aligned} \tag{8}$$

1. To simplify the presentation, we omit the superscript of $x^{i,j}$ as x.

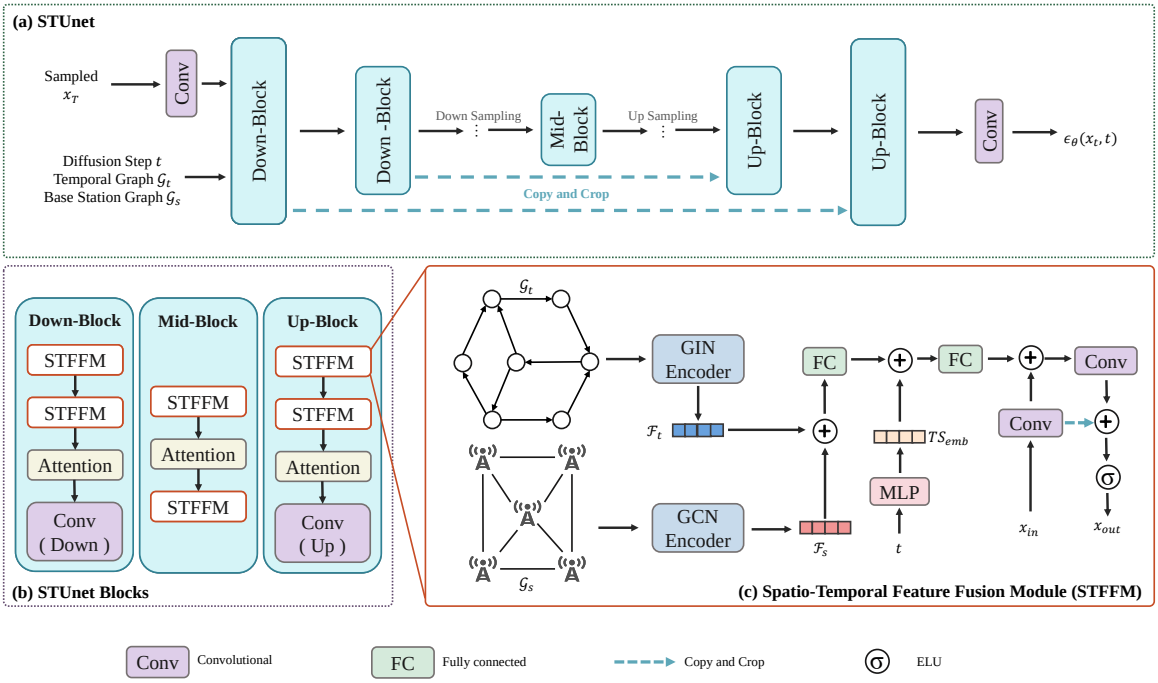


Fig. 6: Denoising network for the noise level prediction in reverse denoising process. Figure (a) provides an overview of the STUnet framework. Figure (b) depicts the internal structural composition of each STUnet block. Figure (c) illustrates the STFFM, which is used for learning the spatio-temporal patterns of cellular traffic.

Based on Equation (6), assuming that x_0 is known, the posterior distribution can be expressed as:

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}\left(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t I\right). \tag{9}$$

Meanwhile, with the Bayesian formula, we have:

$$\begin{aligned} q(x_{t-1}|x_t, x_0) &= \frac{q(x_t, x_{t-1}, x_0)}{q(x_t, x_0)} \\ &= \frac{q(x_t|x_{t-1}, x_0) \cdot q(x_{t-1}, x_0)}{q(x_t|x_0) \cdot q(x_0)} \\ &= \frac{q(x_t|x_{t-1}, x_0) \cdot q(x_{t-1}|x_0) \cdot q(x_0)}{q(x_t|x_0) \cdot q(x_0)} \\ &= \frac{q(x_t|x_{t-1}, x_0) \cdot q(x_{t-1}|x_0)}{q(x_t|x_0)}. \end{aligned} \tag{10}$$

According to Equation (6), we can determine $q(x_{t-1}|x_0)$ and $q(x_t|x_0)$. By combining this information with Equations (7), (9) and (10), we can derive

$$\begin{aligned} \tilde{\mu}_t &= \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right), \\ \tilde{\beta}_t &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t, \end{aligned} \tag{11}$$

where $\tilde{\beta}_t$ can be computed directly from the diffusion hyperparameter β_t. Therefore, the mean $\mu_\theta(x_t, t)$ in Equation (8) is computed as:

$$\mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right), \tag{12}$$

where θ represents the trainable parameters of the denoising neural network. $\epsilon_\theta(x_t, t)$ is the estimated noise component, learned by STUnet, which we introduce later.

Finally, cellular traffic data is reconstructed at $t = 0$ by iteratively applying the reverse Markov chain to remove noise step by step.

5.3 Spatio-temporal fusion denoising network

5.3.1 *Spatio-temporal Unet*

To facilitate the reverse denoising process of the spatio-temporal diffusion model, we design the Spatio-Temporal UNet (STUnet) , as illustrated in Figure 6(a). Within STUnet, we introduce the Spatio-Temporal Feature Fusion Module (STFFM), which guides the model in capturing temporal periodic patterns and spatial patterns of cellular traffic.

The STUnet framework consists of two main components: an encoder that corresponds to the down-sampling process (from Down-Block to Mid-Block) and a decoder that corresponds to the up-sampling process (from Mid-Block to Up-Block). In the Down-Block, extracted features are cropped during down-sampling, while in the Up-Block, the decoder concatenates these cropped features for output. Finally, the predicted $\epsilon_\theta(x_t, t) = STUnet(x_T, t)$ is produced via a convolutional layer.

5.3.2 *Spatio-temporal Unet Blocks*

Figure 6(b) illustrates the three core components of STUnet: Down-Block, Mid-Block, and Up-Block. STUnet consists of Down-Block and Up-Block networks with the same number of layers, which are connected by Mid-Block. Each STUnet block is composed of STFFM and attention networks. The Down-Block ultimately outputs through a down-sampling convolutional layer, while the Up-Block reconstructs the final traffic data through an up-sampling convolutional layer.

In the Mid-Block, the attention network between the two STFFMs is implemented using a multi-head attention mechanism. This mechanism projects the sequence information encoded by the encoder into multiple subspaces and extracts various semantic information output by the STFFM. In our work, we set the number of attention heads to 4, allowing us to focus on the diverse features of the input, including historical traffic, time step, spatial semantics, and temporal semantics. Given $x_{in} \in \mathbb{R}^{n \times m}$ represents the intermediate layer input of STUnet Blocks, where m and n denote the feature dimension and sequence length of the cellular traffic, respectively. The attention network in the Mid-Block can be expressed as:

$$\begin{aligned} Q_i &= x_{in} W_i^Q, \\ K_i &= x_{in} W_i^K, \\ V_i &= x_{in} W_i^V, \\ he_i &= softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i, \\ MHAtt(Q,K,V) &= Concat(\{he_i\}_{i=1}^4) W^O, \end{aligned} \tag{13}$$

where $i \in \{1,2,3,4\}$ and $d_k = m/4$ represent the sequence number and feature dimension of each attention head. $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{m \times d_k}$ and $W^O \in \mathbb{R}^{m \times m}$ are learnable parameter matrices.

In the Down-Block and Up-Block, the attention network following the two STFFMs is implemented using a multi-head linear attention mechanism. This approach reduces the computational complexity of the attention layer during the multi-layer encoding and decoding process while effectively capturing the coarse-grained semantic information of the cellular traffic. Each head of the linear attention mechanism can be expressed as:

$$\begin{aligned} \hat{Q}_i &= Softmax(Q_i, dim = -2), \\ \hat{K}_i &= Softmax(K_i, dim = -1), \\ LAtt(Q_i, K_i, V_i) &= \hat{Q}_i \cdot \left(\hat{K}_i^T V_i\right), \end{aligned} \tag{14}$$

where $\hat{Q}_i$ and $\hat{K}_i$ denote the normalization of the query in the feature dimension and the normalization of the key in the sequence dimension, respectively. The importance weights of spatio-temporal features, time step features, and historical traffic features are dynamically adjusted through the multi-head attention layers in STUnet.

5.3.3 Spatio-temporal feature fusion module

As depicted in Figure 6(c), the temporal graph $\mathcal{G}_t$ and base station graph $\mathcal{G}_s$ are pre-trained to obtain the corresponding time period representation $\mathcal{F}_t$ and base station representation $\mathcal{F}_s$, which serve as spatio-temporal priors for the generated traffic data. To enhance the denoising learning, we design the spatio-temporal feature fusion module (STFFM) for each residual block in STUnet, combining the extracted spatio-temporal information with the diffusion step information t to guide the restoration process.

We employ sinusoidal position encoding to represent the step position information TS_{Emb} for diffusion steps. Subsequently, we utilize a multilayer perceptron (MLP) to embed the diffusion step t:

$$TS_{Emb} = MLP\left(SinPosEmb\left(t\right)\right). \tag{15}$$

Given base station embedding $\mathcal{F}_s$ and time period embedding $\mathcal{F}_t$, the spatio-temporal representation $\mathcal{F}_{st}$ is generated by a fully connected layer. Then $\mathcal{F}_{st}$ is combined with the diffusion step information to serve as guidance for the diffusion generation process:

$$\mathcal{F}_{st} = FC\left(TS_{Emb} + FC\left(Concat\left(\mathcal{F}_s, \mathcal{F}_t\right)\right)\right), \tag{16}$$

For a given randomly sampled x_T, the model first applies a convolution operation and then integrates it with the learned spatio-temporal representation to predict the final noise:

$$\begin{aligned} \hat{x}_{in} &= Conv\left(x_{in}\right), \\ x_{out} &= ELU\left(\hat{x}_{in} + Conv\left(\hat{x}_{in} + \mathcal{F}_{st}\right)\right). \end{aligned} \tag{17}$$

where x_{in} is the input to the current network layer, while x_{out} is the output for the next layer. $ELU(\cdot)$ represents the Exponential Linear Unit (ELU) activation function, which improves stability in training. $Conv(\cdot)$ denotes the convolution operation, essential for feature extraction.

Finally, during the up-sampling process, the output from the corresponding down-sampling residual block is cropped and concatenated, ensuring accurate feature reconstruction.

5.4 Optimization function

During network training, STUnet is optimized by minimizing the difference between the predicted traffic noise and the actual traffic noise. Given the real noisy data as ϵ, ϵ is sampled by Gaussian distribution $\epsilon \sim \mathcal{N}(0, I)$. By incorporating ϵ, the values of x_t at time step t can be computed using Equation (7). The noise component ϵ_t is then calculated as follows:

$$\epsilon_t = \left(1 - \sqrt{\overline{\alpha}_t}\right) x_0 - \sqrt{1 - \overline{\alpha}_t}\epsilon. \tag{18}$$

Given that $\epsilon_\theta(x_t, t)$ denotes the predicted noise component at time step t, as generated by the STUnet, we train the spatio-temporal fusion denoising network using the L1 loss function, which is defined as:

$$\mathcal{L} = \mathbb{E}_{x_0, \epsilon_t}\left[||\epsilon_t - \epsilon_\theta(x_t, t)||^2\right]. \tag{19}$$

6 Performance Evaluation

In this section, we conduct extensive experiments and analysis to validate the effectiveness of our STOUTER using the real-world public datasets.

6.1 Experimental settings

6.1.1 Datasets

We conduct experiments using the public communication dataset known as Call Detailed Records (CDRs) from Italy, curated and provided by the Semantics and Knowledge Innovation Lab. The datasets cover two regions: communication data from **Milan** and **Trentino** [52]. For both the Milan and Trentino datasets, five types of traffic information are recorded in detail: *Internet*, *Received-SMS*, *Sent-SMS*, *Incoming-Call*, and *Outgoing-Call*. Specifically, Internet indicates the network usage for Internet accessing, Incoming-Call and Outgoing-Call correspond to voice call data, and Received-SMS and Sent-SMS correspond to SMS communication data. Milan dataset collection period spanned from

TABLE 1: Statistics about the datasets of Milan and Trentino.

Dataset	Data type	# of BSs
Milan	Internet	9905
	SMS	9916
	Received-SMS	10000
	Sent-SMS	9916
	Call	8376
	Incoming-Call	9901
	Outgoing-Call	9856
Trentino	Internet	4754
	SMS	5698
	Received-SMS	6258
	Sent-SMS	5698
	Call	3209
	Incoming-Call	5551
	Outgoing-Call	5142

November 1, 2013, to January 1, 2014. Trentino dataset collection period spanned from November 1, 2013, to December 30, 2013.

We divide the base station areas according to the grid systems provided in the datasets. In Milan, the number of designated base station areas is 10,000, while in Trentino, it is 11,466. In addition, some existing studies [37] have merged the sent and received data for traffic analysis tasks. We also use *Call* (aggregated from Incoming-Call and Outgoing-Call) and *SMS* (aggregated from Received-SMS and Sent-SMS) as experimental datasets. After data preprocessing, the detailed information of the experimental dataset is presented in Table 1.

6.1.2 Baselines

We compare our model against two categories of cellular traffic generation methods, that is *GAN-based* and *VAE-based* methods. Additionally, to verify the effectiveness of our proposed STUnet, we also compare the diffusion model approach with *WaveNet-based* [53] as the denoising network framework. The baseline methods are described below.

- **TCN-GAN [18]**: Generative Adversarial Network (GAN) consists of two components: a generator and a discriminator. The generator transforms a random noise vector, which is typically sampled from a prior distribution such as a Gaussian distribution, into realistic data samples. The discriminator distinguishes between synthetic traffic samples generated by the generator and real traffic data. In this method, we exclude the classification of long-term and short-term periodic traffic as well as the urban knowledge graph and instead adopt the core TCN-GAN network structure for cellular traffic generation.
- **VAE [54]**: Variational Autoencoder (VAE) consists of two components: an encoder and a decoder. The encoder maps the input data to a low-dimensional latent space, while the decoder reconstructs the vectors from the latent space back to the original data. This method originally employs a hybrid convolutional VAE for text generation. In our experiments, we adapt this model for cellular traffic generation by modifying its architecture to accommodate traffic data characteristics.
- **DiffWave [53]**: DiffWave is a diffusion model primarily used for audio generation. It features a non-autoregressive structure, which effectively leverages the strong capabilities of sound wave modeling. In our experiments, we use WaveNet from DiffWave as the denoising network for a diffusion model based cellular traffic generation method.

Due to DiffTraj [45] focusing on GPS trajectory generation and lacking the necessary information to construct knowledge graphs, we cannot directly compare our method with DiffTraj, KSTDiff [46], STK-Diff [31], and DiffSTG [44].

6.1.3 Evaluation metrics

To measure the numerical discrepancy between the generated cellular traffic and real traffic data, we employ commonly used error metrics, including *Mean Absolute Error (MAE)* and *Root Mean Square Error (RMSE)*.

Let $\mathbf{X}_g$ and $\mathbf{X}_r$ denote the generated traffic and the real traffic across all distributions, respectively. As defined in Section 3.1, $\mathcal{X}_i^g$ and $\mathcal{X}_i^r$ represent the generated traffic and the real traffic of the i-th base station. Thus, MAE is calculated as:

$$MAE(\mathbf{X}_g, \mathbf{X}_r) = \frac{1}{n}\sum_{i=1}^{n}|\mathcal{X}_i^g - \mathcal{X}_i^r|, \tag{20}$$

and RMSE is calculated as:

$$RMSE(\mathbf{X}_g, \mathbf{X}_r) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mathcal{X}_i^g - \mathcal{X}_i^r)^2}, \tag{21}$$

where n is the total number of base stations.

Additionally, we ultilize *Jensen-Shannon Divergence (JSD)* [55] to evaluate the similarity between probability distributions of generated and real traffic. JSD is a symmetric divergence measure based on Kullback-Leibler (KL) divergence, making it well-suited for comparing two distributions. The KL divergence between the generated traffic and the real traffic can be expressed as:

$$KL(\mathbf{X}_g \parallel \mathbf{X}_r) = \sum_i \mathbf{X}_g(i)\log\frac{\mathbf{X}_g(i)}{\mathbf{X}_r(i)}. \tag{22}$$

The JSD between $\mathbf{X}_g$ and $\mathbf{X}_r$ is then calculated as:

$$JSD(\mathbf{X}_g, \mathbf{X}_r) = \frac{KL(\mathbf{X}_g \parallel \mathbf{X}_m) + KL(\mathbf{X}_r \parallel \mathbf{X}_m)}{2}, \tag{23}$$

where $\mathbf{X}_m = \frac{\mathbf{X}_g + \mathbf{X}_r}{2}$ denotes the average distribution between $\mathbf{X}_g$ and $\mathbf{X}_r$.

6.1.4 Implementation

All experiments were executed on a server equipped with an NVIDIA GeForce RTX 3090 GPU (64 GB VRAM), an Intel(R) Core(TM) i7-10700K CPU (3.80 GHz), 80 GB of system RAM, and implemented using Python 3.7 with the PyTorch deep learning framework.

TABLE 2: Performance comparisons of our proposed STOUTER and baseline methods using the metrics of MAE, RMSE, and JSD ($\times 10^{-4}$). Detailed results for Incoming-Call, Outgoing-Call, Received-SMS, and Sent-SMS datasets from Milan and Trentino. The optimal results are highlighted in **bold** and the suboptimal results are underlined.

Methods	Milan											
	Incoming-Call			Outgoing-Call			Received-SMS			Sent-SMS		
	MAE	RMSE	JSD	MAE	RMSE	JSD	MAE	RMSE	JSD	MAE	RMSE	JSD
TCN-GAN	0.1282	0.1887	1.9996	0.2300	0.2752	3.9110	0.2393	0.3487	4.7695	0.2142	0.2640	2.0356
VAE	0.0717	0.0935	0.4032	0.1023	0.1411	0.7663	0.0219	0.0469	**0.3288**	0.0885	0.1156	0.5220
DiffWave	0.0787	0.0993	0.4366	0.0970	0.1326	0.6977	0.0286	0.0518	0.3372	0.0849	0.1119	0.5068
STOUTER	**0.0689**	**0.0894**	**0.3837**	**0.0809**	**0.1120**	**0.5336**	**0.0189**	**0.0459**	0.3295	**0.0772**	**0.1017**	**0.4353**
Methods	**Trentino**											
	Incoming-Call			Outgoing-Call			Received-SMS			Sent-SMS		
	MAE	RMSE	JSD	MAE	RMSE	JSD	MAE	RMSE	JSD	MAE	RMSE	JSD
TCN-GAN	0.1913	0.2326	1.9676	0.2005	0.2612	2.7834	0.1972	0.2941	3.8463	0.1438	0.1709	1.1441
VAE	0.0725	0.0936	0.3818	0.1040	0.1441	0.8057	0.0301	0.0575	**0.3991**	0.0914	0.1169	0.4767
DiffWave	0.0839	0.1040	0.4633	0.0984	0.1362	0.7088	0.0325	0.0620	0.4216	0.1047	0.1326	0.5529
STOUTER	**0.0720**	**0.0909**	**0.3608**	**0.0820**	**0.1105**	**0.5320**	**0.0268**	**0.0564**	0.4006	**0.0824**	**0.1056**	**0.4262**

TABLE 3: Performance comparisons of our proposed STOUTER and baseline methods using the metrics of MAE, RMSE, and JSD ($\times 10^{-4}$). Detailed results for Internet, Call, and SMS datasets from Milan and Trentino. The optimal results are highlighted in **bold** and the suboptimal results are underlined.

Methods	Milan								
	Internet			Call			SMS		
	MAE	RMSE	JSD	MAE	RMSE	JSD	MAE	RMSE	JSD
TCN-GAN	0.4183	0.4693	2.6895	0.1357	0.1598	1.0161	0.2714	0.4335	9.2472
VAE	0.1057	0.1444	0.8017	0.0818	0.1071	0.4765	0.0612	0.0735	0.2652
DiffWave	0.1329	0.1707	0.9199	0.0860	0.1086	0.5573	0.0721	0.0837	0.2213
STOUTER	**0.0781**	**0.1088**	**0.4843**	**0.0697**	**0.0925**	**0.3963**	**0.0583**	**0.0715**	**0.1997**
Methods	**Trentino**								
	Internet			Call			SMS		
	MAE	RMSE	JSD	MAE	RMSE	JSD	MAE	RMSE	JSD
TCN-GAN	0.4786	0.5312	2.9059	0.1267	0.1488	0.8921	0.0977	0.1103	0.3402
VAE	0.1044	0.1440	0.7938	0.0810	0.1018	0.3520	0.0721	0.0896	**0.3369**
DiffWave	0.0986	0.1348	0.8333	0.0809	0.1005	0.5050	0.1540	0.1668	0.4479
STOUTER	**0.0827**	**0.1139**	**0.5196**	**0.0719**	**0.0915**	**0.3076**	**0.0613**	**0.0808**	0.3380

6.2 Overall performance

Table 2 and Table 3 presents the results of the performance evaluation of our model compared to baseline models in multiple datasets from Milan and Trentino. Specifically, Table 2 presents the evaluation results for the Incoming-Call, Outgoing-Call, Received-SMS, and Sent-SMS datasets. Table 3 presents the evaluation results for the Internet, Call and SMS datasets.

The results show that our method outperforms the baselines over the metrics of MAE, RMSE, and JSD. For example, on the Internet traffic datasets from Milan and Trentino, our model on average improves MAE by 19.23%, RMSE by 18.35%, and JSD by 52.77%, when compared to the baselines.

In contrast, TCN-GAN exhibits the worst performance among the models. This is mainly due to the instability of GAN-generated data, which lacks consistency without strong guidance signals such as knowledge graphs and detailed traffic cycle patterns. Meanwhile, VAE and DiffWave demonstrate varying strengths in different datasets. However, their performance is not consistently superior across all datasets, leading to dataset-dependent biases.

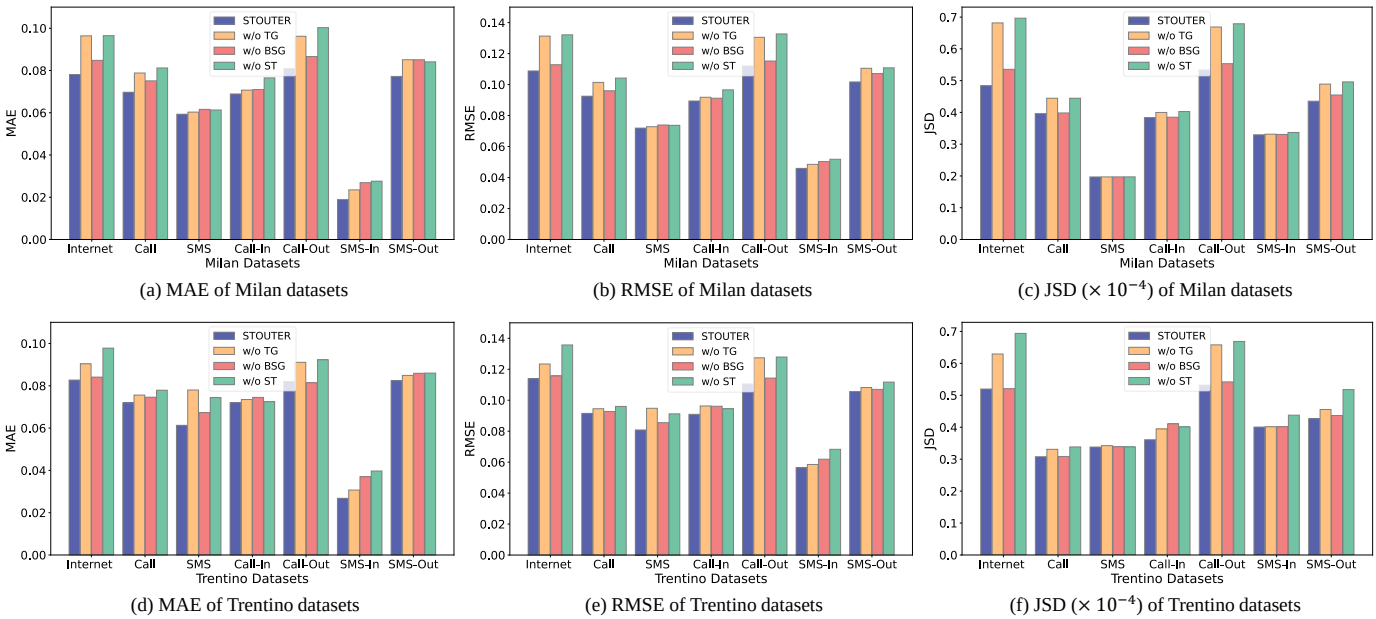
TABLE 4: Comparisons of model overheads.

Methods	Parameters (M)	Training time ($\times 10^4 s$)	Inference time (s)
TCN-GAN	3.3821	13.0474	0.0468
VAE	1.1238	4.4291	0.0159
DiffWave	1.0051	7.5435	0.1082
STOUTER	2.9525	6.5966	0.3783

Our method achieves suboptimal results on the JSD metric for the Received-SMS datasets from both regions and the SMS dataset from Trentino. This is due to the high randomness in the Received-SMS dataset compared to other datasets, where cellular traffic patterns exhibit weak correlations over different time periods and base stations, making distribution modeling more challenging. Since our model integrates spatio-temporal information via STFFM, it introduces less stochastic variability compared to VAE. Despite this, our model consistently achieves the best overall performance in other datasets.

We analyze the time and space complexity of our proposed model and baseline models using the Trentino Internet dataset, including comparisons on model parameters,

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3591183



(a) MAE of Milan datasets (b) RMSE of Milan datasets (c) JSD ($\times 10^{-4}$) of Milan datasets

(d) MAE of Trentino datasets (e) RMSE of Trentino datasets (f) JSD ($\times 10^{-4}$) of Trentino datasets

Fig. 7: Performance impact of temporal graph and base station graph for Internet, Call, SMS, Incoming-call ('Call-In' for short), Outgoing-Call ('Call-Out' for short), Received-SMS ('SMS-In' for short), and Sent-SMS ('SMS-Out' for short) datasets from Milan and Trentino.

training time, and inference time. As shown in Table 4, the TCN-GAN method has the most parameters due to its dual network design (generator and discriminator). STOUTER's spatio-temporal fusion module, utilizing a multi-layer U-Net, results in slightly more parameters than VAE and DiffWave. Training times show that TCN-GAN takes the longest, while STOUTER's is slightly higher than VAE's due to its complex denoising structure. For inference cost, with a batch size of 128, diffusion model-based methods require longer times than TCN-GAN and VAE.

Overall, while STOUTER incurs slightly more memory overhead and inference time, it achieves much better generation results. Given that the purpose of cellular traffic generation is often to create more data for research in areas with limited datasets, the demand for real-time performance could not be high. Therefore, the trade-off of increased inference time for more realistic generated data is acceptable.

6.3 Ablation study and variants analysis

6.3.1 Ablation of spatio-temporal graph module

To access the impact of temporal graph representation learning and base station graph representation learning on the model's performance, we conduct ablation experiments by comparing STOUTER with three modified versions: STOUTER without the time period representation (denoted as **w/o TG**), STOUTER without the base station representation (denoted as **w/o BSG**), and STOUTER without both the time period and base station representations (denoted as **w/o ST**). By evaluating these variants, we analyze how spatio-temporal feature learning enhances the diffusion model's ability to generate realistic cellular traffic patterns.

We present the performance comparison of these models across three evaluation metrics (MAE, RMSE, and JSD) on the Milan and Trentino datasets in Figure 7. The results indicate that removing the temporal and base station graph representation modules degrades performance. Notably, for the SMS data, the optimization effect on JSD is relatively weaker. This is particularly evident in the Received-SMS dataset, where the spatio-temporal correlation of the overall traffic behavior patterns is inherently weak, limiting the effectiveness of the spatio-temporal model in learning its distribution. Despite this, for other metrics, our model consistently demonstrates superior performance, validating the importance of integrating spatio-temporal representations in cellular traffic generation.

6.3.2 Variants of spatio-temporal graph modeling

When modeling spatio-temporal graphs, we employed the GIN [48] graph autoencoder for temporal graphs and the GCN [51] graph autoencoder for base station graphs. To assess their effectiveness, we conducted a variant analysis of graph neural networks. For temporal graphs, we compared three graph node encoding methods: GAT [56], GCN, and GraphSAGE [57]. For base station graphs, we evaluated three variant methods: GAT, GIN, and GraphSAGE.

The results are presented in Table 5. Our methods demonstrate superiority over other variant methods, achieving an improvement of at least 1.74% across various metrics. The GIN model, with its strong ability to capture graph structural information, effectively identifies both long-term and short-term periodicity in temporal graphs. Meanwhile, for the base station graph, modeling the correlations among surrounding base stations is crucial. The GCN excels in this regard by effectively capturing the connections between base station nodes and using the features of neighboring nodes to update the representation of each graph node.

TABLE 5: Performance comparisons of model variants using different graph modeling methods over metrics of MAE, RMSE, and JSD ($\times 10^{-4}$). The optimal results are highlighted in **bold** and the suboptimal results are underlined.

Graph Type	Variants	Milan			Trentino		
		MAE	RMSE	JSD	MAE	RMSE	JSD
Temporal Graph	GAT	0.0850	0.1123	0.5334	0.0874	0.1208	0.5738
	GCN	0.0847	0.1122	0.5352	0.0853	0.1181	0.5577
	GraphSAGE	0.0820	0.1111	0.5308	0.0855	0.1173	0.5636
Base Station Graph	GAT	0.0889	0.1177	0.5392	0.0905	0.1249	0.612
	GIN	0.0815	0.1112	0.5313	0.0846	0.1172	0.5521
	GraphSAGE	0.0808	0.1107	0.5271	0.0852	0.1166	0.5623
Ours		**0.0781**	**0.1088**	**0.4843**	**0.0827**	**0.1139**	**0.5196**

TABLE 6: Performance comparisons of model variants for fusion mechanisms within the STFFM using the metrics of MAE, RMSE, and JSD ($\times 10^{-4}$). The optimal results are highlighted in **bold** and the suboptimal results are underlined.

Variants	Milan			Trentino		
	MAE	RMSE	JSD	MAE	RMSE	JSD
WeightFu	0.0838	0.1128	0.5497	0.0954	0.1309	0.7360
GateFu	0.0824	0.1128	0.5368	**0.0827**	0.1145	**0.5089**
CrossAtt	0.0998	0.1340	0.7089	0.1015	0.1320	0.6998
DecFu	0.1169	0.1540	1.0059	0.0982	0.1317	0.7352
Ours	**0.0781**	**0.1088**	**0.4843**	**0.0827**	**0.1139**	0.5196

6.3.3 *Variants of fusion mechanism*

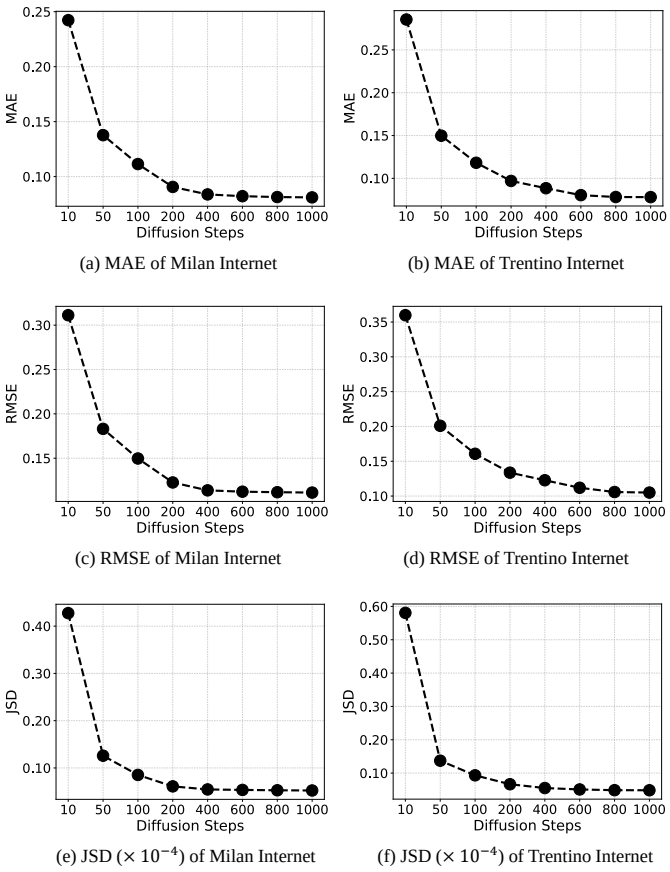
In STFFM, we utilize concatenation and fully connected layers for spatio-temporal feature fusion. To evaluate the effectiveness of feature fusion, we analyze several variants of dynamic feature fusion methods, including dynamic weight fusion (WeightFu) [58], gated mechanism fusion (GateFu) [58], cross-attention mechanism (CrossAtt) [59], and decoupled fusion (DecFu) [60]. The experimental results of the fusion mechanism variants are summarized in Table 6.

Our method effectively retains original temporal and spatio features, leveraging a fully connected layer for complex spatio-temporal interactions. Compared to GateFu, which has weaker feature interaction modeling but lower computational complexity, and WeightFu, better for strong local correlations, our approach excels in capturing long-term traffic features. CrossAtt struggles with noise due to its large parameters, leading to poor performance, while DecFu risks disrupting crucial interactions. Overall, STFFM enhances feature extraction using multi-head attention.

6.4 Parameter study

During training the diffusion model, the diffusion step plays a crucial role in determining the performance of the model. The diffusion step refers to the number of iterations during which noise is added in the forward process and subsequently removed in the reverse process. This parameter directly impacts the quality of generated traffic data. We conduct sensitivity experiments on the diffusion step using the Internet datasets from Milan and Trentino.

Figure 8 shows the performance trends of the model under different diffusion step values by varying t = from 10 to 1000. Our findings indicate that a higher diffusion step improves model performance, leading to generated traffic data that more closely aligns with real traffic patterns. In addition, stability thresholds vary by dataset. In the Milan Internet dataset, performance metrics stabilize at $t = 600$. In the Trentino Internet dataset, stability is achieved at $t = 800$. Overall, increasing the diffusion step enhances data generation quality, but beyond a certain threshold, further increases yield diminishing improvements.



(a) MAE of Milan Internet (b) MAE of Trentino Internet

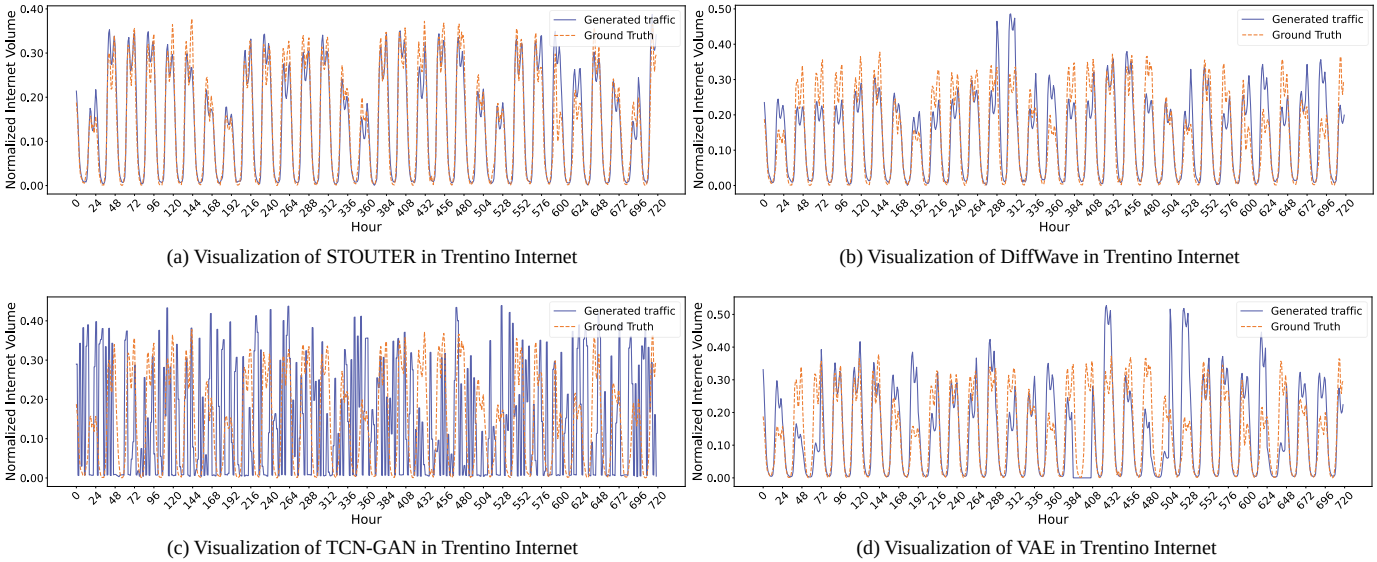
(c) RMSE of Milan Internet (d) RMSE of Trentino Internet

(e) JSD ($\times 10^{-4}$) of Milan Internet (f) JSD ($\times 10^{-4}$) of Trentino Internet

Fig. 8: Performance comparison of different diffusion steps.

6.5 Visualization

We conduct a visualization analysis of long-term generated traffic using the Trentino Internet dataset. Figure 9 presents the visualization results for one month. We compare our STOUTER with three baselines: DiffWave [53], TCN-GAN [18], and VAE [54].



(a) Visualization of STOUTER in Trentino Internet

(b) Visualization of DiffWave in Trentino Internet

(c) Visualization of TCN-GAN in Trentino Internet

(d) Visualization of VAE in Trentino Internet

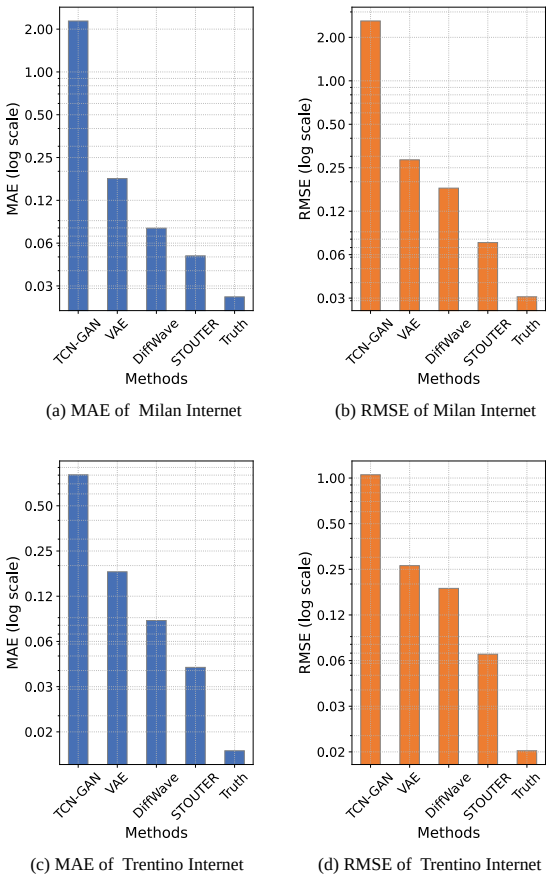
Fig. 9: Visualization comparison on the generated traffic in Trentino's Internet dataset, where 'Generated traffic' represents the traffic generated by different generation models (i.e., STOUTER, DiffWave, TCN-GAN, and VAE), and the 'Ground Truth' represents the real traffic.

Figure 9 reveals that the Internet traffic generated by DiffWave and VAE struggles to capture the periodic patterns of real traffic, resulting in significant deviations where the generated traffic is substantially larger or smaller than the actual traffic during certain hours. The traffic generated by TCN-GAN is generally cluttered and does not align with the real traffic trends, with only a small portion of the generated data closely matching the actual values. In contrast, our method effectively reconstructs the overall trend of real Internet traffic, with only a few instances showing relatively minor deviations. We calculated the indicators for the visualization samples. It shows that STOUTER (MAE:0.18, RMSE:0.2558) outperforms VAE (MAE:0.2923, RMSE:0.4528), DiffWave (MAE:0.2359, RMSE:0.3384), and TCN-GAN (MAE:1.5976, RMSE:2.5625). It indicates that STOUTER has a better understanding of traffic uncertainty and can minimize significant deviations from true values.

6.6 Case study

To evaluate the usability of the generated cellular traffic data, we perform traffic prediction modeling using the Internet datasets from Milan and Trentino. We employ Long Short-Term Memory (LSTM) networks [36] as the traffic predictor. The experimental training set consists of generated traffic data, while real traffic data is used for testing and validation. To facilitate effective comparison, we also train the model using real traffic data and contrast the results with those obtained from training on generated data.

Figure 10 illustrates the prediction results over the following week, evaluated using two metrics: MAE and RMSE. It is evident that the prediction model trained on real traffic data achieves the best performance across both datasets. Furthermore, our model outperforms other generative methods, producing traffic data that yield prediction errors within 0.1 of the model trained on real data in both MAE and RMSE. These results confirm that the cellular traffic data generated by STOUTER is highly usable, making it a viable alternative to real traffic data for supporting downstream applications. In addition, Accurate traffic fore-



(a) MAE of Milan Internet

(b) RMSE of Milan Internet

(c) MAE of Trentino Internet

(d) RMSE of Trentino Internet

Fig. 10: Performance of traffic prediction modeling using the Internet datasets from Milan and Trentino.

casting enables network operators to effectively deploy base stations in high-traffic areas, optimize resource allocation, and enhance network capacity and coverage. By training the predictor with traffic data generated by STOUTER, forecasting accuracy can be improved, providing more reliable data support for base station deployment.

7 Conclusion and Future Work

In this paper, we propose STOUTER, a novel spatio-temporal fusion diffusion model for cellular traffic generation. This approach incorporates spatio-temporal relationships into diffusion model-based cellular traffic generation process, and thus can produce realistic and high-quality synthetic traffic data. We validate the performance of STOUTER through extensive experiments on large-scale and real-world cellular traffic datasets. The results demonstrate significant performance improvements in various metrics compared to existing generative models, and confirm that STOUTER-generated traffic closely aligns with real distributions, making it highly effective for downstream applications such as network optimization, traffic prediction, and resource allocation.

In future work, we would like to explore the diverse characteristics of cellular traffic to generate more realistic traffic patterns when multi-source datasets, including meteorology, population, events, and disasters, are available. Additionally, STOUTER requires historical data for training when generating cellular traffic, we thus consider using transfer learning to apply knowledge from data-rich areas to regions lacking data support for cellular traffic generation studies. Furthermore, we will evaluate the effectiveness of STOUTER in assisting with network planning if feasible.

References

[1] Q. V. Khanh, N. V. Hoai, L. D. Manh, A. N. Le, and G. Jeon, "Wireless communication technologies for iot in 5G: Vision, applications, and challenges," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 3229294, 2022.

[2] M. Attaran, "The impact of 5G on the evolution of intelligent automation and industry digitization," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 5, pp. 5977–5993, 2023.

[3] G. Liu, Y. Huang, N. Li, J. Dong, J. Jin, Q. Wang, and N. Li, "Vision, requirements and network architecture of 6G mobile network beyond 2030," *China Communications*, vol. 17, no. 9, pp. 92–104, 2020.

[4] Y. Zhang and A. Årvidsson, "Understanding the characteristics of cellular data traffic," in *ACM SIGCOMM*, 2012, pp. 13–18.

[5] C. Lin, S.-L. Tung, H.-T. Su, and W. H. Hsu, "CTCam: enhancing transportation evaluation through fusion of cellular traffic and camera-based vehicle flows," in *ACM CIKM*, 2023, pp. 5341–5345.

[6] C. Gao, T. Feng, H. Wang, D. Jin, J. Feng, X. Wang, L. Zhu, and C. Deng, "A multi-scale ensemble learning model for cellular traffic prediction," in *IEEE GLOBECOM*, 2022, pp. 209–214.

[7] C. Lin, S.-L. Tung, H.-T. Su, and W. H. Hsu, "Tel2Veh: fusion of telecom data and vehicle flow to predict camera-free traffic via a spatio-temporal framework," in *ACM WWW*, 2024, pp. 1083–1086.

[8] X. Wang, Z. Wang, K. Yang, Z. Song, C. Bian, J. Feng, and C. Deng, "A survey on deep learning for cellular traffic prediction," *Intelligent Computing*, vol. 3, p. 0054, 2024.

[9] X. Zhou, Y. Zhang, Z. Li, X. Wang, J. Zhao, and Z. Zhang, "Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning," *Neural Computing and Applications*, pp. 1–11, 2022.

[10] L. Yu, M. Li, W. Jin, Y. Guo, Q. Wang, F. Yan, and P. Li, "STEP: a spatio-temporal fine-granular user traffic prediction system for cellular networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 12, pp. 3453–3466, 2020.

[11] Q. Liu, J. Li, and Z. Lu, "ST-Tran: spatial-temporal transformer for cellular traffic prediction," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3325–3329, 2021.

[12] H. Weng, Y. Liu, and L. Chen, "Spatial bottleneck transformer for cellular traffic prediction in the urban city," in *Australasian Joint Conference on Artificial Intelligence*, 2023, pp. 265–276.

[13] W. Jiang, "Cellular traffic prediction with machine learning: a survey," *Expert Systems with Applications*, vol. 201, p. 117163, 2022.

[14] K. Li, W. Wang, and H.-L. Liu, "6G shared base station planning using an evolutionary bi-level multi-objective optimization algorithm," *Information Sciences*, vol. 642, p. 119224, 2023.

[15] M. Kishk, A. Bader, and M.-S. Alouini, "Aerial base station deployment in 6G cellular networks using tethered drones: the mobility and endurance tradeoff," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 103–111, 2020.

[16] S. Xu, M. Marwah, M. Arlitt, and N. Ramakrishnan, "STAN: synthetic network traffic generation with generative neural models," in *Deployable Machine Learning for Security Defense*, 2021, pp. 3–29.

[17] 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.

[18] S. Hui, H. Wang, T. Li, X. Yang, X. Wang, J. Feng, L. Zhu, C. Deng, P. Hui, D. Jin *et al.*, "Large-scale urban cellular traffic generation via knowledge-enhanced gans with multi-periodic patterns," in *ACM SIGKDD*, 2023, pp. 4195–4206.

[19] 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," in *ACM SIGKDD*, 2023, pp. 4842–4851.

[20] Y. Liu, J. Ding, Y. Fu, and Y. Li, "UrbanKG: an urban knowledge graph system," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 4, 2023.

[21] M. J. Shehab, I. Kassem, A. A. Kutty, M. Kucukvar, N. Onat, and T. Khattab, "5G networks towards smart and sustainable cities: a review of recent developments, applications and future perspectives," *IEEE Access*, vol. 10, pp. 2987–3006, 2022.

[22] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *ICLR*, 2018.

[23] A. Brock, T. Lim, J. Ritchie, and N. Weston, "Neural photo editing with introspective adversarial networks," in *ICLR*, 2017.

[24] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *ICLR*, 2019.

[25] 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.

[26] 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," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 3, pp. 65–76, 2006.

[27] J. Zhang, J. Tang, X. Zhang, W. Ouyang, and D. Wang, "A survey of network traffic generation," in *Third International Conference on Cyberspace Technology*, 2015, pp. 1–6.

[28] A. A. Cardoso and F. H. T. Vieira, "Generation of synthetic network traffic series using a transformed autoregressive model based adaptive algorithm," *IEEE Latin America Transactions*, vol. 17, no. 08, pp. 1268–1275, 2019.

[29] Z. Lin, A. Jain, C. Wang, G. Fanti, and V. Sekar, "Using GANs for sharing networked time series data: challenges, initial promise, and open questions," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 464–483.

[30] Y. Yin, Z. Lin, M. Jin, G. Fanti, and V. Sekar, "Practical GAN-based synthetic Ip header trace generation using netshare," in *ACM SIGCOMM*, 2022, pp. 458–472.

[31] 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.

[32] K. Liang, L. Meng, M. Liu, Y. Liu, W. Tu, S. Wang, S. Zhou, X. Liu, F. Sun, and K. He, "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 9456–9478, 2024.

[33] H. Chai, T. Jiang, and L. Yu, "Diffusion model-based mobile traffic generation with open data for network planning and optimization," in *ACM SIGKDD*, 2024, p. 4828–4838.

[34] OpenStreetMap. Accessed in May 2025. [Online]. Available: http://www.openstreetmap.org/

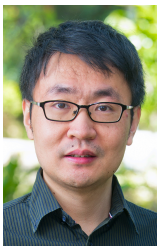
[35] Y. Zhu and S. Wang, "Joint traffic prediction and base station sleeping for energy saving in cellular networks," in *IEEE ICC*, 2021, pp. 1–6.

[36] W. Wang, C. Zhou, H. He, W. Wu, W. Zhuang, and X. Shen, "Cellular traffic load prediction with lstm and gaussian process regression," in *IEEE ICC*, 2020, pp. 1–6.
[37] X. Ma, B. Zheng, G. Jiang, and L. Liu, "Cellular network traffic prediction based on correlation ConvLSTM and self-attention network," *IEEE Communications Letters*, vol. 27, no. 7, pp. 1909–1912, 2023.
[38] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, 2020, pp. 6840–6851.
[39] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *ICML*, 2021, pp. 8857–8868.
[40] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "CSDI: conditional score-based diffusion models for probabilistic time series imputation," in *NeurIPS*, 2021, pp. 24 804–24 816.
[41] M. Liu, H. Huang, H. Feng, L. Sun, B. Du, and Y. Fu, "PriSTI: a conditional diffusion framework for spatiotemporal imputation," in *IEEE ICDE*, 2023, pp. 1927–1939.
[42] J. M. L. Alcaraz and N. Strodthoff, "Diffusion-based conditional ECG generation with structured state space models," *Computers in Biology and Medicine*, vol. 163, p. 107115, 2023.
[43] H. Yuan, S. Zhou, and S. Yu, "EHRDiff: exploring realistic EHR synthesis with diffusion models," *Transactions on Machine Learning Research*, 2024.
[44] H. Wen, Y. Lin, Y. Xia, H. Wan, Q. Wen, R. Zimmermann, and Y. Liang, "DiffSTG: probabilistic spatio-temporal graph forecasting with denoising diffusion models," in *ACM SIGSPATIAL*, 2023, pp. 1–12.
[45] Y. Zhu, Y. Ye, S. Zhang, X. Zhao, and J. Yu, "DiffTraj: generating gps trajectory with diffusion probabilistic model," in *NeurIPS*, 2023, pp. 65 168–65 188.
[46] Z. Zhou, J. Ding, Y. Liu, D. Jin, and Y. Li, "Towards generative modeling of urban flow through knowledge-enhanced denoising diffusion," in *ACM SIGSPATIAL*, 2023, pp. 1–12.
[47] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, "GraphMAE: self-supervised masked graph autoencoders," in *ACM SIGKDD*, 2022, pp. 594–604.
[48] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.
[49] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: a survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.
[50] J. Zhu, Z. Fang, X. Yang, and L. Yin, "Flow synchronization of mobile communication network in cities areas," *Geo-Information Science*, vol. 20, no. 6, pp. 844–853, 2018.
[51] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[52] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of milan and the province of trentino," *Scientific Data*, vol. 2, no. 1, pp. 1–15, 2015.
[53] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: a versatile diffusion model for audio synthesis," in *ICLR*, 2021.
[54] S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," in *EMNLP*, 2017, pp. 627–637.
[55] B. Fuglede and F. Topsoe, "Jensen-shannon divergence and hilbert space embedding," in *Proceedings of International Symposium on Information Theory*, 2004, p. 31.
[56] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.
[57] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017.
[58] X. Ma, J. Yang, T. Hong, M. Ma, Z. Zhao, T. Feng, and W. Zhang, "STNet: Spatial and temporal feature fusion network for change detection in remote sensing images," in *IEEE ICME*, 2023, pp. 2195–2200.
[59] S. Fan, Q. Dong, F. Zhu, Y. Lv, P. Ye, and F.-Y. Wang, "SCF-Net: Learning spatial contextual features for large-scale point cloud segmentation," in *CVPR*, 2021, pp. 14 504–14 513.
[60] G. Wang, F. Fan, S. Shi, S. An, X. Cao, W. Ge, F. Yu, Q. Wang, X. Han, S. Tan, Y. Tan, and Z. Wang, "Multi modality fusion transformer with spatio-temporal feature aggregation module for psychiatric disorder diagnosis," *Computerized Medical Imaging and Graphics*, vol. 114, p. 102368, 2024.

Xiaosi Liu received the B.E. degree in Software Engineering from Nanchang University, Nanchang, China, in 2022. She is currently a Master's degree student at the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, under the supervision of Dr. Zhidan Liu. Her research interests include spatio-temporal data analysis and urban computing.

Xiaowen Xu received the B.E. degree in Computer Science and Technology from Shenzhen University, Shenzhen, China, in 2023. She is currently a first-year PhD student in Intelligient Transportation at The Hong Kong University of Science and Technology (GZ), China, under the supervision of Dr. Zhidan Liu. Her research interests are in the areas of trajectory data analysis and mobile computing.

Zhidan Liu (Senior Member, IEEE) received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2014. After that, he worked as a Research Fellow at Nanyang Technological University, Singapore, and a faculty member at the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is currently an Assistant Professor at Intelligent Transportation Thrust, System Hub, The Hong Kong University of Science and Technology (Guangzhou). His research interests include the Internet of Things, mobile computing, urban computing, and big data analytic. He is a senior member of IEEE and CCF.



Zhenjiang Li (Member, IEEE) received the BE degree from Xi'an Jiaotong University, Xi'an, China, in 2007, and the MPhil and Ph.D. degrees from the Hong Kong University of Science and Technology, Hong Kong, China, in 2009 and 2012, respectively. He is currently an associate professor with the Department of Computer Science, City University of Hong Kong. His research interests include the Internet of Things, edge AI systems, and smart sensing.

Kaishun Wu received his Ph.D. degree in Computer Science and Engineering at The Hong Kong University of Science and Technology. Before joining HKUST(GZ) as a Full Professor at DSA Thrust and IoT Thrust in 2022, he was a distinguished Professor and Director of Guangdong Provincial Wireless Big Data and Future Network Engineering Center at Shenzhen University. Prof. Wu is an active researcher with more than 200 papers published in major international academic journals and conferences, as well as more than 100 invention patents, including 12 from the USA. He received the 2012 Hong Kong Young Scientist Award, and the 2014 Hong Kong ICT Awards: Best Innovation, and 2014 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He is a Fellow of IEEE, IET, and AAIA.