Enhancing Fine-Grained Urban Flow Inference via Incremental Neural Operator

Qiang Gao[1], **Xiaolong Song**[1], **Li Huang**[1]*, **Goce Trajcevski**[2], **Fan Zhou**[3] and **Xueqin Chen**[4]
[1]Southwestern University of Finance and Economics, Chengdu, China, 611130
[2]Iowa State University, Iowa, USA
[3]University of Electronic Science and Technology of China, Chengdu, China, 610054
[4]Delft University of Technology, Delft, Netherlands, 2628CN
qianggao@swufe.edu.cn, 223081200028@smail.swufe.edu.cn, lihuang@swufe.edu.cn, gocet25@iastate.edu, fan.zhou@uestc.edu.cn, nedchen0728@gmail.com

Abstract

Fine-grained urban flow inference (FUFI), which involves inferring fine-grained flow maps from their coarse-grained counterparts, is of tremendous interest in the realm of sustainable urban traffic services. To address the FUFI, existing solutions mainly concentrate on investigating spatial dependencies, introducing external factors, reducing excessive memory costs, etc., – while rarely considering the *catastrophic forgetting* (CF) problem. Motivated by recent operator learning, we present an **U**rban **N**eural **O**perator solution with **I**ncremental learning (**UNOI**), primarily seeking to learn grained-invariant solutions for FUFI in addition to addressing CF. Specifically, we devise an urban neural operator (UNO) in UNOI that learns mappings between approximation spaces by treating the different-grained flows as continuous functions, allowing a more flexible capture of spatial correlations. Furthermore, the phenomenon of CF behind time-related flows could hinder the capture of flow dynamics. Thus, UNOI mitigates CF concerns as well as privacy issues by placing UNO blocks in two incremental settings, i.e., flow-related and task-related. Experimental results on large-scale real-world datasets demonstrate the superiority of our proposed solution against the baselines.

1 Introduction

In the development of smart cities [Zheng *et al.*, 2014], studying fine-grained urban flow is crucial for providing accurate warnings in specific traffic situations such as congestion [Zhou *et al.*, 2020; Liang *et al.*, 2022; Yu *et al.*, 2023]. However, obtaining data at a spatially fine-granularity (FG) is not conducive to urban environments, as it requires deploying a (prohibitively) large number of sensing devices to cover a citywide landscape [Liang *et al.*, 2019]. In contrast, acquiring coarse-grained (CG) data is much simpler. Hence, fine-grained urban flow inference (FUFI) from CG observations has become an essential task for urban planners. In practice, solving FUFI can significantly alleviate the burden associated with deploying massive piezoelectric sensors and loop detectors on road segments.

Typically, FUFI can be viewed as a mapping problem that transforms data from a space with low information content into a more informative space [Liang *et al.*, 2019; Ouyang *et al.*, 2020]. Motivated by the successful methods of image super-resolution (SR) in computer vision (CV) [Dong *et al.*, 2015; Lim *et al.*, 2017], the first convolutional neural network (CNN)-based FUFI framework named UrbanFM was proposed in [Liang *et al.*, 2019], with a primary focus on addressing spatial correlations between coarse-grained and fine-grained maps as well as the complexities introduced by external factors. Grounded in the concept of UrbanFM, recent works enhanced FUFI accuracy by exploring spatial dependencies [Liang *et al.*, 2020; Zhou *et al.*, 2021], inferring on a large scale [Ouyang *et al.*, 2020], and mitigating catastrophic forgetting [Yu *et al.*, 2023]. Although these novel methods have achieved remarkable results, there are still some challenges that have not been resolved yet.

In practice, urban flow maps usually exhibit intricate patterns and structures with varying complexities at different scales. Existing efforts are based on the CG map and target FG map with the view of super-resolution rules, directly learning a mapping function between them, which is naturally scale sensitive [Liang *et al.*, 2019; Zhou *et al.*, 2021]. Since these models lack affluent exploration and adaptation to diverse scales, this hinders the models' ability to capture nuanced spatial relationships and patterns present in urban flow maps across various scales, resulting in the learned neural network not generalizing well to different scales of maps beyond the map size of the training data and even yielding unstable training. Some recent ODE-based solutions (e.g., FODE [Zhou *et al.*, 2020]) can partially solve the instability problem in FUFI – however, by relying on overly complex function solving, we find that they still face the gradient-exploding problem. We consider that exploring the learning of scale-invariant mappings with rich structures and patterns within spatial domains from a more flexible space rather than directly from the data space itself is worthy of investigation.

In addition, a FUFI system usually needs to handle sequentially arriving datasets/tasks. Prior studies primarily strive to either build an isolation model – which, essentially, will retrain the whole model with a newly acquired FUFI dataset; or rely on a fine-tuned approach, – which is to continually

*Corresponding author (Li Huang)

update the parameters of the model trained on the previous dataset. However, recursively retraining the model for each newly coming dataset leads to a lack of insights from the previously learned knowledge, i.e., catastrophic forgetting (CF) [Kirkpatrick *et al.*, 2017]. In fine-tuning manners, one could initially preserve old knowledge from the learned model, which can further boost the learning of the new task. However, due to the back-propagation impact, the old knowledge would also be forgotten with the constant process of fine-tuning. [Yu *et al.*, 2023] follows the rule of continual learning and builds a replay buffer to maintain the old knowledge behind the retained samples from the old datasets, providing a new perspective that the old knowledge can enhance the learning of newly arrived tasks/datasets. Nevertheless, we argue that such a manner could be impractical in various real-world FUFI contexts. For instance, due to privacy concerns, we could not have the right to access the previous datasets after model training, failing to bind/transfer old knowledge to new tasks. Hence, how to tackle the privacy concern in addition to CF becomes another urgent issue in the FUFI problem.

To remedy the above concerns, this study presents a novel **U**rban **N**eural **O**perator solution with **I**ncremental learning (UNOI). Specifically, we draw inspiration from the discretization-invariant capabilities of recent neural operators (NO) and their successful application in super-resolution tasks [Wei and Zhang, 2023], and propose an Urban Neural Operator (**UNO**) that extends NO to capture *grained-invariant features* from the original CG flow map as a supplementary for the input of the urban flow inference module. Unlike the original Neural Operator (NO) methods, UNO focuses on learning mappings between two approximation spaces of continuous functions. In addition, with the CF issue in mind, we operate our UNO in an incremental learning manner with two incremental settings, primarily seeking to transfer old knowledge to the new task/dataset. Specifically, to consider the time-dependent relationship between neighboring temporal flows, we propose a flow-related incremental enhancement method for individual task learning, aiming at mitigating the CF problem in a single task. To address the concerns related to data privacy when handling a series of FUFI tasks/datasets, we propose a task-related incremental enhancement method for sequential task learning. In this procedure, we do not need to build additional replay buffers to enforce the model to remember the old knowledge. Instead, our solution is a data-free paradigm, which treats the earlier trained model as the old knowledge carrier rather than retaining connections to real-world old samples. Our main contributions are summarized as follows:

- We devise an urban neural operator (UNO) to generate grained-invariant solutions for addressing the FUFI task by following the rule of recent operator learning, yielding a more reliable and effective manner of flow inference.
- We extend UNO to UNOI with two incremental enhancement methods to address the catastrophic forgetting problem as well as privacy concerns. To the best of our knowledge, this is the first attempt to alleviate the CF issue with a data-free view in handling a series of FUFI tasks.
- We conduct extensive experiments on four real-world datasets/tasks to demonstrate that UNOI not only outperforms the baselines but also offers a more date-free manner to mitigate the CF problem in sequential task learning.

2 Preliminaries and Problem Statement

We first introduce the technical foundation of neural operators (NO), followed by the specific problem that UNO aims to address, and end with the task statement of FUFI.

Neural Operators. Neural operators (**NO**) are designed to learn mappings between infinite-dimensional function spaces based on a finite set of observed input-output pairs, which are widely employed to generate discretization-invariant solutions for a family of partial differential equations (PDEs) [Li *et al.*, 2020c; Li *et al.*, 2020b; Li *et al.*, 2020a; Kovachki *et al.*, 2023]. Suppose we have: (1) $\mathbb{R}^{d_a}$-valued input functions $a \in \mathcal{A}$ defined on a bounded domain $D_a \subset \mathbb{R}^{d_a}$; and (2) output functions $u \in \mathcal{U}$ which are $\mathbb{R}^{d_u}$-valued and defined on a bounded domain $D_u \subset \mathbb{R}^{d_u}$. Building on the concept of the Green's Function [Greenberg, 2015], the general formula for the NO framework can be described as follows:

$$h_0(x) = \mathcal{L}(a(x)), \tag{1}$$

$$h_{l+1}(x) = \sigma(W_l h_l(x) + \int_{D_l} k_\phi^{(l)}(x, y) h_l(y) dy), \tag{2}$$

$$u(x) = \mathcal{P}(h_L(x)). \tag{3}$$

$\mathcal{L} : \mathbb{R}^{d_a} \mapsto \mathbb{R}^{d_{h_0}}$ in Eq. (1) represents the *Lifting* mapping, which aims at mapping the input function to its first hidden representation: $\{a : D \to \mathbb{R}^{d_a} \mapsto h_0 : D \to \mathbb{R}^{d_{h_0}}\}$. Typically, d_{h_0} is greater than d_a, indicating that this operation serves as a channel expansion step. The *Iterative Kernel Integration* in Eq. (2) involves a local linear operator $W_l \in \mathbb{R}^{d_{h_{l+1}} \times d_{h_l}}$ and a *Kernel Integral* operator $(\mathcal{K}_l(h_l))(x) = \int_{D_l} k_\phi^{(l)}(x, y) h_l(y) dy$, used to map each hidden representation to the next one: $\{h_l : D_l \to \mathbb{R}^{d_{h_l}} \mapsto h_{l+1} : D_{l+1} \to \mathbb{R}^{d_{h_{l+1}}}\}$, while maintaining spatial continuum. σ denotes the activation function. $\mathcal{P}$ in Eq. (3) denotes a point-wise function for *Projecting* the final hidden representation to the output function: $\{h_L : D' \to \mathbb{R}^{d_{h_L}} \mapsto u : D' \to \mathbb{R}^{d_u}\}$.

Problem Statement. A coarse-grained flow map $\boldsymbol{X}$ with size $H \times W$ can be defined as a vector-valued function v in Hilbert space $\mathcal{H}$ with bounded domain $D_{H \times W} \to \mathbb{R}_+$. To obtain the fine-grained flow map $\boldsymbol{\mathcal{X}}$, UNO aims to learn an urban neural operator $\mathcal{G}_\Theta$ between two $\mathcal{H}$-spaces associated with coarse-grained ($D_{I \times J}$) and its arbitrary fine-grained ($D_{NI \times NJ}$) domains, respectively:

$$\mathcal{G}_\Theta : \mathcal{H}(D_{I \times J}) \mapsto \mathcal{H}(D_{NI \times NJ}), \tag{4}$$

where $N \in \mathbb{Z}_+$ is upscaling factor. We can access the cell-wise function values of v at the coordinates (cells) $\{x_i\}_{i=1}^{H \times W} \subset D$, where $x_i \in \boldsymbol{X}$. Typically, we can regard cell flow x_i as "snapped" at the central point of a cell in a grid-based map [Hershberger, 2013]. By doing so, we transfer the infinite-dimensional function mapping problem to learn the mapping between two approximation spaces with finite-dimension and continuous functions:

$$\mathcal{G}_\Theta : \{v(x_i)\}_{i=1}^{I \times J} \mapsto \{u(\mathsf{x}_j)\}_{j=1}^{NI \times NJ}, \tag{5}$$

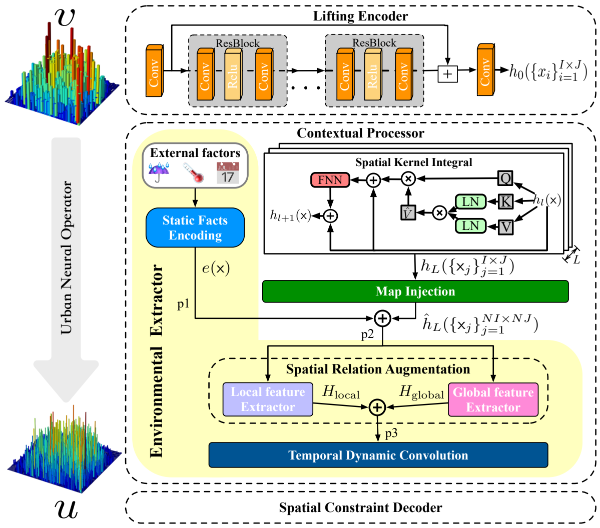

Figure 1: Overview of UNO.

where $\mathsf{x}_j \in \mathcal{X}$ and the vector-valued functions of CG and FG spaces are $v \in \mathcal{H}(D_{I\times J})$ and $u \in \mathcal{H}(D_{NI\times NJ})$, respectively.

Task Statement. In traditional approaches to solving the FUFI problem [Liang *et al.*, 2020; Zhou *et al.*, 2020; Liang *et al.*, 2021], the learning model commonly relies on complete urban flow data. For example, we could obtain years of urban flows and feed them into the learning system all at once. As claimed in [Yu *et al.*, 2023], such a paradigm belongs to offline learning. In contrast, we focus on solving the FUFI problem in an incremental learning manner, assuming the previously used flow data is not available due to privacy concerns. Formally, given a sequence of chronological urban flow datasets, we learn new (e.g., year t) knowledge incrementally with the help of old (e.g., year $(t-1)$) knowledge, without having to revisit the actual previous urban flows, where handling each year's flow maps can be treated as an individual task (e.g., task t).

3 Urban Neural Operator

As Fig.1 shows, while keeping the neural operator in mind, our UNO adopts an *encoder-processor-decoder* architecture. The encoder (i.e., **Lifting Encoder**) corresponds to the *Lifting* mapping; the processor (i.e., **Contextual Processor**) encompasses *Interactive Kernel Integration* and additional *Environmental Extractor*; and the decoder (i.e., **Spatial Constraint Decoder**) is responsible for simply getting us back to the space of the output function – all of which we detail next.

3.1 Lifting Encoder (LE)

In contrast to the traditional neural operators from PDEs that employ a simple function such as a multi-layer perception (MLP) to lift the input scale, our devised LE $\mathcal{E}_\psi$ in UNO is responsible for lifting the input space into a higher-dimensional latent space, which helps the model capture structural interactions and *basis functions* from the coarse-grained flow maps, enhancing the capacity to handle complexly implicit interactions behind the urban flow volumes. For LE architecture, we employ two CNN layers at the head and tail of $\mathcal{E}_\psi$, sequentially connected by several residual blocks. Each residual block comprises two CNN layers connected by an activation layer, and the input features are added to the final output. In this manner, increasing layers rather than feature channels not only enhances model performance but also reduces memory cost(cf. [Lim *et al.*, 2017]). Recall Eq.(1) – now we use $\mathcal{E}_\psi$ to handle each input flow map $\boldsymbol{X}$ as the *lifting* operation in NO, which can be briefly defined as follows:

$$h_0(\boldsymbol{X}) = \mathcal{E}_\psi(v(\{x_i\}_{i=1}^{I\times J})). \tag{6}$$

We will feed the output of Eq.(6) into Contextual Processor.

3.2 Contextual Processor (CP)

CP mainly contains two layers: the Spatial Kernel Integral attempts to capture structural interactions from a spatial perspective, and the Environmental Extractor strives to incorporate multiple factual bases that could affect flow dynamics.

Spatial Kernel Integral. Recent works discussed various possible forms for implementing the kernel integral operator, such as graph neural networks [Li *et al.*, 2020c; Li *et al.*, 2020b], Fourier transformations [Li *et al.*, 2020a], attention-based [Kovachki *et al.*, 2021; Cao, 2021; Li *et al.*, 2022], etc. In contrast, we employ *linear Galerkin-type attention* without Softmax [Cao, 2021], which has an approximation capacity comparable to a Petrov-Galerkin projection [Reddy, 2013] under a Hilbertian setup [Cao, 2021] and is also effective in capturing structural interactions in the spatial domain [Li *et al.*, 2022]. In Galerkin-type attention, the latent representation is interpreted column-wise, where each column represents a basis, in contrast to conventional row-wise attention. Specifically, it can be formulated as:

$$Att(h_l) = Q(LN(K)^{\mathrm{T}} \cdot LN(V))/n, \tag{7}$$

where $h_l \in \mathbb{R}^{(IJ)\times d_h}$ and $h_0 = h_0(\boldsymbol{X})$. Similar to standard Transformer [Vaswani *et al.*, 2017], Q, K and V are the latent representations calculated by multiplication with three trainable projection matrices W_Q, W_K and $W_V \in \mathbb{R}^{d_h\times d_h}$, respectively, i.e., $Q = XW_Q$, $K = XW_K$ and $V = XW_V \in \mathbb{R}^{(IJ)\times d_h}$. And $LN(\cdot)$ represents the layer normalization.

More importantly, we use the results of $Att(h_l)$ to replace $k_\phi^{(l)}(x,y)h_l(y)dy$ in Eq. (2). In UNO, $\boldsymbol{X}_l = \{h_l(x_i)\}_{i=1}^{d_h}$, each item $h_l(x_i)$ is column-wise component. Moreover, we introduce a nonlinear feedforward neural network (FFN) with residual concatenation inside to enrich the bases in Q, K, and V (cf. [Cao, 2021]). Hence, the iterative kernel integral specified in Eq. (2) can be rewritten as:

$$h_{l+1}(x) = h_l(x) + FFN(Att(h_l(x)) + h_l(x)), \tag{8}$$

$$Att(h_l(x)) = \sum_{j=1}^{d_h}\sum_{m=1}^{d_h} \langle v_j, k_m \rangle \, q_m(x), \tag{9}$$

where $\langle v_j, k_m \rangle = (K^T V)_{mj}$. The iterative kernel integration helps capture the relationship between the input and output, and progressively approximates the target function at different scales. Thus, the output $h_L(\{\mathsf{x}^j\}_{j=1}^{I\times J})$ is approximately mapped to the target FG space $\mathcal{H}(D_{NI\times NJ})$. Since the input $h_0(\{x_i\}_{i=1}^{I\times J})$ only possesses $I \times J$ points, it approximates the mapping to $I\times J$ points in the FG space, which is less than the exact number $NI\times NJ$ of points in the FG space. More specifically, considering the spatial constraint between CG and FG maps, each point in the CG space corresponds to $N \times N$ subregions of the approximate space of FG. Therefore, we can approximate each point in $h_L(\{\mathsf{x}_j\}_{j=1}^{I\times J})$ to the value of the central point of each subregion. We use a *map injection* operation (f) on the output $h_L(\{\mathsf{x}_j\}_{j=1}^{I\times J})$ based on the points' coordinates $\{\mathsf{x}_{mn}\}_{n=1,m=1}^{NI,NJ}$ in the FG map to reconstruct the value around x_j, $j = 1, \cdots, I \times J$:

$$\hat{h}_L(\{\mathsf{x}_j\}_{j=1}^{NI\times NJ}) = f(h_L(\{\mathsf{x}_j\}_{j=1}^{I\times J}), \{\mathsf{x}_{mn}\}_{n=1,m=1}^{NI,NJ}). \quad (10)$$

Environmental Extractor. Our Environmental Extractor strives to augment the learned features by incorporating multiple factual grounds, which include three phases (P1-P3):

(P1) *Static Facts Encoding:* With the same settings as in UrbanFM [Liang *et al.*, 2019], we utilize several distinct embedding layers to embed continuous (e.g., temperature, wind speed, etc.) and categorical features (e.g., weather, the day of the week, etc.) into low-dimensional vectors. Specifically, for continuous features, we directly concatenate their values to form the continuous external vector e_{con}. For categorical features, we utilize separate embedding layers and then concatenate those embeddings to construct the categorical vector e_{cat}. The final external factor is given by $e = e_{con} \oplus e_{cat}$. In the end, we employ dense layers to transform it into the same function space as $\hat{h}_L(\mathsf{x})$, denoted as $e(\mathsf{x})$.

(P2) *Spatial Relation Augmentation:* We first concatenate $\hat{h}_L(\mathsf{x})$ and $e(\mathsf{x})$ and then adopt two distinct blocks for local and global features, respectively: local relation block (LRB) and global relation block (GRB). In LRB, we initially slice the input into a set of disjoint sub-spaces with size $N \times N$, leading to $I \times J$ sub-regions in total. Each sub-space is then fed to a learning backbone with two CNN layers followed by an independent component layer [Chen *et al.*, 2019]. Regarding GRB, we directly feed the input to the backbone, which has the same structure as the extraction backbone in LRB but with different kernel sizes in the CNN layers. Consequently, we concatenate the resulting two feature maps $H = \{H_{\text{local}}, H_{\text{global}}\}$.

(P3) *Temporal Dynamic Convolution:* In order to capture the influence of the temporal factor on the urban flow distribution, we introduce a time-related block consisting of K independent CNN layers, where K is the number of time spans in a day. Subsequently, we feed H into the CNN layer corresponding to its specific time span.

3.3 Spatial Constraint Decoder

After we obtain the feature map from the approximate fine-grained space, we directly feed the output of the CP to an N^2 – Normalization layer, used to obey the spatial constraint (cf. UrbanFM [Liang *et al.*, 2019]). Hence, the training objective of UNO is the commonly used mean squared error (MSE) between the inferred flow map $\widetilde{\mathcal{X}}$ and the ground truth $\mathcal{X}$:

$$\mathcal{L}(\Theta) = ||\widetilde{\mathcal{X}} - \mathcal{X}||^2. \quad (11)$$

4 UNO with Incremental Enhancement

Due to the inherent CF issue in tackling a series of FUFI tasks, we introduce two incremental enhancements that allow UNO to handle time-dependent flows more flexibly in individual task training and sequential task learning, respectively.

4.1 Individual Task Training in Incremental

Given a task t ($t \geq 1$), we have access to current urban flow data $\mathcal{D}_t$, which is also ordered over time, e.g., $\mathcal{D}_t = \{\boldsymbol{X}_1^t, \boldsymbol{X}_2^t, \cdots\}$. Usually, existing solutions randomly select a batch of flow maps for model optimization, denoted as:

$$\Theta_t^+ = \Theta_t - \alpha \nabla_{\Theta_t} \mathcal{L}_{\mathcal{I}_t}(\Theta_t), \quad (12)$$

where $\mathcal{I}_t \in \mathcal{D}_t$ is a batch of selected urban flow pairs. However, they ignore the time-dependent relationship between these flow maps. For instance, the current flow at τ is naturally related to the previous $\tau - 1$, yielding the CF issue throughout current urban flows. To address this, we introduce a flow-related incremental enhancement method for each individual task. Given a batch of randomly selected flow pairs $\mathcal{I}_t$, we further select a portion of $\mathcal{I}_t$ (denoted as $\mathcal{I}_t^\tau$) with a sampling rate η, where τ reports each input-target pair's timestamp. In the optimization, we feed coarse-grained flow maps at $\tau - 1$ to the model and treat their predictions as the ground truth of $\mathcal{I}_t^\tau$ at τ, aiming at transferring the old knowledge in $\mathcal{I}_t^{\tau-1}$ to $\mathcal{I}_t^\tau$. Notably, we use $\mathcal{I}_t^{'\tau}$ instead of $\mathcal{I}_t^\tau$ for clarity. Hence, the above Eq.(12) can be rewritten as follows:

$$\Theta^+ = \Theta - \alpha \nabla_\Theta (\mathcal{L}_{\mathcal{I}/\mathcal{I}_\tau}(\Theta) + \mathcal{L}_{\mathcal{I'}_t^\tau}(\Theta)). \quad (13)$$

In implementation, we set $\eta = 0.25$ to avoid excessive interference with new knowledge acquisition.

4.2 Sequential Task Learning in Incremental

After we have trained the task $t - 1$ and obtained an optimal model Θ_{t-1}^*, we strive to use Θ_{t-1}^* to handle the new urban flow dataset, i.e., $\mathcal{D}_t$. Motivated by recent continual learning paradigms [Lopez-Paz and Ranzato, 2017; Buzzega *et al.*, 2020], a pioneering work [Yu *et al.*, 2023] attempts to build a memory buffer that selectively preserves some of the old samples from the previous datasets to maintain the old knowledge for new task learning. However, we consider the settings where any previous datasets (i.e., $\mathcal{D}_{<t}$) are currently not available due to privacy concerns, which renders the current solutions inapplicable. Fortunately, recent zero-shot learning solutions [Nayak *et al.*, 2019; Gao *et al.*, 2023] have shown that we can acquire old knowledge from *previously trained models* rather than retaining connections to *old samples*. However, they still rely on previous targets to help reconstruct old samples, which is impractical for FUFI. The reason for this is that the target (i.e., the FG flow map) already implies coarse-grained flows of different granularities. It is impractical to preserve diverse fine-grained flow maps due to information leakage and memory costs.

To this end, we propose a task-related incremental enhancement method for sequential task learning. Motivated by recent pseudo-labeling resolutions [Troisemaine *et al.*, 2023], we feed the coarse-grained flow maps in the task t to Θ_{t-1}^*. After that, we obtain pseudo-fine-grained flow maps, which

	task-1			task-2			task-3			task-4		
	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE
SRCNN	18.464	2.491	0.714	21.270	2.681	0.689	23.184	2.829	0.727	14.730	2.289	0.665
ESPCN	17.690	2.497	0.732	20.875	2.727	0.732	22.505	2.862	0.773	13.898	2.228	0.711
VDSR	17.297	2.213	0.467	21.031	2.498	0.486	22.372	2.548	0.461	13.351	1.978	0.411
DeepSD	17.272	2.368	0.614	20.738	2.612	0.621	22.014	2.739	0.682	15.031	2.297	0.652
SRResNet	17.338	2.457	0.713	20.466	2.660	0.688	21.996	2.775	0.717	13.446	2.189	0.637
UrbanFM	16.372	2.066	0.335	19.548	2.284	0.328	21.243	2.398	0.336	12.744	1.850	0.311
DeepLGR	17.125	2.103	0.339	21.217	2.386	0.350	23.563	2.497	0.351	13.390	1.916	0.345
FODE	16.473	2.142	0.403	19.884	2.377	0.395	21.425	2.490	0.417	12.840	1.947	0.396
UrbanODE	16.342	2.135	0.406	19.648	2.357	0.394	21.177	2.460	0.408	12.668	1.929	0.391
UrbanPy	16.082	2.026	0.329	19.025	2.232	0.318	20.810	2.333	0.313	12.336	1.810	0.304
CUFAR	14.991	1.952	0.306	18.259	2.186	0.301	19.309	2.243	0.289	11.681	1.758	0.288
UNO	14.691	1.927	0.297	**17.722**	2.148	0.290	19.072	2.217	0.279	**11.514**	1.736	0.276
UNO+	**14.464**	**1.900**	**0.285**	17.756	**2.133**	**0.283**	**18.854**	**2.194**	**0.274**	11.564	**1.732**	**0.272**

Table 1: Performance comparison on four TaxiBJ datasets when using the *single-task* protocol.

		task-2			task-3			task-4		
		MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE
fine-tune	UrbanFM	19.162	2.257	0.322	20.499	2.341	0.325	12.285	1.814	0.314
	DeepLGR	20.571	2.336	0.334	21.845	2.427	0.345	12.820	1.858	0.318
	FODE	19.251	2.323	0.379	20.511	2.410	0.387	12.414	1.895	0.369
	UrbanODE	19.070	2.302	0.371	20.275	2.375	0.372	12.182	1.862	0.361
	UrbanPy	18.822	2.208	0.317	20.117	2.293	0.314	12.088	1.800	0.307
	CUFAR	17.746	2.151	0.293	18.915	2.219	0.287	11.486	1.745	0.290
	UNO	**17.407**	**2.121**	**0.287**	**18.794**	**2.199**	**0.280**	**11.258**	**1.716**	**0.275**
incremental	UrbanFM	18.477	2.215	0.312	19.809	2.290	0.314	11.919	1.778	0.302
	DeepLGR	19.202	2.292	0.342	19.892	2.331	0.330	11.977	1.819	0.314
	FODE	18.799	2.297	0.370	20.012	2.369	0.373	11.997	1.852	0.359
	UrbanODE	18.735	2.289	0.374	19.779	2.340	0.361	11.924	1.836	0.352
	UrbanPy	18.286	2.193	0.311	19.503	2.264	0.314	11.958	1.787	0.304
	CUFAR	17.616	2.141	0.292	18.840	2.213	0.285	11.420	1.735	0.283
	CUFI	17.665	2.125	0.282	19.002	2.203	0.279	11.414	1.717	0.278
	UNOI	**17.276**	**2.112**	**0.282**	**18.600**	**2.188**	**0.276**	**11.191**	**1.712**	**0.274**

Table 2: Performance comparison between *fine-tune* and *incremental* protocols.

can be treated as weak signals reflecting past perceptions. For any flow map $\boldsymbol{X}_t$, we have:

$$\bar{\boldsymbol{X}}_t = \mathcal{G}_{\Theta^*_{t-1}}(\boldsymbol{X}_t). \tag{14}$$

Hence, our objective for the current task t can be denoted as:

$$\mathcal{L}(\Theta_t) = \mathcal{L}((\mathcal{G}_{\Theta_t}(\boldsymbol{X}), \mathcal{X})|\Theta_t) + \mathcal{L}((\mathcal{G}_{\Theta^*_{t-1}}(\boldsymbol{X}), \bar{\boldsymbol{X}})|\Theta_t). \tag{15}$$

Herein, we omit t on either $\boldsymbol{X}$ or $\mathcal{X}$ for simplicity. In practice, we randomly select a small portion ($\eta = 0.25$) of CG flow maps in $\mathcal{D}_t$ to bypass the training instability. Specifically, we record the convergence of actual flow pairs in implementation and sample again if the model performs poorly.

5 Experiments

Datasets. Following [Yu *et al.*, 2023], we conduct experiments on four real-world taxi traffic datasets collected continuously for four years (from 2013 to 2016) in Beijing city.

Baselines. We compare our UNOI with the baselines which include *five image super-resolution methods*: SRCNN [Dong *et al.*, 2015], ESPCN [Shi *et al.*, 2016], VDSR [Kim *et al.*, 2016], DeepSD [Vandal *et al.*, 2017], and SRResNet [Ledig *et al.*, 2017]; and *six FUFI baselines*: UrbanFM [Liang *et al.*, 2019], DeepLGR [Liang *et al.*, 2020], FODE [Zhou *et al.*, 2020], UrbanODE [Zhou *et al.*, 2021], UrbanPy [Ouyang *et al.*, 2020], and CUFAR [Yu *et al.*, 2023].

Training Protocols. In this study, we also use four training protocols: *Single-task* learns each task in isolation; *Fine-tune* and *continual* learn a new task without the help of previous task(s); *Joint* learns previous tasks and new tasks together.

Evaluation Metrics. We employ three widely used FUFI metrics: mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

Implementation. Methods are implemented using PyTorch, accelerated by one NVIDIA GeForce-RTX-4090. The Adam optimizer is employed with an initial learning rate of $1e^{-4}$ and training epoch set to 60. To prevent overfitting, we apply the learning rate decay trick and set the dropout rate to 0.3. The batch size is set to 16. For reproducibility, our source codes are available at https://github.com/Longsuni/UNO.git.

5.1 Model Performance

UNO Performance. Table 1 reports the inference performance of all methods on four FUFI tasks using the *single-task* protocol, where the best result is marked in **bold** while the second best is underlined. We also conducted a variant of UNO, namely UNO+, which uses the flow-related incremental enhancement method to facilitate the learning of every single task but does not break the rule of the *single-task* protocol. We find that UNO consistently outperforms all of the baselines, including the super-resolution methods, demonstrating the effectiveness of our novel operator. In particular, traditional standard super-resolution methods perform poorly due in part to domain conflicts. Additionally, UNO+ performs the best on most of the metrics compared to UNO, indicating that our flow-related incremental enhancement can boost the knowledge transfer from the previouw to the new flow map.

UNOI Performance. Now we turn to evaluate the performance of our incremental enhancement in sequential task learning. We choose *fine-tune* and *incremental* protocols for each FUFI solution, respectively. We note that all methods are initially trained on Task-1. And then each model is fine-tuned or incrementally leaned on new coming tasks. We have the following observations. (1) As shown in the top of Table 2, our UNO without any incremental enhancement still performs the best compared to the baseline FUFI solutions when we use the same *fine-tune* protocol. Compared to the methods that follow the *single-task* protocol (cf. Table 1), we observe that UNO, with the *fine-tune* protocol, allows for the maintenance of old knowledge to some extent. (2) As shown in the bottom of Table 2, we enhance traditional solutions with the same sample replay method proposed in [Yu *et al.*, 2023]. As CUFAR performs the best among the baselines, we build a variant of CUFAR by replacing the replay module of CUFAR with our incremental enhancements, namely CUFI. We can find it achieves a competitive performance against the CUFAR. Meanwhile, we do not allow reuse of old samples from earlier datasets due to privacy concerns. Such competitive performance also demonstrates the superiority of our data-free solution. (3) UNOI performs the best against other incremental learning methods, indicating it can well boost the learning of newly coming tasks even if we do not allow to access any old samples from the previous tasks. In addition, the outperforming results of UNOI against UNO with *single-task* protocol indicate that our incremental enhancement can promote knowledge transfer from old datasets to the new task.

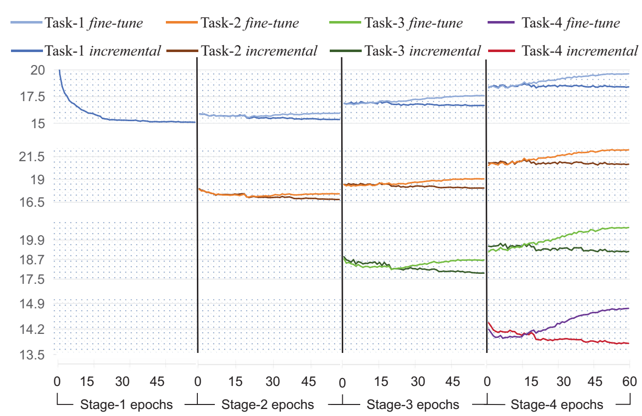

Figure 2: The performance comparison on addressing the CF issue.

CF issue. As we claimed before, due to the *catastrophic forgetting* (CF) concern, the existing solutions could lack the ability to retain the old knowledge from past datasets or tasks. To validate the performance of UNOI in addressing the CF issue, we visualize validation performance (i.e., MSE loss) in the training processes of UNOI by using *fine-tune* and *continual* protocols, respectively. As shown in Fig. 2, as the number of new tasks increases, we clearly discover that the loss gap between these two protocols widens, especially on the last task. Consequently, we can conclude that the *fine-tune* protocol faces a serious CF problem, which in turn suggests that our UNOI is able to alleviate the CF problem effectively.

5.2 Model Analysis

Ablation Study. We first investigate the impact of each proposed module in our proposed UNOI. We correspondingly yield three variants of UNOI, including: w/o NO removes the neural operator (i.e., Spatial Kernel Integral) in UNO block; w/o SE removes the Static Fact Encoding block; w/o TE removes the Temporal Dynamic Convolution block. As shown in Fig. 3, we can find that: (1) removing any components does degrade the model performance. More importantly, removing the neural operator in UNOI brings the largest losses, which indicates it contributes the most. (2) Compared to the UNO with *single-task* protocol (denoted as 'single-task') and the UNO with *fine-tune* protocol (denoted as 'fine-tune'), UNOI performs the best, which also indicates that UNOI successfully transfers the old knowledge from past tasks.

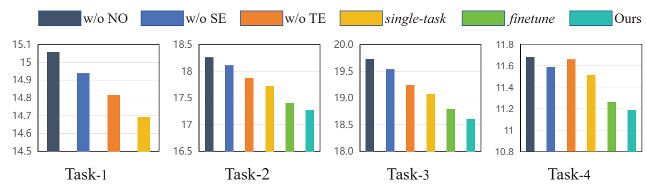

Figure 3: An ablation study with MSE performance.

Convergence Analysis. To evaluate the performance of our UNO module, we compare it with five popular FUFI solutions under the *single-task* protocol. As shown in Fig. 4, UNO converges the fastest and achieves the lowest validation losses on all tasks. In particular, traditional ODE-based methods produce drastic oscillations during model convergence and even multiple gradient explosions. In contrast, ours performs more stable due to the priority of invariant and smooth functional approximation. Compared with CUFAR, we can observe that our method converges significantly faster than it, suggesting the efficiency of our proposed UNO block.

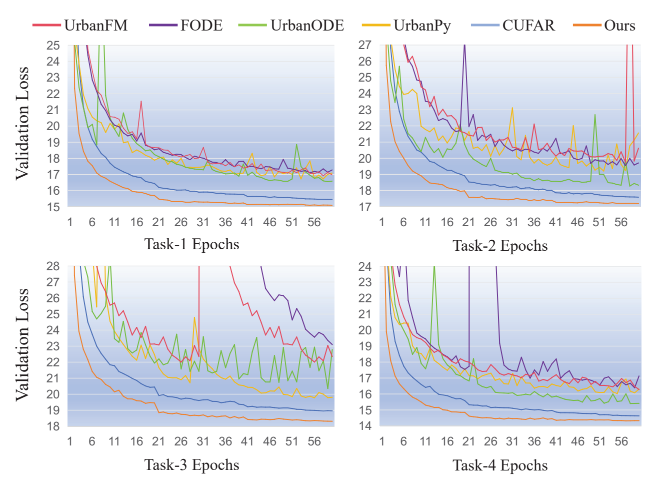

Figure 4: Convergence Analysis of UNO.

Joint Protocol. Now we turn to compare our UNOI with a naive context. That is, we use the *joint* protocol to handle all of the FUFI tasks sequentially while maintaining the used

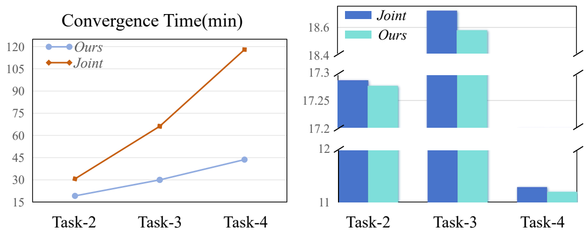

Figure 5: Comparison between *joint* and *continual* protocols.

datasets. *Joint* protocol is a commonly used paradigm that attempts to train all tasks simultaneously, usually yielding more promising results. Fig. 5 shows the performance comparison regarding model efficiency and MSE performance. We can find that as more tasks arrive, the time cost of model convergence under the *joint* protocol increases significantly. Moreover, we also observe that the model under the *joint* protocol underperforms ours. We consider that the possible reason is that reusing all old samples from previous datasets may raise noise interference. Moreover, not all past knowledge is useful due to the impact of uncertainties such as improvements in urban infrastructure, i.e., the presence of *concept drift*.

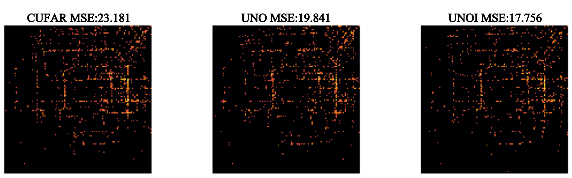

Figure 6: A case study on error visualization.

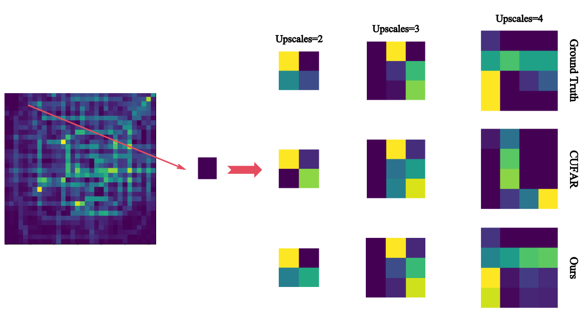

Figure 7: A case study on flow inference.

Case Study. In the end, we conduct two case studies to uncover UNOI performance from an interpretability perspective. First, we visualize the inference errors of three models on a case flow map regarding MSE. As shown in Fig. 6, the brighter the pixels, the larger the errors. We can find that our UNOI has much fewer brighter pixels than CUFAR while UNO, which has no incremental enhancements, performs moderately well. Next, we provide another case study by visualizing a coarse-grained flow map during inference. As shown in Fig. 7, we can find our UNOI can infer the urban flow well from a micro perspective. We especially observe that it performs better in lower-scale flow spaces when inferring coarse-grained and fine-grained flow details.

6 Related Work

The FUFI problem was first formulated in [Liang *et al.*, 2019], motivating the mapping functions between a CG and FG flow maps. Existing works addressing FUFI aim to tackle the following challenges: *(1) Exploring spatial correlations and introducing external factors:* UrbanFM [Liang *et al.*, 2019] proposed the spatial constraint concept and first embedded external features as a supplementary for FUFI. DeepLGR [Liang *et al.*, 2020] enhances the understanding of flow dynamics by revisiting CNN approaches, emphasizing the learning of both global spatial dependencies and local feature representations. FODE [Zhou *et al.*, 2020] and UrbanODE [Zhou *et al.*, 2021] efficiently estimate spatial correlations from a dynamic systems perspective by introducing neural ODEs to overcome the numerically unstable gradient computation problem. *(2) Inference on large-scale map:* UrbanPy [Ouyang *et al.*, 2020] extends UrbanFM to provide outstanding effectiveness and efficiency for large-scale upsampling via introducing a cascading paradigm. *(3) Improving generalization ability of the model.* The latest state-of-the-art work CUFAR [Yu *et al.*, 2023], not only proposes an effective way to extract local and global features but also introduces continual learning with an adaptive knowledge replay strategy to address the catastrophic forgetting [Kirkpatrick *et al.*, 2017] phenomenon across different tasks.

However, the existing works directly learn a scale-sensitive mapping function between fixed data space, limiting generalization across diverse scales. In addition, while CUFAR introduces a continual learning-based approach to address the catastrophic forgetting (CF) problem via introducing the replay buffer mechanism, such a method may be impractical in real-world FUFI contexts. Privacy concerns often restrict access to previous datasets after model training, preventing the transfer of old knowledge to new tasks. In contrast, UNO not only explores a scale-invariant mapping from the function space rather than the data space, but also tackles privacy-related concerns in conjunction with the CF problem.

7 Conclusion

We presented UNOI, a novel urban neural operator solution with incremental learning, for addressing the FUFI problem. Unlike traditional fine-grained urban flow inference approaches, UNOI provides two-granularity views to transfer the knowledge learned from the past to the newly acquired one in terms of handling a series of FUFI tasks/datasets. Specifically, we devise an urban neural operator (UNO), which takes advantage of recent operator learning and learns mappings between approximation spaces by treating the different-grained flows as continuous functions. To address the CF problem under accessibility constraints for past data (e.g., due to privacy concerns), we devise two incremental enhancements to handle individual task training and sequential task learning, respectively. The experiments conducted with four training protocols demonstrate the significant superiority of our solution. As part of our future work, we plan to make more efforts to effectively tackle heterogeneous tasks (e.g., a series of cross-regional tasks) in an incremental manner.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.62102326), the Sichuan Science and Technology Program under Grant (Grant No.2023ZYD0145), the Natural Science Foundation of Sichuan Province (Grant No.2023NSFSC1411), the NSF SWIFT 2030249, and Guanghua Talent Project.

References

[Buzzega *et al.*, 2020] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 15920–15930, 2020.

[Cao, 2021] Shuhao Cao. Choose a transformer: Fourier or galerkin. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

[Chen *et al.*, 2019] Guangyong Chen, Pengfei Chen, Yujun Shi, Chang-Yu Hsieh, Benben Liao, and Shengyu Zhang. Rethinking the usage of batch normalization and dropout in the training of deep neural networks. *arXiv preprint arXiv:1905.05928*, 2019.

[Dong *et al.*, 2015] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[Gao *et al.*, 2023] Qiang Gao, Xiaojun Shan, Yuchen Zhang, and Fan Zhou. Enhancing knowledge transfer for task incremental learning with data-free subnetwork. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Greenberg, 2015] Michael D Greenberg. *Applications of Green's functions in science and engineering*. Courier Dover Publications, 2015.

[Hershberger, 2013] John Hershberger. Stable snap rounding. *Comput. Geom.*, 46(4):403–416, 2013.

[Kim *et al.*, 2016] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[Kirkpatrick *et al.*, 2017] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[Kovachki *et al.*, 2021] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

[Kovachki *et al.*, 2023] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[Ledig *et al.*, 2017] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[Li *et al.*, 2020a] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[Li *et al.*, 2020b] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[Li *et al.*, 2020c] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

[Li *et al.*, 2022] Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022.

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

[Liang *et al.*, 2020] Yuxuan Liang, Kun Ouyang, Yiwei Wang, Ye Liu, Junbo Zhang, Yu Zheng, and David S. Rosenblum. Revisiting convolutional neural networks for citywide crowd flow analytics. In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2020.

[Liang *et al.*, 2021] Yuxuan Liang, Kun Ouyang, Junkai Sun, Yiwei Wang, Junbo Zhang, Yu Zheng, David Rosenblum, and Roger Zimmermann. Fine-grained urban flow prediction. In *Proceedings of the Web Conference 2021*, pages 1833–1845, 2021.

[Liang *et al.*, 2022] Yuxuan Liang, Kun Ouyang, Yiwei Wang, Zheyi Pan, Yifang Yin, Hongyang Chen, Junbo Zhang, Yu Zheng, David S Rosenblum, and Roger Zimmermann. Mixed-order relation-aware recurrent neural networks for spatio-temporal forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Lim *et al.*, 2017] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[Lopez-Paz and Ranzato, 2017] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6470–6479, 2017.

[Nayak *et al.*, 2019] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pages 4743–4751. PMLR, 2019.

[Ouyang *et al.*, 2020] Kun Ouyang, Yuxuan Liang, Ye Liu, Zekun Tong, Sijie Ruan, Yu Zheng, and David S Rosenblum. Fine-grained urban flow inference. *IEEE transactions on knowledge and data engineering*, 34(6):2755–2770, 2020.

[Reddy, 2013] Junuthula Narasimha Reddy. *An introduction to the finite element method*, volume 3. McGraw-Hill New York, 2013.

[Shi *et al.*, 2016] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[Troisemaine *et al.*, 2023] Colin Troisemaine, Vincent Lemaire, Stéphane Gosselin, Alexandre Reiffers-Masson, Joachim Flocon-Cholet, and Sandrine Vaton. Novel class discovery: an introduction and key concepts. *arXiv preprint arXiv:2302.12028*, 2023.

[Vandal *et al.*, 2017] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. Deepsd: Generating high resolution climate change projections through single image super-resolution. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1663–1672, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2017.

[Wei and Zhang, 2023] Min Wei and Xuesong Zhang. Super-resolution neural operator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18247–18256, 2023.

[Yu *et al.*, 2023] Haoyang Yu, Xovee Xu, Ting Zhong, and Fan Zhou. Overcoming forgetting in fine-grained urban flow inference via adaptive knowledge replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[Zheng *et al.*, 2014] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3):1–55, 2014.

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

[Zhou *et al.*, 2021] Fan Zhou, Xin Jing, Liang Li, and Ting Zhong. Inferring high-resolutional urban flow with internet of mobile things. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7948–7952. IEEE, 2021.