
Cross City Traffic Flow Generation via Retrieval Augmented Diffusion Model

Yudong Li¹, Jingyuan Wang^{*1, 2, 3}, Xie Yu¹, Peiyu Wang¹, Qian Huang⁴

¹School of Computer Science and Engineering, Beihang University, Beijing, China

²School of Economics and Management, Beihang University, Beijing, China

³MIT Key Laboratory of Data Intelligence and Management, Beihang University, Beijing, China

⁴Global Technical Service Dept, Huawei Technologies Co., Ltd Beijing, China

{yudongli, yjwang, yuxie_scse, peiyuwang}@buaa.edu.cn
huangqian16@huawei.com

Abstract

Traffic flow data are of great value in smart city applications. However, limited by data collection costs and privacy sensitivity, it is rather difficult to obtain large-scale traffic flow data. Therefore, various data generation methods have been proposed in the literature. Nevertheless, these methods often require data from a specific city for training and are difficult to directly apply to new cities lacking data. To address this problem, this paper proposes a retrieval-augmented diffusion generation model with geographic representation alignment. We use data from multiple source cities for training, extract consistent representations across multiple cities, and leverage retrieval-augmented generation (RAG) technology to incorporate dynamic traffic flow patterns into the condition, aiming to improve the accuracy of data generation in the target city. Experiments on four real-world datasets demonstrate that, compared to existing generation methods, our method achieves best cross-city zero-shot performance. Our code and datasets can be found in <https://github.com/lyd1881310/CRAFT>.

1 Introduction

Background. Traffic flow data is crucial in intelligent transportation systems [26, 17], urban management [16, 49] and smart cities applications [11, 47]. The success of traffic flow analysis [48, 55] is coupled with a data-hungry paradigm, where superior performance and remarkable generalization ability rely on large-scale and high-quality data. However, unlike computer vision (CV) and natural language processing (NLP), where large-scale datasets are readily available from public sources, collecting traffic flow data faces strict privacy constraints and much higher costs. As a result, existing public traffic flow datasets are usually limited in both scale and quality. Overcoming this challenge to acquire larger-scale, higher-quality traffic flow data has emerged as a critical bottleneck.

Motivation of This Work. In response, traffic flow generation has become an increasingly urgent research direction [58]. It aims to synthesize realistic traffic flow by learning the conditional mapping from static urban geographic context to dynamic traffic flow distributions. Early work predominantly employed physics-driven models [60, 52, 1, 42, 28], which only estimate static average flow volumes and struggled to capture the complex temporal dynamics observed in real-world traffic. More recently, deep generative approaches have been dominant, leveraging richer geographic context as input to model flow distributions with greater fidelity. Their modeling paradigm has shifted from variational

*Corresponding author: Jingyuan Wang

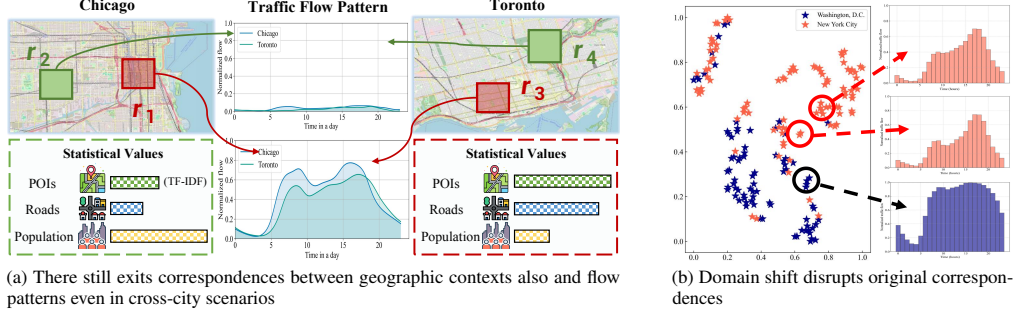


Figure 1: In 1a, For each region, we aggregate the statistical values of its geographic contexts to form the raw geographic features, finding that regions with similar geographic features may share flow patterns even across different cities. From 1b, we further observe that, overall, this correspondence becomes blurred in cross-city scenarios. Although geographic features and traffic flow are still roughly correlated, the presence of cross-city domain shift prevents it from being applied directly.

autoencoders (VAEs) [22] and generative adversarial networks (GANs) [8] to diffusion models [12], which now lead the field due to their strong conditional generation capabilities. Consequently, recent studies emphasize incorporating stronger urban priors into conditions [58], thereby supporting more fine-grained and accurate traffic flow generation.

While incorporating stronger geographic priors into model inputs boost performance, these priors are often city-specific, which limits the model’s ability to generalize. Such priors are essentially the dependencies between local geographic structures and their corresponding flow patterns, which typically require training on city-specific historical flow data. This reliance hinders model deployment in cities with insufficient traffic flow records. In this paper, we attempt to address this issue, and explore directly generating dynamic traffic flow in entirely unseen cities via a cross-city transfer learning framework.

Problem Analysis. To achieve our motivation, we must restrict model learning from common geographic contexts shared across cities, thus avoiding the aforementioned city-specific priors.

In specific, they denote points of interest (POIs), roads, and population. However, utilizing these contexts for cross-city transfer faces two difficulties:

- *Domain Shift:* Within a single city, regions with similar geographic features typically exhibit analogous traffic-flow patterns. However, in cross-city settings, domain shifts in geographic representations disrupt this correspondence (as illustrated in Fig. 1). Consequently, regions from different cities—even if their geographic features are close in representation space—may exhibit markedly different traffic-flow behaviors (see in Fig. 1b).
- *Insufficient Condition:* static geographic contexts alone cannot support accurate cross-city flow generation. Regions with similar spatial characteristics may share similar periodicity and trends but their stochastic properties can vary significantly (e.g., absolute values of mean, peak and variance). Such dynamics are provided by historical flow records but are absent in cross-city scenarios.

Remark: Unlike traffic flow prediction [18, 50, 15, 46], which forecasts future traffic based on historical flow data, traffic flow generation synthesizes realistic traffic flow mainly on static geographic features. Departing from previous methods, our model is explicitly designed for cross-city generalization: it is trained on source cities and directly deployed to unseen target cities.

Design Insights and Contributions. Accordingly, we propose a Cross-city Retrieval-Augmented traffic Flow generaTion model (CRAFT), which is a simple yet effective DDPM-based [12] model. Fundamentally, we propose the Geographic Feature Alignment (GFA) to address *domain shift*; For *Insufficient Condition*., we propose the Retrieval-based Condition Augmentation (RCA), which integrates traffic flow from source cities to supplement temporal dynamics. Our contributions can be summarized as:

- Our work is an initial probe to tackle cross-city traffic flow generation, proposing a transfer learning framework that enables traffic flow generation without historical data in target cities.

- Specifically, we proposed GFA and RCA to tackle two fundamental challenges in cross-city flow generation. Both are lightweight, plugin components without altering the backbone architecture, revealing the simplicity of our approach.
- Extensive experiments on four real-world urban datasets demonstrate the state-of-the-art (SOTA) zero-shot generation performance of our model and further validate its strong generalization ability.

2 Related Work

Traffic Flow Generation Models has evolved along two key trajectories: leveraging more comprehensive geographic features as inputs, and evolving from static to dynamic flow generation. This development can be grouped into three stages: physics-based, static, and dynamic flow generation models. *Physics-based Models* rely on empirical rules. The Gravity Models [1, 52, 60] and Radiation Models [28, 42] generate origin-destination (OD) flows based on gravitational laws and radiative diffusion, respectively. These methods use simple geographic features as inputs and are limited to coarse-grained OD flow prediction; Subsequently, neural networks became prevalent due to their ability to incorporate more comprehensive geographic information. Initially, *Static Flow Generation Models* learn the dependencies between static traffic flows and the geographic characteristics of urban regions [27, 40, 34, 37, 43]. For instance, DeepGravity [43] extracts features from OpenStreetMap (OSM) to estimate OD flows, while DeepFlowGen [40] incorporates abundant POI data. Although these methods capture richer geo-features than physics-based models, they still fall short in modeling temporal dynamics; Further, *Dynamic Flow Generation Models* address this limitation by modeling temporal variations through generative frameworks [58, 14, 56, 3, 36, 33, 35]. The field has advanced from generative adversarial networks (GANs) [14] to diffusion models [3, 58], which now lead the field due to their strong conditional generation capabilities.

Comparison: Current models perform well within individual cities as they tend to involve city-specific priors, but this reliance hinders their cross-city transfer abilities. In contrast, our model is designed to enable cross-city generalization by leveraging the common features shared across cities.

Retrieval-Augmented Generation (RAG) is widely used in large language models (LLMs) [24, 7], as it can dynamically integrate knowledge from external databases and enhance generation accuracy. Recently, time series analysis involves RAG to provide meaningful guidance [25]. For example, [25, 30, 44, 20] applied similarity retrieval based on time series embeddings to improve the prediction accuracy and enhance the zero-shot capabilities of time series foundation models (TSFM) [5, 54]. RAG is helpful for enhancing the model’s capabilities in unseen scenarios. However, in the field of spatio-temporal data generation [21, 14, 59, 51], attempts using RAG still remain limited.

3 Preliminaries

Definition 1: (Region) We divided each city into N non-overlapping basic rectangular grids or polygons, denoted as $\mathcal{R} = \{r_i | i = 1, 2, \dots, N\}$, where N is the total number of regions. Each r_i is characterized by various geographical features including population, road network and points of interest (POIs). These features serve as the basic static condition for traffic flow generation.

Definition 2: (Region Graph) A region graph is denoted as $\mathcal{G} = \langle \mathcal{R}, \mathcal{E} \rangle$, where $\mathcal{E} \in \{0, 1\}^{|N| \times |N|}$ is the binary adjacency matrix. Specifically, $\mathcal{E}_{i,j} = 1$ if r_i and r_j are adjacent, and $\mathcal{E}_{i,j} = 0$ otherwise.

Definition 3: (Traffic Flow) Given a region r_i , assuming it has C traffic flow features, such as traffic inflow and outflow. We denote the traffic flow of r_i at t -th time slice with time length T as $\mathbf{X}_{i,t} \in \mathbb{R}^{C \times T}$. The traffic flow dataset of all regions staring at all time slice is denoted as $\mathcal{X}$.

Problem Statement: (Traffic flow generation). The objective of traffic flow generation is to train a model $\mathcal{F}$ to generate dynamic traffic flow of regions based on their static geographic features, including population, road network and POIs. Specifically, $\mathcal{F}$ takes a region graph $\mathcal{G}$ as input and outputs $\hat{\mathcal{X}}$ as

$$\hat{\mathcal{X}} = \mathcal{F}(\mathcal{G}; \theta), \quad (1)$$

where θ are trainable parameters of $\mathcal{F}$, and $\hat{\mathcal{X}}$ is evaluated by measuring the similarity to the real-world traffic flow data $\mathcal{X}$.

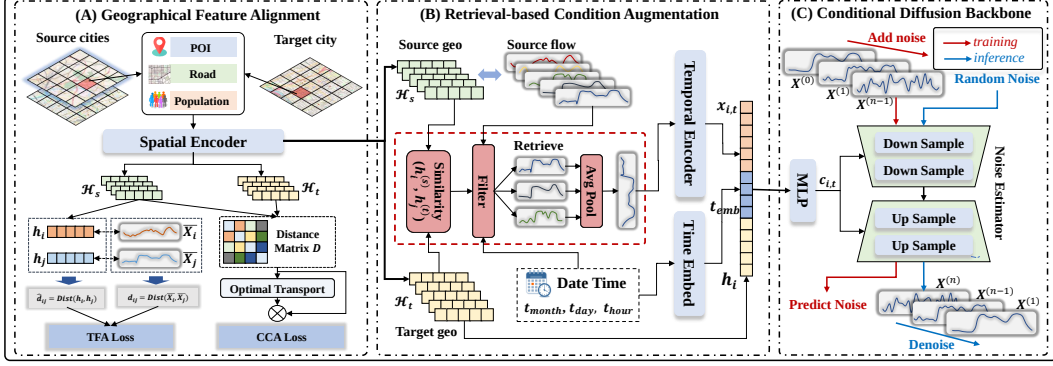


Figure 2: The framework of CRAFT

Problem Statement: (Cross-city traffic flow generation). Models are trained on data from multiple source cities and make zero-shot generation on unseen target cities. Formally, let $\mathcal{G}^{(t)}$ denote the region graph of a target city and $\theta^{(s)}$ denote parameters trained from source cities and the process is

$$\hat{\mathcal{X}}^{(t)} = \mathcal{F}(\mathcal{G}^{(t)}; \theta^{(s)}). \quad (2)$$

4 Methodology

4.1 Framework

The framework of CRAFT is in Figure. 2, which applies the DDPM as the backbone. Specifically, CRAFT contains three modules: (1) Geographical Feature Alignment (GFA); (2) Retrieval-based Condition Augmentation (RCA); (3) Conditional Diffusion Backbone.

Initially, the datasets are divided into source and target cities: source cities include both traffic flow data and regional geographic features, while the target city only has regional geographic features. GFA is first pre-trained on source and target cities to provide cross-city geographic feature alignment. Subsequently, RCA is designed to retrieve relevant historical data from source cities via the aligned regional representations, supplying an augmented condition for the diffusion backbone. Finally, Conditional Diffusion Backbone takes augmented condition to make zero-shot traffic flow generation in the target city.

4.2 Geographic Feature Alignment

To address the challenge of *domain shift* (in Section 1), we propose the geographic feature alignment module. Unlike existing geographic representation learning methods [19, 4, 9, 57, 53], which learn representations based on entity embeddings, we begin by selecting common geographic features shared across cities and use them to construct basic representations for each region. Subsequently, we adopt the Graph Transformer [41] as the spatial encoder, which takes each city’s region graph $\mathcal{G}$ as input to generate higher-level representations. Finally, the encoder is trained through traffic flow alignment (TFA) and cross-city alignment (CCA) to enable transferable geographic representations.

Basic Geographic Representation. For most cities, both static maps (OpenStreetMap) and population data (WorldPop) are publicly available and highly correlated with traffic flow patterns. Therefore, the basic geographic representations are derived from the following three aspects:

- **POIs:** POIs are highly correlated with the region functions. For each region, we use the TF-IDF algorithm [39] to construct POI representations $\mathbf{f}^{(poi)}$ based on POIs’ categories and numbers.
- **Roads:** For each region, we compute the total length of road segments across all road categories, as this captures the region’s transportation capacity. Regions with dense networks and a high proportion of trunk roads typically offer better accessibility and attract heavier volume of traffic flow. The road feature is denoted as $\mathbf{f}^{(road)}$.

- **Population:** For each region, we collect statistics on its population size and density, as population implicitly reflect the scale of its traffic flow. Regions with higher population density tend to generate larger volumes of traffic flow. The population feature is denoted as $\mathbf{f}^{(pop)}$.

For each region r_i , its $\mathbf{f}^{(poi)}$, $\mathbf{f}^{(road)}$, and $\mathbf{f}^{(pop)}$ are concatenated and projected through a multi-layer perceptron (MLP) to generate the *basic geographic representation*

$$\mathbf{z}_i = \mathbf{MLP} \left(\mathbf{f}_i^{(pop)} \parallel \mathbf{f}_i^{(poi)} \parallel \mathbf{f}_i^{(road)} \right). \quad (3)$$

Further, the spatial encoder takes $\mathbf{z}_i$ as input and models the correlations among regions. For a specific city with its $\mathcal{G}$. Region representations are further embedded as

$$\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N\} = \text{GraphTransformer}(\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N\}, \mathcal{G}), \quad (4)$$

where N is the region numbers of this city and $\mathcal{G}$ is the region graph. To obtain cross-city transferable region representations, we pre-train the spatial encoder through the proposed traffic flow alignment and cross-city alignment methods. During the pre-training stage, geographic representations of both source and target cities are jointly involved to facilitate cross-city alignment.

Traffic Flow Alignment (TFA). Only regions from source cities are involved in the TFA. We aim to ensure that the differences in region representations reflect the differences in their corresponding traffic flows. Specifically, we first compute the average traffic flow for each region over the time period. Then, all regions from the source cities are collected as the set $\mathcal{R}_s = \{r_1, r_2, \dots, r_{N_s}\}$, with their corresponding representations denoted as $\mathcal{H}_s = \{\mathbf{h}_1^{(s)}, \mathbf{h}_2^{(s)}, \dots, \mathbf{h}_{N_s}^{(s)}\}$, where N_s is the total number of regions from all source cities. The traffic flow alignment loss is defined as

$$\mathcal{L}_{FA} = \frac{1}{N_s^2} \sum_{i,j} \left(\hat{d}_{ij} - d_{ij} \right)^2, \quad (5)$$

where $\hat{d}_{ij} = \text{MinMaxNorm}_{i,j} \left(\left\| \mathbf{h}_i^{(s)} - \mathbf{h}_j^{(s)} \right\|_2 \right)$, $d_{ij} = \text{MinMaxNorm}_{i,j} \left(\left\| \bar{\mathbf{X}}_i - \bar{\mathbf{X}}_j \right\|_2 \right)$, and $\bar{\mathbf{X}}_i$ is the average traffic flow data of region r_i .

Cross-City Alignment (CCA). We seek to project semantically similar regions from both source and target cities into proximity within the representation space. Since explicit regional correspondence labels are unavailable, our model needs to learn these correspondences adaptively. By leveraging flow patterns, TFA alone draws regions with similar flows together. Although it does not fully resolve domain shift, it groups regions with both similar flow dynamics and geographic context. Consequently, in the representation space each region's counterparts naturally reside in its neighborhood. Therefore, we formulate the cross-city alignment as an optimal transport (OT) problem [32], seeking the minimum-cost distribution mapping between regions in the source and target cities under the Wasserstein distance.

Specifically, the representations of all regions in source cities and target city are collected into the set $\mathcal{H}_s$ and $\mathcal{H}_t = \{\mathbf{h}_1^{(t)}, \mathbf{h}_2^{(t)}, \dots, \mathbf{h}_{N_t}^{(t)}\}$, respectively. N_t is the number of regions in the target city. As both $\mathcal{H}_s$ and $\mathcal{H}_t$ are enumerable, we can directly calculate the Wasserstein distance and tackle this problem through the optimal transport solver [2, 6]. First, Euclidean distance is used to get the initial representation distance matrix $\mathbf{D}$ between regions in source cities and the target city, and $\mathbf{D}_{ij}$ is

$$\mathbf{D}_{ij} = \left\| \mathbf{h}_i^{(s)} - \mathbf{h}_j^{(t)} \right\|_2, \text{ where } \mathbf{h}_i^{(s)} \in \mathcal{H}_s, \mathbf{h}_j^{(t)} \in \mathcal{H}_t. \quad (6)$$

Then, the transport matrix $\mathbf{T}$ is calculated by the OT solver, where $\mathbf{T} = \text{OTSolver}(\mathbf{D})$. The cross-city alignment loss $\mathcal{L}_{CA}$ is defined as Wasserstein distance, which is equal to the dot product of the transport matrix $\mathbf{T}$ and the distance matrix $\mathbf{D}$, as follows

$$\mathcal{L}_{CA} = \sum_{i,j} \mathbf{T}_{ij} \cdot \mathbf{D}_{ij}. \quad (7)$$

For more details on the formulation of optimal transport problem and the structure of the OT solver, please refer to Appendix A. The total alignment loss is the weighted sum of $\mathcal{L}_{FA}$ and $\mathcal{L}_{CA}$, where λ_1 and λ_2 are the balance weights, as follows

$$\mathcal{L}_A = \lambda_1 \mathcal{L}_{FA} + \lambda_2 \mathcal{L}_{CA}. \quad (8)$$

Remark: As shown in Fig. 1, basic geographic representations are too coarse to guide cross-city flow generation. To refine this issue, we propose GFA with two complementary losses: **(1)** Traffic-flow alignment: refine regional representations in the correct direction according to regional flow patterns from source cities; **(2)** Tackle cross-city alignment without explicit regional correspondence labels, it enables our model to adaptively determine the most accurate counterpart by leveraging an optimal transport loss.

4.3 Retrieval-based Condition Augmentation

To address the challenge of *insufficient condition* (Section 1), we proposed the retrieval-augmented condition augmentation strategy. For each target region, our method retrieves relevant historical data from source cities according to its geographic representation, supplementing the input conditions for the diffusion backbone. For each target region r_i , the generated input condition contains three parts: geographic features h_i , time embedding t_{emb} , and historical data from source cities $x_{i,t}$.

t_{emb} contains three periodic temporal components: the month in the year $t_{month} \in [1, 12]$, the day in the week $t_{day} \in [1, 7]$ and the hour in the day $t_{hour} \in [1, 24]$. These three components are encoded into vectors through the embedding layers and concatenated to obtain the time embedding $t_{emb} = (t_{month} || t_{day} || t_{hour})$, which indicates the start time of the generated flow.

$x_{i,t}$ denotes the retrieved results for r_i according to $t_{month}, t_{day}, t_{hour}$ and h_i . Traffic flows from source cities are first segmented into sequences of length T . We then filter these segments using two criteria: (1) Starting time information t_{month}, t_{day} and t_{hour} ; (2) Region representation similarity to h_i . The matched top K flow sequences are selected as $\bar{\mathcal{X}}_{i,t} = \{\bar{\mathbf{X}}_{1,t}, \bar{\mathbf{X}}_{2,t}, \dots, \bar{\mathbf{X}}_{K,t}\}$, where $\bar{\mathbf{X}}_{k,t} \in \mathbb{R}^{C \times T}$, where $\bar{\mathbf{X}}_{i,t}$ is the average flow of region r_i staring at time slice t . We employ a self-attention [45] block $\text{Attn}(\cdot)$ to extract flow patterns from $\bar{\mathcal{X}}_{i,t}$, which takes the averaged $\bar{\mathcal{X}}_{i,t}$ as input to mitigate the impact of noise. The extracted retrieval result is

$$x_{i,t} = \text{Attn} \left(\frac{1}{K} \sum_{k=1}^K \bar{\mathbf{X}}_{k,t} \right). \quad (9)$$

Finally, above features are concatenated and projected by a Multi-Layer Perceptron (MLP) block to generate the input condition $c_{i,t}$ for the target region r_i , as follows

$$c_{i,t} = \text{MLP} (h_i || x_{i,t} || t_{emb}). \quad (10)$$

4.4 Conditional Diffusion Backbone

As shown in Fig. 2, we adopt the Denoising Diffusion Probabilistic Model (DDPM) as the backbone.

Training. In the training phase, for the traffic flow sample $\mathbf{X}_{i,t}$ in source city, we add random noise through the forward diffusion process and use a 1D-U-Net [38] as a noise estimator to predict the added noise. Following most of the literature about diffusion model, we use $\mathbf{X}$ to represent $\mathbf{X}_{i,t}$ in the rest of this paper. Following [12], the noise-adding process of k steps can be simplified as:

$$\mathbf{X}^{(k)} = \sqrt{\bar{\alpha}_k} \mathbf{X} + \sqrt{1 - \bar{\alpha}_k} \epsilon, \quad (11)$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\bar{\alpha}_k$ is a hyper parameter. The noise estimator ϵ_θ takes the noised sample $\mathbf{X}^{(k)}$, the step k and the condition vector $c_{i,t}$ as inputs to predict the noise. The Mean Squared Error (MSE) loss function is used to train the noise estimator, as follows

$$\mathcal{L}_{NE} = \mathbb{E}_{k, \mathbf{X}_{i,t}, \epsilon} \left\| \epsilon_\theta \left(\mathbf{X}^{(k)}, k, c_{i,t} \right) - \epsilon \right\|^2. \quad (12)$$

Inference. In the inference phase, for each r_i in the target city, the model takes a random Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as input and generates its corresponding traffic flow $\hat{\mathbf{X}}$ conditioned on the $c_{i,t}$ through an n -step denoising process:

$$p_\theta \left(\hat{\mathbf{X}}^{(k-1)} | \hat{\mathbf{X}}^{(k)} \right) := \mathcal{N} \left(\mu_\theta \left(\hat{\mathbf{X}}^{(k)}, k, c_{i,t} \right), \sigma^2 \left(\hat{\mathbf{X}}^{(k)}, k \right) \mathbf{I} \right), \quad 1 \leq k \leq n, \quad (13)$$

In the equation 13, μ_θ denotes the mean variable and σ^2 is the variance:

$$\mu_\theta \left(\mathbf{X}^{(k)}, k, c_{i,t} \right) = \frac{1}{\sqrt{\bar{\alpha}_k}} \left(\mathbf{X}^{(k)} - \frac{\beta_k}{\sqrt{1 - \bar{\alpha}_k}} \epsilon_\theta \left(\mathbf{X}^{(k)}, k, c_{i,t} \right) \right); \quad \sigma^2 \left(\mathbf{X}^{(k)}, k \right) = \frac{1 - \bar{\alpha}_{k-1}}{1 - \bar{\alpha}_k} \beta_k. \quad (14)$$

Table 1: Cross-city traffic flow generation results

Method	City	Inflow			Outflow			City	Inflow			Outflow		
		CPC(↑)	NMAE(↓)	NRMSE(↓)	CPC(↑)	NMAE(↓)	NRMSE(↓)		CPC(↑)	NMAE(↓)	NRMSE(↓)	CPC(↑)	NMAE(↓)	NRMSE(↓)
GMEL	Chicago(CHI)	0.730	0.173	0.227	0.725	0.175	0.236	Washington,D.C.(DC)	0.741	0.273	0.319	0.716	0.281	0.316
DFG		0.162	0.306	0.438	0.159	0.310	0.441		0.691	0.240	0.343	0.690	0.239	0.342
KSTDiff		0.006	0.334	0.467	0.132	0.322	0.453		0.613	0.519	0.610	0.608	0.564	0.673
CGAN		0.230	0.379	0.511	0.203	0.364	0.495		0.514	0.384	0.509	0.507	0.398	0.523
Diffwave		0.332	0.486	0.598	0.532	0.496	0.609		0.650	0.403	0.538	0.570	0.458	0.584
DiT		0.509	0.357	0.451	0.528	0.343	0.436		0.634	0.351	0.420	0.607	0.352	0.422
DDPM		0.415	0.273	0.393	0.416	0.275	0.396		0.346	0.387	0.514	0.351	0.388	0.512
CVAE		0.490	0.267	0.385	0.488	0.269	0.385		0.468	0.343	0.461	0.471	0.347	0.461
CRAFT		0.785	0.140	0.216	0.786	0.140	0.216		0.815	0.158	0.240	0.816	0.159	0.240
GMEL	Toronto(TRT)	0.735	0.224	0.283	0.744	0.217	0.276	New York City(NYC)	0.585	0.310	0.379	0.672	0.188	0.242
DFG		0.278	0.395	0.514	0.278	0.394	0.512		0.581	0.182	0.278	0.585	0.183	0.279
KSTDiff		0.006	0.469	0.597	0.248	0.413	0.540		0.040	0.226	0.364	0.030	0.228	0.364
CGAN		0.561	0.358	0.468	0.572	0.354	0.461		0.368	0.330	0.476	0.407	0.382	0.527
Diffwave		0.521	0.421	0.531	0.546	0.456	0.574		0.363	0.475	0.608	0.434	0.440	0.575
DiT		0.552	0.394	0.489	0.601	0.381	0.471		0.419	0.416	0.508	0.449	0.383	0.478
DDPM		0.592	0.358	0.474	0.595	0.359	0.474		0.523	0.316	0.431	0.525	0.324	0.438
CVAE		0.699	0.289	0.394	0.695	0.295	0.398		0.568	0.280	0.403	0.571	0.280	0.401
CRAFT		0.804	0.178	0.267	0.804	0.179	0.268		0.782	0.103	0.170	0.786	0.102	0.165

where β_k is a hyper-parameter and $\bar{\alpha}_k = \prod_{i=1}^k (1 - \beta_i)$. Through an n-step denoising process, the traffic flow data $\hat{X}^{(0)}$ is generated.

5 Experiment

Dataset: We conducted experiments on four real-world bicycle trip datasets, namely Chicago (CHI)¹, Washington, D.C. (DC)², Toronto (TRT)³, and New York City (NYC)⁴. We manually partition each city into grid-based regions and count the number of bicycles entering and exiting each region within each hour to obtain the traffic flow. In our experiments, we use normalized traffic flow values; **Baseline:** We employ GMEL [27], DFG [40], KSTDiff [58], CGAN [29], Diffwave [23], DiT [31], DDPM [12] and CVAE [10] as our baselines; **Evaluation Metric:** We evaluate our model on the metrics of Common Part of Commuters (CPC), Normalized Mean Absolute Error (NMAE) and Normalized Root Mean Square Error (NRMSE). *Details about dataset pre-procession, baselines, and evaluation metrics are in Appendix B.*

5.1 Overall Generation Performance

Generation Performance: Table. 1 reports the zero-shot cross-city performance comparison between our method and selected baselines. The experiments are organized into four groups. Each city is sequentially assigned as the target city while the remaining three cities serve as source cities. Models are trained exclusively on the source cities and evaluated in a zero-shot manner on the unseen target city. For each target city, both inflow and outflow data are generated. From the results, our method consistently outperforms existing approaches and achieves state-of-the-art (SOTA) performance across various settings. It yields an improvement of 59.7% compared with the average level of all baselines and an improvement of 22.5% compared with the second-best baseline (GMEL). When compared with ordinary DDPM, our approach yields an average improvement of 61.5%.

Utility of the Generated Data: To further assess the quality of the generated flow data, we evaluate them on a downstream traffic flow task across multiple target cities. Specifically, we use the synthetic data produced by CRAFT and other baselines to train two representative models for flow prediction: a vanilla LSTM [13] and a Transformer [45]. As a reference, we also train the same models on real flow data from the corresponding target cities. As shown in Table. 2, our model consistently outperforms all baseline methods across various evaluation metrics. It achieves an improvement of 55.9% over the average level of all baselines and an improvement of 14.9% over the second-best baseline (DDPM). Furthermore, our model achieves performance that is closest to the results of direct training on real traffic flow data. Specifically, it demonstrates an average performance degradation of only 10.4%, with a minimum drop of 3.8% and a maximum drop of 22.2%. These results underscore the model's

¹<https://divvy-tripdata.s3.amazonaws.com>

²<https://s3.amazonaws.com/capitalbikeshare-data>

³<https://ckan0.cf.opendata.inter.prod-toronto.ca>

⁴<https://s3.amazonaws.com/tripdata/2023-citibike-tripdata.zip>

Table 2: Data utility comparison on traffic flow prediction

Gen	Pred	Chicago(CHI)				Washington, D.C.(DC)				Toronto(TRT)				New York City(NYC)			
		Inflow		Outflow		Inflow		Outflow		Inflow		Outflow		Inflow		Outflow	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Real		0.102	0.152	0.105	0.157	0.105	0.156	0.109	0.163	0.132	0.191	0.132	0.191	0.063	0.107	0.063	0.106
GMEL	LSTM	0.198	0.264	0.195	0.262	0.242	0.325	0.239	0.303	0.265	0.329	0.258	0.321	0.144	0.216	0.140	0.201
DFG		0.303	0.443	0.307	0.446	0.157	0.232	0.154	0.233	0.318	0.436	0.319	0.437	0.082	0.141	0.081	0.139
KSTDiff		0.339	0.472	0.306	0.441	0.514	0.604	0.560	0.672	0.478	0.606	0.418	0.551	0.225	0.363	0.229	0.365
CGAN		0.435	0.562	0.414	0.540	0.450	0.571	0.447	0.566	0.340	0.445	0.347	0.451	0.291	0.432	0.334	0.475
Diffwave		0.275	0.383	0.277	0.376	0.321	0.418	0.347	0.442	0.373	0.445	0.363	0.437	0.211	0.306	0.177	0.278
DiT		0.266	0.315	0.251	0.307	0.307	0.361	0.301	0.347	0.334	0.382	0.323	0.363	0.246	0.280	0.250	0.282
DDPM		0.117	0.176	0.119	0.177	0.163	0.240	0.165	0.245	0.150	0.221	0.161	0.234	0.081	0.130	0.084	0.129
CVAE		0.252	0.344	0.256	0.348	0.293	0.383	0.289	0.376	0.243	0.344	0.244	0.344	0.142	0.233	0.142	0.231
CRAFT		0.109	0.164	0.111	0.166	0.124	0.180	0.130	0.189	0.141	0.203	0.145	0.206	0.067	0.112	0.069	0.110
Real		0.098	0.151	0.100	0.154	0.095	0.149	0.100	0.157	0.131	0.196	0.132	0.197	0.056	0.103	0.056	0.103
GMEL	Transformer	0.194	0.268	0.189	0.263	0.229	0.297	0.232	0.295	0.256	0.325	0.256	0.327	0.141	0.202	0.127	0.185
DFG		0.337	0.484	0.311	0.451	0.251	0.338	0.236	0.314	0.331	0.447	0.336	0.453	0.083	0.145	0.089	0.148
KSTDiff		0.336	0.469	0.338	0.477	0.514	0.605	0.560	0.672	0.479	0.607	0.418	0.550	0.208	0.338	0.218	0.357
CGAN		0.373	0.487	0.380	0.496	0.393	0.503	0.403	0.508	0.286	0.382	0.297	0.388	0.264	0.391	0.276	0.406
Diffwave		0.282	0.384	0.261	0.351	0.318	0.396	0.303	0.382	0.362	0.432	0.366	0.436	0.178	0.271	0.175	0.258
DiT		0.275	0.317	0.253	0.304	0.297	0.352	0.295	0.347	0.333	0.381	0.322	0.363	0.229	0.267	0.233	0.270
DDPM		0.117	0.181	0.122	0.186	0.163	0.247	0.164	0.251	0.150	0.222	0.159	0.235	0.078	0.133	0.081	0.131
CVAE		0.224	0.308	0.224	0.308	0.271	0.351	0.267	0.344	0.253	0.339	0.255	0.341	0.160	0.269	0.163	0.268
CRAFT		0.105	0.161	0.108	0.165	0.116	0.171	0.122	0.182	0.141	0.206	0.144	0.207	0.065	0.111	0.067	0.111

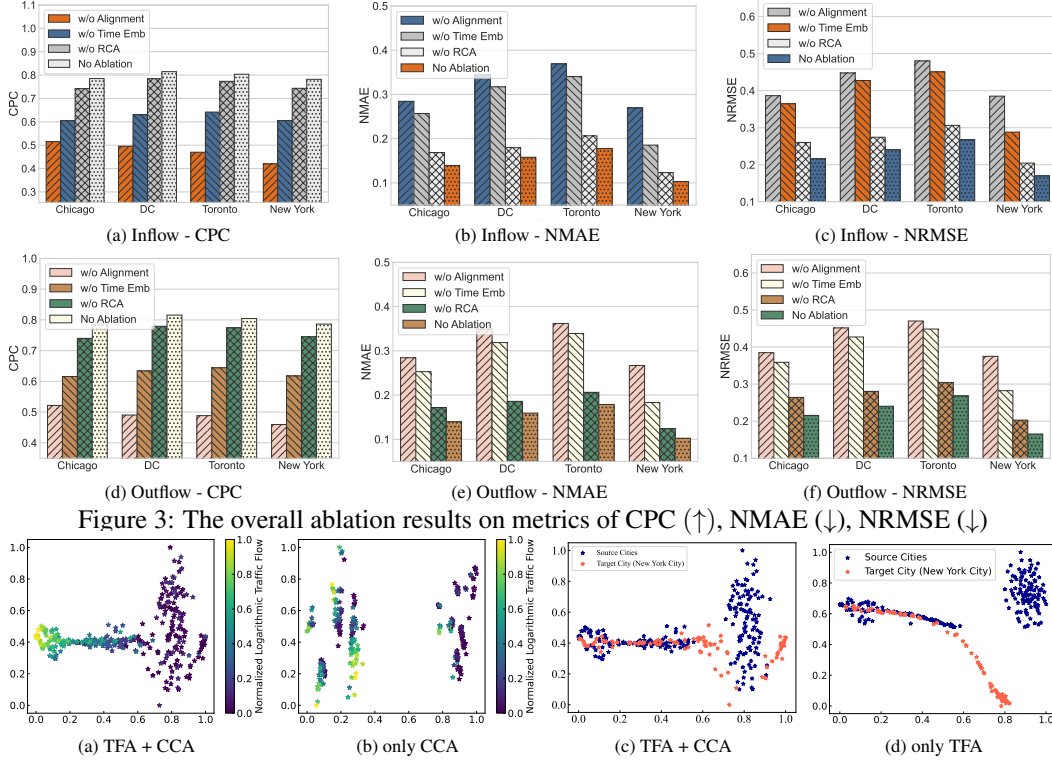


Figure 4: Alignment Analysis on TFA and CCA (New York City is the Target City). All scatter points in this figures represent the regions' geographic features. In (a) and (b), the color intensity indicates the value of traffic flow. After TFA, regions with similar flow patterns exhibit similar geographic representations. By comparing (c) and (d), we observe that CCA alleviates the distribution shift between the target city and source cities.

superior capability in traffic flow generation, outperforming existing baselines and highlighting its strong potential for practical deployment.

5.2 Model Analysis

Ablation Study: Recall that our CRAFT relies on two main components: GFA and RCA. In Fig. 3, we first conducted overall ablation studies to testify their effectiveness based on following settings. (1) *w/o Alignment*: remove GFA by directly using target region's original geographic features for cross-

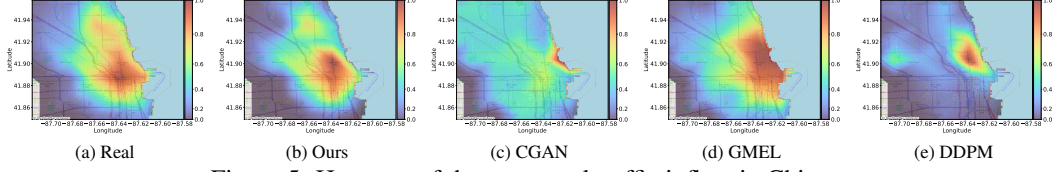


Figure 5: Heatmap of the generated traffic inflow in Chicago

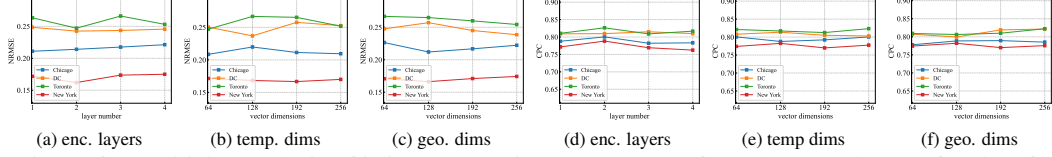


Figure 6: Sensitivity analysis of inflow generation on NRMSE (fig. a, b, c), and CPC (fig. d, e, f).

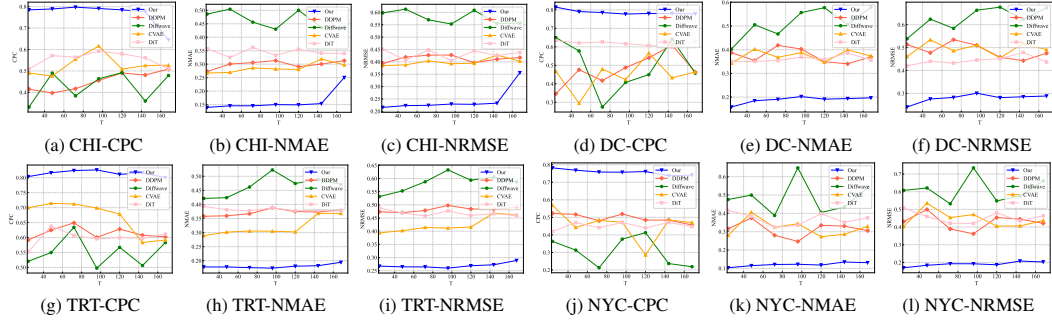


Figure 7: Temporal length extension of all four cities on metrics of CPC ($\uparrow$), NMAE ($\downarrow$), NRMSE ($\downarrow$). All metrics are evaluated on inflow generation.

city retrieval and generation of conditions. (2) *w/o RCA*: remove the retrieval-augmented features $\mathbf{x}_{i,t}$ from $\mathbf{c}_{i,t}$. (3) *w/o Temporal embedding*: remove the temporal embeddings $\mathbf{t}_{emb}$ from $\mathbf{c}_{i,t}$, which supplements the periodic patterns of the traffic flow. Both *w/o RCA* and *w/o Temporal embedding* are related to RCA. From the experimental results in Fig. 3 we can notice: (1) GFA provides the most improvement, highlighting that distribution shift is indeed the major problem in the cross-city generation. (2) In RCA, $\mathbf{t}_{emb}$ contributes the most to the performance gain, highlighting the value of periodic temporal patterns. Additionally, $\mathbf{x}_{i,t}$ consistently improves performance, highlighting the value of supplementing dynamic patterns from source cities.

Further, we explore the specific role of the traffic flow alignment (TFA) and the cross-city alignment (CCA) in GFA. In Fig. 4, we apply t-SNE to all regions’ geographic features in New York. We observe that TFA indeed aligns geographic features with traffic flow patterns. CCA mitigates the distribution shift in geographic features between source cities and the target city, thereby enhancing cross-city generalization. Experiments on all four city are provided in Appendix C.

Sensitivity Analysis: We made various sensitivity analyses on four important hyperparameters, including (1) the layer number of the sequence encoder; (2) the dimension of temporal embedding; (3) the dimension of the geographic representation. As shown in Fig. 6, our model demonstrates strong robustness, with a maximum performance fluctuation of 8.8%. Detailed results for all evaluation metrics are provided in Appendix C.

5.3 Additional Experiments and Discussion

Geographic Visualization: Fig. 5 compares generated flow distributions of CRAFT and baselines in Chicago (please refer Appendix C for all cities). CRAFT and GMEL are most identical to the real distributions. Notably, CRAFT outperforms other baselines with more realistic generated flow. This highlights of CRAFT’s strong zero-shot generation ability and better alignment with geographic and flow patterns.

Temporal Length Extension: In general, most models default to a horizon of $T = 24$, where most baselines remain stable. Increasing T both raises task difficulty and enriches sample information, so an ideal model should exploit the enriched information from bigger T while remaining stable in performance loss. To further explore the potential of each model’s long horizon generation, we trained

several baselines and CRAFT on $\{2T, 3T, \dots, 7T\}$ as $(48, 72, \dots, 168)$. As Fig. 7 shows: (1) All models’ performance drops as T grows, revealing a common long-horizon bottleneck. (2) Diffwave exhibits high variance and poor accuracy, while diffusion-based DiT and DDPM remain steadier and more accurate. (3) CRAFT consistently achieves SOTA and outperforms all these baselines, demonstrating its superior potential in long-horizon generation. More results are in Appendix C.

6 Conclusion

In this paper, we propose CRAFT, a model capable of zero-shot traffic flow generation. Unlike existing approaches, CRAFT is specifically designed for cross-city scenarios and advantage in generating high-quality traffic flow data in cities with limited or no historical flow records. Specifically, we identify two key challenges: *domain shift* and *insufficient condition*. To address these, we introduce Geographic Feature Alignment (GFA) for domain shift and Retrieval-based Condition Augmentation (RCA) for insufficient condition. As both GFA and RCA serve as concise plug-and-lay modules, our method requires no additional modifications to the diffusion backbone. Extensive experiments on four real-world datasets demonstrate the broad effectiveness of CRAFT, and further illustrates the great potential of CRAFT in the fields of urban planning and traffic management.

Limitations: Restrict by available datasets, CRAFT is currently validated on in-out flow generation, and we have not yet explored its ability on other data types like origin-destination (OD) flows. Furthermore, restricted by computational resources, we do not conduct experiments on temporal horizon more than 168. Actually, from Fig. 7, CRAFT remains the most stable across varying temporal horizons, demonstrating its great potential for extension to longer temporal horizons.

Acknowledgements

Jingyuan Wang’s work was partially supported by the National Natural Science Foundation of China (No. 72222022, 72171013, 72242101) and the Fundamental Research Funds for the Central Universities (JKF-2025017226182).

References

- [1] Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J Ramasco, Filippo Simini, and Marcello Tomasini. Human mobility: Models and applications. *Physics Reports*, 734:1–74, 2018.
- [2] Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia conference*, pages 1–12, 2011.
- [3] Haoye Chai, Tong Li, Fenyue Jiang, Shiyuan Zhang, and Yong Li. Knowledge guided conditional diffusion model for controllable mobile traffic generation. In *Companion Proceedings of the ACM Web Conference 2024*, pages 851–854, 2024.
- [4] Jiawei Cheng, Jingyuan Wang, Yichuan Zhang, Jiahao Ji, Yuanshao Zhu, Zhibo Zhang, and Xiangyu Zhao. Poi-enhancer: An llm-based semantic enhancement framework for poi representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 11509–11517, 2025.
- [5] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.
- [6] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL <http://jmlr.org/papers/v22/20-451.html>.

- [7] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2:1, 2023.
- [8] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.
- [9] Chengkai Han, Jingyuan Wang, Yongyao Wang, Xie Yu, Hao Lin, Chao Li, and Junjie Wu. Bridging traffic state and trajectory for dynamic road network and trajectory representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11763–11771, 2025.
- [10] William Harvey, Saeid Naderiparizi, and Frank Wood. Conditional image generation by conditioning variational auto-encoders. *arXiv preprint arXiv:2102.12037*, 2021.
- [11] Kethmi Hirushini Hettige, Jiahao Ji, Shili Xiang, Cheng Long, Gao Cong, and Jingyuan Wang. Airphynet: Harnessing physics-guided neural networks for air quality prediction. *arXiv preprint arXiv:2402.03784*, 2024.
- [12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997.
- [14] Shuodi Hui, Huandong Wang, Tong Li, Xinghao Yang, Xing Wang, Junlan Feng, Lin Zhu, Chao Deng, Pan Hui, Depeng Jin, et al. 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*, pages 4195–4206, 2023.
- [15] Jiahao Ji, Jingyuan Wang, Zhe Jiang, Jiawei Jiang, and Hu Zhang. Stden: Towards physics-guided neural networks for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4048–4056, 2022.
- [16] Jiahao Ji, Jingyuan Wang, Junjie Wu, Boyang Han, Junbo Zhang, and Yu Zheng. Precision cityshield against hazardous chemicals threats via location mining and self-supervised learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3072–3080, 2022.
- [17] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4356–4364, 2023.
- [18] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4365–4373, 2023.
- [19] Jiawei Jiang, Dayan Pan, Houxing Ren, Xiaohan Jiang, Chao Li, and Jingyuan Wang. Self-supervised trajectory representation learning with temporal regularities and travel semantics. In *2023 IEEE 39th international conference on data engineering (ICDE)*, pages 843–855. IEEE, 2023.
- [20] Baoyu Jing, Si Zhang, Yada Zhu, Bin Peng, Kaiyu Guan, Andrew Margenot, and Hanghang Tong. Retrieval based time series forecasting. *arXiv preprint arXiv:2209.13525*, 2022.
- [21] Alexandra Kapp, Julia Hansmeyer, and Helena Mihaljević. Generative models for synthetic urban mobility data: A systematic literature review. *ACM Computing Surveys*, 56(4):1–37, 2023.
- [22] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.
- [23] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

- [24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- [25] Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. Retrieval-augmented diffusion models for time series forecasting. *Advances in Neural Information Processing Systems*, 37:2766–2786, 2024.
- [26] Zehua Liu, Jingyuan Wang, Zimeng Li, and Yue He. Full bayesian significance testing for neural networks in traffic forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.
- [27] Zhicheng Liu, Fabio Miranda, Weiting Xiong, Junyan Yang, Qiao Wang, and Claudio Silva. Learning geo-contextual embeddings for commuting flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 808–816, 2020.
- [28] A Paolo Masucci, Joan Serras, Anders Johansson, and Michael Batty. Gravity versus radiation models: On the importance of scale and heterogeneity in commuting flows. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 88(2):022812, 2013.
- [29] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [30] Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Y Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. Ts-rag: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster. *arXiv preprint arXiv:2503.07649*, 2025.
- [31] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- [32] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: with applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- [33] Can Rong, Jingtao Ding, Zhicheng Liu, and Yong Li. Complexity-aware large scale origin-destination network generation via diffusion model. *arXiv preprint arXiv:2306.04873*, 2023.
- [34] Can Rong, Jie Feng, and Jingtao Ding. Goddag: Generating origin-destination flow for new cities via domain adversarial training. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):10048–10057, 2023.
- [35] Can Rong, Huandong Wang, and Yong Li. Origin-destination network generation via gravity-guided gan. *arXiv preprint arXiv:2306.03390*, 2023.
- [36] Can Rong, Jingtao Ding, Yan Liu, and Yong Li. A large-scale benchmark dataset for commuting origin-destination matrix generation. *arXiv preprint arXiv:2407.15823*, 2024.
- [37] Can Rong, Zhicheng Liu, Jingtao Ding, and Yong Li. Learning to generate temporal origin-destination flow based-on urban regional features and traffic information. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–17, 2024.
- [38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.
- [39] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
- [40] Erzhao Shao, Huandong Wang, Jie Feng, Tong Xia, Hedong Yang, Lu Geng, Depeng Jin, and Yong Li. Deepflowgen: Intention-aware fine grained crowd flow generation via deep neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5693–5707, 2021.

- [41] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.
- [42] Filippo Simini, Marta C González, Amos Maritan, and Albert-László Barabási. A universal model for mobility and migration patterns. *Nature*, 484(7392):96–100, 2012.
- [43] Filippo Simini, Gianni Barlacchi, Massimiliano Luca, and Luca Pappalardo. A deep gravity model for mobility flows generation. *Nature communications*, 12(1):6576, 2021.
- [44] Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Retrieval augmented time series forecasting. *arXiv preprint arXiv:2411.08249*, 2024.
- [45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [46] Jingyuan Wang, Yating Lin, Junjie Wu, Zhong Wang, and Zhang Xiong. Coupling implicit and explicit knowledge for customer volume prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- [47] Jingyuan Wang, Xiaojian Wang, and Junjie Wu. Inferring metapopulation propagation network for intra-city epidemic control and prevention. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining*, 2018.
- [48] Jingyuan Wang, Jiawei Jiang, Wenjun Jiang, Chao Li, and Wayne Xin Zhao. Libcity: An open library for traffic prediction. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, pages 145–148, 2021.
- [49] Jingyuan Wang, Xin Lin, Yuan Zuo, and Junjie Wu. Dgeye: Probabilistic risk perception and prediction for urban dangerous goods management. *ACM Transactions on Information Systems (TOIS)*, 39(3):1–30, 2021.
- [50] Jingyuan Wang, Jiahao Ji, Zhe Jiang, and Leilei Sun. Traffic flow prediction based on spatiotemporal potential energy fields. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 9073–9087, 2022.
- [51] Jingyuan Wang, Yujing Lin, and Yudong Li. Gtg: Generalizable trajectory generation model for urban mobility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 834–842, 2025.
- [52] Alan Geoffrey Wilson. A family of spatial interaction models, and associated developments. *Environment and Planning A*, 3(1):1–32, 1971.
- [53] Yifan Yang, Jingyuan Wang, Xie Yu, and Yibang Tang. Hygmap: Representing all types of map entities via heterogeneous hypergraph.
- [54] Jiexia Ye, Weiqi Zhang, Ke Yi, Yongzi Yu, Ziyue Li, Jia Li, and Fuguee Tsung. A survey of time series foundation models: Generalizing time series representation with large language model. *arXiv preprint arXiv:2405.02358*, 2024.
- [55] Xie Yu, Jingyuan Wang, Yifan Yang, Qian Huang, and Ke Qu. Bigcity: A universal spatiotemporal model for unified trajectory and traffic state data analysis. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pages 4455–4469. IEEE, 2025.
- [56] Shiyuan Zhang, Tong Li, Shuodi Hui, Guangyu Li, Yanping Liang, Li Yu, Depeng Jin, and Yong Li. Deep transfer learning for city-scale cellular traffic generation through urban knowledge graph. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4842–4851, 2023.
- [57] Wentao Zhang, Jingyuan Wang, Yifan Yang, et al. Veccity: A taxonomy-guided library for map entity representation learning. *arXiv preprint arXiv:2411.00874*, 2024.

- [58] Zhilun Zhou, Jingtao Ding, Yu Liu, Depeng Jin, and Yong Li. Towards generative modeling of urban flow through knowledge-enhanced denoising diffusion. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pages 1–12, 2023.
- [59] Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. Difftraj: Generating gps trajectory with diffusion probabilistic model. *Advances in Neural Information Processing Systems*, 36:65168–65188, 2023.
- [60] George Kingsley Zipf. The p_1 p_2 / d hypothesis: on the intercity movement of persons. *American Sociological Review*, 11(6):677–686, 1946.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main claims made in the abstract and introduction accurately summarize the key contributions of our work.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We include the limitations of our work in Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: This is not a theoretical paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Our method is well described in Section 4 and detailed experiment settings, code and data are provided by the URLs in Abstract.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The URLs of the anonymized code and data have been provided in the Abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Our method is well described in Section 4 and detailed experiment setting is provided in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We reported the best results within the range of adjustable parameters for each method.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The computer resources needed to reproduce the experiments are presented in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We read the NeurIPS Code of Ethics and strictly adhere to it.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss impacts of our method in Section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited the relevant papers and provided links to download the data.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We release the code URLs in Abstract and provide documentation in the form of README.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLM is used only for polishing writing.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Model Details & Hyperparameters

A.1 Basic Geographic Features

We extract urban geographic features from OpenStreetMap (OSM) ⁵ and WorldPop ⁶ data. The original geographic features include the following three aspects.

- POI features (denoted as $f^{(poi)}$) reflect the function of a region. We attempt to construct the POI semantic features of a region by counting the quantities of various POI categories. However, the quantities of different POI categories exhibit significant imbalance, for instance, the total number of commercial POIs is far larger than that of residential POIs. This imbalance can easily cause the model to overlook the influence of POI categories with smaller quantities. Therefore, we employ the TF-IDF algorithm [39] to extract POI features. Specifically, we treat each POI category as a "word", consider all POIs within a region as a "document", and define the entire city as the "corpus". When calculating the importance of a word for a document, the algorithm automatically incorporates the word's frequency in the corpus for weighted analysis.
- Road features (denoted as $f^{(road)}$) reflect the transportation attribute of a region. We calculate the total length of all categories of road segments within a region. Regions with dense road networks and a large number of trunk roads usually have convenient transportation and tend to generate a higher volume of traffic flow.
- Population features (denoted as $f^{(pop)}$) reflect the traffic potential of a region. Regions with higher population density are more likely to generate a higher volume of traffic flow. We obtained the United Nations (UN)-adjusted 100m resolution national population data from *WorldPop* and counted the population number in each manually partitioned rectangular regions.

A.2 Optimal Transport Problem

Optimal Transport (OT) is a mathematical problem aiming to find the most efficient way to move mass from source distribution to target distribution. It was introduced by *Gaspard Monge* in 1781. When both the source and target distributions are represented by enumerable samples (N_s samples for source and N_t samples for target), the OT problem can be formally defined as

$$\begin{aligned} \mathbf{T}^* &= \arg \min_{\mathbf{T} \in \mathbb{R}_+^{N_s \times N_t}} \sum_{i,j} \mathbf{T}_{i,j} \cdot \mathbf{D}_{i,j}, \\ s.t. \quad \mathbf{T}\mathbf{1} &= \mathbf{w}_s \text{ and } \mathbf{T}^\top \mathbf{1} = \mathbf{w}_t, \end{aligned} \quad (15)$$

where $\mathbf{D} \in \mathbb{R}_+^{N_s \times N_t}$ is the cost matrix (distance matrix) defining the cost to move mass from source distribution to target distribution, $\mathbf{w}_s \in \mathbb{R}^{N_s}$ and $\mathbf{w}_t \in \mathbb{R}^{N_t}$ are the weights of each samples in the source and target distribution. The total weights of both $\mathbf{w}_s$ and $\mathbf{w}_t$ are equal to 1. The objective of the OT problem is to find a transportation plan $\mathbf{T}^*$ that minimizes the total transportation cost under the weights-equal constraint.

OT problem has two main functions: (1) Measuring the distance between two distributions; (2) Finding the correspondences between two distributions. We employed both. Specifically, we treated the geographic representations of regions in the source and target cities as two mass distributions, with each region assigned the same weight. we used the solution of the OT problem (also known as the Wasserstein distance) to measure the geographic representation distance of the source and target cities, and treated it as a loss to optimize the spatial encoder, thereby pulling the correspondent regions in representation space.

As illustrated in Equation 15, the OT problem is a linear programming problem, and we use an OT solver based on the network simplex algorithm [2] to address it. Thanks to the Python Optimal Transport (POT) ⁷ tool, we can conveniently calculate the solution of OT problem.

⁵<https://www.openstreetmap.org>

⁶<https://www.worldpop.org/>

⁷<https://github.com/PythonOT/POT>

Table 3: Hyperparameters setting for CRAFT

Hyperparameter	Setting value
Diffusion steps (n)	500
$\beta_1 \sim \beta_n$	0.0002 $\sim$ 0.04 (linear)
GraphTransformer (GT) layers	3
Temporal encoder layers	2
GT attention heads	4
Retrieval top-K value	5
Batch size	256
Learning rate	5×10^{-6}
Training epochs	300
Regional geographic representation (h_i) dimensions	128
Temporal encoder hidden dimensions (dimensions of $x_{i,t}$)	256
Hour embedding (t_{hour}) dimensions	64
Weekday embedding (t_{week}) dimensions	64
Month embedding (t_{month}) dimensions	64
Condition ($c_{i,t}$) dimensions	256

A.3 Implementation Details

For the proposed CRAFT method, we provide the hyperparameter settings in Table 3 to facilitate the reproducibility by researchers. All these parameters are recommended values, not fixed, and can be adjusted according to the dataset and experimental environment. During training, the AdamW optimizer was used. To enhance stability, the EMA (Exponential Moving Average) mechanism was adopted to train the diffusion model.

B Details of Experimental Settings

B.1 Experimental Environment

All neural network models (including CRAFT and other baselines) are implemented in PyTorch and trained on a single NVIDIA RTX 3090 GPU. The experimental machine ran on Ubuntu 20.04.6 LTS, was equipped 24-core Intel(R) Xeon(R) Silver CPU, and had 503 GB of RAM. The training time for all models on a single dataset did not exceed 16 hours.

B.2 Datasets and Pre-processing

Table 4: Data description

Datasets	Chicago	Washington D.C.	Toronto	New York City
# Trips ($\times 10^3$)	5136	4011	2395	35080
Time range	2023.01-2023.12	2023.01-2023.12	2020.01-2020.12	2023.01-2023.12
# Regions	73	82	61	96
# POIs	17205	14070	20621	50776

We conducted experiments using the traffic flow datasets of four cities, namely Chicago, Washington D.C., Toronto, and New York City. The original data are all trip records of shared bicycles, which include the latitude and longitude of the starting and ending points of users’ trips as well as timestamps. We associated the trips with the manually partitioned urban regions, and counted the number of bicycles entering and leaving each region within each hour, which served as the traffic flow values. Details of the datasets are presented in Table 4.

In fact, the user trip data is sparse, which leads to the instability of the traffic flow trend in original data. This is also a common problem in researches about traffic flow data. In response to this, we have adopted two processing methods: (1) We have filled in the missing values in the traffic flow sequence through linear interpolation. For the situation where there are values at the previous and subsequent time steps but missing values in the middle, we have filled them with the average of the

previous and subsequent values. (2) We have used the moving average method to smooth the values within a window with a time length of 3. For the processed data, we have set a sliding window with a length of T to extract the training and test samples. If there are still missing values exceeding 5% in a certain sample, we will discard that sample.

B.3 Baseline Methods

The baseline models used in the experiment include two types of deep learning traffic flow generation models that have emerged in the literature in recent years: *Static Flow Generation Models* and *Dynamic Flow Generation Models*. *Static Flow Generation Models* directly estimate traffic flow using the geographic features and temporal information of urban regions, and they include the following two models.

- GMEL [27]: It uses two graph neural networks to extract features and can simultaneously predict the inflow and outflow as well as the OD (origin-destination) flow between regions.
- DFG [40]: It uses a deep cross network to extract the POI (point of interest) features inside and outside the region, conducts supervised training using the intention-aware pedestrian flow, and predicts the inflow and outflow of the region.

Dynamic Flow Generation Models use deep generative models to learn the complex distribution of data and map random noise into data samples. The selected baseline models are as follows.

- KSTDiff [58]: It uses the urban knowledge graph to extract the representations of urban geographic entities, and then constructs a knowledge-enhanced spatiotemporal diffusion model to generate the inflow and outflow of regions.
- CGAN [29]: This is a conditional generative adversarial network. In the experiment, we use the original static geographic features and time embeddings as conditions to guide the generation of the GAN.
- Diffwave [23]: This is a diffusion model suitable for generating time series data and is often used in speech synthesis. In this experiment, we use it to learn the distribution of traffic flow data.
- DiT [31]: This is a diffusion probabilistic model with a Transformer as the noise estimator.
- DDPM [12]: This is a diffusion probabilistic model U-Net as noise estimator for image synthesizing. To be applied in traffic flow data generation, we replaced 2D convolutions with 1D convolutions.
- CVAE [10]: This is a conditional variational autoencoder. Condition information is added to both the Encoder and the Decoder to guide it to learn the conditional probability distribution of the data. We use the original static geographic features and time embeddings as conditions.

B.4 Evaluation metrics

We used three commonly used metrics in traffic flow generation research to evaluate the quality of generated data, including Common Part of Commuters (CPC), Normalized Mean Absolute Error (NMAE), Normalized Root Mean Square Error (NRMSE). Since the actual traffic flow values in different cities vary greatly, the traffic flow data generated by our model are all normalized, which can reflect the relative magnitudes of traffic flows in different regions and at different time in the same city. All the metrics are calculated based on the normalized data, and the calculation formulas are as follows

$$\begin{aligned} \text{CPC} &= \frac{2 \sum_{i=1}^M \min(\hat{y}_i, y_i)}{\sum_{i=1}^M \hat{y}_i + \sum_{i=1}^M y_i}, \\ \text{NMAE} &= \frac{\frac{1}{M} \sum_{i=1}^M |\hat{y}_i - y_i|}{\max(y_i) - \min(y_i)}, \\ \text{NRMSE} &= \frac{\sqrt{\frac{1}{M} \sum_{i=1}^M (\hat{y}_i - y_i)^2}}{\max(y_i) - \min(y_i)}, \end{aligned} \quad (16)$$

where $\hat{y}_i$ is the generated value, y_i is the real value, and M is the number of values of all samples in the test dataset. To enhance the stability of evaluation results, we grouped the data by the Region ID, month, weekday and hour of the target city's test samples and calculated the differences between the group-averaged values.

C Additional Experiments

C.1 Sensitivity Analysis Results

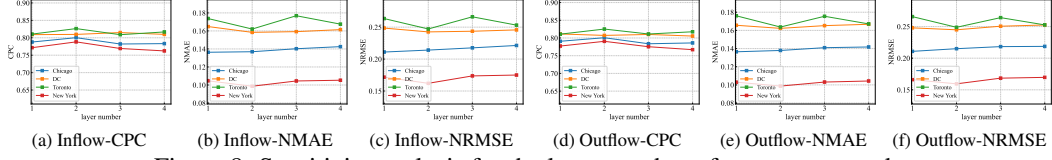


Figure 8: Sensitivity analysis for the layer number of sequence encoder

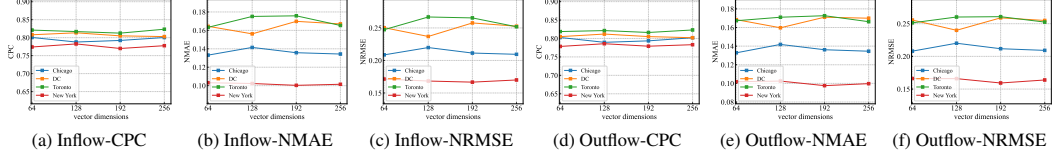


Figure 9: Sensitivity analysis for the dimension of temporal embedding

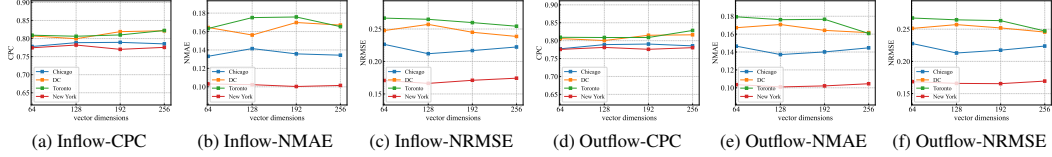


Figure 10: Sensitivity analysis for the dimension of geographic representation

To validate how different hyperparameter configurations affect model performance, we plotted the variation curves of evaluation metrics against three key hyperparameters.: (1) the layer number of the sequence encoder; (2) the dimension of temporal embedding; (3) the dimension of the geographic representation. The results of sensitivity analyses across all datasets and evaluation metrics are shown in the Fig. 8, Fig. 9 and Fig. 10. We can observe that CRAFT demonstrates excellent robustness across all datasets and evaluation metrics, eliminating the need for meticulous hyperparameter tuning to ensure superior model performance.

C.2 Temporal Length Extension Results

Overall Zero-shot Generation Performance in Target City. In the main experiments of the paper, we set the generation time window length to $T = 24$. We then progressively extended the time length T to the following values: $\{48, 72, 96, 120, 144, 168\}$. The experimental setup remained consistent: training the model using data from three cities and performing zero-shot generation on the fourth city. Detailed evaluation results are presented in Table 6. From these tables, we can conclude that: (1) CRAFT achieved best performance in over 98.6% of cases across all datasets and evaluation metrics, demonstrating that CRAFT exhibits state-of-the-art (SOTA) zero-shot generalization capabilities for sequence data generation of varying temporal lengths. (2) CRAFT demonstrates stable performance, while methods such as Diffwave and DDPM exhibit significant performance fluctuations under different temporal length settings.

Data Utility Comparison. We compared the utility of generated data across different temporal lengths for traffic flow prediction tasks. Regardless of the value of T , we used the first $T/2$ historical sequence as input to predict the $T/2$ future flow sequence. The downstream models are trained on generated data from various methods and tested on real data. Detailed results are presented in Table 7 and 8. We found that in 86.5% of cases, the downstream model trained with CRAFT’s generated data achieved best performance, indicating that CRAFT’s generated data has the better utility than other baselines.

Computational Cost. As the length of temporal length increases, the model size and training time will also grow correspondingly. To observe this phenomenon in detail, we conducted experiments on a single NVIDIA RTX 3090 GPU, collected relevant statistical data during the training phase, and the results are shown in Table 5. We observe that our model’s computational cost grows linearly with temporal length increases, but this effect is weak. Extending the time length is acceptable in terms of computational cost.

Table 5: The relationship between computational cost and temporal length

T	Model size (Byte)	Train time (s/epoch)	Valid time (s/epoch)	Avg memory (GB)
24	51781096	40.999	5.685	0.343
48	51830248	41.156	5.531	0.344
72	51879400	43.729	6.053	0.344
96	51928552	44.555	5.631	0.345
120	51977704	46.712	6.429	0.345
144	52026856	48.883	6.666	0.345
168	52076008	51.896	6.351	0.346

C.3 Visualization of the Traffic Flow Spatial Heatmap

We display the average traffic flow of real data and generated data on maps. The visualizations for all datasets and baselines are shown in Fig. 11. This comparison intuitively demonstrates that the traffic data generated by CRAFT exhibits the highest similarity to real data in terms of spatial distribution, indicating that CRAFT can effectively capture the universal mapping relationship between geographic representations and traffic flow across different cities.

C.4 Visualization of Geographic Feature Alignment (GFA)

We employed t-SNE analysis to visualize the impact of the Traffic Flow Alignment (TFA) and Cross City Alignment (CCA) modules in Geographic Feature Alignment. The results of the four experiments are presented in Fig. 12. Regardless of which city was chosen as the target, the alignment results exhibited similar conclusions: (1) Under the combined action of TFA and CCA, regions with high and low traffic volumes were well-separated in the representation space. (2) Without TFA, representations from different cities tended to cluster into multiple groups, with high- and low-traffic regions intermingled within the same cluster, making it difficult to distinguish and reducing the quality of conditions. (3) Without CCA, significant domain shift occurred between the source and target cities. Specifically, a portion of the target city’s region representations deviated from the concentrated representation area of the source city, potentially leading to poorer transferability.

Table 6: Cross-city traffic flow generation results with extended temporal length

Method	City	T = 48			T = 72			T = 96			T = 120			T = 144			T = 168		
		CPC	NMAE	NRMSE	CPC	NMAE	NRMSE	CPC	NMAE	NRMSE	CPC	NMAE	NRMSE	CPC	NMAE	NRMSE	CPC	NMAE	NRMSE
GMEL	Chicago(Inflow)	0.590	0.215	0.310	0.645	0.201	0.288	0.741	0.212	0.268	0.490	0.246	0.351	0.700	0.183	0.257	0.512	0.246	0.358
DFG		0.155	0.328	0.457	0.153	0.334	0.462	0.152	0.340	0.467	0.156	0.326	0.454	0.156	0.328	0.455	0.155	0.330	0.457
KSTDiff		0.000	0.358	0.488	0.460	0.288	0.399	0.003	0.370	0.499	0.540	0.444	0.533	0.552	0.539	0.616	0.032	0.360	0.489
CGAN		0.584	0.379	0.488	0.539	0.371	0.487	0.521	0.386	0.500	0.461	0.359	0.476	0.546	0.492	0.582	0.509	0.437	0.535
Diffwave		0.490	0.504	0.616	0.384	0.456	0.570	0.464	0.430	0.533	0.492	0.500	0.609	0.360	0.437	0.555	0.479	0.437	0.556
DT		0.572	0.325	0.407	0.564	0.362	0.448	0.593	0.333	0.405	0.580	0.356	0.445	0.561	0.341	0.425	0.510	0.340	0.438
DDPM		0.398	0.300	0.419	0.417	0.306	0.427	0.455	0.313	0.428	0.490	0.291	0.397	0.481	0.300	0.411	0.509	0.313	0.417
CVAE		0.476	0.269	0.388	0.555	0.286	0.404	0.616	0.282	0.392	0.509	0.280	0.395	0.525	0.319	0.427	0.527	0.297	0.402
CRAFT		0.789	0.145	0.223	0.798	0.145	0.224	0.791	0.150	0.229	0.785	0.149	0.228	0.777	0.154	0.233	0.645	0.251	0.356
GMEL	Chicago(Outflow)	0.667	0.197	0.278	0.455	0.266	0.373	0.720	0.199	0.259	0.606	0.213	0.310	0.429	0.266	0.371	0.502	0.248	0.352
DFG		0.152	0.333	0.460	0.150	0.339	0.465	0.149	0.346	0.470	0.153	0.332	0.457	0.152	0.333	0.458	0.152	0.335	0.460
KSTDiff		0.000	0.363	0.491	0.214	0.335	0.462	0.006	0.375	0.501	0.533	0.628	0.708	0.569	0.478	0.554	0.151	0.356	0.479
CGAN		0.587	0.367	0.476	0.552	0.370	0.483	0.529	0.379	0.493	0.438	0.368	0.486	0.540	0.526	0.612	0.505	0.433	0.534
Diffwave		0.459	0.445	0.561	0.350	0.469	0.584	0.421	0.482	0.591	0.298	0.428	0.544	0.353	0.460	0.575	0.500	0.490	0.601
DT		0.546	0.346	0.434	0.531	0.346	0.437	0.579	0.346	0.423	0.550	0.349	0.438	0.550	0.364	0.449	0.536	0.342	0.434
DDPM		0.405	0.301	0.418	0.423	0.307	0.427	0.446	0.319	0.434	0.491	0.292	0.398	0.479	0.302	0.413	0.510	0.315	0.420
CVAE		0.472	0.274	0.392	0.562	0.287	0.404	0.620	0.283	0.392	0.507	0.281	0.395	0.534	0.316	0.422	0.532	0.294	0.398
CRAFT		0.791	0.146	0.223	0.797	0.147	0.225	0.797	0.148	0.225	0.792	0.146	0.223	0.789	0.148	0.224	0.656	0.246	0.349
GMEL	Washington, D.C.(Inflow)	0.728	0.234	0.284	0.609	0.289	0.383	0.649	0.272	0.360	0.764	0.223	0.261	0.726	0.227	0.292	0.711	0.234	0.301
DFG		0.697	0.246	0.348	0.696	0.252	0.354	0.698	0.254	0.355	0.692	0.252	0.354	0.692	0.254	0.356	0.691	0.256	0.358
KSTDiff		0.000	0.465	0.595	0.494	0.356	0.458	0.289	0.421	0.542	0.039	0.464	0.591	0.000	0.477	0.603	0.641	0.477	0.584
CGAN		0.596	0.342	0.465	0.510	0.420	0.538	0.575	0.440	0.550	0.583	0.391	0.507	0.653	0.480	0.593	0.442	0.391	0.514
Diffwave		0.578	0.505	0.624	0.274	0.466	0.584	0.407	0.557	0.663	0.450	0.577	0.677	0.641	0.506	0.623	0.457	0.580	0.673
DT		0.621	0.351	0.440	0.627	0.354	0.432	0.617	0.369	0.446	0.608	0.359	0.451	0.598	0.386	0.479	0.630	0.357	0.437
DDPM		0.476	0.355	0.477	0.417	0.418	0.534	0.488	0.403	0.510	0.541	0.349	0.458	0.609	0.340	0.443	0.463	0.370	0.475
CVAE		0.296	0.402	0.534	0.480	0.368	0.486	0.423	0.388	0.511	0.566	0.347	0.455	0.433	0.399	0.518	0.464	0.374	0.491
CRAFT		0.791	0.185	0.275	0.786	0.191	0.282	0.777	0.202	0.301	0.781	0.191	0.281	0.781	0.194	0.285	0.778	0.197	0.288
GMEL	Washington, D.C.(Outflow)	0.683	0.239	0.321	0.557	0.303	0.401	0.616	0.297	0.390	0.534	0.310	0.408	0.633	0.271	0.363	0.688	0.245	0.317
DFG		0.696	0.246	0.347	0.694	0.252	0.353	0.695	0.254	0.355	0.689	0.252	0.353	0.689	0.254	0.355	0.688	0.256	0.357
KSTDiff		0.000	0.470	0.598	0.581	0.338	0.412	0.004	0.488	0.612	0.000	0.476	0.601	0.536	0.503	0.622	0.653	0.513	0.629
CGAN		0.583	0.354	0.476	0.543	0.427	0.540	0.596	0.441	0.551	0.554	0.408	0.519	0.653	0.497	0.610	0.423	0.401	0.519
Diffwave		0.502	0.481	0.605	0.321	0.434	0.562	0.254	0.546	0.650	0.289	0.476	0.596	0.202	0.546	0.655	0.450	0.421	0.549
DT		0.580	0.371	0.461	0.579	0.361	0.450	0.623	0.377	0.464	0.583	0.357	0.452	0.589	0.387	0.478	0.621	0.372	0.463
DDPM		0.474	0.359	0.479	0.422	0.416	0.531	0.496	0.395	0.503	0.543	0.353	0.460	0.607	0.345	0.448	0.462	0.372	0.475
CVAE		0.301	0.404	0.533	0.484	0.368	0.483	0.429	0.386	0.507	0.551	0.355	0.461	0.436	0.400	0.516	0.464	0.376	0.489
CRAFT		0.790	0.188	0.277	0.783	0.195	0.285	0.773	0.207	0.304	0.781	0.193	0.280	0.781	0.196	0.284	0.778	0.199	0.288
GMEL	Toronto(Inflow)	0.748	0.230	0.281	0.721	0.240	0.309	0.639	0.283	0.366	0.558	0.313	0.407	0.672	0.260	0.344	0.703	0.255	0.326
DFG		0.275	0.421	0.535	0.271	0.432	0.544	0.269	0.441	0.551	0.275	0.423	0.536	0.274	0.425	0.537	0.272	0.430	0.542
KSTDiff		0.002	0.500	0.622	0.672	0.398	0.472	0.685	0.479	0.606	0.670	0.393	0.468	0.378	0.427	0.536	0.268	0.485	0.601
CGAN		0.631	0.358	0.474	0.394	0.436	0.546	0.528	0.399	0.513	0.546	0.378	0.478	0.556	0.387	0.495	0.421	0.404	0.512
Diffwave		0.550	0.424	0.553	0.634	0.462	0.588	0.498	0.523	0.633	0.567	0.475	0.594	0.506	0.488	0.605	0.583	0.471	0.587
DT		0.638	0.381	0.469	0.608	0.378	0.458	0.595	0.388	0.477	0.600	0.377	0.460	0.597	0.378	0.467	0.611	0.377	0.460
DDPM		0.625	0.360	0.471	0.649	0.367	0.479	0.600	0.388	0.498	0.628	0.375	0.484	0.608	0.373	0.483	0.602	0.381	0.490
CVAE		0.714	0.302	0.402	0.712	0.305	0.414	0.699	0.305	0.412	0.679	0.302	0.416	0.583	0.368	0.469	0.592	0.368	0.461
CRAFT		0.817	0.178	0.265	0.824	0.176	0.264	0.826	0.174	0.260	0.811	0.181	0.269	0.816	0.182	0.273	0.800	0.194	0.289
GMEL	Toronto(Outflow)	0.736	0.235	0.292	0.632	0.286	0.370	0.716	0.248	0.317	0.734	0.236	0.297	0.688	0.254	0.334	0.722	0.240	0.308
DFG		0.274	0.420	0.533	0.270	0.432	0.543	0.268	0.441	0.550	0.274	0.422	0.534	0.274	0.425	0.536	0.271	0.430	0.540
KSTDiff		0.002	0.498	0.619	0.682	0.411	0.492	0.685	0.479	0.604	0.673	0.402	0.482	0.646	0.419	0.513	0.310	0.492	0.609
CGAN		0.637	0.359	0.472	0.523	0.420	0.526	0.526	0.405	0.515	0.544	0.379	0.478	0.547	0.393	0.499	0.559	0.368	0.457
Diffwave		0.469	0.432	0.548	0.617	0.446	0.564	0.558	0.438	0.561	0.420	0.471	0.589	0.574	0.432	0.547	0.555	0.467	0.581
DT		0.619	0.381	0.469	0.588	0.394	0.482	0.619	0.381	0.465	0.611	0.365	0.444	0.595	0.397	0.489	0.604	0.380	0.467
DDPM		0.628	0.360	0.470	0.650	0.373	0.485	0.603	0.390	0.498	0.633	0.375	0.483	0.609	0.376	0.486	0.607	0.382	0.490
CVAE		0.709	0.307	0.408	0.705	0.313	0.420	0.695	0.310	0.415	0.680	0.305	0.417	0.591	0.365	0.463	0.606	0.358	0.449
CRAFT		0.814	0.182	0.270	0.821	0.180	0.270	0.822	0.181	0.269	0.810	0.182	0.271	0.814	0.188	0.280	0.803	0.195	0.289
GMEL	New York City(Inflow)	0.676	0.178	0.238	0.621	0.256	0.336	0.598	0.297	0.362	0.611	0.263	0.331	0.602	0.282	0.347	0.511	0.249	0.308
DFG		0.184	0.382	0.279	0.182	0.382	0.282	0.184	0.382	0.282	0.181	0.378	0.279	0.181	0.378	0.279	0.181	0.378	0.279
KSTDiff		0.355	0.442	0.552	0.035	0.232	0.362	0.000	0.232	0.370	0.372	0.471	0.578	0.282	0.169	0.226	0.356	0.184	0.236
CGAN		0.425	0.561	0.667	0.419	0.585	0.682	0.396	0.610	0.703	0.458	0.639	0.759	0.411	0.549	0.654	0.444	0.380	0.496
Diffwave		0.313	0.500	0.621	0.213	0.388	0.531	0.376	0.468	0.735	0.413	0.405	0.548	0.236	0.425	0.576	0.219	0.540	0.661
DT		0.470	0.384	0.499	0.441	0.324	0.422	0.470	0.335	0.420	0.440	0.398	0.480	0.471	0.351	0.435	0.447	0.375	0.464
DDPM		0.513	0.375	0.499	0.481	0.381	0.389	0.519	0.247	0.342	0.485	0.333	0.454	0.434	0.330	0.465	0.446	0.306	0.421
CVAE		0.442	0.406	0.534	0.483	0.323	0.454	0.473	0.347	0.471	0.492	0.273	0.406	0.487	0.438	0.323	0.470	0.430	0.473
CRAFT		0.769	0.115	0.															

Table 7: Data utility comparison on traffic flow prediction (LSTM)

Gen	Pred	Chicago				Washington, D.C.				Toronto				New York City			
		Inflow		Outflow		Inflow		Outflow		Inflow		Outflow		Inflow		Outflow	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Real	LSTM ($T = 48$)	0.097	0.151	0.100	0.154	0.094	0.144	0.097	0.150	0.109	0.163	0.111	0.164	0.056	0.101	0.056	0.101
GMEL		0.167	0.230	0.166	0.228	0.202	0.254	0.210	0.268	0.241	0.303	0.251	0.320	0.092	0.144	0.091	0.142
DFG		0.283	0.418	0.289	0.423	0.134	0.202	0.136	0.206	0.319	0.426	0.321	0.427	0.070	0.126	0.069	0.125
KSTDiff		0.362	0.493	0.368	0.497	0.469	0.599	0.475	0.603	0.507	0.629	0.505	0.627	0.348	0.389	0.762	0.814
CGAN		0.199	0.292	0.212	0.309	0.283	0.394	0.286	0.399	0.287	0.388	0.286	0.382	0.151	0.248	0.163	0.257
Diffwave		0.221	0.299	0.232	0.297	0.276	0.354	0.331	0.441	0.327	0.434	0.286	0.362	0.141	0.211	0.138	0.211
DiT		0.232	0.285	0.246	0.292	0.266	0.311	0.288	0.330	0.299	0.337	0.307	0.343	0.252	0.280	0.209	0.245
DDPM		0.113	0.174	0.115	0.171	0.124	0.183	0.122	0.183	0.117	0.171	0.122	0.175	0.065	0.113	0.066	0.110
CVAE		0.175	0.255	0.172	0.249	0.308	0.429	0.304	0.421	0.158	0.230	0.162	0.233	0.138	0.219	0.135	0.213
CRAFT		0.099	0.153	0.103	0.158	0.103	0.156	0.111	0.167	0.120	0.173	0.127	0.180	0.063	0.107	0.064	0.105
Real	LSTM ($T = 72$)	0.104	0.156	0.107	0.159	0.095	0.145	0.099	0.151	0.114	0.169	0.115	0.170	0.059	0.106	0.059	0.105
GMEL		0.169	0.234	0.175	0.246	0.227	0.285	0.240	0.302	0.258	0.329	0.270	0.349	0.093	0.140	0.097	0.153
DFG		0.303	0.435	0.306	0.438	0.142	0.213	0.151	0.226	0.368	0.481	0.370	0.482	0.073	0.129	0.073	0.129
KSTDiff		0.256	0.340	0.282	0.381	0.320	0.400	0.314	0.372	0.336	0.380	0.340	0.403	0.224	0.361	0.205	0.343
CGAN		0.208	0.300	0.206	0.294	0.294	0.401	0.294	0.396	0.407	0.510	0.409	0.519	0.262	0.327	0.267	0.335
Diffwave		0.244	0.310	0.239	0.342	0.312	0.352	0.296	0.363	0.360	0.462	0.405	0.492	0.175	0.236	0.142	0.214
DiT		0.259	0.300	0.228	0.283	0.292	0.332	0.306	0.354	0.321	0.366	0.312	0.355	0.169	0.231	0.204	0.244
DDPM		0.116	0.171	0.117	0.176	0.133	0.186	0.134	0.190	0.121	0.177	0.127	0.183	0.072	0.122	0.074	0.119
CVAE		0.175	0.256	0.177	0.258	0.229	0.319	0.233	0.321	0.174	0.248	0.177	0.250	0.108	0.178	0.109	0.180
CRAFT		0.109	0.165	0.111	0.167	0.111	0.166	0.116	0.171	0.124	0.179	0.127	0.180	0.066	0.111	0.067	0.108
Real	LSTM ($T = 96$)	0.108	0.161	0.110	0.163	0.094	0.144	0.098	0.149	0.117	0.175	0.119	0.175	0.062	0.111	0.063	0.110
GMEL		0.174	0.245	0.182	0.248	0.231	0.288	0.248	0.312	0.261	0.319	0.262	0.313	0.102	0.152	0.139	0.198
DFG		0.322	0.460	0.326	0.462	0.152	0.224	0.158	0.230	0.351	0.455	0.351	0.454	0.081	0.140	0.081	0.142
KSTDiff		0.371	0.502	0.376	0.505	0.416	0.540	0.494	0.617	0.480	0.608	0.480	0.606	0.234	0.372	0.235	0.371
CGAN		0.261	0.372	0.264	0.376	0.397	0.515	0.409	0.519	0.335	0.453	0.336	0.454	0.170	0.240	0.164	0.232
Diffwave		0.252	0.341	0.236	0.321	0.319	0.378	0.297	0.350	0.311	0.384	0.327	0.412	0.149	0.229	0.182	0.271
DiT		0.234	0.287	0.242	0.287	0.297	0.335	0.296	0.333	0.316	0.357	0.302	0.341	0.230	0.260	0.230	0.260
DDPM		0.116	0.172	0.119	0.177	0.122	0.179	0.122	0.184	0.128	0.183	0.135	0.190	0.074	0.125	0.074	0.118
CVAE		0.152	0.225	0.159	0.233	0.301	0.405	0.300	0.402	0.170	0.240	0.174	0.241	0.129	0.202	0.131	0.203
CRAFT		0.112	0.170	0.115	0.171	0.111	0.162	0.115	0.171	0.129	0.188	0.134	0.191	0.069	0.118	0.068	0.112
Real	LSTM ($T = 120$)	0.106	0.160	0.108	0.161	0.095	0.145	0.099	0.149	0.121	0.183	0.122	0.183	0.062	0.111	0.063	0.110
GMEL		0.184	0.243	0.187	0.256	0.213	0.260	0.218	0.271	0.277	0.333	0.262	0.312	0.099	0.152	0.105	0.155
DFG		0.306	0.445	0.311	0.448	0.168	0.249	0.172	0.252	0.352	0.467	0.356	0.472	0.089	0.154	0.089	0.154
KSTDiff		0.390	0.434	0.624	0.702	0.458	0.588	0.473	0.599	0.334	0.377	0.337	0.388	0.767	0.819	0.390	0.444
CGAN		0.275	0.374	0.283	0.387	0.293	0.404	0.286	0.395	0.314	0.409	0.313	0.410	0.136	0.228	0.137	0.227
Diffwave		0.236	0.294	0.233	0.299	0.387	0.495	0.324	0.413	0.307	0.383	0.294	0.358	0.151	0.245	0.143	0.235
DiT		0.231	0.284	0.242	0.286	0.276	0.319	0.277	0.321	0.302	0.344	0.299	0.344	0.270	0.297	0.175	0.219
DDPM		0.116	0.172	0.118	0.177	0.123	0.181	0.122	0.181	0.132	0.188	0.141	0.196	0.070	0.118	0.069	0.112
CVAE		0.151	0.217	0.155	0.222	0.214	0.304	0.220	0.310	0.166	0.239	0.164	0.234	0.133	0.215	0.130	0.208
CRAFT		0.110	0.168	0.112	0.170	0.110	0.166	0.116	0.170	0.129	0.187	0.131	0.190	0.073	0.115	0.074	0.114
Real	LSTM ($T = 144$)	0.107	0.160	0.110	0.162	0.095	0.144	0.098	0.148	0.125	0.186	0.127	0.186	0.062	0.109	0.061	0.108
GMEL		0.184	0.242	0.221	0.327	0.209	0.262	0.238	0.303	0.261	0.317	0.263	0.317	0.086	0.137	0.091	0.138
DFG		0.315	0.454	0.319	0.456	0.155	0.227	0.157	0.229	0.368	0.487	0.372	0.490	0.086	0.145	0.086	0.145
KSTDiff		0.524	0.585	0.456	0.505	0.481	0.607	0.350	0.386	0.341	0.419	0.377	0.423	0.200	0.334	0.188	0.294
CGAN		0.394	0.495	0.420	0.515	0.459	0.579	0.470	0.590	0.362	0.460	0.355	0.452	0.172	0.255	0.176	0.260
Diffwave		0.244	0.335	0.312	0.425	0.290	0.358	0.295	0.346	0.333	0.440	0.366	0.459	0.167	0.251	0.142	0.210
DiT		0.226	0.280	0.224	0.276	0.306	0.345	0.305	0.342	0.304	0.344	0.311	0.352	0.195	0.234	0.206	0.240
DDPM		0.117	0.176	0.121	0.183	0.123	0.182	0.125	0.185	0.132	0.188	0.137	0.194	0.076	0.121	0.078	0.117
CVAE		0.209	0.293	0.210	0.293	0.278	0.370	0.279	0.370	0.226	0.314	0.228	0.316	0.154	0.255	0.153	0.252
CRAFT		0.113	0.170	0.115	0.170	0.114	0.169	0.118	0.171	0.130	0.189	0.137	0.196	0.075	0.122	0.077	0.118
Real	LSTM ($T = 168$)	0.112	0.169	0.115	0.172	0.098	0.146	0.101	0.152	0.121	0.179	0.122	0.179	0.061	0.107	0.061	0.107
GMEL		0.204	0.280	0.205	0.283	0.198	0.256	0.211	0.258	0.225	0.268	0.226	0.279	0.098	0.148	0.097	0.148
DFG		0.299	0.431	0.304	0.433	0.153	0.223	0.156	0.228	0.312	0.414	0.313	0.414	0.089	0.154	0.089	0.153
KSTDiff		0.348	0.480	0.335	0.462	0.443	0.524	0.505	0.623	0.401	0.516	0.379	0.481	0.199	0.339	0.195	0.305
CGAN		0.412	0.504	0.426	0.525	0.395	0.496	0.390	0.484	0.335	0.412	0.340	0.416	0.181	0.269	0.173	0.261
Diffwave		0.238	0.306	0.227	0.295	0.334	0.388	0.444	0.573	0.307	0.354	0.331	0.407	0.166	0.260	0.147	0.220
DiT		0.235	0.296	0.236	0.295	0.289	0.326	0.285	0.324	0.315	0.351	0.317	0.357	0.218	0.250	0.168	0.214
DDPM		0.118	0.175	0.121	0.182	0.131	0.181	0.134	0.188	0.125	0.182	0.137	0.195	0.073	0.119	0.075	0.119
CVAE		0.228	0.312	0.228	0.311	0.316	0.416	0.313	0.411	0.187	0.251	0.190	0.251	0.180	0.260	0.180	0.256
CRAFT		0.114	0.170	0.117	0.173	0.111	0.165	0.115	0.171	0.124	0.180	0.133	0.188	0.078	0.128	0.078	0.122

Table 8: Data utility comparison on traffic flow prediction (Transformer)

Gen	Pred	Chicago				Washington, D.C.				Toronto				New York City			
		Inflow		Outflow		Inflow		Outflow		Inflow		Outflow		Inflow		Outflow	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Real	Transformer ($T = 48$)	0.099	0.158	0.103	0.162	0.094	0.149	0.098	0.155	0.111	0.170	0.114	0.172	0.053	0.102	0.054	0.103
GMEL		0.165	0.228	0.165	0.227	0.201	0.260	0.199	0.253	0.220	0.274	0.223	0.284	0.080	0.133	0.081	0.131
DFG		0.300	0.424	0.300	0.418	0.184	0.250	0.176	0.240	0.341	0.453	0.328	0.435	0.085	0.134	0.081	0.130
KSTDiff		0.367	0.500	0.377	0.510	0.469	0.604	0.476	0.609	0.571	0.723	0.530	0.670	0.337	0.378	0.756	0.807
CGAN		0.159	0.221	0.171	0.237	0.282	0.379	0.284	0.382	0.239	0.310	0.235	0.304	0.128	0.193	0.135	0.201
Diffwave		0.220	0.292	0.212	0.279	0.278	0.345	0.284	0.354	0.291	0.356	0.285	0.347	0.156	0.228	0.140	0.211
DiT		0.235	0.289	0.236	0.287	0.269	0.312	0.282	0.320	0.296	0.339	0.298	0.337	0.181	0.225	0.158	0.212
DDPM		0.111	0.169	0.115	0.173	0.113	0.166	0.115	0.172	0.119	0.174	0.123	0.178	0.063	0.111	0.065	0.110
CVAE	Transformer ($T = 72$)	0.162	0.228	0.164	0.229	0.230	0.311	0.237	0.318	0.172	0.222	0.176	0.226	0.114	0.181	0.116	0.183
CRAFT		0.097	0.148	0.100	0.152	0.097	0.149	0.105	0.161	0.124	0.174	0.124	0.173	0.060	0.103	0.060	0.102
Real	Transformer ($T = 96$)	0.102	0.159	0.105	0.162	0.094	0.147	0.098	0.153	0.116	0.176	0.118	0.177	0.055	0.100	0.056	0.100
GMEL		0.186	0.251	0.186	0.258	0.249	0.311	0.240	0.298	0.252	0.303	0.248	0.298	0.083	0.138	0.083	0.132
DFG		0.362	0.491	0.350	0.486	0.166	0.222	0.166	0.222	0.391	0.511	0.390	0.509	0.069	0.116	0.070	0.118
KSTDiff		0.349	0.514	0.364	0.519	0.383	0.511	0.376	0.488	0.356	0.401	0.352	0.413	0.216	0.355	0.207	0.359
CGAN		0.179	0.243	0.187	0.249	0.254	0.345	0.252	0.333	0.391	0.493	0.359	0.458	0.146	0.190	0.149	0.194
Diffwave		0.239	0.297	0.228	0.294	0.299	0.352	0.279	0.328	0.324	0.369	0.339	0.382	0.143	0.211	0.143	0.209
DiT		0.243	0.291	0.225	0.281	0.285	0.328	0.287	0.331	0.321	0.364	0.311	0.354	0.177	0.234	0.188	0.234
DDPM		0.108	0.161	0.113	0.167	0.132	0.187	0.131	0.186	0.128	0.184	0.130	0.185	0.067	0.117	0.070	0.118
CVAE	Transformer ($T = 120$)	0.175	0.243	0.176	0.244	0.224	0.305	0.227	0.306	0.192	0.255	0.198	0.259	0.106	0.165	0.109	0.170
CRAFT		0.103	0.155	0.106	0.158	0.106	0.158	0.109	0.161	0.121	0.174	0.124	0.176	0.060	0.104	0.060	0.104
Real	Transformer ($T = 144$)	0.105	0.162	0.108	0.165	0.093	0.143	0.096	0.147	0.115	0.175	0.117	0.176	0.059	0.105	0.059	0.104
GMEL		0.186	0.256	0.180	0.240	0.218	0.276	0.232	0.294	0.231	0.281	0.236	0.286	0.091	0.141	0.145	0.184
DFG		0.341	0.467	0.323	0.442	0.175	0.237	0.168	0.228	0.354	0.467	0.360	0.472	0.068	0.116	0.069	0.117
KSTDiff		0.387	0.526	0.397	0.538	0.420	0.546	0.492	0.616	0.448	0.564	0.484	0.604	0.243	0.386	0.236	0.375
CGAN		0.195	0.268	0.197	0.267	0.362	0.447	0.385	0.476	0.300	0.388	0.296	0.380	0.124	0.175	0.133	0.187
Diffwave		0.240	0.297	0.235	0.304	0.272	0.320	0.274	0.319	0.294	0.342	0.297	0.348	0.144	0.209	0.139	0.205
DiT		0.231	0.287	0.227	0.280	0.285	0.327	0.278	0.321	0.308	0.346	0.297	0.339	0.146	0.208	0.151	0.208
DDPM		0.114	0.169	0.118	0.173	0.122	0.175	0.121	0.176	0.130	0.183	0.138	0.191	0.070	0.116	0.071	0.115
CVAE	Transformer ($T = 168$)	0.153	0.214	0.158	0.219	0.255	0.332	0.261	0.336	0.170	0.230	0.174	0.231	0.113	0.169	0.117	0.169
CRAFT		0.106	0.160	0.108	0.162	0.104	0.150	0.108	0.158	0.122	0.178	0.128	0.183	0.063	0.109	0.064	0.108
Real	Transformer ($T = 192$)	0.100	0.155	0.103	0.159	0.090	0.140	0.093	0.144	0.116	0.175	0.118	0.177	0.057	0.105	0.057	0.105
GMEL		0.192	0.256	0.193	0.261	0.222	0.290	0.232	0.304	0.249	0.305	0.250	0.306	0.094	0.150	0.098	0.156
DFG		0.310	0.425	0.303	0.413	0.150	0.203	0.151	0.208	0.349	0.460	0.354	0.465	0.071	0.122	0.072	0.123
KSTDiff		0.390	0.436	0.621	0.699	0.509	0.654	0.507	0.640	0.328	0.380	0.329	0.386	0.721	0.771	0.341	0.420
CGAN		0.270	0.369	0.282	0.390	0.283	0.375	0.288	0.381	0.273	0.349	0.261	0.338	0.108	0.175	0.109	0.173
Diffwave		0.221	0.281	0.222	0.284	0.270	0.342	0.262	0.321	0.309	0.350	0.294	0.343	0.140	0.207	0.137	0.202
DiT		0.222	0.280	0.222	0.276	0.265	0.318	0.268	0.315	0.306	0.349	0.303	0.347	0.173	0.220	0.149	0.208
DDPM		0.109	0.161	0.113	0.166	0.112	0.163	0.113	0.166	0.129	0.184	0.135	0.190	0.067	0.111	0.066	0.109
CVAE	Transformer ($T = 216$)	0.153	0.214	0.158	0.220	0.202	0.272	0.212	0.283	0.170	0.231	0.173	0.231	0.106	0.171	0.105	0.168
CRAFT		0.107	0.162	0.109	0.163	0.103	0.152	0.107	0.156	0.120	0.173	0.123	0.177	0.063	0.104	0.063	0.104
Real	Transformer ($T = 252$)	0.101	0.155	0.104	0.159	0.091	0.140	0.093	0.144	0.118	0.177	0.120	0.180	0.055	0.104	0.056	0.104
GMEL		0.211	0.281	0.209	0.285	0.242	0.301	0.234	0.284	0.264	0.317	0.263	0.312	0.088	0.140	0.090	0.142
DFG		0.323	0.450	0.307	0.427	0.188	0.248	0.185	0.247	0.399	0.521	0.391	0.510	0.069	0.120	0.069	0.119
KSTDiff		0.524	0.586	0.459	0.510	0.488	0.618	0.369	0.423	0.367	0.462	0.385	0.446	0.190	0.315	0.166	0.239
CGAN		0.342	0.431	0.392	0.480	0.371	0.463	0.352	0.449	0.284	0.360	0.288	0.368	0.133	0.186	0.135	0.184
Diffwave		0.221	0.280	0.215	0.273	0.291	0.334	0.296	0.353	0.290	0.357	0.288	0.359	0.139	0.205	0.136	0.202
DiT		0.223	0.279	0.221	0.276	0.290	0.331	0.283	0.322	0.297	0.338	0.301	0.342	0.147	0.208	0.143	0.203
DDPM		0.113	0.167	0.117	0.173	0.122	0.173	0.124	0.176	0.127	0.181	0.131	0.185	0.071	0.113	0.069	0.110
CVAE	Transformer ($T = 288$)	0.192	0.260	0.193	0.261	0.277	0.359	0.283	0.363	0.234	0.297	0.240	0.305	0.124	0.198	0.124	0.196
CRAFT		0.107	0.160	0.110	0.163	0.106	0.154	0.111	0.159	0.125	0.180	0.129	0.183	0.066	0.107	0.065	0.105
Real	Transformer ($T = 336$)	0.104	0.160	0.108	0.164	0.090	0.140	0.093	0.144	0.115	0.173	0.118	0.176	0.054	0.102	0.055	0.103
GMEL		0.216	0.288	0.217	0.287	0.219	0.281	0.218	0.277	0.238	0.301	0.241	0.297	0.093	0.148	0.095	0.150
DFG		0.315	0.427	0.313	0.424	0.162	0.217	0.167	0.226	0.377	0.493	0.374	0.492	0.071	0.122	0.072	0.123
KSTDiff		0.344	0.473	0.340	0.467	0.444	0.526	0.510	0.625	0.395	0.508	0.374	0.473	0.199	0.339	0.196	0.308
CGAN		0.385	0.469	0.385	0.476	0.324	0.418	0.323	0.412	0.319	0.385	0.346	0.413	0.140	0.200	0.145	0.210
Diffwave		0.228	0.288	0.221	0.283	0.281	0.328	0.267	0.321	0.290	0.339	0.289	0.341	0.142	0.206	0.136	0.202
DiT		0.231	0.288	0.227	0.282	0.283	0.325	0.267	0.318	0.305	0.342	0.308	0.347	0.192	0.231	0.154	0.206
DDPM		0.115	0.169	0.119	0.174	0.129	0.178	0.132	0.182	0.127	0.178	0.136	0.187	0.071	0.112	0.069	0.110
CVAE	Transformer ($T = 384$)	0.193	0.257	0.195	0.259	0.285	0.369	0.287	0.370	0.196	0.247	0.202	0.252	0.138	0.189	0.138	0.188
CRAFT		0.106	0.158	0.108	0.162	0.102	0.149	0.106	0.154	0.122	0.176	0.125	0.179	0.069	0.114	0.067	0.111

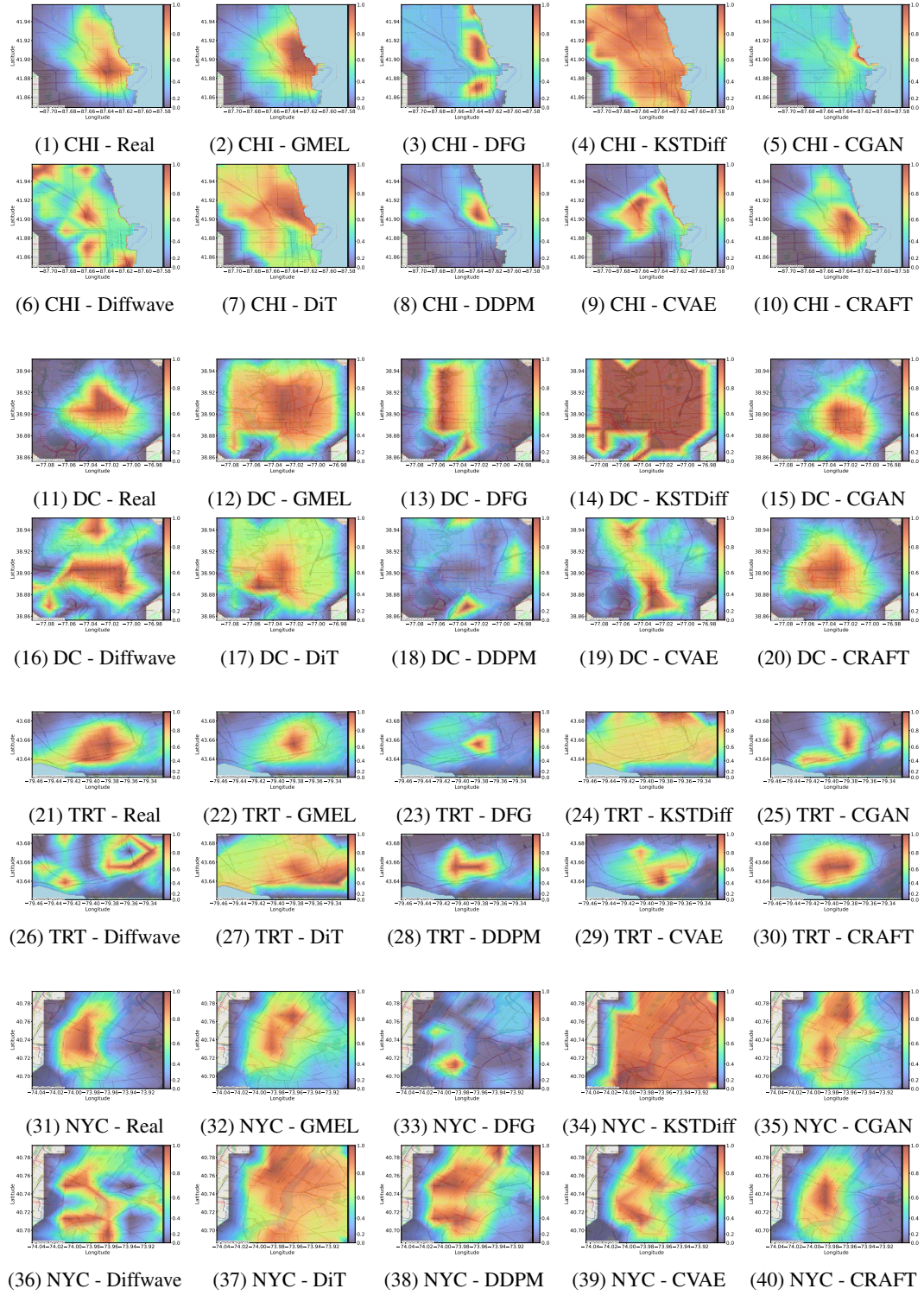


Figure 11: Heatmap of Traffic Flow in Different Cities (CHI stands for Chicago, DC stands for Washington, D.C., TRT stands for Toronto and NYC stands for New York City)

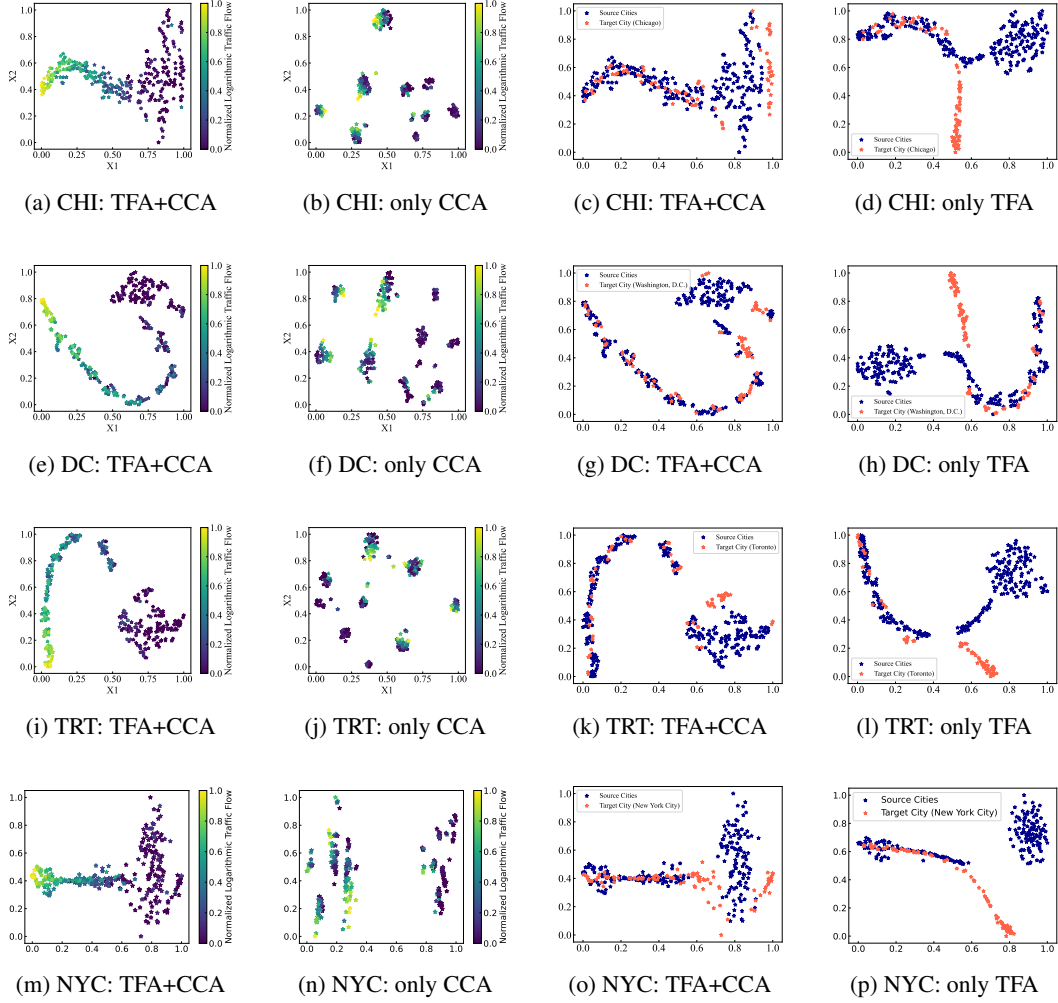


Figure 12: Visualization analysis for TFA and CCA (CHI stands for Chicago, DC stands for Washington, D.C., TRT stands for Toronto and NYC stands for New York City)