Fine-Grained Data Inference via Incomplete Multi-Granularity Data

Anonymous Author(s)*

Abstract

Urban fine-grained data map inference, leveraging information from coarse-grained maps, has emerged as a significant area of research due to the growing complexity and data heterogeneity in urban environments. Existing methods have a priori assumption that a coarse-grained data map, one fixed-size granularity, transforms into a fine-grained data map, also one fixed-size granularity. However, in the actual scenarios, the collected coarse-grained data maps are often incomplete and have significantly distinct granularities in various urban areas, which results in incomplete heterogeneous data, i.e., multi-granularity data maps in terms of spatial information. Meanwhile, different granularity data maps are needed for various urban downstream tasks, which is a multi-task problem. To that end, this paper proposes a novel framework, a multi-granularity super-resolution data map inference framework (MGSR), designed to harness spatio-temporal information to transform incomplete coarse-grained multi-granularity data maps into fine-grained multi-granularity data maps. Specifically, we design a granularity alignment network to align multi-granularity information and address missing data on each granularity map by leveraging the other granularity maps with a well-designed self-supervised task. Then, we introduce a feature extraction network to capture spatio-temporal dependencies and extract features. Finally, we devise a recurrent super-resolution network with shared parameters to infer multi-granularity data maps. We conduct extensive experiments on three real-world benchmark datasets and demonstrate that MGSR significantly outperforms the state-of-the-art methods for multi-granularity urban data map inference, and reduces RMSE and MAE by up to 40.1% and 50.3%, respectively. The source code has been released at https://anonymous.4open.science/r/MGSR-7E5C.

CCS Concepts

• **Information systems** → **Spatial-temporal systems**.

Keywords

Multi-Granularity Data, Super-Resolution, Fine-Grained Inference, Spatio-Temporal Data

ACM Reference Format:

Anonymous Author(s). 2018. Fine-Grained Data Inference via Incomplete Multi-Granularity Data. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

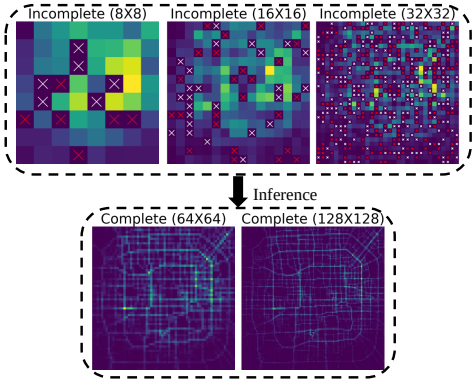

Figure 1: The brightness of colors in data maps indicate the flow value in BJTaxi P1. The above three data maps are available data maps, where the red and white "×" denote missing data due to sensor distribution and random, respectively. The bottom two data maps are the inferred data maps.

1 Introduction

The fine-grained data map inference is crucial for smart city applications like intelligent transportation management, urban planning, etc. To sense urban information, numerous sensors with different models and functions have been employed city-wide for various purposes by organizations and agencies, resulting in heterogeneous and spatially disorganized sensor networks. Therefore, it is not possible to obtain fine-grained and standard urban data maps for the whole city, especially in terms of constructing sensor grids.

The raw data available is often **multi-granularity** and **incomplete** due to the following reasons: 1) *Sensor Heterogeneity*. Different sensors collect data at varying spatial granularities. This diversity arises from the use of various sensor models and technologies deployed across the city for distinct purposes, e.g., traffic monitoring, environmental sensing, etc. The resulting heterogeneous data collection creates a complex dataset that poses challenges for inferring fine-grained data. 2) *Spatial Disorganization*. Sensors are deployed without a unified plan, leading to uneven coverage where some urban areas are over-monitored while others lack sufficient data. Consequently, integrating data from these disorganized sensors is challenging, as the spatial distribution of the collected data is irregular and further complicating fine-grained data inference. 3) *External Factors*. Data collection can be disrupted due to sensor malfunctions or system breakdowns, leading to gaps in the data.

For the above reasons, we can usually obtain multi-granularity incomplete data. As shown in Fig. 1, we take incomplete multi-granularity data maps as input to infer multi-granularity fine-grained data maps. This process is crucial for enhancing the quality and usability of data for downstream applications, ensuring comprehensive coverage and accuracy.

Numerous approaches have been developed to transform coarse-grained data maps into fine-grained data maps, addressing the challenges of spatial granularity and data completeness. Traditional methods, such as statistical techniques [26] and tensor decomposition [4], have been extensively applied to fine-grained data. However, these methods often fall short in handling large-scale urban data due to their inherent limitations in capturing complex spatial correlations. With advancements in computer vision, super-resolution techniques have been applied to fine-grained data map inference. These methods utilize stacked convolutional layers and super-resolution layers to effectively capture spatio-temporal dependencies and infer fine-grained data maps. UrbanFM [18], the first to introduce super-resolution networks into this task, addresses two primary challenges: (1) the spatial correlations between coarse- and fine-grained urban data maps and (2) the complexities of external impacts. UrbanPy [23] improves upon this by designing a cascading model for progressive inference and decomposing the original task into multiple subtasks to enhance accuracy. MT-CSR [17] specifically targets fine-grained data inference from incomplete coarse-grained data. UrbanSTC [24] uses a contrastive self-supervision method to account for all correlated spatial and temporal patterns, training massive learnable parameters effectively.

While existing methods address various factors for inferring fine-grained data from coarse-grained data, they assume uniform granularity, which is less realistic compared to the complexities of multi-granularity data scenarios. In real-world scenarios, the challenge of inferring various granularities from incomplete multi-granularity data is more pertinent yet complex compared to traditional single-granularity inference problems. Transforming incomplete multi-granularity data into fine-grained data maps involves several challenges: 1) Spatia-temporal Dependencies. The relationships between data maps at input and output various granularities are diverse and complex. 2) Incomplete Multi-granularity Data. The collected multi-granularity data often contains gaps due to various missing patterns and presents distinct views that must be aligned and integrated for accurate inference. 3) Multi-task Inference. Inferring data across different granularities inherently constitutes a multi-task problem. Each granularity represents a distinct task, and effectively balancing these tasks is critical.

To bridge the gap between real-world scenarios and existing methods, we propose MGSR, a multi-granularity super-resolution data map inference framework designed to align incomplete multi-granularity data and infer fine-grained multi-granularity data maps. Specifically, we first introduce the Granularity Alignment Network to align incomplete multi-granularity data maps and impute missing data at each granularity based on the others using a well-designed self-supervised task. Next, we present the Feature Extraction Network, capturing spatio-temporal dependencies across multiple granularities and extracting feature maps. Finally, a Recurrent Super-Resolution Network employs shared parameters to infer fine-grained multi-granularity data maps, enhancing parametric efficiency and the framework's generalization capabilities. The contributions of this paper are summarized as follows:

- We design an innovative multi-granularity super-resolution data map inference framework that infers fine-grained multi-granularity data based on incomplete coarse-grained multi-granularity data in an end-to-end way. MGSR effectively harmonizes data of varying granularities, leveraging their interdependencies to enhance the accuracy and completeness of fine-grained data. It addresses the complexity of real-world data scenarios.
- We propose GLAN and FEN to align and impute multi-granularity information and to extract and integrate multi-granularity features using a well-designed self-supervised task, enhancing representation ability and generalization.
- We devise RSRN to infer fine-grained multi-granularity data, thereby improving scalability and parameter efficiency. The recursive structure of RSRN leverages outputs from previous steps to progressively infer more fine-grained data to handle complex multi-task scenarios.
- We evaluate MGSR by extensive experiments on six datasets under various settings. The results demonstrate that MGSR significantly outperforms state-of-the-art methods, reducing RMSE and MAE by up to 42.3% and 52.3%, respectively.

2 Notions and Problem Definition

In this section, we give the mathematical definitions and problem statements discussed in this paper for convenience.

Region. As shown in Fig. 1, a city is partitioned into $I^g \times J^g$ grid map based on longitude and latitude at a granularity g, where each grid element represents a region. The region set is denoted as R^g, where R^g_{ij} correspond to the i-th row and the j-th column grid map.

Data Map. Given a particular time and a granularity g, X^g denotes the data map formed by the data collected from the corresponding sensors. X^g_{ij} is a d-dimensional vector representing data observed in R^g_{ij}, e.g., temperature and air quality.

Multi-Granularity Data Map. A city can be partitioned into multiple data maps under distinct granularities (*Sensor Heterogeneity*). We define coarse- and fine-grained granularity sets are $\mathcal{G}_c = \{\nu | \nu = g_0, \ldots, g_{(N_c-1)}\}$ and $\mathcal{G}_f = \{\mu | \mu = g_0, \ldots, g_{(N_f-1)}\}$. For example, in Fig. 1, the coarse- and fine-grained granularity sets are $\{8, 16, 32\}$ and $\{64, 128\}$, respectively.

Structural Constraint. Structural Constraint typically refers to the rule that ensures consistency between coarse- and fine-grained data. Specifically, It can be defined as:

$$X^{\nu}_{i'j'} = \sum_{ij} X^{\mu}_{ij} \quad s.t. \lfloor \frac{i}{u^{\mu}_{\nu}} \rfloor = i', \ \lfloor \frac{j}{u^{\mu}_{\nu}} \rfloor = j', \tag{1}$$

where $\mu = u^{\mu}_{\nu}\nu$, where u^{μ}_{ν} is an upscaling factor, for example, $u^{\mu}_{\nu} = 2$ when $\nu = 16$ and $\mu = 32$.

Missing Pattern. The missing pattern is determined by two factors, as follows: (1) **Sensor Spatial Distribution**. It makes fixed-position regions miss data at all times due to the disorganization of sensor positions. (2) **Missing at Random** [3]. When data is missing at random, the probability of a data point being missing is related to the observed data but not to the missing data itself due to External Factors. In conclusion, we formulate the mask operation as follows:

$$\bar{X}^{\nu}_{ij} = \begin{cases} 0, & M^{\nu}_{ij} = 0 \\ X^{\nu}_{ij}, & M^{\nu}_{ij} = 1 \end{cases}, \tag{2}$$

where the $\bar{X}^{\nu}_{ij}$ is the collected available data as the input.

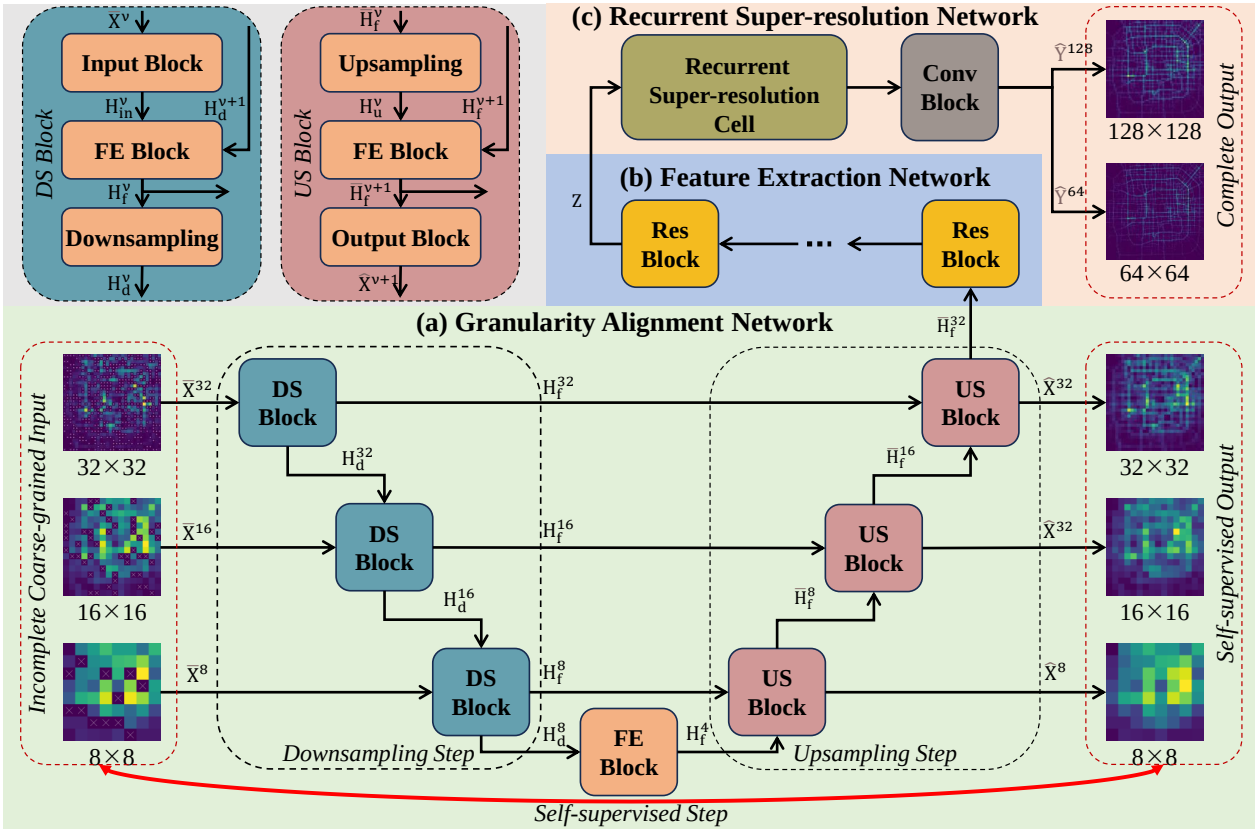

Figure 2: An overall architecture of MGSR.

Problem Statement. Given incomplete multi-granularity data maps $\{\bar{X}^{\nu}|\nu \in \mathcal{G}_c\}$ and inferred granularities $\mathcal{G}_f$, our objective is to infer the multi-granularity data maps $\{\hat{Y}^{\mu}|\mu \in \mathcal{G}_f\}$ with maximum accuracy. This process must adhere to the ground truth $\{X^{\mu}|\mu \in \mathcal{G}_f\}$ as possible subject to the structural constraints.

3 Methodology

This section details the proposed overall framework, as illustrated in Fig. 2, consisting of three main components: (a) Granularity Alignment Network (GLAN), (b) Feature Extraction Network (FEN), and (c) Recurrent Super-Resolution Network (RSRN). Additionally, we incorporate three loss functions according to the problem definition to optimize MGSR, ensuring its robustness and accuracy.

3.1 Granularity Alignment Network

Inspired by U-net [1, 25], GLAN aligns and integrates incomplete multi-granularity data maps by facilitating cross-granularity information passing, thus completing missing data in each map. As illustrated in Fig. 2(a), the GLAN mainly consists of three steps: downsampling, upsampling, and self-supervised step. While architectures like Swin Transformer [21] could have been chosen for cutting-edge performances, we opted for convolutional networks to maintain the simplicity and efficiency of the framework without compromising effectiveness.

3.1.1 Downsampling step. The downsampling step is a process in which high-granularity feature maps transform into low-granularity feature maps, which aims to align high- to low-granularity feature maps. It stacks downsampling (DS) blocks with each DS block reducing the feature map by 2×. As the feature map's granularity decreases, the amount of spatial information it can represent diminishes. To counteract this, we increase the number of channels, enhancing the feature map's capacity to represent detailed information. The DS block consists of three layers: input block, fusion and extraction information (FE) block, and downsampling.

Input Block. The input block transforms input data maps into multi-channel feature maps using stacked convolution layers, enhancing representation capacity. It is expressed as:

$$H_{in}^{\nu} = \psi_i(\bar{X}^{\nu}) \quad s.t. \quad \nu \in \mathcal{G}_c, \tag{3}$$

where $\bar{X}^{\nu} \in \mathbb{R}^{c\times\nu\times\nu}$ is the input data map and $H_{in}^{\nu} \in \mathbb{R}^{d\times\nu\times\nu}$ is the output of the input block.

Fusion and Extraction Information (FE) Block. The first layer of the FE block is a 1×1 convolution that aligns the feature maps and reduces their dimensionality. If high-granularity feature maps exist, the FE block concatenates a d-channel high-granularity feature map with a d-channel low-granularity feature map to form a 2d-channel input, which is then processed to output a d-channel feature map, completing the low-granularity information. The rest layers of FE block are stacked residual blocks that capture spatial

relationships between regions. It is formulated as:

$$H_f^{\nu} = \phi([H_{in}^{\nu},\ H_d^{\nu+1}]), \tag{4}$$

where $H_f^{\nu},\ H_d^{\nu+1} \in \mathbb{R}^{d\times\nu\times\nu}$ and [] is the concatenated operation. $H_d^{\nu+1}$ is the output of the previous downsampling.

Downsampling. The Downsampling is to down-sample the feature maps using a max-pooling layer, which is expressed as:

$$H_d^{\nu} = \varphi_d(H_f^{\nu}), \tag{5}$$

where $H_d^{\nu} \in \mathbb{R}^{d\times\nu/2\times\nu/2}$.

This process allows information to pass from high to low granularities, enabling low-granularity feature maps to utilize higher-granularity information to fill in missing data. Meanwhile, an inverse architecture is similarly designed for upsampling.

3.1.2 Upsampling step. The upsampling step reverses the downsampling process, passing the low-granularity information to the high-granularity feature maps. To that end, the upsampling step that stacks upsampling (US) blocks is employed to expand the low-granularity feature maps. The US block increases feature maps by 2×, consisting of upsampling, FE block, and output block.

Upsampling. The upsampling that transforms a low-granularity feature map into a high-granularity feature map is realized by a super-resolution convolution, which is expressed as:

$$H_u^{\nu} = \varphi_u(\bar{H}_f^{\nu}), \tag{6}$$

where $H_u^{\nu} \in \mathbb{R}^{d\times2\nu\times2\nu}$. The input of the first upsampling is the output of the individual FE block.

Meanwhile, we design skip connections to introduce corresponding granularity information from the downsampling step, allowing shallow layer features to be directly passed to deeper layers. This preserves original information, retaining more details in deep networks. To integrate low-granularity feature maps, the FE blocks are similar to the downsampling step, but align the low-granularity feature map to the high-granularity feature map in upsampling step, expressed as:

$$\bar{H}_f^{\nu} = \phi([H_u^{\nu-1},\ H_f^{\nu}]), \tag{7}$$

where $H_u^{\nu-1},\ H_f^{\nu},\ \bar{H}_f^{\nu} \in \mathbb{R}^{d\times\nu\times\nu}$. We assume the $\bar{H}_f^{\nu}$ is the completed granularity ν feature map under available data, which is because $H_u^{\nu-1},\ H_f^{\nu}$ contain $\bar{X}^{g_0},\ldots,\bar{X}^{\nu/2}$ and $\bar{X}^{\nu},\ldots,\bar{X}^{g_{(N_c-1)}}$, respectively. Therefore, we take $\bar{H}_f^{\nu}$ as the upsampling block and the self-supervised task input.

Output Block. It projects feature maps into a data maps for self-supervised tasks, typically consisting of one or more convolution layers. It enhances self-supervised task performance by learning better feature representations, which can be expressed as:

$$\hat{X}^{\nu} = \psi_o(\bar{H}_f^{\nu}), \tag{8}$$

where $\hat{X}^{\nu} \in \mathbb{R}^{c\times\nu\times\nu}$.

With the downsampling and upsampling steps, every granularity feature map obtains the other granularity information to complete its missing values. Compared to all granularity feature maps, the information in granularity g_{N_c-1} is the most abundant and highest, which integrates all granularity information and retains details. Therefore, the final output $\bar{H}_f^{g_{(N_c-1)}}$ serves as the input to infer fine-grained data maps.

3.1.3 Self-supervised step. Self-supervised learning is a form of unsupervised learning. Integrating self-supervised learning into GLAN is beneficial for the following reasons: 1) **Improved Representation Learning**. Self-supervised task makes the representations capture underlying structures and patterns, which guides the component to learn corresponding granularity representations. 2) **Regularization and Robustness**. Self-supervised tasks can act as a form of regularization, which leads to more robust models that perform better on downstream tasks. 3) **Improved Data Efficiency**. This efficient use of data resources leads to better performance and faster convergence during training.

We design the self-supervised task based on the coarse-grained data map inference task. Meanwhile, the Mean Squared Error (MSE) is applied to measure the discrepancy between predicted values $\hat{X}^{\nu}$ and ground truths $\bar{X}^{\nu}$, where $M_{ij}^{\nu} = 1, \nu \in \mathcal{G}_c$. The self-supervised loss is defined as:

$$\mathcal{L}_{self} = \sum\nolimits_{\nu}^{\mathcal{G}_c} \frac{\sum_{i=0}^{\nu}\sum_{j=0}^{\nu} M_{ij}^{\nu}\|\bar{X}_{ij}^{\nu} - \hat{X}_{ij}^{\nu}\|_F^2}{\sum_{i=0}^{\nu}\sum_{j=0}^{\nu} M_{ij}^{\nu}}, \tag{9}$$

where $\|\cdot\|_F$ is the Frobenius norm.

3.2 Feature Extraction Network

The FEN is a critical component within our framework for super-resolution inference. Its primary role is to capture the spatio-temporal features required to infer fine-grained data maps from coarse-grained inputs. The introduction of FEN is driven by several key reasons inherent in working with incomplete coarse-grained data maps: (1) **Preservation of Fine Details**. Coarse-grained data maps often lack the detailed information necessary for fine-grained data maps. FEN is designed to extract these fine details from the input data, enabling the reconstruction of fine-grained maps. (2) **Complex Spatial Relationships**. The relationships within the data across granularities can be complex and nuanced. FEN helps to capture and model these intricate interactions, providing a robust feature set for further processing.

The FEN, implemented by stacking residual blocks, effectively captures deep spatio-temporal features while mitigating the commonly encountered vanishing gradient problem, expressed as

$$Z = FEN(\bar{H}_f^{g_{(N_c-1)}}), \tag{10}$$

where $Z \in \mathbb{R}^{d\times g_{(N_c-1)}\times g_{(N_c-1)}}$.

3.3 Recurrent Super-resolution Network

The RSRN aims to infer fine-grained multi-granularity data maps based on the above components. Inspired by Recurrent Neural Networks (RNNs) [22], RSRN is designed with a recurrent structure to address this multi-task problem.

RNN-like structures excel at capturing sequential dependencies, crucial for progressively refining data maps from coarse to fine. This fashion ensures that each fine-grained map benefits from the contextual information provided by previous inferences. Specifically, by incorporating hidden states, outputs and labels from previous iterations, RNN-like structures ensure that the network integrates

robust features for subsequent inferences. This continuous integration helps maintain consistency and accuracy across different levels of granularity.

The RSRN operates iteratively inferring fine-grained data maps at increasingly higher granularity levels. At each iteration, it utilizes the output and hidden state from the previous step to guide the inference of higher-granularity data map, described as follows:

$$\mathbf{z} = \sigma_s(Conv_z([\mathbf{s}^{\mu}, \hat{Y}^{\mu}])), \quad (11)$$

$$\mathbf{r} = \sigma_s(Conv_r([\mathbf{s}^{\mu}, \hat{Y}^{\mu}])), \quad (12)$$

$$\mathbf{h} = \sigma_t(Conv_h([(\mathbf{r} \odot \mathbf{s}^{\mu}), \hat{Y}^{\mu}])), \quad (13)$$

$$\mathbf{s}^{2\mu} = (1 - \mathbf{z}) \odot \mathbf{s}^{\mu} + \mathbf{z} \odot \mathbf{h}, \quad (14)$$

$$\hat{Y}^{2\mu} = SConv(\mathbf{s}^{2\mu}), \quad (15)$$

where $Conv$ and $SConv$ represent convolution and super-resolution convolution operations, respectively. σ_s and σ_t denote the sigmoid and tanh activation functions, respectively. $\odot$ indicates the element-wise multiplication. The $\mathbf{s}$ is assigned Z in the first step.

This iterative process enables the RSRN to effectively utilize the hierarchical dependencies within the data, ensuring accurate and detailed reconstruction of fine-grained data maps.

3.4 Training

This subsection details the training procedure of our framework, which leverages a combination of self-supervised loss, task loss, and structural constraint loss. The self-supervised loss focuses on parameter optimization of the GLAN, while the task and structural constraint losses contribute to the overall training of the framework.

The self-supervised loss function is described in the above discussion. In addition to the self-supervised loss, our framework is trained using the task loss function and the structural constraint loss function. The overall loss function is expressed as:

$$\mathcal{L} = \mathcal{L}_{task} + \alpha\mathcal{L}_{con} + \beta\mathcal{L}_{self}, \quad (16)$$

where α and β are hyper-parameters used to balance the contributions of the loss functions. $\mathcal{L}_{task}$ and $\mathcal{L}_{con}$ denote the task and structural constraint, respectively.

$\mathcal{L}_{task}$ ensures that the predicted fine-grained data maps closely match the ground truths. Mathematically, the task loss function can be defined as:

$$\mathcal{L}_{task} = \sum\nolimits_{\mu \in \mathcal{G}_f} \|X^{\mu} - \hat{Y}^{\mu}\|_F^2. \quad (17)$$

$\mathcal{L}_{con}$ is incorporated to maintain structural consistency across different granularity levels. This loss ensures that the inferred fine-grained data maps adhere to the known structural relationships inherent in the coarse-grained data. To avoid repetitive computation, we design a chained structural constraint loss function. It can be expressed as:

$$\mathcal{L}_{con} = \sum\nolimits_{\nu}^{\mathcal{G}_c} \delta(\bar{X}^{\nu}, \hat{Y}^{g_f^0}) + \sum\nolimits_{\mu}^{\mathcal{G}_f} \delta(X^{\mu}, \hat{Y}^{\mu+1}), \quad (18)$$

$$\delta(\mathbf{X}^m, \mathbf{Y}^n) = \sum\nolimits_{i'j'} (\mathbf{X}^m_{i'j'} - \sum\nolimits_{ij} \mathbf{Y}^n_{ij}),\ s.t. \lfloor \frac{i}{u^n_m} \rfloor = i', \lfloor \frac{j}{u^n_m} \rfloor = j', \quad (19)$$

where the first term formulates the structural constraints between the inputted and inferred data maps and the other term formulates the structural constraints between the inferred data maps.

Table 1: Dataset Description.

Dataset	BJTaxi P1	NYCTaxi	MHK
Time span	7/1-10/31, 2013	1/1-1/7, 2015	1/21-1/27, 2023
Time interval	30 minutes	1 hour	30 minutes
Input	8, 16, 32	8, 16	8, 16
Output	64, 128	32, 64	32

4 Experiments

In this section, we conduct extensive experiments to benchmark the effectiveness and generalization ability of our MGSR across real-world datasets. Our experiments are designed to answer the following research questions: **RQ1.** Can our MGSR provide superior performance compared to several state-of-the-art baselines? **RQ2.** Can the GLAN effectively align and integrate information in multi-granularity data maps? **RQ3.** Can the RSRN address the multi-task problem and have better generalization in multi-granularity data map inference? **RQ4.** What is the impact of various components in the MGSR on different datasets? **RQ5.** How well does MGSR perform for missing values generated by different missing patterns?

4.1 Datasets

In this subsection, we conduct experiments on three real-world datasets. The details of the datasets are introduced as follows:

BJTaxi [33]. It consists of trajectory data from taxicab GPS data in Beijing from four different periods: P1 to P4, where the values denote the number of taxis in each grid. We select P1 to evaluate our method (the other in Appendix B).

NYCTaxi [1]. The dataset provides detailed trip records of yellow taxis within New York City. We construct the dataset using inflow and outflow data of grids as data maps.

MHK. This dataset contains 108.1 GB of cellular signaling data collected in Meihekou, Jilin Province, China. A record denotes that a user was present in a base station's coverage area at a specific time. The cover areas of different base stations are distinct, which results in multi-granularity data maps. We consider the number of users in each grid as a value for data maps.

The specifications of the datasets are summarized in Table 1.

4.2 Experimental Settings

4.2.1 Evalutaion Metrics. In line with existing works on fine-grained data map inference, we employ two widely used metrics: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

4.2.2 Baselines. We compare with two categories of methods, as detailed below: (1) **Heuristic Methods**: **Mean Partition** (Mean) [18] evenly distributes each coarse-grained value across the corresponding fine-grained positions under structural constraints. **Historical Average** (HA) [18] allocates the coarse-grained flow to fine-grained flow by historical proportions. (2) **Fine-grained Inference Methods**: **UrbanFM** [18] formalizes the fine-grained inference problem and introduces super-resolution into the field. **UrbanPy** [23] identifies the limitation of this preliminary work for large-scale

[1] https://www.nyc.gov/site/tlc/index.page

Table 2: Aligned missing rates of datasets.

Missing Rate	20%	40%	60%	80%
BJTaxi P1	98.86%	94.28%	78.47%	50.24%
NYCTaxi	95.50%	84.96%	65.08%	38.08%
MHK	96.14%	85.03%	64.88%	37.67%

upsampling and presents the improved method. **UrbanSG** [35] employs a conditional GAN [10] as the backbone, considering external factors as the specified condition. **MT-CSR** [17] makes the first attempt to infer fine-grained flows based on the incomplete coarse-grained urban flow observations. **UrbanSTC** [24] formulates a self-supervision to predict fine-grained urban flows, taking into account all correlated spatial and temporal contrastive patterns.

4.2.3 Training Details and Hyper-parameters Settings. We employ the Adam [7] optimizer with a learning rate of $1e^{-3}$ and a weight decay of $5e^{-5}$. We set the number of residual blocks in the FE blocks and the feature extraction network as 1 and 3, respectively. For convenience in comparing with baselines, we apply the same number of channels, i.e., 128. The self-supervised loss weight is searched within the range $\{0.002, 0.01, 0.02, 0.05, 0.1, 0.5\}$, and the structural constraint loss weight is chosen from $\{0.01, 0.02, 0.05, 0.08, 0.1, 0.2, 0.5\}$.

Dataset Settings. We set the missing rates of input data maps at 20%, 40%, 60%, and 80% to evaluate MGSR with baselines where the contribution of the two missing modes is the same. For example, with a missing rate of 20%, both the sensor distribution and the random missing pattern generate 10% missing data each.

Baseline Settings. For baselines, we initially preserve the settings as provided in the original papers and fine-tune hyperparameters across the datasets. Due to the baselines not being proficient in multi-granularity data maps, for convenience of comparison, we manually align data maps and generate one data map. Additionally, the baselines cannot end-to-end infer multi-granularity data maps while inferring one granularity data map at a time.

For the input, the granularity of the aligned data map is the same as the highest granularity in input data maps. We fill the data map with ground truths if the corresponding position is covered by an arbitrary data map in the input data maps (see detailed description in Appendix A). The manual alignment method is the performance upper bound of GLAN. For the output, each baseline method requires a separate model to be trained for each granularity data map. In this way, baselines avoid the multi-task problem but increase the number of parameters and computing power consumption.

4.3 Performance Results

To demonstrate the effectiveness of MGSR (**RQ1**), we compare it with baselines. For convenience of comparison, we design a variant of MGSR, named MGSR-S, where the input and output ways are the same as the baselines. From the performances summarized in Table 3, we have the following findings.

We observe the performances of MGSR and baselines across various missing rates, from low to high. As shown in Table 2, the aligned missing rates are low and approximately non-missing when setting the missing rate at 20%. As the missing rate grows to 60% and above, the aligned missing rate significantly decreases. MGSR and baselines exhibit similar or slightly inferior performance when setting 20% missing rates. This can be attributed to the baselines' use of GAN and self-supervised tasks, which are effective at increasing performance levels when the missing data rate is relatively low. However, as the missing rates increase, the performance of most baselines declines significantly. MT-CSR shows a minor decline, but MGSR exhibits only a slight decrease in performance as the missing rates rise. MT-CSR exhibits limitations in addressing the few-shot problem, which is particularly evident in its performance on the MHK dataset. MGSR consistently outperforms the baselines, especially in scenarios with higher missing data rates. This consistent performance advantage underscores the robustness of our method in handling varying degrees of data incompleteness, highlighting its capability to maintain high-quality inference across different granularity levels, even under challenging conditions.

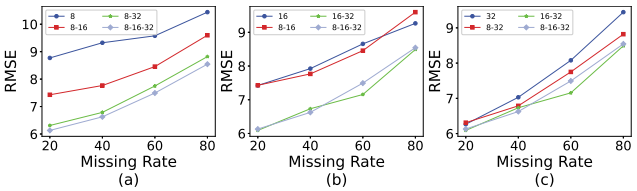

Figure 3: Illustration of GLAN performances on BJTaxi P1. (a), (b), and (c) are 8×8, 16×16, and 32×32, respectively.

The performance of Mean surpasses that of HA because HA relies on historical mappings that can be disrupted by missing data, leading to performance degradation. UrbanFM and UrbanPy exhibit performance drops with increased missing data because they do not account for missing values. UrbanSG improves performance with the GAN architecture and handles missing data better. UrbanSTC alleviates missing data issues through self-supervised tasks. MT-CSR, which considers missing values, performs well but cannot handle different missing patterns.

Additionally, the input and output ways of MGSR differ from those of MGSR-S and the baselines, where MGSR is at a disadvantage in terms of input and output. This is why MGSR cannot perform as well as MGSR-S. For the input, MSRG-S obtains fine-grained information under the same coverage area. For the output, MGSR is an end-to-end framework that infers multi-granularity data maps with fewer parameters, which is a multi-task problem. Meanwhile, the multi-task labels provide additional supervised signals for parameter optimization, enhancing its generalization capability. This is why MGSR performance is degraded compared to MGSR-S performance, especially the former.

4.4 GLAN Analysis

To evaluate whether the GLAN can align information in incomplete multi-granularity data maps (**RQ2**), we conduct experiments with 8, 16, and 32 as base, gradually incorporating other granularity information. The results, as illustrated in Fig. 3, show that RMSE and MAE for inferring 64 and 128 granularity data maps decrease as more data maps are aligned. Integrating coarse-grained data maps with fine-grained data maps dramatically boosts performance, shown in Fig. 3(a). However, performance improvements also risk introducing noise when coarse-grained data maps integrate into

Table 3: Performance comparison of MGSR and baselines.

Model			Mean	HA	UrbanFM	UrbanPy	UrbanSG	MT-CSR	UrbanSTC	MGSR-S	MGSR	Δ
BJTaxi P1	20%	RMSE	26.73	38.07	6.06	6.13	6.09	6.03	5.87	**5.40**	6.13	8.0%
		MAE	15.70	18.26	2.79	2.81	2.79	3.86	**2.62**	3.10	3.23	-18.3%
	40%	RMSE	27.59	38.25	10.97	11.08	10.97	7.15	10.79	**5.74**	6.63	19.8%
		MAE	15.90	18.36	3.73	3.82	3.71	4.64	3.48	3.48	**3.41**	2.0%
	60%	RMSE	29.64	38.71	18.28	18.72	18.27	8.76	18.14	**6.40**	7.49	27.0%
		MAE	16.41	18.65	6.32	7.27	6.28	5.64	6.01	**3.53**	3.80	37.4%
	80%	RMSE	33.50	39.69	27.92	28.03	27.91	12.42	27.80	**7.44**	8.54	40.1%
		MAE	17.37	19.22	11.18	11.47	11.10	8.12	10.81	**4.04**	4.09	50.3%
NYCTaxi	20%	RMSE	5.09	6.63	2.55	2.57	2.92	2.36	2.51	**2.25**	2.42	5.0%
		MAE	2.49	3.00	1.25	1.26	1.43	1.40	**1.21**	1.23	1.27	-1.6%
	40%	RMSE	5.38	6.70	3.48	3.51	3.72	2.53	3.44	**2.34**	2.53	7.3%
		MAE	2.58	3.03	1.52	1.56	1.68	1.47	1.49	**1.27**	1.32	13.7%
	60%	RMSE	5.93	6.85	4.68	4.72	4.83	2.74	4.64	**2.45**	2.66	10.6%
		MAE	2.81	3.12	2.03	2.07	2.14	1.59	1.98	**1.31**	1.34	17.2%
	80%	RMSE	6.59	7.05	5.97	5.97	6.03	3.16	5.94	**2.67**	2.90	15.7%
		MAE	3.09	3.24	2.63	2.64	2.70	1.80	2.59	**1.40**	1.40	16.6%
MHK	20%	RMSE	48.96	58.24	20.89	21.37	22.48	24.42	19.98	**19.74**	22.41	1.2%
		MAE	17.40	15.89	**5.17**	5.22	5.54	12.89	5.43	5.61	5.57	-7.7%
	40%	RMSE	50.10	58.65	26.70	28.11	26.93	27.89	26.84	**19.81**	28.21	25.8%
		MAE	17.13	15.83	5.47	5.86	5.55	14.70	**4.87**	5.05	6.57	-3.7%
	60%	RMSE	53.57	59.87	39.27	38.92	40.58	35.67	38.64	**28.29**	32.17	20.7%
		MAE	16.89	15.89	8.40	8.06	8.84	18.09	7.63	**7.07**	7.69	7.3%
	80%	RMSE	57.76	61.40	51.50	51.06	51.96	43.82	50.83	**32.66**	34.92	25.5%
		MAE	16.58	16.01	11.84	11.74	11.86	22.84	11.18	**9.25**	10.44	17.3%

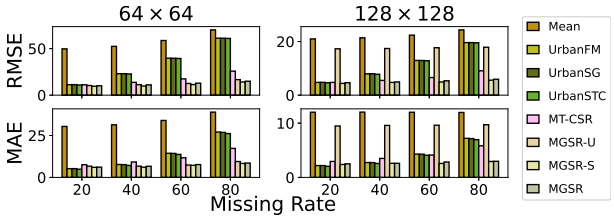

Figure 4: Illustration of RSRN performances on BJTaxi P1.

fine-grained data maps, as shown in Fig. 3(b) and (c). This suggests that GLAN effectively integrates multi-granularity data.

Additionally, Table 2 reveals that GLAN remains robust despite substantial fluctuations in missing rates. Even with varying degrees of data incompleteness, GLAN maintains its performance, demonstrating its resilience.

Overall, these findings indicate that GLAN not only improves the precision of inferred data maps by effectively utilizing information from various granularities but also offers a robust solution against data incompleteness. These findings underscore GLAN's value in multi-granularity data imputation and inference tasks, highlighting its potential for real-world urban flow monitoring applications.

4.5 RSRN Analysis

To answer **RQ3**, this subsection conducts various experiments about RSRN to validate the effectiveness in the multi-task problem and the generalization capability. We introduced a new variant, MGSR-U, which is trained using only the 64 granularity as the ground truth while simultaneously inferring both 64 and 128 granularities.

As shown in Fig. 4, the results indicate that performances of baselines significantly decline with missing rates increasing. This highlights their limited robustness and adaptability to incomplete data. MGSR-S and MGSR exhibit comparable performance, demonstrating RSRN effectiveness for the multi-task problem. This is attributed to the relatedness of the tasks and the appropriately designed structure, enhancing generalization when trained together.

For 64, MGSR-U's performance is close to MGSR, indicating effective learning and inference at this granularity even without explicit multi-granularity training. However, performance of inferring 128 is a significant drop, due to the lack of 128 granularity labels during training. Despite this, MGSR-U outperforms Mean and surpasses some fine-grained inference models. The chained structural constraint provides an optimized supervisory signal that aids better inference even without training data. Meanwhile, RSRN, leveraging its recursive nature, effectively learns patterns of inference during training, significantly contributing to its superior performance.

These experiments demonstrate that, while the multi-task problem can complicate the learning process, properly designed network structures and training strategies can significantly enhance generalization capabilities. The RSRN module, through its ability to leverage structural constraints and recursive learning, shows promising results in the multi-task problem and generalization.

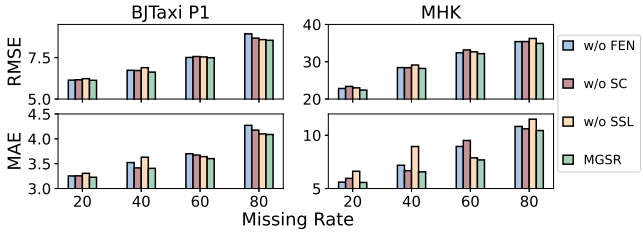

Figure 5: Ablation study of MGSR on BJTaxi P1 and MHK.

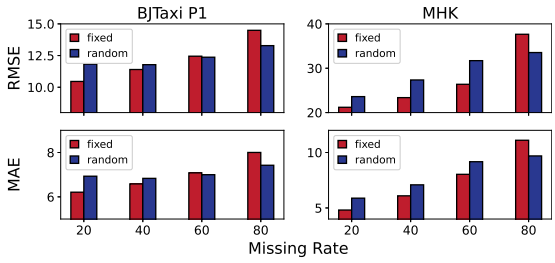

Figure 6: Performance comparison of different missing patterns. The "fixed" and the "random" denote that sensor distribution and random result in missing values, respectively.

4.6 Ablation Study

In this subsection, we present ablation studies to evaluate the contributions of components within our proposed framework (**RQ4**). As depicted in Fig. 5, we consider the following variants as below:

w/o FEN. This variant excludes the FEN. It leads to a significant drop in performance due to the reduced ability, capturing complex spatio-temporal correlations. **w/o SC**. The structural constraint is removed. This results in a performance decline, emphasizing its importance in preserving structural relationships across different granularities. The structural constraint acts as a regularizer that guides the framework to maintain the inherent structure of the data. **w/o SSL**. This variant removes the self-supervised loss. The performance drop suggests that it plays a significant role in enhancing the ability to infer fine-grained data maps. It utilizes the available data more effectively by learning from the data itself, improving the robustness and accuracy of the inference.

4.7 Missing Patten Analysis

We analyze the impact of different missing patterns (**RQ5**), as illustrated in Fig. 6. When the missing rates for BJTaxi P1 and MHK are below 60% and 80%, respectively, random missing patterns exhibit a more significant negative impact. The random missing pattern creates more randomness, resulting in spatial discontinuities, and disrupting the framework's ability to infer high-granularity data maps effectively. Conversely, when the missing rates exceed these thresholds, the sensor distribution pattern becomes more detrimental. The fixed-position missing poses a more considerable challenge, which cannot capture spatio-temporal relationships between regions leading to greater performance degradation. These findings demonstrate the varying impacts of missing data patterns, underscoring the importance of accounting for the missing patterns.

5 Related Work

5.1 Fine-grained Data Map Inference

This fine-grained data inference focuses on enhancing the data granularity to provide more detailed and accurate information. UrbanFM [18] was the pioneering work introducing super-resolution convolution in the context of urban data map inference. Subsequent models, such as UrbanPy [23], have modified this framework to further enhance performance. DeepLGR [19] employed temporal features to infer fine-grained data maps. UrbanSG [35] integrated the GAN framework to aid in optimizing parameters. MT-CSR [17] considered fine-grained data map inference in the missing data map at a single level of granularity. To address limited data challenges, models like UrbanSTC [24] and STCF [31] employed self-supervised learning, thereby improving generalization and reducing reliance on data. STCF attracted spatial-temporally similar flow maps while distancing dissimilar ones within the representation space. Inspired by diffusion models, DiffUFlow [36] overlay the extracted spatial-temporal feature onto the coarse-grained flow map, serving as a conditional guidance for the reverse diffusion process.

In addition, other techniques have been explored for fine-grained inference. UFI-Flow [31] learned the conditional distributions of coarse- and fine-grained map pairs. FODE [37] extended neural Ordinary Differential Equations.

In conclusion, the fine-grained has evolved significantly with advancements in image super-resolution techniques, multi-task and self-supervised learning. Despite the notable progress made, challenges remain in achieving fine-grained multi-granularity inference, particularly in the alignment of multi-granularity data maps.

5.2 Self-supervised Learning

Self-supervised learning is widely used in many fields, e.g., computer vision [5, 11–13], spatio-temporal data [28, 34], etc. In computer vision, self-supervised learning methods like contrastive learning [6, 8, 9, 29, 38] and masked image modeling [2, 16, 27, 30] have demonstrated remarkable success in tasks such as image classification, segmentation, and object detection by leveraging large unlabeled datasets. In the spatio-temporal data [14, 15, 20, 32], self-supervised learning has proven useful for enhancing model robustness and performance in the absence of extensive labeled data.

In particular, fine-grained data map inference has benefited from self-supervised techniques that aim to enhance model generalization and reduce reliance on large labeled datasets.

6 Conclusion

This paper proposes MGSR to address fine-grained multi-granularity data inference by leveraging incomplete multi-granularity data. Incomplete and multi-granularity problems are widely seen in real-life scenarios. We design the GLAN to align incomplete multi-granularity information. Additionally, we employ a well-designed self-supervised task to impute missing data at each granularity. Meanwhile, we propose the RSRN to tackle the multi-task nature of multi-granularity data map inference and enhance the framework's generalization capabilities. Extensive experiments validate our framework's superiority, demonstrating substantial improvements in multi-granularity data map inference.

References

[1] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. 2022. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*. Springer, 205–218.

[2] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. 2022. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11315–11325.

[3] Jialei Chen, Yuanbo Xu, Pengyang Wang, and Yongjian Yang. 2023. Deep Generative Imputation Model for Missing Not At Random Data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 316–325.

[4] Longbiao Chen, Jérémie Jakubowicz, Dingqi Yang, Daqing Zhang, and Gang Pan. 2016. Fine-grained urban event detection and characterization based on tensor cofactorization. *IEEE Transactions on Human-Machine Systems* 47, 3 (2016), 380–391.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[6] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. 2022. When does contrastive visual representation learning work?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14755–14764.

[7] P Kingma Diederik. 2014. Adam: A method for stochastic optimization. *(No Title)* (2014).

[8] Rizhao Fan, Matteo Poggi, and Stefano Mattoccia. 2023. Contrastive learning for depth prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3226–3237.

[9] Songwei Ge, Shlok Mishra, Simon Kornblith, Chun-Liang Li, and David Jacobs. 2023. Hyperbolic contrastive learning for visual representations beyond objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6840–6849.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems* 33 (2020), 21271–21284.

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16000–16009.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[14] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. 2023. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 4356–4364.

[15] Junzhong Ji, Fan Yu, and Minglong Lei. 2022. Self-supervised spatiotemporal graph neural networks with self-distillation for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* 24, 2 (2022), 1580–1593.

[16] Xiangwen Kong and Xiangyu Zhang. 2023. Understanding masked image modeling via learning occlusion invariant feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6241–6251.

[17] Jiyue Li, Senzhang Wang, Jiaqiang Zhang, Hao Miao, Junbo Zhang, and Philip S. Yu. 2023. Fine-Grained Urban Flow Inference With Incomplete Data. *IEEE Transactions on Knowledge and Data Engineering* 35, 6 (2023), 5851–5864.

[18] Yuxuan Liang, Kun Ouyang, Lin Jing, Sijie Ruan, Ye Liu, Junbo Zhang, David S Rosenblum, and Yu Zheng. 2019. Urbanfm: Inferring fine-grained urban flows. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 3132–3142.

[19] Yuxuan Liang, Kun Ouyang, Yiwei Wang, Ye Liu, Junbo Zhang, Yu Zheng, and David S Rosenblum. 2021. Revisiting convolutional neural networks for citywide crowd flow analytics. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*. Springer, 578–594.

[20] Chang Liu, Yuan Yao, Dezhao Luo, Yu Zhou, and Qixiang Ye. 2022. Self-supervised motion perception for spatiotemporal representation learning. *IEEE Transactions on Neural Networks and Learning Systems* 34, 12 (2022), 9832–9846.

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10012–10022.

[22] Larry R Medsker, Lakhmi Jain, et al. 2001. Recurrent neural networks. *Design and Applications* 5, 64-67 (2001), 2.

[23] Kun Ouyang, Yuxuan Liang, Ye Liu, Zekun Tong, Sijie Ruan, Yu Zheng, and David S. Rosenblum. 2022. Fine-Grained Urban Flow Inference. *IEEE Transactions on Knowledge and Data Engineering* 34, 6 (2022), 2755–2770.

[24] Hao Qu, Yongshun Gong, Meng Chen, Junbo Zhang, Yu Zheng, and Yilong Yin. 2023. Forecasting Fine-Grained Urban Flows Via Spatio-Temporal Contrastive Self-Supervision. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2023), 8008–8023.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. 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*. Springer, 234–241.

[26] Olivier Rukundo and Hanqiang Cao. 2012. Nearest neighbor value interpolation. *arXiv preprint arXiv:1211.1768* (2012).

[27] Linus Scheibenreif, Michael Mommert, and Damian Borth. 2023. Masked vision transformers for hyperspectral image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2166–2176.

[28] Qingmei Wang, Minjie Cheng, Shen Yuan, and Hongteng Xu. 2023. Hierarchical contrastive learning for temporal point processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 10166–10174.

[29] Xuehui Wang, Kai Zhao, Ruixin Zhang, Shouhong Ding, Yan Wang, and Wei Shen. 2022. Contrastmask: Contrastive learning to segment every thing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11604–11613.

[30] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. 2022. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9653–9663.

[31] Haoyang Yu, Xovee Xu, Ting Zhong, and Fan Zhou. 2022. Fine-Grained Urban Flow Inference via Normalizing Flow (Student Abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 13101–13102.

[32] Liangzhe Yuan, Rui Qian, Yin Cui, Boqing Gong, Florian Schroff, Ming-Hsuan Yang, Hartwig Adam, and Ting Liu. 2022. Contextualized spatio-temporal contrastive learning with self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13977–13986.

[33] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[34] Qianru Zhang, Chao Huang, Lianghao Xia, Zheng Wang, Zhonghang Li, and Siuming Yiu. 2023. Automated spatio-temporal graph contrastive learning. In *Proceedings of the ACM Web Conference 2023*. 295–305.

[35] Xv Zhang, Yuanbo Xu, Ying Li, and Yongjian Yang. 2023. Fine-Grained Urban Flow Inferring via Conditional Generative Adversarial Networks. In *Web and Big Data*, Bohan Li, Lin Yue, Chuanqi Tao, Xuming Han, Diego Calvanese, and Toshiyuki Amagasa (Eds.). Springer Nature Switzerland, Cham, 420–434.

[36] Yuhao Zheng, Lian Zhong, Senzhang Wang, Yu Yang, Weixi Gu, Junbo Zhang, and Jianxin Wang. 2023. Diffuflow: Robust fine-grained urban flow inference with denoising diffusion model. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 3505–3513.

[37] Fan Zhou, Liang Li, Ting Zhong, Goce Trajcevski, Kunpeng Zhang, and Jiahao Wang. 2020. Enhancing urban flow maps via neural odes. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,{ IJCAI} 2020*.

[38] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. 2022. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6908–6917.

Table 4: Dataset Description.

Dataset	Time span	Time interval	#Channel	Input Granularity	Output Granularity	#Train	#Valid	#Test
BJTaxi	P1: 7/1/2013-10/31/2013	30 minutes	1	8×8, 16×16, 32×32	64×64, 128×128	1530	765	765
	P2: 2/1/2014-6/30/2014					1779	889	891
	P3: 3/1/2015-6/30/2015					1746	873	873
	P4: 11/1/2015-3/31/2016					2122	1061	1061
NYCTaxi	1/1/2015-1/7/2015	1 hour	2	8×8, 16×16	32×32, 64×64	2880	744	720
MHK	1/21/2023-1/27/2023	30 minutes	1	8×8, 16×16	32×32	240	48	48

Table 5: Performance comparison of MGSR and baselines on BJTaxi P2-4 Dataset.

Model			Mean	HA	UrbanFM	UrbanPy	UrbanSG	MT-CSR	UrbanSTC	MGSR-S	MGSR	Δ
BJTaxi P2	20%	RMSE	34.29	47.07	7.55	7.91	7.56	6.67	7.33	**6.14**	7.17	8.0%
		MAE	20.11	22.56	3.17	3.47	3.18	4.25	**2.97**	3.62	3.74	-21.9%
	40%	RMSE	35.19	47.26	13.26	13.68	13.26	7.80	13.09	**6.41**	7.58	17.8%
		MAE	20.28	22.67	4.29	4.94	4.30	5.05	4.05	**3.73**	3.91	7.9%
	60%	RMSE	37.99	47.87	23.23	23.37	23.23	10.05	23.11	**7.00**	8.56	30.4%
		MAE	21.00	23.03	7.68	8.00	7.65	6.56	7.37	**3.97**	4.19	39.5%
	80%	RMSE	43.09	49.15	36.22	36.78	36.21	14.87	36.16	**8.58**	9.76	42.3%
		MAE	22.22	23.79	14.23	15.56	14.16	9.68	13.95	**4.62**	4.87	52.3%
BJTaxi P3	20%	RMSE	35.09	46.81	6.91	7.47	6.92	6.96	6.63	**6.44**	7.15	2.9%
		MAE	20.90	22.86	3.19	3.68	3.18	4.51	**2.96**	3.71	3.79	-25.3%
	40%	RMSE	36.29	47.06	14.35	14.56	14.36	8.08	14.17	**6.71**	8.02	16.9%
		MAE	21.17	23.01	4.67	4.97	4.66	5.28	4.39	**3.87**	4.22	11.8%
	60%	RMSE	39.67	47.80	25.78	25.93	25.78	10.72	25.63	**7.31**	8.60	31.8%
		MAE	22.01	23.43	8.57	8.98	8.55	7.01	8.18	**4.03**	4.29	42.5%
	80%	RMSE	43.86	48.81	36.25	36.55	36.24	14.79	36.12	**8.66**	9.98	41.5%
		MAE	23.15	24.09	14.56	15.44	14.51	9.76	14.14	**4.71**	4.90	51.8%
BJTaxi P4	20%	RMSE	24.41	32.09	5.68	6.22	5.69	5.40	5.49	**5.02**	5.62	7.0%
		MAE	14.44	15.60	2.53	3.03	2.55	3.56	**2.38**	3.04	2.99	-25.6%
	40%	RMSE	25.25	32.26	10.46	10.69	10.46	6.25	10.32	**5.09**	5.97	18.5%
		MAE	14.62	15.70	3.51	3.87	3.51	4.13	3.32	**3.02**	3.14	9.0%
	60%	RMSE	27.27	32.68	17.27	17.52	17.27	7.84	17.16	**5.53**	6.76	29.4%
		MAE	15.10	15.93	5.80	6.36	5.79	5.16	5.55	**3.18**	3.45	38.3%
	80%	RMSE	31.03	33.57	26.28	26.37	26.28	11.09	26.20	**6.76**	8.07	39.1%
		MAE	16.13	16.48	10.47	10.75	10.44	7.42	10.21	**3.69**	4.03	50.2%

A Manual Alignment Method

The granularity of the aligned data map is determined by the highest granularity present in the input data maps, ensuring that the granularity of the aligned map matches that of the finest data available. The alignment process involves filling the aligned data map with ground truths, but only in regions that are covered by at least one of the input data maps.

To clarify this, as depicted in Fig. 7, consider a scenario where two data maps of different granularity, one at 2×2 and another at 4×4, are manually aligned to produce a unified 4×4 data map. The initial step, illustrated by the orange section, involves using the high-granularity data map to populate the aligned map, filling in all corresponding cells where data is available at the high granularity. After this, the low-granularity data map is applied to complete the remaining areas, represented in the green section. In cases where the low-granularity map lacks precise values for certain regions (e.g., 13 and 19), these grids are instead filled using ground truths to maintain data consistency and ensure the aligned map is fully populated.

This manual alignment process sets an upper bound on performance, meaning it represents the most accurate possible reconstruction of the data. By combining high- and low-granularity inputs and supplementing missing values with ground truths, this method provides an optimal solution for map alignment, allowing the resulting map to maintain the highest fidelity achievable with the given data.

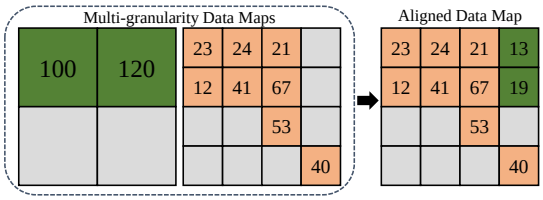

Figure 7: Illustration of manually aligning multi-granularity data maps. The left and right are multi-granularity data maps and an aligned data map, respectively. The gray grids denote missing data. The green and orange grids in the aligned data map indicate the values from coarse- and fine-grained data maps.

Table 6: Aligned missing rates of datasets. It illustrates aligned missing rates that are constructed according to the way in Fig. 7 at various setting missing rates.

Missing Rate	20%	40%	60%	80%
BJTaxi P2	98.99%	94.00%	78.58%	50.06%
BJTaxi P3	99.46%	93.43%	78.91%	50.16%
BJTaxi P4	99.16%	93.93%	79.55%	50.09%

B Experimental Details and Further Results

B.1 Statistics of Datasets

In Section 4, we have described three datasets used for validating the performance of our model. To further strengthen our experiments, we now introduce three additional datasets as supplementary evaluation benchmarks.

Table 4 presents a detailed summary of all six datasets, including key statistics such as the time span each dataset covers and the granularity of both the input and output. In addition, we partition the dataset into non-overlapping training, validation, and testing sets, following a chronological order. The table also shows the number of samples in the training, validation, and test sets for each dataset, ensuring a clear understanding of the dataset distributions used throughout our experiments.

B.2 Extended results

Table 6 presents the aligned missing rates across the datasets, further supporting and validating the results discussed in Subsection 4.2. These supplementary experiments reinforce the findings from the initial tests by confirming that the aligned missing rates consistently behave as expected across different scenarios and datasets, regardless of the variations in missing data rates.

In addition, Table 5 provides performance results for the BJTaxi P2-P4, offering a more granular breakdown of model performance. For further insights into the methodology and settings for these tests, readers can refer to Section 4.3. The supplementary results align closely with the original findings, confirming that the conclusions reached in Section 4.2 remain valid. These additional experiments help to strengthen the robustness and generalizability of the results across different experimental conditions and datasets.