Spatial-Temporal Contrasting for Fine-Grained Urban Flow Inference

Xovee Xu, *Graduate Student Member, IEEE*, Zhiyuan Wang, Qiang Gao, Ting Zhong, Bei Hui, Fan Zhou, *Member, IEEE*, and Goce Trajcevski, *Member, IEEE*

***Abstract*—Fine-grained urban flow inference (FUFI) problem aims to infer the fine-grained flow maps from coarse-grained ones, benefiting various smart-city applications by reducing electricity, maintenance, and operation costs. Existing models use techniques from image super-resolution and achieve good performance in FUFI. However, they often rely on supervised learning with a large amount of training data, and often lack generalization capability and face overfitting. We present a new solution: Spatial-Temporal Contrasting for Fine-Grained Urban Flow Inference (STCF). It consists of (i) two pre-training networks for spatial-temporal contrasting between flow maps; and (ii) one coupled fine-tuning network for fusing learned features. By attracting *spatial-temporally similar* flow maps while distancing dissimilar ones within the representation space, STCF enhances efficiency and performance. Comprehensive experiments on two large-scale, real-world urban flow datasets reveal that STCF reduces inference error by up to 13.5%, requiring significantly fewer data and model parameters than prior arts.**

***Index Terms*—Contrastive learning, traffic management, urban computing, urban flow inference.**

I. Introduction

THE rapid development and miniaturization of sensing systems (e.g., mobile devices, surveillance cameras, and particle concentration sensors), along with the advances in communication and information infrastructures, have enabled a plethora of novel applications in the realm of smart cities and urban computing [1], [2], [3]. An important consequence is the generation of vast volumes of heterogeneous spatial-temporal data that could be used for improved management of quality of life. However, one of the main challenges, from both economic and technical perspectives, is how to balance the trade-offs between the number of devices and communication overheads versus maintaining a satisfactory inference/prediction performance in various transportation systems [4].

In many real-world urban computing and traffic applications, e.g., transportation managing and planning, city resource scheduling and allocating, real-time decision making and city construction planning [5], a specific problem that attracts considerable attention is the so-called Fine-grained Urban Flow Inference (FUFI), which aims to upscale the coarse-grained, low-resolution urban flow map into fine-grained, high-resolution ones. FUFI can be employed in large-scale urban transportation systems [6], e.g., citywide traffic flow monitoring [7] and radio map reconstruction [8]. In traditional sensing systems, deploying and maintaining a large number of devices require large amounts of electricity and human labor. The FUFI problem can effectively reduce the expensive maintenance/operation costs while also upholding an acceptable accuracy when inferring the real-time fine-grained flow maps, contributing to environmental protection, energy-saving, and emission reduction.

Existing Work: Researchers have addressed the FUFI problem by mainly utilizing deep convolutional neural networks (especially in the point view of single image super-resolution in computer vision [9]) tailored for urban crowd flow data. UrbanFM [7] is among the first to formalize the FUFI problem, which adopts a deep residual architecture [10] as the main building block, and designs a distributional upsampling layer and a feature fusion sub-net for capturing the spatial constraint and external influence factors, respectively. FODE [11] extends UrbanFM by introducing neural ordinary differential equations [12] in an affine coupling layer, balancing between inference accuracy and computational efficiency. UrbanODE [13] further enhances FODE by introducing a pyramid attention network for spatial-temporal feature extraction.

Limitation: Although existing methods surpass traditional image super-resolution models such as SRCNN [14] and VDSR [15], being fully supervised, they rely on special and complex architectural designs and often require massive training data to guarantee satisfactory inference performance. Furthermore, stacking convolutional layers to increase the receptive fields is also inefficient according to [16]. Supervised methods with large architectures run substantial risks of overfitting and low generalization capabilities [17], [18], limiting their practical use in FUFI systems.

Manuscript received 28 March 2023; revised 27 August 2023; accepted 4 September 2023. Date of publication 18 September 2023; date of current version 13 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 62273071, 62072077, and 62176043, in part by the Natural Science Foundation of Sichuan Province, China, under Grant 2022NSFSC0505, and in part by the National Science Foundation under Grant SWIFT 2030249 and in part by Kingland Foundation. Recommended for acceptance by Y. Tong. *(Corresponding author: Bei Hui.)*

Xovee Xu, Ting Zhong, and Fan Zhou are with the University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China, and also with the Kash Institute of Electronics and Information Industry, Kashi, Xinjiang 84400, China (e-mail: xovee.xu@gmail.com; zhongting@uestc.edu.cn; fan.zhou@uestc.edu.cn).

Zhiyuan Wang and Bei Hui are with the University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China (e-mail: zhy.wangcs@gmail.com; bhui@uestc.edu.cn).

Qiang Gao is with the Southwestern University of Finance and Economics, Chengdu, Sichuan 611130, China (e-mail: qianggao@swufe.edu.cn).

Goce Trajcevski is with Iowa State University, Ames, IA 50011 USA (e-mail: gocet25@iastate.edu).

Digital Object Identifier 10.1109/TBDATA.2023.3316471

Challenge: To address the above limitations, self-supervised representation learning is a promising direction, which has shown excellent results in many language and vision tasks [19], [20], primarily due to their impressive power in learning universal features, requiring fewer data, generalizing well, and mitigating the overfitting issues [21], [22]. However, directly applying self-supervised techniques into FUFI models faces several vital obstacles: (i) data augmentation procedures – which are commonly used in visual representation learning, such as crop, resize, and rotation – would greatly disrupt the spatial structure of flow maps; (ii) existing pre-training strategies, e.g., solving jigsaw puzzles [23], predicting relative position [24], global-local contrast [25], would break the spatial-constraint required in FUFI [7]. Instance discrimination approaches [21], [26], on the one hand, ignore the temporal correlations between flow maps and, on the other hand, rely on data augmentations to create similar data views for contrasting.

Present Work: In this article, we propose a new FUFI solution – **S***patial-***T***emporal* **C***ontrasting for Fine-Grained Urban* **F***low Inference* (**STCF**), which follows a pre-training & fine-tuning paradigm. We design two new augmentation-agnostic pretext tasks specifically for urban flow data: (i) spatial-contrasting between coarse- and fine-grained flow maps; and (ii) temporal-contrasting between time-related coarse-grained flow maps. The fine-grained flow maps during self-supervised pre-training are served as discrimination samples rather than prediction labels, facilitating the model to learn better flow map representations. The proposed two pre-training networks are able to attract positive samples in the representation space without destructing the *spatial structures* of flow maps. Moreover, we propose a new external influence factor aggregation module that accounts for the influence of factors on individual map cells. Finally, we adopt a coupled fine-tuning network to link the pre-trained feature maps for fine-grained flow map inference. STCF has a simple structure, requires no special/complex architecture designs [7], [11] nor stacked deep convolutional layers [27], [28], but achieves significant performance boosting. Our main contributions are as follows:

- *Spatial-temporal contrastive pretext tasks are designed for urban flow map feature learning:* To the best of our knowledge, STCF is the first attempt to introduce self-supervised learning into fine-grained urban flow inference. We design two novel augmentation-agnostic pretext tasks in a contrastive manner: multiple (spatially or temporally correlated) positive flow map samples are attracted together in the representation space, while negative samples are repelled away from the positive samples.
- *Combining spatial-temporal correlations for fine-grained urban flow map inference:* We jointly model the flow map relationships in a coupled fine-tuning network without special and complex architecture designs or stacked deep convolutional layers that may result in overfitting. We further propose a new external influence factor aggregation module that accounts for fine-grained factor influence on individual map cells.
- *Our framework is data-efficient, lightweight, and outperforms prior arts by non-trivial margins:* We conducted extensive experiments on two large-scale real-world urban flow datasets demonstrate that STCF significantly improves urban flow inference performance compared to state-of-the-art approaches.

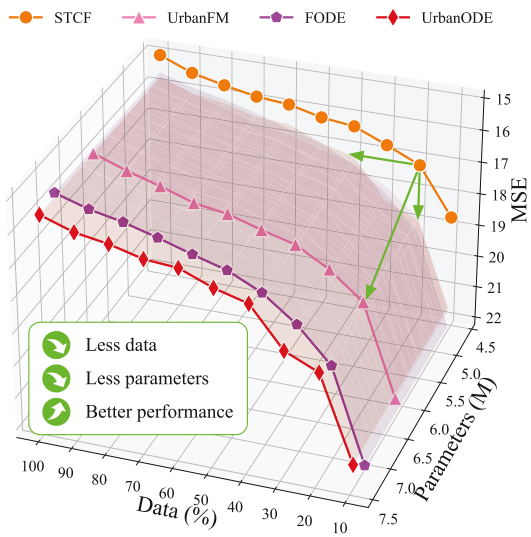

Fig. 1. Urban flow inference performance comparison between our proposed STCF and three state-of-the-art baselines on TaxiBJ dataset, with different data fractions and model parameters. STCF is data- and parameter-efficient and achieves significant performance improvements (indicated by the green arrows in three axis directions).

As shown in Fig. 1, combining our contributions, STCF surpasses three strong baselines on TaxiBJ P1 dataset, decreasing the inference error up to 13.5% (or 22.9% without external factors) compared to UrbanODE [13]. STCF generally needs 20–50% fewer data to perform on par with or even better than baselines and uses much fewer model parameters. These traits of STCF provide obvious advantages when computing resources or training data are limited, which is critical in FUFI that obtaining a large amount of data or maintaining many sensors require expensive costs or heavy crowd-sourcing, contributing to a green and sustainable transportation system.

II. Related Work

A. Fine-Grained Urban Flow Inference

Fine-grained urban flow inference (FUFI) problem is first formulated by [7] aiming at reducing the high costs of long-term operation and maintenance, and bridging the gap between storage/processing efficiency and data usability for large-scale transportation systems [6]. Closely related to single-image super-resolution (SISR) in computer vision, FUFI has several common concepts in comparison to SISR, e.g., they both study the regular data (urban flow maps can be seen as a special type of image), utilizing convolutional layers and upsampling techniques to infer high-resolution "images" from low-resolution "images". However, FUFI problem, despite its similarity to SISR, poses distinct characteristics and new challenges. First, SISR is an ill-posed problem, while FUFI has a unique solution that we want to infer the fine-grained urban flows as accurately as possible (often measured by mean squared errors other than PSNR and SSIM [29]). Moreover, FUFI problem has a spatial-constraint in

urban maps and is associated with complex external influence factors which greatly affect the urban flow distributions.

Existing FUFI models proposed several dedicated learning modules for addressing the challenges and improving the inference performance [30]. UrbanFM [7], as the first one to formulate and tackle the FUFI problem, distinguishing itself from SISR models by two essential components: an inference network with distributional upsampling and an external factor fusing subnet. Subsequent works mainly follow the learning paradigm of UrbanFM, such as stacked convolution layers and skip-connections between the low-level and high-level feature maps, but also propose new mechanisms to improve FUFI from different aspects [6], [31], [32], [33], [34].

For example, Zhou et al. [11] designed an affine coupling layer to overcome the gradient computation instability and introduced neural ordinary differential equations (ODE) [35], [36] into their model. The ODE-based module balances the trade-off between FUFI performance and computational overhead. Li et al. [8] explored the spatial-temporal relationships between historical urban maps by using stacked ST-Residual blocks and masked loss function. UrbanODE [13] adopted pyramid attention blocks to improve the flow map feature extraction.

Li et al. [31] studied the FUFI problem from the sparsity and incomplete data perspectives, i.e., the coarse-grained flow maps are unevenly distributed and incomplete. In such cases, accurately inferring the urban flows becomes extremely difficult. The authors proposed a multi-task urban flow completion and super-resolution model named MT-CSR, which first completes the coarse-grained urban flows and then upscales the flow map. MT-CSR incorporates the local spatial dependencies, global POI similarities, and the complex associations between coarse- and fine-grained urban flows.

In [16], the authors analyzed the characteristics of crowd flows and revisited the shortcomings of convolutional neural network (CNN)-based methods: inefficiency in learning global spatial dependencies and ignoring latent region functions.

These approaches often made strong assumptions and adopted complex architecture designs (e.g., the skip connections and pyramid attentions), which suffers from severe overfitting problems. In this work, we use a simple model architecture and self-supervised pre-training to improve the generalizability of our model.

B. Contrastive Self-Supervised Learning

Being fully-supervised, classical deep learning-based models achieving great success in research community in tremendous prediction tasks due to their outstanding ability to learn representations from large-scale datasets. However, such models often require massive data to guarantee a good performance and face several obstacles such as generalization capability, adversarial attack, and spurious data [37], especially when labels are expensive (or even impossible) to obtain.

Self-supervised learning (SSL), as a subset of unsupervised learning, provides researchers an alternative approach to learn good representations without human-annotated supervision. In recent years, SSL gained immense attention and showed noticeable results on downstream tasks. Researchers proposed several novel mechanisms that are enabling model training from data itself. They are often categorized into two directions: predictive or contrastive. Predictive SSL is focuses on designing hand-crafted pretext tasks, such as solving jigsaw puzzles [23], maximizing mutual information between local and global [25], predicting relative patch [24] and rotation [38]. Pretext tasks provide pseudo-labels for model training and pattern learning, and have shown promising results in computer vision (CV), natural language processing (NLP), and graph learning [22], [37], [39]. However, artificially designed pretext tasks rely on *ad-hoc* heuristics and domain-specific knowledge and therefore limiting their generalization capability [26].

Contrastive SSL can be seen as an instance discrimination method, aiming to train an encoder capable of discriminating positive and negative pairs by using a contrastive loss [40], [41], [42], [43], [44], [45], [46]. Several augmentation techniques and negative sampling strategies have been proposed to create different views of the same sample and optimize the feature-learning process. For example, He et al. [21] proposed momentum contrast mechanism to decouple the batch size from dictionary size; Chen et al. [26] used data augmentations (e.g., color distort, crop, resize, and Gaussian blur) to create two related views for instance discrimination; and Veličković et al. [47] presented DIM model which maximizes the mutual information between global and local representations of graphs.

Unlike existing contrastive SSL models, we study the urban flow inference problem and propose modeling the spatial-temporal relationships between urban flow maps in both coarse- and fine-granularities. The temporal-contrasting module in STCF considers temporally correlated flow maps as positive samples and lets them be close in representation space, while the spatial-contrasting module directly utilizes the fine-grained map as a related "view" of the coarse-grained map. It is worth noting that fine-grained flow maps were utilized during pre-training, but they served as self-supervision signals for contrastive learning rather than labels for supervised learning. These two modules are essentially different from previous SSL pretext tasks. They avoid the structure-destructive data augmentations and are suitable for the FUFI problem. To the best of our knowledge, STCF is the first work to study spatial-temporal contrastive learning and is also the first work to create sample views in different granularity in contrastive learning.

III. Preliminaries

FUFI problem is essentially different from the visual super-resolution since the former models the city flows rather than image pixels, colors, or video frames. In addition, FUFI problem obeys spatial constraint where the sum of fine-grained flow volume in a region should equal the coarse-grained flow volume in that region. Also, urban city flows are variable to external influence factors such as weather, geographical location, time, and holidays. Inferring the fine-grained flow maps exposes new challenges compared to visual SR problem.

TABLE I
MATHEMATICAL NOTATIONS (IN ALPHABETICAL ORDER)

Symbol	Description
C	Channel numbers of convolution layer.
F	External influence factors of map $\mathcal{M}$.
H	(Height) Map $\mathcal{M}$ has H lines of grid-cells.
$\mathbf{H}$	Hidden feature maps.
r_{ij}	A cell that locates in the i^{th} row and j^{th} column of map $\mathcal{M}$.
S	Upscaling factor of flow map inference.
T	Time span of how long we observe urban flows in a map.
$\mathcal{T}$	A collection of flow maps that temporally correlated.
W	(Width) Map $\mathcal{W}$ has W columns of grid-cells.
x_{ij}^t	Flow volume in region r_{ij} of map $\mathcal{M}$ during time $[t-T,t]$.
w_{ij}^t	A weight parameter for x_{ij}^t in distributional upsampling.
$\mathbf{X}^t$	The flow map of $\mathcal{M}$ during time $[t-T,t]$. $\widetilde{\mathbf{X}}^t$ is the output of EIF aggregation module. $\hat{\mathbf{X}}^t$ is the inferred flow map.
$\mathbf{z}$	Dense representation of map $\mathcal{M}$.

We now formally define the FUFI problem. Mathematical notations in this work are listed in Table I. Assuming that the urban area of interest $\mathcal{M}$ consists of $H \times W$ spatial grid-cells, a cell r_{ij} denotes the i^{th} row and the j^{th} column of $\mathcal{M}$, and $\mathbf{X}^t \in \mathbb{R}_+^{H\times W}$ denotes the flow map, at a given time t, where each entry $x_{ij} \in \mathbb{R}_+$ corresponds to the volume of the flow in the region r_{ij}.

Intuitively, the flow volume x_{ij}^t captures the number of entities inside the respective region r_{ij} over time window of $[t-T,t]$. Following previous models [7], [11], [13], [48], we formally define the FUFI problem as follows:

Definition 1 (Fine-Grained Urban Flow Inference): For a city map $\mathcal{M}$, given its coarse-grained, low-resolution flow map $\mathbf{X}_{\text{coarse}}^t$ at time t, the FUFI problem is to learn a mapping function $\mathcal{F}$ to transform the coarse-grained flow map $\mathbf{X}_{\text{coarse}}^t \in \mathbb{R}_+^{H\times W}$ into fine-grained, high-resolution flow map:

$$\mathbf{X}_{\text{fine}}^t = \mathcal{F}_\Theta(\mathbf{X}_{\text{coarse}}^t), \tag{1}$$

where $\mathbf{X}_{\text{fine}}^t \in \mathbb{R}_+^{SH\times SW}$, Θ denotes the learnable model parameters, and S is an upscaling factor.

Different from visual SR problem, FUFI obeys the following spatial constraint:

Definition 2 (Spatial Constraint): The flow volume $x_{ij,\text{coarse}}^t$ in cell r_{ij} of the coarse-grained flow map $\mathbf{X}_{\text{coarse}}^t$ should **equal** the overall flow volume in the corresponding $S \times S$ cells in fine-grained flow map $\mathbf{X}_{\text{fine}}^t$, i.e.,

$$x_{ij,\text{coarse}}^t = \sum_{\substack{i'\in[(i-1)S+1,iS]\\ j'\in[(j-1)S+1,jS]}} x_{i'j',fine}^t, \tag{2}$$

where $i \in [1,H], j \in [1,W]$ and $i' \in [1,SH], j' \in [1,SW]$.

An illustration of FUFI problem and spatial constraint for Beijing city is shown in Fig. 2. We note there is no temporal constraint in FUFI.

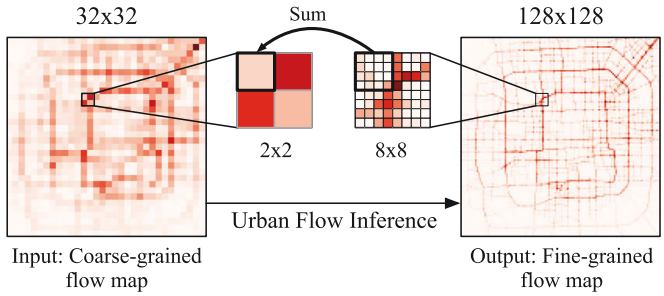

Fig. 2. Illustration of fine-grained urban flow inference. Take Beijing City as an example, given a coarse-grained, low-resolution flow map (32×32), we aim to infer a fine-grained, high-resolution flow map (128×128) while also obeying the structural constraint.

IV. STCF: METHODOLOGY

In this section, we present a Spatial-Temporal Contrasting model for fine-grained urban Flow inference (STCF). Specifically, we first introduce the two pre-training networks, which effectively extract useful feature maps in the aspects of spatial-temporal city flow dynamics, capture spatial-temporal flow map correlations, and require no data augmentations which would break the spatial constraint of FUFI. Then we describe STCF's external influence factor (EIF) aggregation module. Finally, we show how the coupled fine-tuning network can infer better fine-grained flow maps. A sketch of STCF framework is shown in Fig. 3.

A. Spatial-Contrasting Pre-Training Network

The input of the spatial-contrasting (SC) network is the coarse-grained flow map $\mathbf{X}_{\text{coarse}}^t$, which we encode by two convolutional layers with C channels and 3×3 kernel size, each layer followed by ReLU nonlinearity. The two convolutional layers are taken as a feature learning network to map coarse-grained flow map $\mathbf{X}_{\text{coarse}}^t$ to low-level hidden feature maps $\mathbf{H}_{\text{coarse}}^{t,\text{spatial}} \in \mathbb{R}^{H\times W\times C}$. We call this network a spatial-contrasting encoder $\mathbf{Enc}_{\text{coarse}}^{\text{spatial}}(\cdot)$ later used in fine-tuning stage. Then we adopt a batch normalization layer, another convolutional layer (with SC channels and 3×3 kernel size) followed by ReLU and global average pooling layer. Finally, we use a fully-connected layer with SC hidden units to get a dense representation $\mathbf{z}_{\text{coarse}}^{t,\text{spatial}} \in \mathbb{R}^{SC}$.

The global average pooling layer and dense layer are similar to the *MLP-based projection head* used in previous self-supervised learning (SSL) models [26], [49], which purpose is to bridge the learning objective gap between the two representations (one for downstream task and another for contrasting). However, the projection head in STCF aims to project feature maps $\mathbf{H}^t$ to flow map representation $\mathbf{z}^t$, while previous SSL models project one dense representation to another.

Similarly, we build another convolutional neural network to model the fine-grained flow map $\mathbf{X}_{\text{fine}}^t$, which has an identical structure but with different layer weights. It outputs $\mathbf{z}_{\text{fine}}^{t,\text{spatial}} \in \mathbb{R}^{SC}$ along with $\mathbf{z}_{\text{coarse}}^{t,\text{spatial}}$ for spatial contrastive learning. Both coarse- and fine-grained flow maps at the same time are compared in the representation space. Now we have defined two pre-training networks and we train them with InfoNCE contrastive loss function [26].

Given N urban flow samples in the pre-training set (i.e., $\mathbf{X}_{\text{coarse}}^t$ and $\mathbf{X}_{\text{fine}}^t$, $t \in [1,N]$), we randomly sample a mini-batch of flow maps, batch size is B. Then the loss function for our

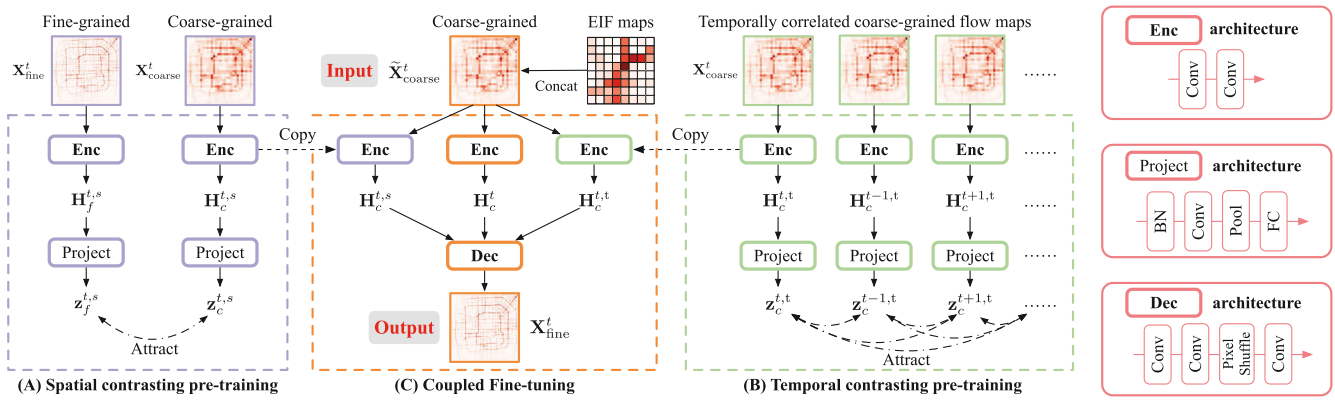

Fig. 3. Illustration of our proposed STCF framework. It has two pre-training stages for modeling spatial-temporal correlations between urban flow maps (a) and (b), and one coupled fine-tuning stage for urban flow inference (c). For simplicity, the $_c$, $_f$, ^{s}, and ^{t} in superscript/subscript denote *coarse*, *fine*, *spatial*, and *temporal*, respectively. The encoders and projection layers in the temporal contrasting pre-training network share their parameters. More details of external influence factor (EIF) aggregation module are depicted in Fig. 4.

spatial-contrasting pre-training network is defined as:

$$\mathcal{L}_{\text{SC},i} = -\log \frac{\exp\left(\text{sim}(\mathbf{z}^t_{\text{coarse},i}, \mathbf{z}^t_{\text{fine},i})/\theta\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp\left(\text{sim}(\mathbf{z}^t_{\text{coarse},i}, \mathbf{z}^t_{k})/\theta\right)} - \log \frac{\exp\left(\text{sim}(\mathbf{z}^t_{\text{fine},i}, \mathbf{z}^t_{\text{coarse},i})/\theta\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp\left(\text{sim}(\mathbf{z}^t_{\text{fine},i}, \mathbf{z}^t_{k})/\theta\right)}, \quad (3)$$

where $\text{sim}(\cdot,\cdot)$ is a similarity function between two representations (e.g., cosine similarity or inner dot product), $\theta \in \mathbb{R}^+$ is a scalar temperature hyper-parameter, $\mathbb{1}_{[\cdot]}$ is an indicator function (we omit the $^{\text{spatial}}$ superscript for brevity). The final loss is computed by $\mathcal{L}_{\text{SC}} = \frac{1}{2B}\sum_{i=1}^{B}\mathcal{L}_{\text{SC},i}$.

This loss function compares all the mini-batch samples in the latent representation space, where the positive pairs $(\mathbf{z}^t_{\text{coarse},i}, \mathbf{z}^t_{\text{fine},i})$ and $(\mathbf{z}^t_{\text{fine},i}, \mathbf{z}^t_{\text{coarse},i})$ are attracted to be close, while negative samples are repelled away from positive samples. Minimizing the (3) is usually understood as maximizing a lower bound on mutual information between the representations of positive samples [41].

The motivation behinds SC network is that, by pre-training an encoder capable of capturing the spatial-correlations between the same flow maps with different granularity, the trained encoder should help the model overcoming the problem of overfitting and improve the urban flow inference performance, especially when training data are few (which is critical for reducing the deployment/maintenance costs of transportation monitoring system).

B. Temporal-Contrasting Pre-Training Network

Now we introduce how to model the temporal correlations between flow maps in the representation space. Different from the spatial-contrasting (SC) network that uses two distinct flow map encoders, the encoders in temporal-contrasting (TC) network share their parameters since they only encode the coarse-grained flow maps at different times. The architecture of TC network is similar to SC network, the temporal-contrasting encoder $\mathbf{Enc}^{\text{temporal}}_{\text{coarse}}(\cdot)$ consists of two convolutional layers followed by ReLU activation, along with the global average pooling layer and dense layer which project the low-level temporal hidden feature maps $\mathbf{H}^{t,\text{temporal}}_{\text{coarse}} \in \mathbb{R}^{H\times W\times C}$ to a high-level flow map representation $\mathbf{z}^{t,\text{temporal}}_{\text{coarse}} \in \mathbb{R}^{SC}$ for temporal contrasting pretext task.

In the TC network, there are multiple choices to determine the temporal correlated flow maps, e.g., given a flow map $\mathbf{X}^t_{\text{coarse}}$ observed during time $[t-T, t]$, its temporal neighboring flow maps $\mathbf{X}^{t-T}_{\text{coarse}}$ and $\mathbf{X}^{t+T}_{\text{coarse}}$ can be both considered as positive samples. The periodic flow map is another type of positive samples, e.g., the flow map at the same time of the next day or of the next week. These time-close flow maps or periodic flow maps inherently exhibit similar urban flow dynamics under the regular situations.

Considering a set of temporally correlated flow maps $\mathcal{T} = \{\mathbf{X}^t_{\text{coarse}}\} \cup \{\mathbf{X}^\tau_{\text{coarse}}\}_\tau$, whenever $|\mathcal{T}| = 2$ the loss function (3) can be readily used. However, when $|\mathcal{T}| > 2$, a new loss function should be defined, supporting more than two positive samples in a mini-batch for the TC network. We generalize (3) to support arbitrary number of positive samples.

Specifically, given a batch of B urban flow maps in a pre-training dataset, for an anchor flow map $\mathbf{X}^t_{\text{coarse},i}$, we have $\mathcal{T}-1$ positive samples and $B - |\mathcal{T}|$ negative samples w.r.t. the anchor sample. Then we design the following loss to optimize the TC network:

$$\mathcal{L}_{\text{TC},i} = \frac{-1}{|\mathcal{T}'|}\sum_{\tau}^{\mathcal{T}'} \log \frac{\exp\left(\text{sim}(\mathbf{z}^t_{\text{coarse},i}, \mathbf{z}^\tau_{\text{coarse},i})/\theta\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp\left(\text{sim}(\mathbf{z}^t_{\text{coarse},i}, \mathbf{z}_{k})/\theta\right)} + \frac{-1}{|\mathcal{T}'|}\sum_{\tau}^{\mathcal{T}'} \log \frac{\exp\left(\text{sim}(\mathbf{z}^\tau_{\text{coarse},i}, \mathbf{z}^t_{\text{coarse},i})/\theta\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp\left(\text{sim}(\mathbf{z}^\tau_{\text{coarse},i}, \mathbf{z}_{k})/\theta\right)}, \quad (4)$$

where $\mathcal{T}' = \{\mathbf{X}^\tau_{\text{coarse}}\}_\tau$ (for brevity, we omit the $^{\text{temporal}}$ superscript of $\mathbf{z}^{t,\text{temporal}}_{\text{coarse}}$). The final loss is computed by $\mathcal{L}_{\text{TC}} = \frac{1}{2B}\sum_{i=1}^{B}\mathcal{L}_{\text{TC},i}$.

The intuition behind this loss function is that for multiple temporally correlated flow maps, their feature map projections should also be correlated (close) in the latent space.

C. Coupled Fine-Tuning Network for STCF

After the two pre-training networks pre-trained, we have two encoders in hand for downstream fine-tuning stage, i.e., the spatial contrasting encoder $\mathbf{Enc}^{\text{spatial}}_{\text{coarse}}(\cdot)$ and temporal contrasting encoder $\mathbf{Enc}^{\text{temporal}}_{\text{coarse}}(\cdot)$. We now introduce STCF's coupled fine-tuning network, which fuses the feature maps learned from two pre-training networks.

In the fine-tuning stage, the two encoders are directly used for coarse-grained flow map feature learning. We then build another encoder $\mathbf{Enc}_{\text{coarse}}$ initialized from scratch for fine-tuning. This encoder takes the input $\widetilde{\mathbf{X}}^t_{\text{coarse}}$ combining flow maps and dense EIF maps (cf. next subection IV-D). The fine-grained flow map inference process is defined as:

$$\mathbf{H}^{t,\text{spatial}}_{\text{coarse}} = \mathbf{Enc}^{\text{spatial}}_{\text{coarse}}\left(\mathbf{X}^t_{\text{coarse}}\right), \tag{5}$$

$$\mathbf{H}^{t,\text{temporal}}_{\text{coarse}} = \mathbf{Enc}^{\text{temporal}}_{\text{coarse}}\left(\mathbf{X}^t_{\text{coarse}}\right), \tag{6}$$

$$\mathbf{H}^t_{\text{coarse}} = \mathbf{Enc}_{\text{coarse}}\left(\widetilde{\mathbf{X}}^t_{\text{coarse}}\right), \tag{7}$$

$$\mathbf{H}^t_{\text{fine}} = \mathbf{Dec}\left(\text{Concat}\left(\mathbf{H}^{t,\text{spatial}}_{\text{coarse}}, \mathbf{H}^{t,\text{temporal}}_{\text{coarse}}, \mathbf{H}^t_{\text{coarse}}\right)\right), \tag{8}$$

where $\text{Concat}(\cdot)$ is the concatenate operation, $\mathbf{Dec}(\cdot)$ is a decoder network, which starts with a convolutional layer (C channels and 3×3 kernel size), followed by another convolutional layer (S^2 C channels and 3×3 kernel size). Then the extracted feature maps are fed into a PixelShuffle layer [50] to upsample the coarse-grained feature map ($\in \mathbb{R}^{H \times W \times S^2\ C}$) to fine-grained feature map ($\in \mathbb{R}^{SH \times SW \times C}$). The decoder ends with a single channel convolutional layer with 3×3 kernel size.

D. External Influence Factor Aggregation

External influence factors, such as time, weather, and temperature, are of great importance for inferring the fine-grained flow maps. Prior works found that incorporating EIFs into flow inference can improve model's rebustness and performance [7], [8], [11], [13], [51].

Similar results can also be found in [8], [13]. The EIF aggregation module used in prior arts generally consists of three steps: (i) EIFs are fed into dense layers after pretreatment; (ii) obtained dense representations are reshaped into feature map or used in sub-pixel blocks for upsampling; and (iii) concatenation between flow map and feature map. By doing so, implicit spatial relationships of external influence factors with urban flow volumes are readily linked. Although existing EIF aggregation modules improved FUFI performance, they cannot fully exploit the spatial correlation between EIF and flow volume in finer level of granularity, i.e., the influence of EIF on individual map cells is neglected.

As shown in the top of Fig. 4, hidden EIF representations are directly transformed into single feature map, which ignores the rich cell-interactions and the influence of individual EIFs. Motivated by this, we propose a new EIF aggregation module.

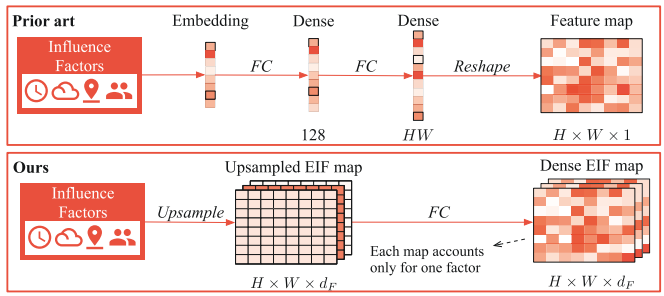

Fig. 4. Illustration of our external influence factor (EIF) aggregation module compared to prior arts [7], [8], [11], [13].

Given the coarse-grained flow map X^t_{coarse} and its corresponding EIF vector F^t which dimension is d_F, our module first expands the EIF vector to d_F-channels $H \times W$ feature maps. The expanded feature maps are then fed into full-connected layers to obtain the dense EIF maps $\mathbf{F}^t \in \mathbb{R}^{H \times W \times C/2}$. In this way, all EIFs are considered and modeled for each individual map cell. Furthermore, to strengthen the robustness of our model, we copy the flow map X^t_{coarse} for $C/2$ times, and apply a small Gaussian noise on all the replicas. Finally, we concatenate the Dense EIF maps and flow maps as the input of base encoder: $\widetilde{\mathbf{X}}^t_{\text{coarse}} \in \mathbb{R}^{H \times W \times C}$.

We note some of the previous models [7], [11] further upsample the EIF feature map to fine-granularity $\mathbf{F}^t_{\text{fine}} \in \mathbb{R}^{SH \times SW \times 1}$ which used in the later part of their models as an information highway to prevent information perishing. Next, we illustrate how to infer the final fine-grained flow map.

E. Inferring Fine-Grained Urban Flow Map

Base Encoder: sim used in [7], [11]. Fig. 5 provides a visual encoder comparison between UrbanFM, FODE, and STCF.

Distributional Upsampling: Prior FUFI methods often adopt this upsampling technique at the end of their networks [7], [11], which, the prediction of the network is not the exact fine-grained flow map but a flow distribution. We also use distributional upsampling (i.e., S^2-Normalization) in STCF: at the end of the encoder network. We take the final single channel feature map as input and then predict a distributional flow map where the values fall into the range $[0, 1]$. Specifically, a weighted flow distributional map is defined as:

$$w^t_{i'j'} = \frac{x^t_{i'j',\mathbf{H}}}{\sum_{\substack{i'' \in \left(\lfloor \frac{i'}{S} \rfloor S, (\lfloor \frac{i'}{S} \rfloor + 1)S\right] \\ j'' \in \left(\lfloor \frac{j'}{S} \rfloor S, (\lfloor \frac{j'}{S} \rfloor + 1)S\right]}} x^t_{i''j'',\mathbf{H}}}, \tag{9}$$

$$\hat{x}^t_{i'j',\text{fine}} = w^t_{i'j'} x^t_{ij,\text{coarse}}, \tag{10}$$

where $x^t_{i'j',\mathbf{H}}$ is the i'th row and j'th column cell in $\mathbf{H}^t_{\text{fine}}$, $w^t_{i'j'} \in [0, 1]$ is a weighted parameter for $x^t_{i'j',\text{fine}}$, $\hat{x}^t_{i'j',\text{fine}}$ is the predicted fine-grained flow volume, and $x^t_{ij,\text{coarse}}$ is the coarse-grained flow volume in $\mathbf{X}^t_{\text{coarse}}$, $i \in \lfloor i'/S \rfloor, j \in \lfloor j'/S \rfloor$. Now we have the inferred fine-grained urban flow map $\hat{\mathbf{X}}^t_{\text{fine}} = (x^t_{i'j',\text{fine}}) \in \mathbb{R}^{SH \times SW \times 1}$.

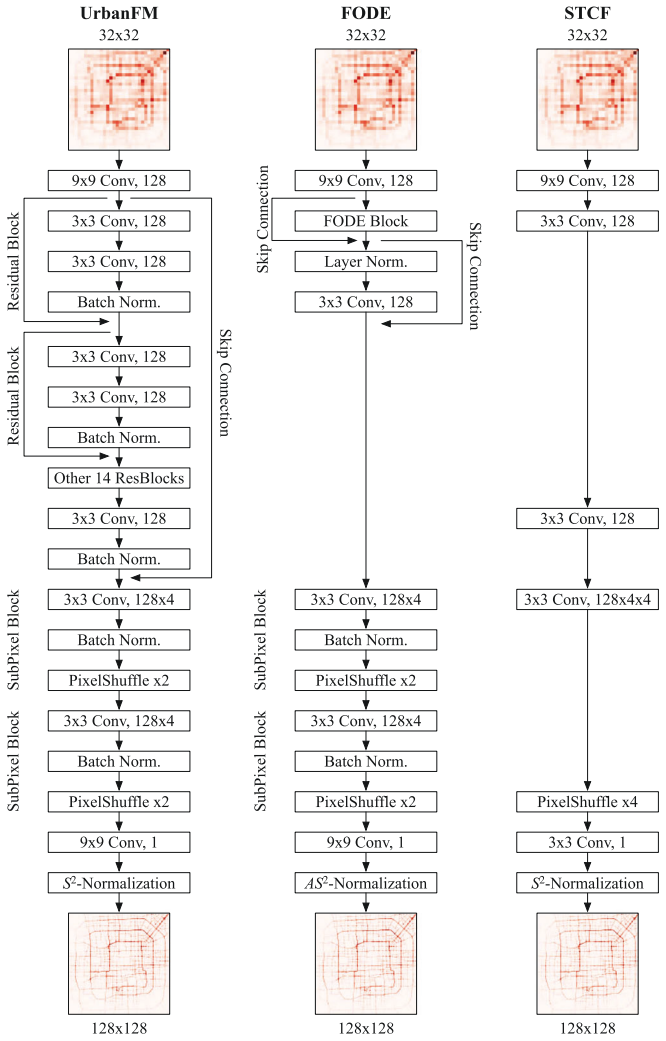

Fig. 5. Encoder network architecture comparison between UrbanFM [7], FODE [11], and STCF. Our proposed model does not have stacked residual blocks, skip connections, or FODE block, and only requires a single layer of PixelShuffle, resulting fewer parameters and better generalizability.

Optimization: The overall optimization of STCF is to minimize the mean squared error (MSE) between ground-truth flow map $\mathbf{X}^t_{\text{fine}}$ and the predicted flow map $\hat{\mathbf{X}}^t_{\text{fine}}$ at time t:

$$\mathcal{L}_{\text{STCF}}(\Theta) = \frac{1}{N}\sum_{t=1}^{N}\left\|\mathbf{X}^t_{\text{fine}} - \mathcal{F}_{\Theta}(\mathbf{X}^t_{\text{coarse}})\right\|^2, \qquad (11)$$

where N is the number of training coarse-grained flow maps, Θ denotes the learnable parameters.

V. Experiments

Now we present the experiments examining the inference performance of STCF compared with state-of-the-arts. Our experiments cover: (i) two large-scale urban flow datasets; (ii) two SISR baselines and three FUFI baselines; (iii) empirical evaluations conducted on different data fractions; (iv) ablation study, case study, and complexity analysis. For the ease of reproducing our results, the source code and datasets are publicly available at https://github.com/Xovee/stcf

A. Datasets

We perform experiments on two public urban flow datasets, including four taxi flow sub-datasets in Beijing City and one bike flow dataset in New York City. Detailed statistics of datasets are shown in Table II and Fig. 6.

- *TaxiBJ* datasets (https://github.com/yoshall/UrbanFM) are originally released in [7] containing urban taxi flow in Beijing within four different periods from 2013 to 2016. The resolution of flow map in TaxiBJ is 128 × 128, each cell in the flow map indicates the taxi flow volume in 30 minutes. The upscale factor S is 4, i.e., we use a downsampled 32 × 32 coarse-grained map to infer the 128 × 128 fine-grained map. The taxi flow maps are collected between 7 AM and 9 PM, and some extremely noisy data are removed [7].
- *BikeNYC* dataset is initially released by Citi Bike (https://citibikenyc.com/system-data) and processed by [11]. This dataset contains bike flow in New York City from Jan 1 to Mar 31, 2019. The resolution of flow map in BikeNYC is 80 × 32, each cell in the flow map indicates the bike flow volume in an hour. The upscale factor S is 2, i.e., we use a downsampled 40 × 16 coarse-grained map to infer the 80 × 32 fine-grained map.

Both datasets are associated with external influence factors which may affect the city flow in certain areas or at a certain time, e.g., *weather*, *date*, and *temperature*. Each dataset was divided into training (50%), validation (25%), and test (25%) sets. To evaluate the data-efficiency capabilities of our model and baselines, we use different training data percentages from 10% to 100% when conducting experiments.

B. Baselines

We compare STCF with the following six baselines, a comparison between them is shown in Table III.

- *Historical Average (HA):* is a simple baseline proportionally predicting the flow volume by its historical average.
- *ESPCN [50]:* is a real-time single image model which extracts feature maps in the low-resolution space and proposes an efficient sub-pixel convolution layer for aggregating features maps.
- *SRResNet [28]:* is a deep residual SISR model which stacks many residual blocks for image super-resolution.
- *UrbanFM [7]:* is the first work to study fine-grained urban flow inference (FUFI) problem. It proposes three key modules: a deep inference network to learn spatial-correlations, a distributional upsampling layer to impose spatial-constraint, and an EIF fusing subnet to improve FUFI performance.
- *FODE [11]:* leverages neural ordinary differential equations (ODE) for memory-efficient flow map feature learning, and proposes an augmented S^2-Normalization layer.
- *UrbanODE [13]:* introduces a pyramid attention network to infer high-quality flow maps based on neural ODEs.

TABLE II
STATISTICS OF TWO URBAN FLOW INFERENCE DATASETS

Dataset	TaxiBJ-P1	TaxiBJ-P2	TaxiBJ-P3	TaxiBJ-P4	BikeNYC
time range	Jul 1 - Oct 31, 2013	Feb 1 - Jun 30, 2014	Mar 1 - Jun 30, 2015	Nov 1 2015 - Mar 31 2016	Jan 1 - Mar 31, 2019
time interval	30 mins	30 mins	30 mins	30 mins	1 h
# maps	3,060	3,559	3,492	4,244	2,159
coarse map	32x32	32x32	32x32	32x32	40x16
fine map	128x128	128x128	128x128	128x128	80x32
latitude	39.82-39.99	39.82-39.99	39.82-39.99	39.82-39.99	40.65-40.81
longitude	116.26-116.49	116.26-116.49	116.26-116.49	116.26-116.49	74.00-74.07
avg flow volume	12.438	14.809	16.309	13.078	1.172
volume std dev	25.671	30.790	33.120	26.818	4.921

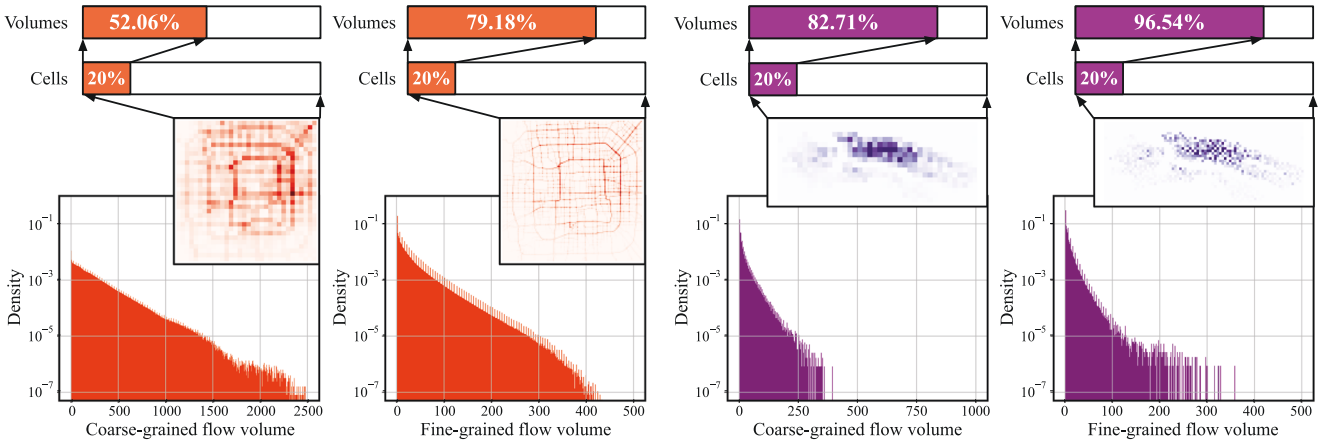

Fig. 6. Flow volume distributions for two datasets (left part is TaxiBJ P1 and right part is BikeNYC). Take fine-grained flow volumes in TaxiBJ dataset as an example, top 20% volumed map cells account for 79.18% total flow volumes.

TABLE III
BASELINE COMPARISON

Model	Backend	EIF	Tempo.	Channel	# Para
ESPCN [50]	CNN			768-384	2.7 M
SRResNet [28]	ResNet			128	6.1 M
UrbanFM [7]	ResNet	✓		128	6.2 M
FODE [11]	ODE	✓		128	6.9 M
UrbanODE [13]	ODE	✓		128	7.4 M
STCF (our)	CNN	✓	✓	128	4.6 M

C. Experimental Settings

Spatial Constraint: We note that for all SISR baselines, we add a S^2-Normalization layer at the end of their architectures to obey the spatial constraint of FUFI problem.

Configurations: For baseline and STCF configurations, to be fair, we uniformly adopt the following settings whenever possible. We use Adam optimizer, initial learning rate is $1e^{-4}$, batch size is 16, default number of base channels is 128 (equal to 4x model size). Except following specific configurations, we keep other hyper-parameters unchanged. For

- ESPCN, the number of channels of two convolution layers are set to 768 and 384, as suggested in [7].
- UrbanFM, we use the official PyTorch implementation (https://github.com/yoshall/UrbanFM), the number of residual blocks is 16.
- FODE [11] and UrbanODE [13], we implement their architectures by PyTorch (cf. https://github.com/Anewnoob/FODE), the number of DE block is 1, the ODESolver is Dopri5 numerical method. The θ in UrbanODE's PA block is embedded Gaussian and ρ is a linear embedding.

We train STCF as well as baselines on training set at most 1,000 epochs, and when validation loss is not declined for 100 consecutive epochs, we early stop the training process and report the inference performance on test set. The default scalar temperature θ is 0.1. For pre-training networks, the batch size is 32. Following [7], the values of max-scalar for coarse- and fine-grained flows in TaxiBJ datasets are set to 1,500 and 100, respectively. For BikeNYC, the max-scalar values are set to 400 and 25.

Computing Infrastructure: For STCF and baselines, we run the experiments on Ubuntu 20.04, 64 GB RAM, Intel Core$^{\text{TM}}$ i7-8700 K CPU, and single NVIDIA 1080 with 8 GB RAM. STCF is implemented with `TensorFlow 2.9`.

Metrics: Different from image super-resolution evaluating metrics, in FUFI problem the prediction target is flow volume of map cell rather than image pixel color – metrics such as PSNR and SSIM [29] are inappropriate here. We mainly use the mean squared error (MSE) for model evaluation, which is defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{X}^t_{\text{fine},i} - \hat{\mathbf{X}}^t_{\text{fine},i}\right\|^2, \tag{12}$$

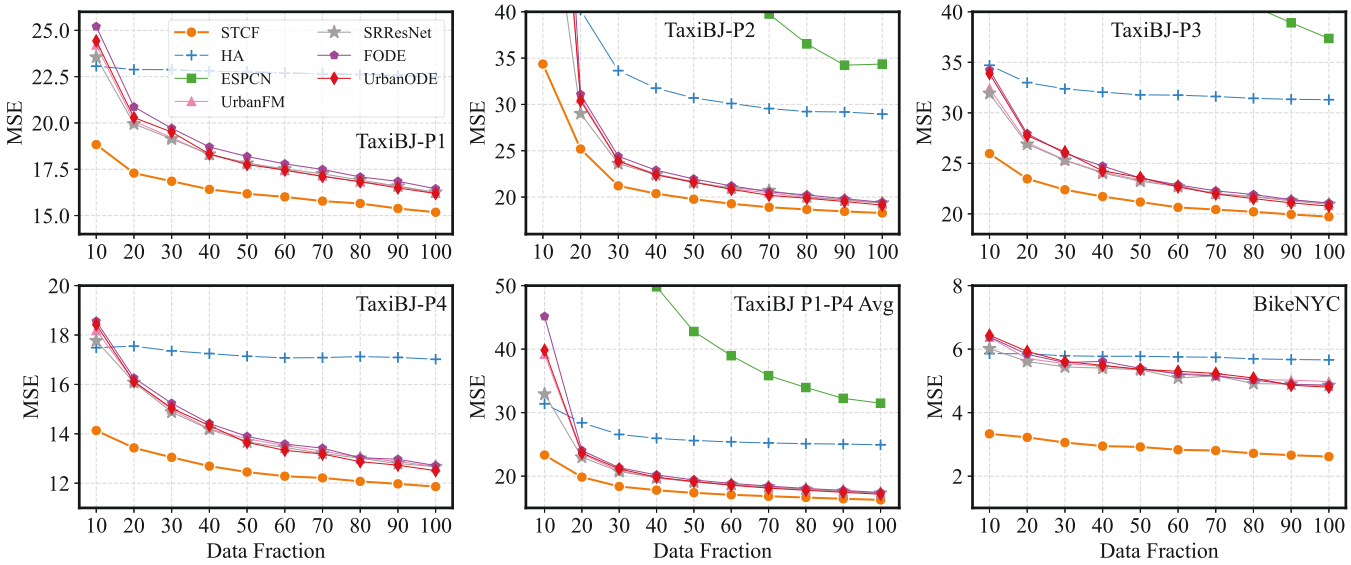

Fig. 7. Performance comparison between STCF and all baselines with different data fractions without using external influence factors (EIF). We note some results of ESPCN are not showed due to their significantly higher MSEs compared to other models.

where N is the number of test samples. We additionally include four more metrics for evaluation: mean absolute error (MAE), mean absolute percentage error (MAPE), mean squared logarithmic error (MSLE), and accuracy with 20% tolerance (ACC@20%), they are defined as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{X}^t_{\text{fine},i} - \hat{\mathbf{X}}^t_{\text{fine},i}\right\| \tag{13}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i-1}^{N}\left\|\frac{\mathbf{X}^t_{\text{fine},i} - \hat{\mathbf{X}}^t_{\text{fine},i}}{\mathbf{X}^t_{\text{fine},i}}\right\| \tag{14}$$

$$\text{MSLE} = \frac{1}{N}\sum_{i=1}^{N}\left\|\log \mathbf{X}^t_{\text{fine},i} - \log \hat{\mathbf{X}}^t_{\text{fine},i}\right\|^2 \tag{15}$$

$$\text{ACC@20\%} = \frac{100}{N}\sum_{i=1}^{N}\mathbb{1}\left(\left\|\frac{\mathbf{X}^t_{\text{fine},i} - \hat{\mathbf{X}}^t_{\text{fine},i}}{\mathbf{X}^t_{\text{fine},i}}\right\| \leq 0.2\right) \tag{16}$$

where $\mathbb{1}(\cdot)$ is indicator function. For MAPE and ACC@20%, we add 1 to all flow volumes to avoid division by zero.

D. Experimental Results and Analysis

We now compare our proposed STCF with baselines and report the experimental results. For fairness, we use two evaluation protocols: with or without external influence factors. The FUFI performances are shown in Figs. 7 and 8. Specifically, we have the following observations:

- Fig. 7 shows MSE on five datasets with different data fractions. We can observe that, STCF consistently and significantly outperform all other baselines. When comparing to UrbanODE, STCF achieves relative performance gains of 22.9%, 59.8%, 23.4%, 23.3%, and 48.2% on TaxiBj P1-P4 and BikeNYC datasets, respectively, with only 10% data. The improvements become larger when using fewer data, which indicates that STCF is data-efficient and greatly reduces the risk of overfitting problem when data are limited, hopefully saving the operation and maintenance costs of urban sensor equipment. This is in line with our motivation that contrastive pre-training and coupled fine-tuning can help learn robust map features and improve FUFI performance.
- For baselines, ESPCN is incapable of tackling FUFI problem and results in the worst performance. As a simple heuristic model, HA neglects temporal and dynamic characteristics of urban flows and only considers the average flow distribution, which, making it uncompetitive. FUFI baselines, powered with deep residual blocks or ODE solvers, outdo HA and ESPCN. However, their performances are rather similar. STCF distinguishes itself from competitors due to its advantages of learning expressive EIF and flow map features, and the ability to overcome overfitting and reduce training data demand.
- Fig. 8 shows the inference performances of STCF and FUFI baselines (UrbanFM, FODE, and UrbanODE) when using EIFs. Again, our proposed STCF greatly outperforms all the baselines on five datasets, up to 13.5%, 36.6%, 15.6%, 17.2%, and 46.8% relative MSE gains, respectively, compared to UrbanODE with only 10% data. The improvement gaps become smaller, this might be owning to the performance saturation.
- External influence factors generally improve the FUFI performance on all five datasets (as shown in Table IV), and the improvements are statistical significant with a level of $p < 0.05$ (student's t-test). EIFs have proven to be effective for inferring the urban flow map in fine-granularity. Specifically, our proposed new EIF aggregation module provides additional performance boost.
- For all datasets, FUFI models perform better on BikeNYC and worse on TaxiBj-P3. We speculate this is because the

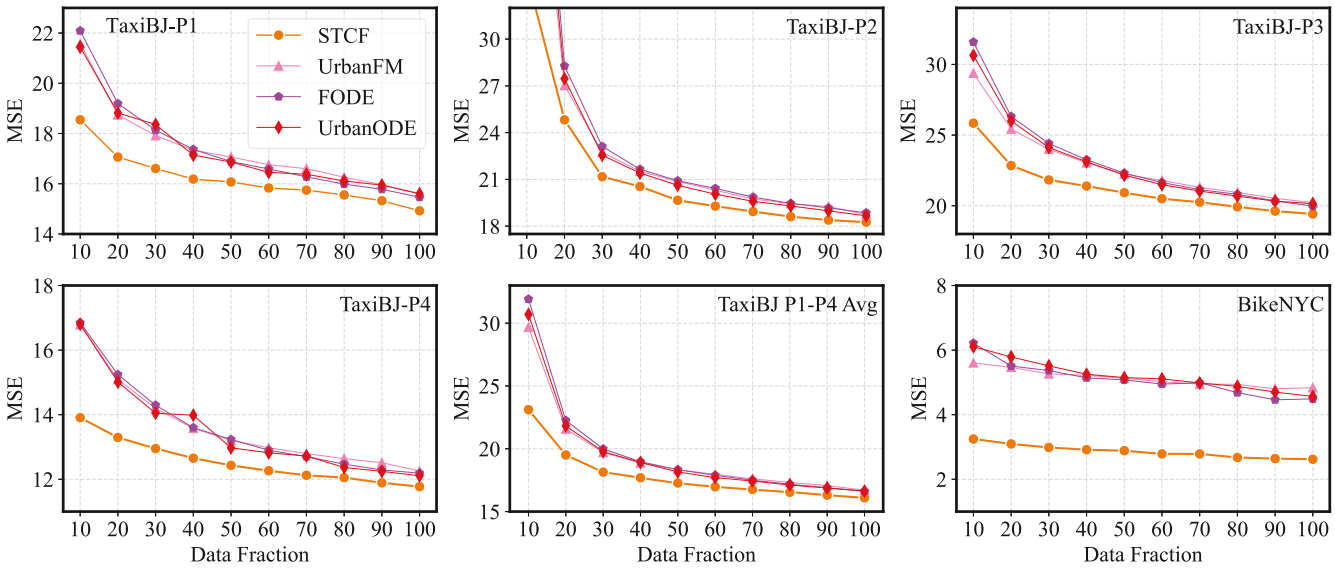

Fig. 8. Performance comparison between STCF and FUFI baselines with different data fractions. External influence factors (EIF) are used.

TABLE IV
EIF IMPROVES FUFI PERFORMANCE

Dataset (50% Data)	P1	P2	P3	P4	BikeNYC
UrbanFM w/o EIF	17.857 ±0.007	21.633 ±0.009	23.283 ±0.012	13.757 ±0.003	5.469 ±0.019
UrbanFM	**17.020*** ±0.004	**20.814*** ±0.014	**22.316*** ±0.010	**13.214*** ±0.006	**5.055*** ±0.013
UrbanODE w/o EIF	17.615 ±0.023	21.040 ±0.016	22.899 ±0.107	13.182 ±0.032	5.217 ±0.037
UrbanODE	**16.872*** ±0.109	**20.860*** ±0.012	**22.208*** ±0.021	**12.929*** ±0.031	**5.152*** ±0.054
STCF w/o EIF	16.119 ±0.023	20.103 ±0.030	21.114 ±0.032	12.482 ±0.024	2.937 ±0.025
STCF w/ prior EIF	16.093 ±0.036	19.871 ±0.100	20.945 ±0.037	12.432 ±0.026	2.899 ±0.032
STCF w/ new EIF	**16.024*** ±0.071	**19.720*** ±0.190	**20.875*** ±0.059	**12.423*** ±0.025	**2.887*** ±0.019

Experiments were conducted on 50% of training data. Each model was run five times and we report the mean MSE with standard deviation. Asterisk (*) indicates statistical significance with level $p < 0.05$.

average flow volume and variance of P3 are larger than others, making it hard for inference.

Overall, we conclude that STCF is data-efficient (generally requires 20–50% fewer data to obtain the same level of results), capable of addressing the overfitting problem, and sets a new state-of-the-art on fine-grained urban flow inference.

E. Ablation Study

- *Impact of model size:* Fig. 9 reports the inference performance under different model sizes (2×, 4×, 6×, 8×). Here, 2× denotes the model layers have 64 channels. In general, the best performance is achieved using 6× model size, further increasing the model size (8×) does not bring additional improvements. Notably, STCF 2× is better or comparable with all other baselines while also saving ∼90% model parameters (compared to UrbanODE 6x).
- *Impact of learning rate:* Fig. 10 records the performance and training loss of STCF using different learning rates. We can see that learning rate has a big impact on performance and $1e^{-4}$ clearly won out over others.

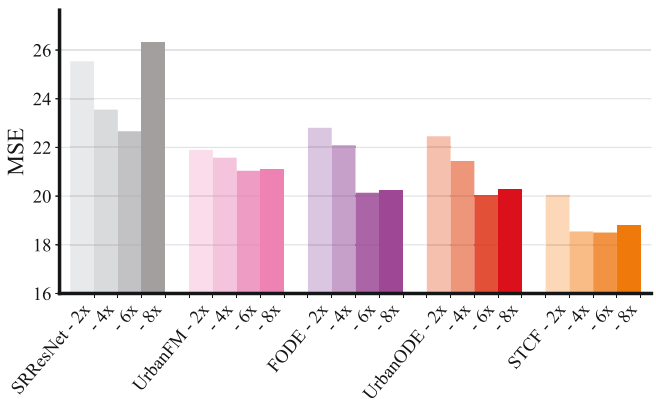

Fig. 9. Impact of model size of STCF and baselines on 10% TaxiBJ-P1 dataset. We use 4x as the default setting of model size in this article.

- *Impact of pre-training network and batch size:* We test the performance of spatial-contrasting network (SC), temporal-contrasting network (TC), Joint pre-training of the SC and TC, and STCF in the left of Fig. 11. We find that SC does better than TC. When training data are

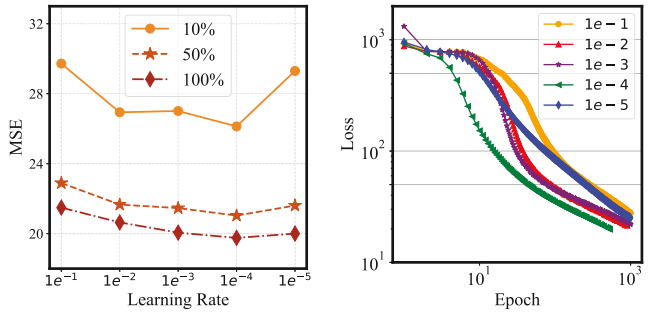

Fig. 10. Impact of learning rate with different data fractions on TaxiBJ-P3. Left: performance; Right: training losses (10% data).

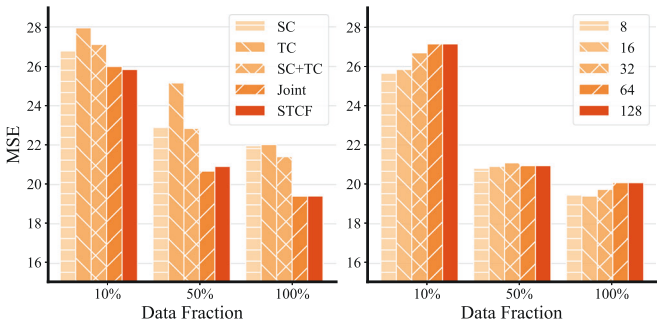

Fig. 11. Impact of pre-training network (left) and batch size (right) of STCF on TaxiBJ-P3 with different data fractions.

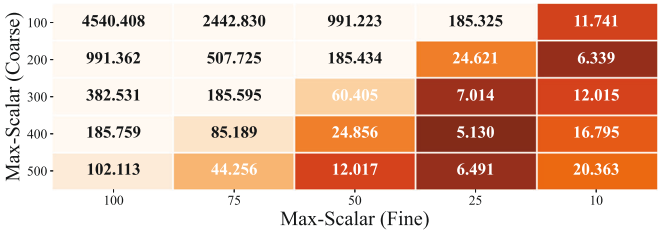

Fig. 12. Impact of max-scalar for both coarse- and fine-grained flow maps in BikeNYC dataset. We use UrbanFM trained on 50% data. UrbanFM without using max-scalar, i.e., set max-scalars to 1, results a poor MSE of 4540.

sufficient, the performance of TC catches up. Overall, Joint pre-training and fine-tuned STCF have the lowest and comparable MSEs. The right of Fig. 11 shows the performance of STCF under different batch size settings. We can see that a small batch size is sufficient for obtaining a good performance. Prior contrastive methods [20] value the importance of large model & batch sizes, we conjecture that such settings are not necessary for FUFI since the variances and information in the flow maps are less than that in large-scale image dataset.

- *Impact of max-scalar:* Since baselines (UrbanFM, FODE, UrbanODE) require an important hyper-parameter max-scalar as a normalization method to speed model training (make loss small enough), we ablate the impact of max-scalar for UrbanFM on BikeNYC dataset, the results are shown in Fig. 12. We found that max-scalars – whether used for coarse- or fine-grained flow map – greatly influenced the FUFI performance. We use the best-performed combination (400 and 25) throughout the paper. Notably, STCF does not need this hyper-parameter.
- *Impact of model parameters:* Now we analyze the space complexity of STCF. We report MSE in terms of model parameters in Fig. 13. It is easy to see that STCF significantly performs better than baselines within the level of model parameters. This mainly attributes to STCF's lightweight architecture, i.e., without complex stacked residual blocks, ODE solvers, and/or pyramid attention. We observe 4× is a better choice for model size in terms of *efficiency* versus *performance* trade-off. For a small model size (2×), the performance drops severely.

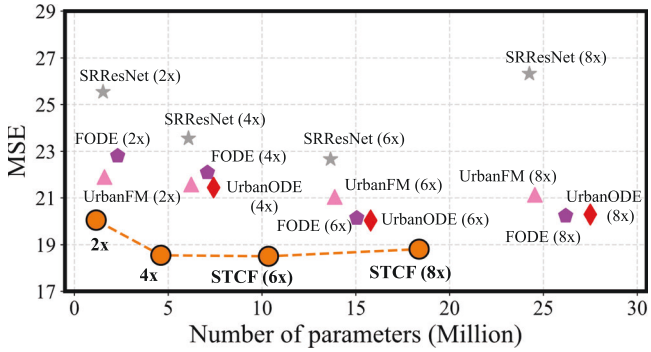

Fig. 13. Complexity analysis on model parameters. We report MSE in terms of model parameter on 10% TaxiBJ-P1.

TABLE V
RESULTS OF TRANSFER LEARNING (MSE)

Dataset	STCF	UrbanODE	FODE	UrbanFM
Original results of each model trained on 50% TaxiBJ P2/P3/P4.				
P2	**19.661** [-1.0]	20.624	20.914	20.873
P3	**20.923** [-1.2]	22.164	22.300	22.240
P4	**12.431** [-0.5]	12.973	13.230	13.207
All trained on 50% TaxiBJ-P1 and tested on P2/P3/P4.				
P2	**21.991** [-2.1]	24.113	23.979	24.061
	(+2.3)	(+3.5)	(+3.1)	(+3.2)
P3	**27.634** [-2.5]	30.134	29.960	30.294
	(+6.7)	(+8.0)	(+7.7)	(+8.1)
P4	**17.766** [-0.9]	18.620	18.668	18.938
	(+5.3)	(+5.6)	(+5.4)	(+5.7)

F. Results of Transfer Learning

The generalization capability of FUFI models is crucial for practical applicability. One of the most important goals of FUFI problem is to save the maintenance and electricity costs of urban sensing systems. On the one hand, the fine-grained flow maps become unavailable once the deployed sensors are closed or dismantled. On the other hand, re-training FUFI model requires new data, which might be expensive. Thus, training a generalized and efficient model that can be effectively transferred to other datasets is vital. We train our model and baselines on one dataset and test them separately on other datasets; The transferring results are shown in Table V. Green numbers in

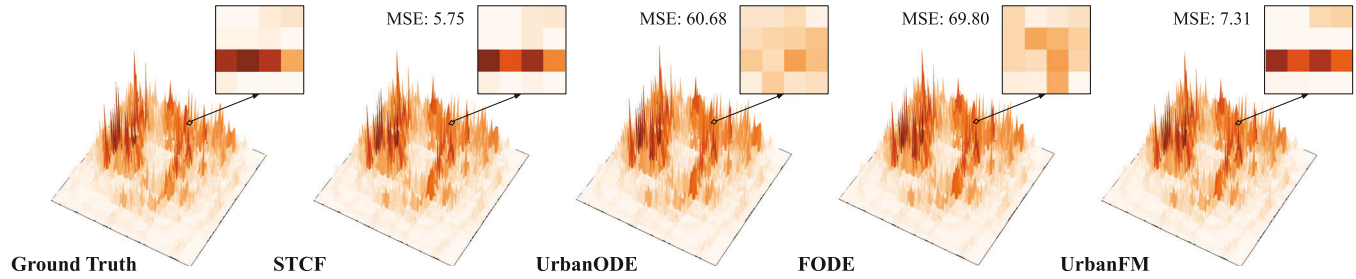

Fig. 14. Case Study: FUFI inference visualization of STCF and baselines on TaxiBJ P1 dataset. Darker (higher) cell indicates larger flow volume.

TABLE VI
MODEL COMPARISON IN TERMS OF MAE, MAPE, MSLE, AND ACC@20% ON 10% TRAINING DATA

Dataset	TaxiBJ P1				BikeNYC			
Metric	MAE	MAPE	MSLE	ACC@20%	MAE	MAPE	MSLE	ACC@20%
SRResNet	2.553	0.336	0.277	57.41%	0.738	0.197	0.146	80.93%
UrbanFM	2.585	0.367	0.290	55.70%	0.708	0.194	0.137	81.67%
FODE	2.551	0.345	0.280	56.38%	0.682	0.191	0.136	80.15%
STCF	**2.135**	**0.234**	**0.160**	**64.50%**	**0.479**	**0.127**	**0.073**	**83.43%**

brackets indicate performance improvement of STCF compared to the best baseline. Orange numbers in parentheses indicate performance change of current model compared to original result (train and test on the same dataset).

We can see that for all models, not surprisingly, their performances declined about 18–50%. For example, there is a 26.6% performance decrease for UrbanODE on P2. The deterioration becomes larger for later datasets. This points out that FUFI predictions for future flow maps become harder. In comparison to baselines, STCF effectively reduces the performance degradation, showing superior generalization capability.

G. Case Study

Here we present two case studies for better understanding the urban flow inference problem, our proposed model, and prediction interpretation.

1) Inference Error Visualization: To explore the effectiveness of STCF on FUFI inference, we visualize the inferred urban flow map as well as a 4×4 sample region in Fig. 14. We can observe that compared to ground truth, the prediction of STCF best approximates the true flow distribution. ODE-based models (UrbanODE and FODE) fail to reconstruct the flow distribution, resulting in the highest inference error. UrbanFM, in contrast, successfully predicts the high-volume cells. However, it uniformly treats the low-volume cells and thus loses information.

2) More Metrics: We provide additional experiments in Table VI, where we show the performance comparison of STCF with baselines on 10% TaxiBJ P1 and BikeNYC datasets, in terms of MAE, MAPE, MSLE, and ACC@20%. Different metrics have different emphases on the prediction. For example, MSE pays more attention to the performance of the model in regions with high flow volumes, while MAPE values the overall performance of the model in all regions. We can see that STCF significantly outperforms the baselines on all metrics.

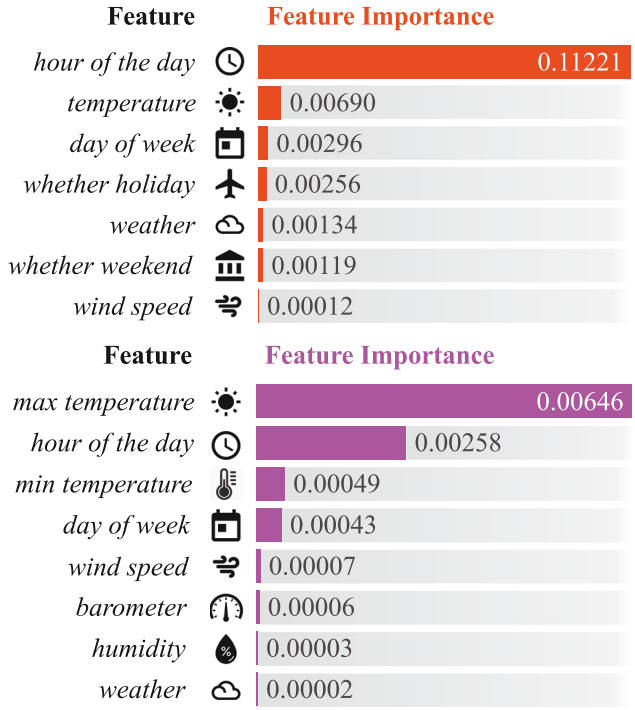

Fig. 15. Importance scores of external influence factors on two datasets. Top: TaxiBJ. Bottom: BikeNYC.

3) EIF importance/influence on Flows: We conduct random feature permutation [52] on external influence factors and report the mean performance change of STCF by 10 runs on the test set, the results are shown in Fig. 15, higher values represent higher importance. It was easy to see that *hour of the day* and *temperature* are two most important factors for STCF. For taxi flow in Beijing, the peak flow volumes occur during office hours. In New York, people are more likely to ride a bike

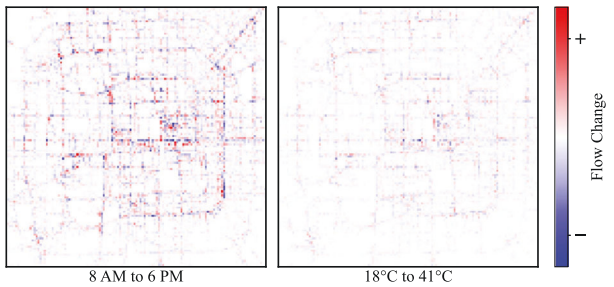

Fig. 16. Visualization: the flow volume distribution change by altering external influence factors on a sample map in TaxiBJ P1.

during commuting hours and are less likely to ride a bike at low temperatures. EIFs affect not only the total volumes of flows but also the distribution of flows. For example, when we manually alter the *hour of the day* and *temperature* values for a sample map in TaxiBJ, the flow distribution changes (cf. Fig. 16). Beyond our expectation, the influence of *weather* factors are minimal, which might be the cause of data sparseness. Overall, the improvements of the EIF module are not as significant as that brought by the contrasting networks. We speculate it is because: (i) the sparsity of the factors, e.g., some weather conditions rarely occurred; (ii) there may exist other EIFs that we did not take advantage of, e.g., traffic control signals and mega events such as football match and concert; (iii) EIFs are not involved in the pre-training stage; (iv) spurious correlations are learned during training. Data augmentation and feature engineering on EIFs may address these limitations and further improve the FUFI performance.

VI. Conclusion

We presented STCF, a novel spatial-temporal framework for data- and parameter-efficient fine-grained urban flow inference. STCF follows a pre-training & fine-tuning paradigm, upscaling the coarse-grained flow map by spatial-temporal contrastive pre-trainings. The pre-trained feature maps and external influence factors are effectively fused via a coupled fine-tuning network, yielding non-complex architecture and yet achieving significant performance improvements over the state-of-the-art.

Our future work will focus on designing more helpful pretext tasks and effective external influence factors, further improving STCF by explicitly modeling dynamical flow distributions by 3D convolution or sequential models, and transferring the learned knowledge to other types of urban flows and/or other cities [53], [54].

References

[1] A. Gharaibeh et al., "Smart cities: A survey on data management, security, and enabling technologies," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 2456–2501, Fourth Quarter 2017.

[2] Y. Liu, Y. Liang, K. Ouyang, S. Liu, D. S. Rosenblum, and Y. Zheng, "Predicting urban water quality with ubiquitous data-a data-driven approach," *IEEE Trans. Big Data*, vol. 8, no. 2, pp. 564–578, Apr. 2022.

[3] M. Zhang, T. Li, Y. Yu, Y. Li, P. Hui, and Y. Zheng, "Urban anomaly analytics: Description, detection, and prediction," *IEEE Trans. Big Data*, vol. 8, no. 3, pp. 809–826, Jun. 2020.

[4] G. D'Amico, P. L'Abbate, W. Liao, T. Yigitcanlar, and G. Ioppolo, "Understanding sensor cities: Insights from technology giant company driven smart urbanism practices," *Sensors*, vol. 20, no. 16, 2020, Art. no. 4391.

[5] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 1–55, 2014.

[6] K. Ouyang et al., "Fine-grained urban flow inference," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2755–2770, Jun. 2022.

[7] Y. Liang et al., "UrbanFM: Inferring fine-grained urban flows," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 3132–3142.

[8] K. Li, J. Chen, B. Yu, Z. Shen, C. Li, and S. He, "Supreme: Fine-grained radio map reconstruction via spatial-temporal fusion network," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2020, pp. 1–12.

[9] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3365–3387, Oct. 2021.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[11] F. Zhou, L. Li, T. Zhong, G. Trajcevski, K. Zhang, and J. Wang, "Enhancing urban flow maps via neural ODEs," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 1295–1302.

[12] Y. Liang, K. Ouyang, H. Yan, Y. Wang, Z. Tong, and R. Zimmermann, "Modeling trajectories with neural ordinary differential equations," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1498–1504.

[13] F. Zhou, X. Jing, L. Li, and T. Zhong, "Inferring high-resolutional urban flow with internet of mobile things," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 7948–7952.

[14] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[15] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654.

[16] Y. Liang et al., "Revisiting convolutional neural networks for citywide crowd flow analytics," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2020, pp. 578–594.

[17] X. Yang, X. He, Y. Liang, Y. Yang, S. Zhang, and P. Xie, "Transfer learning or self-supervised learning? a tale of two pretraining paradigms," 2020, *arXiv:2007.04234*.

[18] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4182–4192.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[20] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 22243–22255.

[21] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[22] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[23] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.

[24] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1422–1430.

[25] R. D. Hjelm et al., "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Representations*, 2018.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[27] J. Liu, J. Tang, and G. Wu, "Residual feature distillation network for lightweight image super-resolution," in *Proc. Eur. Conf. Comput. Vis. AIM Workshop*, 2020, pp. 41–55.

[28] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[30] G. Jin, Y. Liang, Y. Fang, J. Huang, J. Zhang, and Y. Zheng, "Spatio-temporal graph neural networks for predictive learning in urban computing: A survey," 2023, *arXiv:2303.14483*.
[31] J. Li, S. Wang, J. Zhang, H. Miao, J. Zhang, and P. Yu, "Fine-grained urban flow inference with incomplete data," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5851–5864, Jun. 2023.
[32] T. Zhong, H. Yu, R. Li, X. Xu, X. Luo, and F. Zhou, "Probabilistic fine-grained urban flow inference with normalizing flows," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 3663–3667.
[33] X. Xu, Y. Wei, P. Wang, X. Luo, F. Zhou, and G. Trajcevski, "Diffusion probabilistic modeling for fine-grained urban traffic flow inference with relaxed structural constraint," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
[34] H. Yu, X. Xu, T. Zhong, and F. Zhou, "Overcoming forgetting in fine-grained urban flow inference via adaptive knowledge replay," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 5393–5401.
[35] X. Xu, Z. Wang, F. Zhou, Y. Huang, T. Zhong, and G. Trajcevski, "Dynamic Transformer ODEs for large-scale reservoir inflow forecasting," *Knowl.-Based Syst.*, vol. 276, 2023, Art. no. 110737.
[36] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6527–6583.
[37] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Jan. 2023.
[38] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Representations*, 2018.
[39] Y. Liu, S. Pan, M. Jin, C. Zhou, F. Xia, and P. S. Yu, "Graph self-supervised learning: A survey," 2021, *arXiv:2103.00111*.
[40] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 297–304.
[41] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv: 1807.03748*.
[42] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1735–1742.
[43] X. Liu, Y. Liang, Y. Zheng, B. Hooi, and R. Zimmermann, "Spatio-temporal graph contrastive learning," 2021, *arXiv:2108.11873*.
[44] X. Xu, F. Zhou, K. Zhang, and S. Liu, "CCGL: Contrastive cascade graph learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4539–4554, May 2023.
[45] F. Zhou, P. Wang, X. Xu, W. Tai, and G. Trajcevski, "Contrastive trajectory learning for tour recommendation," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 1, pp. 1–25, 2021.
[46] Q. Gao, J. Hong, X. Xu, P. Kuang, F. Zhou, and G. Trajcevski, "Predicting human mobility via self-supervised disentanglement learning," 2022, *arXiv:2211.09625*.
[47] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *Proc. Int. Conf. Learn. Representations*, 2019, Art. no. 4.
[48] Y. Yang, J. Hou, and Y. Xu, "Super resolution deduction: Inferring fine-grained capacity for urban signal station deployment," *IEEE Access*, vol. 9, pp. 23335–23343, 2021.
[49] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv: 2003.04297*.
[50] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.
[51] K. Li et al., "Model and transfer spatial-temporal knowledge for fine-grained radio map reconstruction," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 828–841, Jun. 2022.
[52] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," *J. Mach. Learn. Res.*, vol. 20, no. 177, pp. 1–81, 2019.
[53] Q. Gao, Z. Luo, D. Klabjan, and F. Zhang, "Efficient architecture search for continual learning," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2022.3151511.
[54] S. Wang, H. Miao, J. Li, and J. Cao, "Spatio-temporal knowledge transfer for urban crowd flow prediction via deep attentive adaptation networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4695–4705, May 2022.

Xovee Xu (Graduate Student Member, IEEE) received the BS and MS degrees in software engineering from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 2018 and 2021, respectively. He is currently working toward the PhD degree in computer science. His research focuses on understanding spatial-temporal data, information diffusion, user-generated content, and human social behaviors.

Zhiyuan Wang received the BS and MS degree in software engineering from the University of Electronic Science and Technology of China. He is currently working toward the doctoral student in computer engineering in Texas A&M University. His current research interests include neural network, data mining, time series forecasting, and deep generative models.

Qiang Gao received the PhD degree in Software Engineering from University of Electronic Science and Technology of China (UESTC), in 2020. He is currently an associate professor with the Southwestern University of Finance and Economics. He was a visiting scholar at the Northwestern University, supervised by Dr. Diego Klabjan and Dr. Goce Trajcevski, during 2019-2020. His current research interests include spatio-tempoeal data mining and deep learning. He currently serves as the PC member or reviewer in several international conferences and journals, e.g., TKDE, TNNLS, KDD, ACM SIGSPATIAL, GeoInformatica.

Ting Zhong received the BS degree in computer application and the MS degree in computer software and theory from Beijing Normal University, Beijing, China, in 1999 and 2002, respectively, and the PhD degree in information and communication engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009. She is currently a professor of UESTC. Her current research interests include deep learning, social networks, and cloud computing.

Bei Hui received the PhD degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2009. He is now an associate professor with the School of Information and Software Engineering, UESTC. His research interests include machine learning and knowledge reasoning.

Fan Zhou (Member, IEEE) received the BS degree in computer science from Sichuan University, China, in 2003, and the MS and PhD degrees from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2006 and 2011, respectively. He is currently a full professor with the School of Information and Software Engineering, UESTC. His research interests include machine learning, neural networks, spatio-temporal data management, recommender systems, and social network mining.

Goce Trajcevski (Member, IEEE) received the BSc degree in informatics and automation from the University of Sts. Kiril i Metodij, Skopje, North Macedonia, in 1989, and the MS and PhD degrees in computer science from the University of Illinois at Chicago, Chicago, IL, USA, in 1995 and 2002, respectively. He is currently a Kingland associate professor with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. His research has been funded by the NSF, ONR, BEA, and Northrop Grumman Corporation. In addition to a book chapter and three encyclopedia chapters, he has coauthored more than 220 publications in refereed conferences and journals. His main research interests are in the areas of spatiotemporal data management, uncertainty and reactive behavior management in different application settings, and incorporating multiple contexts. He was the general co-chair of the IEEE International Conference on Data Engineering, 2014 and ACM SIGSPATIAL 2019, the PC Co-Chair of the ADBIS 2018, ACM SIGSPATIAL 2016 and 2017, and IEEE MDM 2023. He has served in various roles in organizing committees in numerous conferences and workshops. He is an associate editor of the ACM Transactions on Spatial Algorithms and Systems and the Geoinformatica Journals.