

Fine-Grained Urban Flow Inferring via Conditional Generative Adversarial Networks

Xv Zhang^{*}[0000-0002-2410-1282], Yuanbo Xu^{**}[0000-0001-8370-5011],
Ying Li^{*}[0000-0002-7804-149X], and Yongjian Yang^{*}[0000-0002-0056-3626]

^{*}Mobile Computing Intelligent (MIC) Laboratory; Key Laboratory of Symbolic Computation and Knowledge Engineering for the Ministry of Education, Jilin University

xuz20@mails.jlu.edu.cn
{yuanbox, liying, yyj}@jlu.edu.cn

Abstract. Urban flow super-resolution (UFSR) can deduce fine-grained urban flow heatmap (UFH) based on coarse-grained observations and plays an essential role in urban planning (traffic prediction, public facility deployment, for instance). However, existing methods fail to capture the internal structural features of sparse UFHs and the external factors that lead to a significant waste of urban resources. To this end, we propose an enhanced super-resolution framework (Urban Flow-aware Super Resolution - Generative Adversarial Network, UrbanSG) to deduce fine-grained UFH for urban resource allocation. Specifically, we employ a conditional-GAN as the backbone, considering external factors as the specified condition. To capture the implicit urban structural correlation, we integrate the flow self-attention mechanism into our model, which focuses on urban grids with active traffic volumes. The evaluations of extensive experiments on two real-world datasets demonstrate the superiority of our framework. Especially when dealing with a sparse dataset, our method reduces error by 15.02% to the state-of-the-art baselines.

Keywords: Urban Computing · Super Resolution · Deep Learning · Attention Mechanism.

1 Introduction

With the rapid development of intelligent terminals and wireless communication, an unprecedented amount of data springs up, improving numerous essential applications in the city, such as intelligent traffic management and public facility deployment. To sufficiently obtain fine-grained data, an intelligent urban system requires many sensing devices to cover the entire urban landform, which simultaneously imposes significant O&M (operation and maintenance) overhead and becomes a prohibitive factor for the global intelligence without prompt resource alignment strategies and proper allocation method. Hence, reconstructing the fine-grained urban situation through available coarse-grained observation becomes an urgent issue.

Urban flow super-resolution (UFSR) aims at inferring fine-grained traffic flows with an available coarse-grained observation. UFSR has brought significant urban planning

^{*} Corresponding Author: Yuanbo Xu

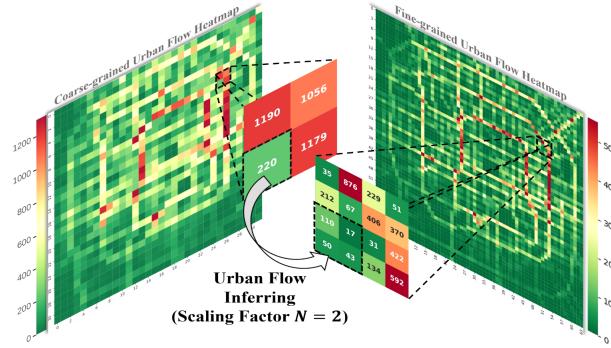


Fig. 1. UFSR: coarse-grained to fine-grained.

and traffic management improvements as a variant and application of image SR for transportation. For example, recovering fine urban traffic flow volume with coarse-grained observed data captured from limited traffic monitoring devices reduces redundant resource allocation. Although UFSR is closely related to image SR, it confronts a unique structural constraint. Specifically, the sum of the fine-grained traffic volumes of a super-resolution region is strictly equal to that of the corresponding coarse-grained super-resolution region, as shown in Figure 1. In other words, we consider UFSR as a mapping problem, which maps low information entropy data to that of high information entropy [11].

Nevertheless, in the complex context of the smart city, UFSR still suffers from the following challenges:

1. External Factors: Urban flow distribution is generally affected by external factors, such as time and weather. For example, office area usually has a higher density than attractions on weekdays, while the opposite is true on weekends; People prefer to be at home on stormy days and outside on sunny days. Hence, the same functional area shows various flow distributions under varying external factors. Without adequate consideration of external factors, the model will not yield a well-detailed fit.
2. Flow distribution sparsity: The high sparsity of UFH in the real world is often insufficient to support existing FSR models. Figure 2(a) shows that the base stations (represented by red dots) cannot be assigned to all urban grids due to the limited quantity and uneven distribution, which leads to data sparsity. Figure 2(b) reflects the sparsity difference under different time slices. The model without sparse data processing capability will degrade performance when facing real-world urban data.

To this end, we devise an enhanced super-resolution framework that can deduce fine-grained UFH from a coarser one. To the best of our knowledge, UrbanSG is the first attempt to exploit CGAN to complete the works above. Notably, the contributions of our work lie in the following aspects:

1. To jointly consider the city's internal relations and external factors, we present UrbanSG, a novel deep neural network model. It employs a CGAN as a backbone that considers external factors as the city's specific condition.

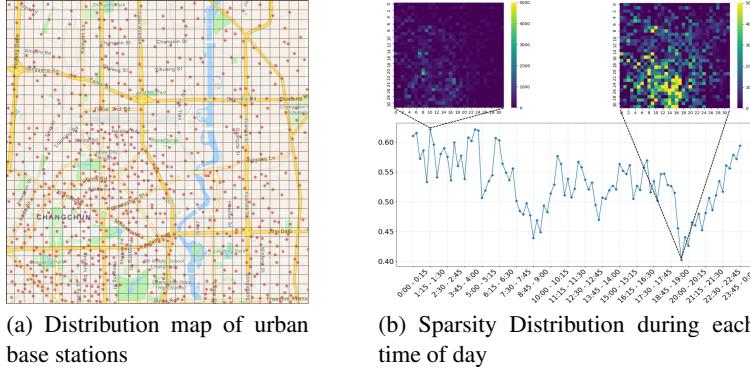


Fig. 2. Mobile signaling data in Changchun.

2. We integrate the self-attention mechanism into our model, which conducts UrbanSG focus on the active urban grids and effectively alleviates the sparsity problem.
3. We process, analyze, and experiment in two urban scenarios. Our experimental results verify the significant advantages of UrbanSG in both effectiveness and efficiency. Moreover, the experiments from multiple perspectives validate the rationale for different components of our model.

2 Related Work

2.1 Image Super Resolution

Image super-resolution (ISR) aims to recover low-resolution images to the original high-resolution images, which has emerged as a popular research topic in CV for decades and has a promising application in image recognition [5, 14], image augmentation [2, 8], and urban computing [11]. Over the years, researchers have successfully proposed many ISR algorithms, mainly categorized into super-resolution recovery [17, 18] and super-resolution reconstruction [1]. To fully utilize the hierarchical features on residual branches, Liu et al. [12] proposed a novel residual feature aggregation framework for more efficient feature extraction. Similarly, for optimizing the residual networks, [19] introduced the global context module to streamline the residual network effectively and [22] proposed a connection block group to prevent an excess of residual network parameters by distillation technique and attention mechanism. Yang et al. [18] proposed a novel texture transformer network consisting of four closely-related modules for texture transfer and synthesis. Compared with the non-local sparse attention pattern in [16]. Yan et al. [17] proposed a graph attention network with recurrent feature mapping blocks that take advantage of the internal patch-recurrence in a natural image.

2.2 Urban Flow Analysis

Urban flow forecasting plays a significant role in several aspects, such as traffic management [11] and risk assessment [4]. Gu et al. [7] developed an interpretable bicycle flow forecasting method, which first divided the entire city into zones based on flow density to provide effective bicycle flow forecasting and interpretable flow patterns. Pan et al. [15] proposed a spatial-temporal relationship network, which accurately predicts fine-grained traffic volume at a specific time by modeling coarse-grained traffic data. Zhang et al. [20] proposed a novel graph neural network that integrates local spatial semantics into a global pattern representation of the whole city through a city’s custom graph diffusion paradigm representation. Gong et al. [6] proposed three spatial-temporal models to solve the above problem to explore the dependence between time and space in urban computing and improve the prediction accuracy. In addition to the applications mentioned above, we aim to address a novel problem regarding urban transportation in this study.

3 Formulation

In this section, we declare four specific concepts and then define the problem of UFSR.

Definition 1 (Urban Grid) We divide the urban sampling area into $I \times J$ square regions based on latitude and longitude. As shown in Figure 1, each square region denotes an urban grid. For an area of interest with a given size, smaller urban grids (i.e., using more significant I, J) suggest that we can obtain a lower granularity of signal data to build a finer-grained UFH.

Definition 2 (Urban Flow Heatmap) Let $M \in \mathbb{R}^{I \times J}$ represents an urban flow heatmap, where each element $m^{i,j} \in N_+$ denotes the traffic volume of the urban grid (I, J) . We define ‘Urban flow’ in traffic volume and people flow, counting the number of vehicles (taxis, for example) passed and the number of check-in records for mobile users in each urban grid separately.

Definition 3 (Super-Resolution Region) We define adjacent $N \times N$ urban grids as a super-resolution region given a scaling factor N . Figure 1 illustrates an example consisting of 128×128 high granularity urban grids. We can obtain 32×32 non-overlapping super-resolution regions, which finally transform into the corresponding coarse-grained UFH with the guide of urban structural constraint.

Definition 4 (Urban Structural Constraint) Similar to [11], let $m_C^{i,j} \in M_C$ be the traffic volume of the urban grid in coarse-grained UFH and $m_F^{i',j'} \in M_F$ be the flows in the corresponding super-resolution region. The structural constraint between $m_C^{i,j}$ and $m_F^{i',j'}$ defines as follows:

$$m_C^{i,j} = \sum_{i',j'} m_F^{i',j'} \quad s.t. \lfloor \frac{i}{N} \rfloor = i, \lfloor \frac{j}{N} \rfloor = j, \quad (1)$$

where N denotes the scaling factor of our UFSR problem, $i = 1, 2, \dots, I$ and $j = 1, 2, \dots, J$.

Problem Statement (Urban Flow Super-Resolution) Given a coarse-grained UFH M_C , a scaling factor N , and the external factor E (holiday, weather, for example), the UFSR task is to learn a model G , which maps $M_C \in \mathbb{R}^{I \times J}$ into a fine-grained UFH $M_F \in \mathbb{R}^{NI \times NJ}$:

$$M_F = G(M_C | E, N; \theta), \quad (2)$$

where θ denotes all trainable parameters of model G .

4 Methodology

Figure 3 depicts the framework of UrbanSG, which infers a high-resolution, super-resolved UFH M_F from a low-resolution input UFH M_C . In this section, we first provide a detailed description of our framework’s workflow. Then we present our strategy for dealing with the adverse effects of sparse urban flow and complex external factors separately.

4.1 Adversarial Network Architecture

We employ a Generative Adversarial Network as the backbone to conduct the structurally constrained impact with an adversarial training process. Following the traditional GANs, UrbanSG consists of two parts: a generator G_{θ_G} and a discriminator D_{θ_D} . We optimize them together in an alternating manner to solve the adversarial min-max problem:

$$\begin{aligned} & \min_{\theta_G} \max_{\theta_D} (\mathbb{E}_{M_F \sim p_{train}(M_F)} [\log D_{\theta_D}(M_F)] \\ & + \mathbb{E}_{M_C \sim p_{train}(M_C)} [\log(1 - D_{\theta_D}(G_{\theta_G}(M_C)))]), \end{aligned} \quad (3)$$

which aims to learn a generator G to fool the discriminator D , which is trained to distinguish between super-resolution UFHs $G_{\theta_G}(M_C)$ and honest UFHs M_F . The generator G can learn the structural constraints and the distribution associations within super-resolution regions.

First, inspired by [9], we employ B residual blocks with the same layout as the deep core of our generator network G . To make the model better suited to the UFSR task, we modify the convolution kernel filter size, the number of input channels, and the activation function of the stacked residual blocks separately. Then, we employ a upsample block to double the size of the feature map (i.e., $M'_C \in \mathbb{R}^{2I \times 2J}$). Next, we present the flow self-attention module, which conducts the model focus on active urban grids from both inter-channel and intra-channel. Considering the urban structural constraint, we present the sum restriction module that corrects all urban grids’ values in UFH according to the volume distribution inside each super-resolution region. Finally, with the workflow of G , we obtain the fine-grained UFH $G_{\theta_G}(M_C) \in \mathbb{R}^{NI \times NJ}$ inferred from coarse-grained UFH M_C .

To discriminate real fine-grained UFH M_F from generated super-resolution samples, we train a discriminator network that solves the min-max problem in Equation 2. Inspired by the VGG network, we adopt a similar model architecture that stacks H convolutional blocks with an increasing number of convolutional kernels in each block, simple but efficient.

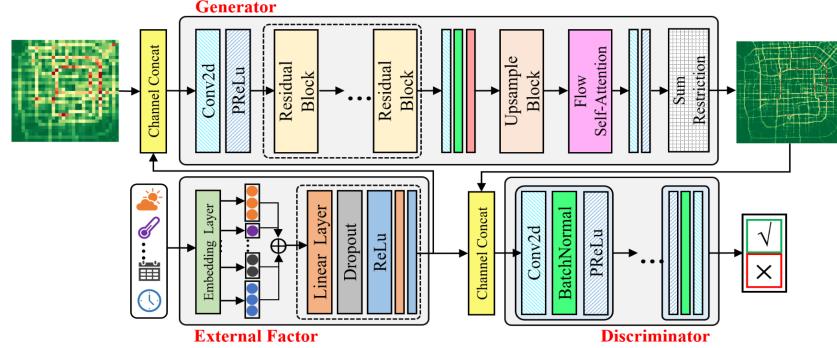


Fig. 3. The framework of UrbanSG with $4 \times$ up-scaling ($N=4$).

Feature Extraction Module In this section, we specify the feature extraction process and the implementation details of the residual block. We first extract the feature representation of external factors and treat it as an additional channel of the input UFH $M_C \sim P_{train}(M_C)$ for further feature extraction. Before entering the residual network, we apply a single convolution layer (filter size 9×9 , channel C) to extract the low-level features. Then the extracted latent features serve as input of the following B Residual Blocks. As shown in Figure 3, the layout of the residual block follows the guideline in [9], which employs two convolutional layers, a Batch Normalization and a PReLU function. Notably, to improve the model performance, we follow the N^2 -normalization operation proposed in [11], which calculates the distribution of each urban grid; this allows the convolution process to focus further on the distribution within each super-resolution region. The detailed process is as follows:

$$\begin{aligned} Sum_{\tilde{M}}^C &= SumPooling(\tilde{M}, D), \\ Sum_{\tilde{M}}^F &= NNupsampling(Sum_{\tilde{M}}, D), \\ \tilde{M}' &= \tilde{M} \oslash Sum_{\tilde{M}}^F, \end{aligned} \quad (4)$$

where \tilde{M} represents a feature map in the convolution process, D is the upscale factor of our UFSR task, and \oslash represents element-wise division.

Finally, considering that the output flow distribution can demonstrate inter-regional dependence on the original coarse-grained UFH M_C and achieve efficient gradient back-propagation, we apply a skip connection which creates an information highway and brings an identity mapping into the feature extraction process.

$$\begin{aligned} m_C^{i,j} &= \sum_{i',j'} \lambda_{i',j'} m_F^{i',j'} \\ s.t. \sum_{i',j'} \lambda_{i',j'} &= 1, \lambda \in N \quad \lfloor \frac{i'}{N} \rfloor = i, \lfloor \frac{j'}{N} \rfloor = j, \end{aligned} \quad (5)$$

where λ denotes the current urban grid traffic volume distribution in the corresponding super-resolution region.

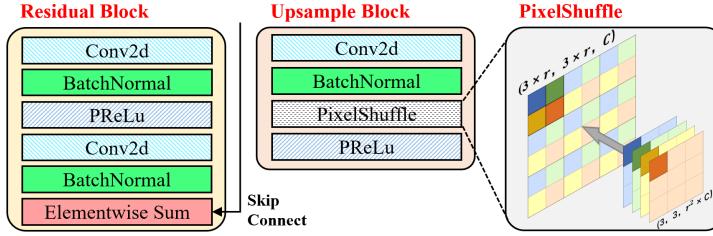


Fig. 4. The workflow of Residual Block and Upsample Block.

4.2 External Factor Integration

Significantly, it is insufficient to consider merely historical traffic data for our UFSR task. The existence of external factors can also make a vital and complicated impact on the urban grid's traffic volume. The actual impact on traffic flow becomes even implicit and non-negligible when different external factors are entwined. To this end, we build our model based on CGAN [13]: this variant of the original GAN allows generator G to produce condition-specific data by providing the critical condition (holiday, weather, for example) as an additional input to G along with a random noise input (Coarse-grained UFH in our UFSR task). Thus, we reformulate Equation 3 as follows:

$$\begin{aligned} & \min_{\theta_G} \max_{\theta_D} (\mathbb{E}_{M_F \sim p_{train}(M_F)} [\log D_{\theta_D}(M_F | \mathbf{c})] \\ & + \mathbb{E}_{M_C \sim p_{train}(M_C)} [\log(1 - D_{\theta_D}(G_{\theta_G}(M_C | \mathbf{c})))]), \end{aligned} \quad (6)$$

where \mathbf{c} corresponds to a conditional vector, such as the one-hot vector for a specific class label ("a rainy Saturday in May at 2 pm," for example).

Specifically, we first divide the available external factors into two feature sets based on the continuity of their values: categorical features and continuous features. We directly concatenated the continuous features (wind speed, temperature, for example) into one vector, \mathbf{f}_{con} . We followed the guideline of [10], which transforms the available categorical features (weather, time of the day, and day of the week, for example) into a low-dimensional vector \mathbf{f}_{cat} by applying different embedding methods to them separately. Then we concatenate the two embeddings as the ultimate latent feature representation of external factors.

$$\mathbf{f}_{ext} = [\mathbf{f}_{con} \oplus \mathbf{f}_{cat}], \quad (7)$$

where \oplus denotes concatenation operation.

After getting the concatenated vector \mathbf{f}_{ext} , we feed it into the Linear Layer and employ the *ReLU* activation function to introduce nonlinear correlation. We employ the Dropout operation to randomly inactive 20% neural connections to avoid over-fitting. Then we employ two linear layers, which enable mapping the low-dimensional external factor into the high-dimensional space to explore the implicit features.

4.3 Flow Self-Attention Module

Most UFSR models [11, 21, 22] are built with convolutional layers, which inevitably lead to drawbacks: 1) Convolution normally focuses on the local neighbors of feature

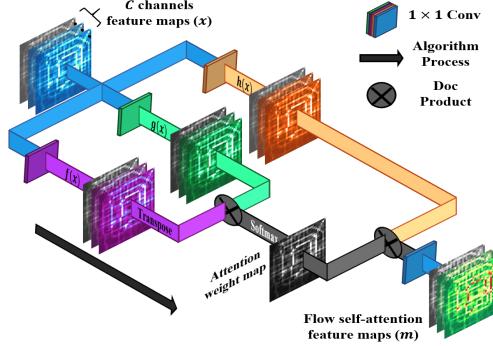


Fig. 5. The workflow of the Flow Self-Attention Module.

maps, which is computationally inefficient to model long-range dependencies. It is unreliable if we only consider the flow effects of adjacent F^2 (filter size F) urban when dealing with UFH because even if two urban grids are far apart, there can be a strong correlation between their traffic flows. For example, suppose there are two railway stations far apart in a city. If one of them breaks down, the urban grid nearby the other will experience a dramatic increase in traffic. 2) Due to the limits of observation, UFH typically contains a considerable amount of static data, which aggravates the data sparsity and hinders exploring the internal data dependency; this seriously disrupts the performance of the UFSR task.

Figure 5 shows the structure of the proposed flow self-attention module. First, to calculate the attention weights, we transform the feature maps obtained from the previous hidden layer $x \in \mathbb{R}^{C \times N}$ (C channels, N feature locations from the previous hidden layer) into two feature spaces, g and f by 1×1 convolution, which satisfies the following equation:

$$g(x) = W_g x, \quad f(x) = W_f x, \quad (8)$$

$$\beta_{ij} = \frac{\exp(t_{ij})}{\sum_{i=1}^N \exp(t_{ij})}, \quad t_{ij} = g(x_i)^T f(x_j), \quad (9)$$

where β_{ij} indicates how much attention the model pays to the i th grid when synthesizing the j th grid. After a 1×1 convolution with the element product of the feature map under another feature space h , we obtain the output of the attention layer: $o_j = (o_1, o_2, \dots, o_j, \dots, o_N) \in \mathbb{R}^{C \times N}$.

$$o_j = v \left(\sum_{i=1}^N \beta_{ij} h(x_i) \right), \quad h(x_i) = W_h x_i, \quad v(x_i) = W_v x_i, \quad (10)$$

In the equation 8 to 10, respectively, $W_f \in \mathbb{R}^{C \times C}$, $W_g \in \mathbb{R}^{C \times C}$, $W_v \in \mathbb{R}^{C \times \bar{C}}$ and $W_h \in \mathbb{R}^{\bar{C} \times C}$ are learned trainable weight matrices parameters which are implemented with 1×1 convolution layers. In order to minimize the computational complexity, we attempt multiple sets of channel number \bar{C} , which is reduced by C/k , where $k \in \{1, 2, 4, 8\}$. Since no significant performance decrease occurs, we uniformly set $k = 8$ in subsequent experiments.

After deriving the output of the attention layer, we do the product with a scaling parameter and sum it with the original input feature map \mathbf{x} . Therefore, the final output is obtained from the following equation:

$$\mathbf{y}_i = \gamma \mathbf{o}_i + \mathbf{x}_i, \quad (11)$$

where γ is a trainable scalar with an initial value of 0. This way allows the model to be more dependent on the relevance of local neighbors in the early training period and then gradually assign more weights to non-local neighbors' relevance as the training epoch increases.

5 Experiment

We conduct extensive experiments on two real-world datasets to validate generalization and inference abilities under the sparse urban flow context. Then, we perform a series of ablation experiments to verify the importance of each module for the UFSR task and the inter-constraint between the modules.

5.1 Dataset

We employ two real-world datasets in our experiments to better examine our model's capabilities, namely TaxiBj and CCMobile. Table 1 details both datasets, containing two sub-datasets: urban flows and external factors.

Table 1. Detailed composition of the datasets

Dataset	TaxiBJ	CCMobile
Time Span	P1:2013.7.1-2013.10.31	
	P2:2014.2.1-2014.6.30	2017.7.5-2017.10.15
	P3:2015.3.1-2015.6.30	
	P4:2015.11.1-2016.3.31	
Time Interval	30 minutes	15 minutes
Coarse-Grained Size	32×32	32×32
Fine-Grained Size	128×128	64×64
Up-scaling Factor	4	2
External factors (time, event and meteorology)		
# Weathers	16 types	10 types
# Holidays	41	13
Temperature / °C	$[-24.6, 41.0]$	$[10.4, 34.1]$
Wind Speed / mph	$[0, 48.6]$	/

1. **CCMobile:** We collect and aggregate check-in records from CCMobile terminal check-in data in each first week of six months. Figure 2(a) shows the experimental area, divided into 32×32 urban grids with red points indicating signal stations. Then we statistic the signal station capacity in each grid and obtain coarse/fine-grained resolution heatmaps within time slices of 15 minutes.

2. **TaxiBJ**: A taxi GPS dataset in Beijing, including driving trajectory and external factors. While the difference between taxiBI and CCMobile lies in: 1) TaxiBJ possesses complete external factor information. 2) TaxiBJ’s data coverage is extensive, which divides the 12-month time into four different period sub-datasets. 3) TaxiBJ has a more significant up-scaling factor $N = 4$. In addition, we split both datasets without overlapping, following the ratio of 7:2:1 (training, validation, and test).

5.2 Experimental Settings

Evaluation Metrics Following the traditional guideline, we employ three standard urban traffic data metrics (RMSE, MAE, and MAPE) to evaluate performance in different aspects, which are separately defined as:

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{n} \sum_{i=1}^n \left\| \mathbf{X}_f^i - \hat{\mathbf{X}}_f^i \right\|_F^2}, \\ MAE &= \frac{1}{n} \sum_{i=1}^n \left\| \mathbf{X}_f^i - \hat{\mathbf{X}}_f^i \right\|_F, \\ MAPE &= \frac{1}{n} \sum_{i=1}^n \left\| (\mathbf{X}_f^i - \hat{\mathbf{X}}_f^i) \odot \mathbf{X}_f^i \right\|_F, \end{aligned} \quad (12)$$

where n denotes the samples’ total number, \mathbf{X}_f^i and $\hat{\mathbf{X}}_f^i$ denote the ground truth and the corresponding i th inferred value separately. RMSE and MAE are commonly used to measure the absolute magnitude of ground truth from the predicted value, while MAPE measures the relative magnitude (i.e., percentage). MAE and MAPE are less susceptible to extreme values and focus more on smooth outcomes, while RMSE employs square calculations to amplify prediction errors and is more sensitive to outlier data; thus, it favors spiky distribution.

Baselines We compare UrbanSG with the following 6 baselines, which can be categorized into: (1) heuristic, (2)image super-resolution, (3)urban super-resolution as follows:

- **Mean Partition (Mean)**: We distribute the traffic volume evenly within an N^2 super-resolution, where N denotes the up-scaling factor.
- **Historical Average (HA)**: HA calculates the weights of each urban grid in the super-resolution region from the training data and assigns them fixedly.
- **SRCCNN** [3]: SRCCNN makes the first attempt to incorporate Convolutional Neural Network into ISR by scaling a coarse-grained image into high-resolution space.
- **SRResNet** [9]: SRResNet introduces the residual block, enabling the model to stack more layers and providing the theoretical basis for subsequent ISR works.
- **UrbanFM** [11]: UrbanFM innovatively incorporates the external factor into urban super-resolution and deploys the distributional up-sampling module.
- **UrbanODE** [21]: UrbanODE employs neural-ODEs and a pyramid attention mechanism for learning spatial correlations, the SOTA in the UFSR task.

5.3 Result on TaxiBJ

Model Comparison In this subsection, we perform a comprehensive comparison test with 6 baselines and 3 variants. We report the result with $C - F$ (Coarse-Fine Grained) as $16 - 64$ as our default setting, $N = 4$.

Table 2. Performance comparison among different models on TaxiBJ.

Method	Upscale	P1				P2				P3				P4			
		RMSE	MAE	MAPE	Δ												
MEAN	4	20.918	12.019	4.469	-	20.466	13.91	5.364	-	27.442	16.029	5.6112	-	19.049	11.07	4.192	-
HA	4	4.741	2.251	0.336	-7.35%	5.381	2.551	0.334	-9.87%	5.594	2.674	0.328	-3.47%	4.125	2.203	0.313	-1.95%
SRCCNN	4	4.297	2.491	0.714	-128.12%	4.612	2.681	0.689	-126.64%	4.815	2.829	0.727	-129.34%	3.838	2.289	0.667	-117.23%
SRResNet	4	4.164	2.457	0.713	-127.8%	4.524	2.66	0.688	-126.32%	4.69	2.775	0.717	-126.18%	3.667	2.189	0.637	-107.49%
UrbanFM	4	3.991	2.036	0.331	-5.75%	4.374	2.256	0.322	-5.92%	4.539	2.348	0.323	-1.89%	3.526	1.831	0.31	-0.98%
UrbanODE	4	3.86	1.963	0.313	0	4.391	2.213	0.304	0	4.479	2.287	0.317	0	3.439	1.81	0.307	0
UrbanSG	4	3.782	1.891	0.287	+8.31%	4.129	2.114	0.279	+8.22%	4.285	2.173	0.293	+7.57%	3.107	1.624	0.288	+6.19%
UrbanSG-attn	4	3.784	1.891	0.291	+7.03%	4.136	2.12	0.286	+5.92%	4.311	2.189	0.295	+6.94%	3.192	1.688	0.29	+5.54
UrbanSG-attn-ext	4	3.841	1.959	0.302	+3.51%	4.307	2.193	0.297	+2.3%	4.391	2.247	0.308	+2.84%	3.355	1.729	0.301	+1.95%
UrbanSG-attn-disc	4	4.016	2.144	0.343	-9.58%	4.397	2.268	0.33	-8.55%	4.604	2.363	0.339	-6.94%	3.606	1.839	0.317	-3.26%

Table 2 summarizes the results of the comparison experiments on TaxiBJ, where -attn, -ext, and -disc denote the deletion of the self-attention module, the external factor integration module, and the generator from UrbanSG, respectively. According to Table 2, We have the following observations: (1) UrbanSG and most of its variants outperform all baselines over all time periods in all validation metrics. Moreover, the advance of UrbanSG-attn-ext over all baselines suggests that employing generative adversarial networks as the backbone plays a leading role in improving the inference performance. (2) The performance of UrbanSG-attn shows no significant decline on TaxiBJ. We argue that it is attributed to the high data coverability of TaxiBJ limits the function of the self-attention module. Figure 6 illustrates a set of experiments for up-scaling factor $N = 2, 4, 6, 8$. We can above that: (1) UrbanSG outperforms all other baselines for all N s. (2) UrbanSG performs superior in the high-resolution UFSR task.

Study on External Factor Integration It is insufficient to extrapolate from historical traffic volume in the complex urban computing context. Thus, we experiment on the effectiveness of external factors, where we randomly subsample the external factor from the original training set in four ratios: 100%, 50%, 30%, and 10%.

To verify the effectiveness of the external factor integration module, we compare UrbanSG with UrbanSG-ext and UrbanSG-up, as shown in Figure 7. When we reduce the scale of external factors in training data, the distance between UrbanSG and UrbansG-ext becomes more remarkable, which indicates that external factor integration plays the role of a meta learner and provides prior knowledge to the model training. It is the same as UrbanSG and UrbanSG-up. Our model may recover some external impacts when training data (external factor) is enough.

5.4 Result on CCMobile

Model Comparison Table 3 illustrates the experiment results on CCMobile, which leads to the following observations: (1) Data sparsity reduces the model performance on CCMobile. However, UrbanSG still outperforms all the baselines in all three metrics. (2) A more significant up-scaling factor decreases the prediction performance. Take UrbanODE as an example; when the up-scaling factor N increases from 2 to 4, its MAE increases from 3.578 to 4.332. (3) Our model outperforms all other baselines in adapting to sparse data. For instance, when increasing N from 2 to 4, UrbanSG’s performance improvement on MAE improves from 13.84% to 15.02%. By comparing the performance of UrbanSG and UrbanSG-attn, the adaption ability mainly comes from the flow self-attention module.

Table 3. Performance Comparison among different models on CCMobile.

Method	#Params/M	N = 2						N = 4					
		MAE	Δ	RMSE	Δ	MAPE	Δ	MAE	Δ	RMSE	Δ	MAPE	Δ
HA	X	10.622	-196.87%	16.408	-87.05%	0.927	-106%	11.032	-154.66%	23.015	-119.04%	0.918	-98.27%
SRCCNN	7.4	4.502	-25.82%	10.006	-14.07%	0.51	-13.4%	5.232	-20.77%	12.148	-15.62%	0.544	-17.58%
SRResNet	5.5	4.403	-23.06%	9.667	-10.2%	0.508	-12.89%	5.123	-18.26%	11.81	-12.4%	0.536	-15.72%
UrbanFM	6.2	4.027	-12.55%	9.196	-4.83%	0.488	-8.44%	4.749	-9.62%	10.925	-3.98%	0.505	-9.05%
UrbanODE	2.1	3.578	0	8.772	0	0.45	0	4.332	0	10.507	0	0.463	0
UrbanSG	2.8	3.083	+13.84%	8.017	+8.61%	0.395	+12.19%	3.681	+15.02%	9.395	+10.58%	0.399	+13.87%
UrbanSG-attn	2.4	3.417	+4.51%	8.603	+1.93%	0.377	+2.8%	4.111	+5.11%	10.483	+2.3%	0.441	+4.84%

Study on Parameter Size Table 3 illustrates the parameter settings for all baselines (heuristics methods without statistics of parameters). In addition, we test multiple sets of hyper-parameters (residual block, B and base channel, C) and argue that larger B and C could effectively improve model performance and bring a training time delay and increase memory space. Therefore, considering the trade-off between model performance and training cost, we set the default B-C as 16-256.

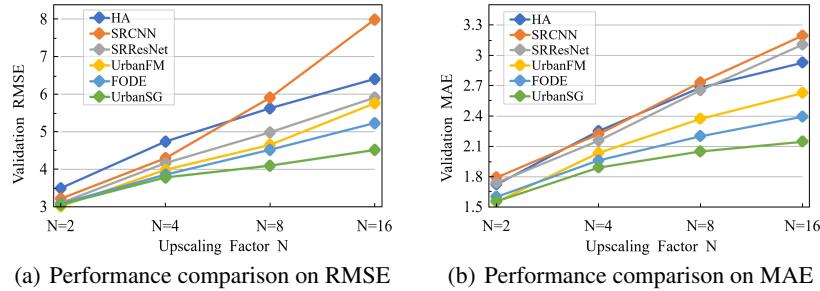
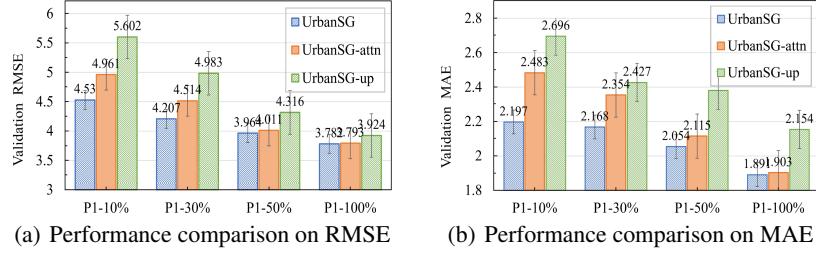
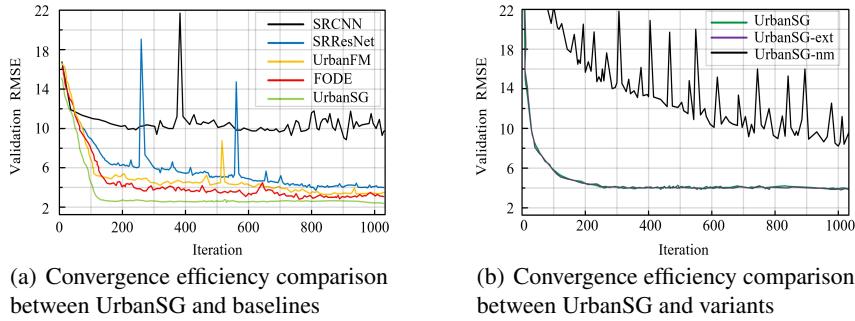


Fig. 6. Performance comparison over various up-scaling factors.

**Fig. 7.** Performance comparison over four-level external factor training sets.**Fig. 8.** Convergence efficiency comparison in 1000 iterations.

Study on Efficiency We conduct a set of comparison experiments on P1, as shown in Figure 8, which shows that UrbanSG converges faster and smoother than its variants and all baselines. UrbanSG reaches convergence at 300 iterations and can maintain a relatively stable convergence trend. Additionally, the similar performance of UrbanSG and UrbanSG-ext indicates that the external factor integration module only slightly affects the model’s convergence but significantly improves performance.

Study on Flow Self-Attention Module We sum up the attention values of all instances during training and normalize them for visualization. Figure 9 shows the heatmap of attention weights and the corresponding physical urban flow heatmap. We observe that: (1) As we can see from regions 1 and 2, the flow self-attention module enables the model to focus on active urban grids and weights the inactive urban grid as 0, hence alleviating or ignoring the interference that sparsity brings to UFSR task. (2) Learning from region 3, the model avoids assigning too much attention weight to urban grids with dense traffic because if each urban grid within a super-resolution region has a large and similar traffic volume, they will share all the attention weights equally, resulting in each being small, which is consistent with common sense.

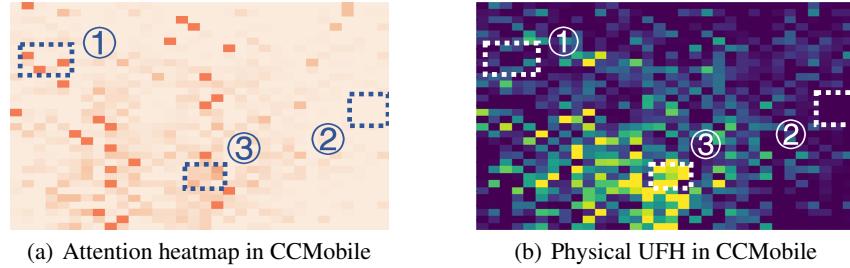


Fig. 9. Heatmap comparison between attention weights and urban flow.

6 Conclusion

This work formalizes the UFSR problem and presents a CGAN-based framework (UrbanSG) to solve it. UrbanSG has addressed the two challenges specific to this problem, i.e., flow distribution sparsity and the complexity of external factors, by leveraging the flow self-attention module and external factor fusion sub-net. Experiment shows that our approach advances baselines by at least 4.5%, 17.0% and 54.1% on TaxiBJ and 3.5%, 7.8% and 22% on CCMobile in terms of three metrics. Various empirical studies and visualizations have confirmed the advantages of UrbanSG in both efficiency and effectiveness.

7 Acknowledgement

This work was supported by the General Project of National Natural Science Foundation of China under Grant 62072209, the National Natural Science Foundation of authority Youth Fund under Grant 62002123, the Key project of Science and technology development Plan of Jilin Province Grant 20210201082GX, the Jilin Provincial Development and Reform Commission Project Grant 2020C017-2, the Science and technology project of Education Department of Jilin Province under Grand JJKH20221010KJ, and the CCF-Baidu Open Fund under Grant 2021PP15002000.

References

1. Cai, J., Meng, Z., Ho, C.M.: Residual channel attention generative adversarial network for image super-resolution and noise reduction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2020)
2. Chen, W., Long, G., Yao, L., Sheng, Q.Z.: Amrnn: attended multi-task recurrent neural networks for dynamic illness severity prediction. World Wide Web p. 2753–2770 (2020)
3. Chen, W., Wang, S., Long, G., Yao, L., Sheng, Q.Z., Li, X.: Dynamic illness severity prediction via multi-task rnns for intensive care unit. In: 2018 IEEE International Conference on Data Mining (ICDM). pp. 917–922. IEEE (2018)
4. Chen, W., Yue, L., Li, B., Wang, C., Sheng, Q.Z.: Damtrnn: A delta attention-based multi-task rnn for intention recognition. In: International Conference on Advanced Data Mining and Applications. pp. 373–388. Springer (2019)

5. Dun, Y., Da, Z., Yang, S., Qian, X.: Image super-resolution based on residually dense distilled attention network. *Neurocomputing* **443**, 47–57 (2021)
6. Gong, Y., Li, Z., Zhang, J., Liu, W., Zheng, Y.: Online spatio-temporal crowd flow distribution prediction for complex metro system. *IEEE Transactions on Knowledge and Data Engineering* **34**, 865–880 (2020)
7. Gu, J., Zhou, Q., Yang, J., Liu, Y., Zhuang, F., Zhao, Y., Xiong, H.: Exploiting interpretable patterns for flow prediction in dockless bike sharing systems. *IEEE Transactions on Knowledge and Data Engineering* **34**(2), 640–652 (2020)
8. Jo, Y., Kim, S.J.: Practical single-image super-resolution using look-up table. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 691–700 (2021)
9. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 105–114 (2017)
10. Liang, Y., Ke, S., Zhang, J., Yi, X., Zheng, Y.: Geoman: Multi-level attention networks for geo-sensory time series prediction. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. pp. 3428–3434 (2018)
11. Liang, Y., Ouyang, K., Jing, L., Ruan, S., Liu, Y., Zheng, Y., et al.: Urbanfm: Inferring fine-grained urban flows. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery. p. 3132–3142 (2019)
12. Liu, J., Zhang, W., Tang, Y., Tang, J., Wu, G.: Residual feature aggregation network for image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2359–2368 (June 2020)
13. Mirza, M., Osindero, S.: Conditional generative adversarial nets. *Computer Science* pp. 2672–2680 (2014)
14. Noor, D.F., Li, Y., Li, Z., Bhattacharyya, S., York, G.: Gradient image super-resolution for low-resolution image recognition. In: ICASSP 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2332–2336 (2019)
15. Pan, Z., Zhang, W., Liang, Y., Zhang, W., Yu, Y., Zhang, J., Zheng, Y.: Spatio-temporal meta learning for urban traffic prediction. *IEEE Transactions on Knowledge and Data Engineering* **34**(3), 1462–1476 (2022). <https://doi.org/10.1109/TKDE.2020.2995855>
16. Wang, R., Lei, T., Zhou, W., Wang, Q., Meng, H., Nandi, A.K.: Lightweight non-local network for image super-resolution. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1625–1629 (2021)
17. Yan, Y., Ren, W., Hu, X., Li, K., Shen, H., Cao, X.: Srgat: Single image super-resolution with graph attention network. *IEEE Transactions on Image Processing* **30**, 4905–4918 (2021)
18. Yang, F., Yang, H., Fu, J., Lu, H., Guo, B.: Learning texture transformer network for image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5791–5800 (2020)
19. Yue, L., Tian, D., Chen, W., Han, X., Yin, M.: Deep learning for heterogeneous medical data analysis. *World Wide Web* **23**(5), 2715–2737 (2020)
20. Zhang, X., Huang, C., Xu, Y., Xia, L., Dai, P., Zheng, Y., et al.: Traffic flow forecasting with spatial-temporal graph diffusion network. In: In Proceedings of the AAAI Conference. vol. 35, pp. 15008–15015 (2021)
21. Zhou, F., Jing, X., Li, L., Zhong, T.: Inferring high-resolutinal urban flow with internet of mobile things. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7948–7952 (2021)
22. Zhou, F., Li, L., Zhong, T., Trajcevski, G., Zhang, K., Wang, J.: Enhancing urban flow maps via neural odes. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. pp. 1295–1302 (2020)