Boosting Fine-Grained Urban Flow Inference via Lightweight Architecture and Focalized Optimization

Yuanshao Zhu[1, 2], **Xiangyu Zhao**[2, *], **Zijian Zhang**[3], **Xuetao Wei**[1], **James Jianqiao Yu**[4, *]

[1]Southern University of Science and Technology, China
[2]City University of Hong Kong, China
[3]Jilin University, China
[4]Harbin Institute of Technology, Shenzhen, China
yuanshao@ieee.org, xianzhao@cityu.edu.hk, zhangzijian@jlu.edu.cn, weixt@sustech.edu.cn, jqyu@ieee.org

Abstract

Fine-grained urban flow inference is crucial for urban planning and intelligent transportation systems, enabling precise traffic management and resource allocation. However, the practical deployment of existing methods is hindered by two key challenges: the prohibitive computational cost of over-parameterized models and the suboptimal performance of conventional loss functions on the highly skewed distribution of urban flows. To address these challenges, we propose a unified solution that synergizes architectural efficiency with adaptive optimization. Specifically, we first introduce **PLGF**, a lightweight yet powerful architecture that employs a Progressive Local-Global Fusion strategy to effectively capture both fine-grained details and global contextual dependencies. Second, we propose **DualFocal** Loss, a novel function that integrates dual-space supervision with a difficulty-aware focusing mechanism, enabling the model to adaptively concentrate on hard-to-predict regions. Extensive experiments on 4 real-world scenarios validate the effectiveness and scalability of our method. Notably, while achieving state-of-the-art performance, PLGF reduces the model size by up to 97% compared to current high-performing methods. Furthermore, under comparable parameter budgets, our model yields an accuracy improvement of over 10% against strong baselines.

1 Introduction

Fine-grained urban flow data provides the foundational insights for modern intelligent transportation systems and smart city infrastructures, enabling applications like precise traffic management and responsive urban planning (Wang, Cao, and Philip 2020). However, acquiring such high spatial resolution data directly through dense sensor deployment is often infeasible due to prohibitive long-term costs for equipment, operation, and maintenance (Xie et al. 2020). To address this, Fine-Grained Urban Flow Inference (FUFI), which infers fine-grained flow maps from available coarse-grained observations, has emerged as a pressing and cost-effective solution for developing sustainable and economically viable smart cities (Qu et al. 2022; Wang et al. 2023).

To accurately infer fine-grained urban flow, early FUFI methods draw from the super-resolution techniques of computer vision, where urban flow maps can be viewed as an image. Foundational works like UrbanFM (Liang et al. 2019) and UrbanPy (Ouyang et al. 2020) established the core paradigm, introducing deep learning frameworks with crucial spatial constraints and pyramid architectures to handle the mapping between coarse and fine-grained flows. Follow-up efforts are mostly focused on refining architectural designs for better city-wide traffic profile capturing (Zhou et al. 2020). More recently, the field has shifted from pursuing pure accuracy to addressing practical deployment challenges, such as catastrophic forgetting in dynamic systems (Yu et al. 2023), robustness to noisy data (Zheng et al. 2023), and cross-city knowledge transfer (Zheng et al. 2024a).

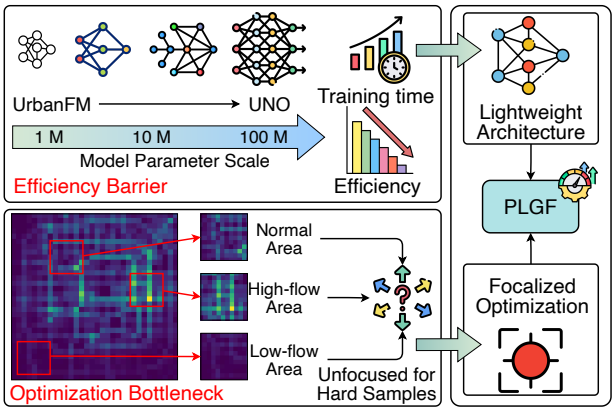

Figure 1: The proposed PLGF addresses efficiency barriers and optimization bottleneck issues with a lightweight architecture design and focalized optimization.

Despite these significant advancements, the very pursuit of higher accuracy has given rise to two critical limitations that hinder practical deployment, as shown in Figure 1. First, an **efficiency barrier** has emerged. The push for higher performance has led to increasingly complex and over-parameterized architectures to capture intricate spatio-temporal dependencies (Gao et al. 2024). The scale of model parameters has increased from 1 million to 100 million. This trend towards "model bloat" incurs extreme computational costs for both training and inference, posing significant chal-

*Corresponding authors

lenges for real-world applications. Second, a persistent **optimization bottleneck** remains. Existing models are typically optimized with conventional regression losses, such as Mean Squared Error, which treat all prediction errors equally. By failing to account for the highly skewed, non-uniform nature of urban flow data, this generic approach leaves the optimization process unfocused, limiting the ultimate accuracy and robustness potential of the model, especially when dealing with high-variance samples (Liang et al. 2019).

To address these fundamental limitations, we propose a unified framework that rethinks both architectural design and optimization strategy. First, we introduce **PLGF** (**P**rogressive **L**ocal-**G**lobal **F**usion), a lightweight yet powerful architecture designed for parameter efficiency. Specifically, PLGF adopts an efficient progressive local-global fusion framework and a context-aware integration mechanism. This design enables it to capture complex multi-scale spatial dependencies with remarkable parameter efficiency, while ensuring that the entire process is subject to modulation by specific spatio-temporal contexts. In addition, we propose the **DualFocal** Loss, a novel loss function for focalized optimization. By integrating dual-scale (linear and logarithmic) supervision with a difficulty-aware focusing mechanism, our loss allows the model to adaptively concentrate on hard-to-predict, low-flow regions and high-variance samples that are often ignored by traditional methods.

The main contributions of this paper are as follows:

- We propose PLGF, a novel, lightweight, and parameter-efficient architecture. It effectively captures multi-scale spatial dependencies based on a progressive local-global fusion and context-aware strategy, thus significantly reducing the computational cost.
- We introduce DualFocal Loss, a universal and plug-and-play loss function with a difficulty-aware focusing mechanism. It can be flexibly applied to various FUFI models, improving both accuracy and robustness.
- Extensive experiments on 4 real-world scenarios demonstrate that PLGF achieves over 10% performance improvement under the same parameter budget and reduces parameter count by approximately 97% while maintaining state-of-the-art performance.

2 Related Work

Fine-grained urban flow inference seeks to recover high-resolution urban flow maps from coarse-grained observations. The foundational paradigm was established by UrbanFM (Liang et al. 2019), which introduced a distributional upsampling module enforcing strict spatial constraints, ensuring that the sum of fine-grained flows matches their corresponding coarse-grained region. Subsequent works advanced FUFI from several perspectives. UrbanPy (Ouyang et al. 2020) proposed a cascading pyramid architecture for progressive upsampling, improving scalability to large-scale urban maps. DeepLGR (Liang et al. 2021) further enhanced spatial modeling by introducing a dual-path network that jointly learns global and local features. Alternative approaches, such as FODE (Zhou et al. 2020) and UrbanODE (Zhou et al. 2021), leveraged neural ordinary differential equations to capture continuous urban dynamics and address numerical instability. More recently, research has shifted to practical deployment challenges in dynamic, real-world environments. CUFAR (Yu et al. 2023) addressed catastrophic forgetting in dynamic environments via an adaptive knowledge replay strategy, while UNO (Gao et al. 2024) proposed data-free incremental learning using neural operators for privacy-preserving, scale-invariant modeling. Robustness to data quality has also been explored: multi-task frameworks have been developed for simultaneous missing data imputation and inference (Li et al. 2022), and denoising strategies have been integrated to enhance resilience to noisy sensor inputs (Zheng et al. 2023). To tackle data scarcity in target cities, cross-city transfer learning methods have been introduced to leverage knowledge from data-rich source cities (Zheng et al. 2024b,a).

Despite these advancements, most existing approaches remain limited by large model sizes and high computational costs. Furthermore, optimization strategies tailored to the highly skewed and non-uniform nature of urban flow data are still lacking. In this work, we address these gaps by proposing a lightweight architecture and a focused optimization strategy, aiming to improve both efficiency and adaptability for real-world FUFI applications.

3 Preliminary

This section formally defines the key concepts and the problem of FUFI, providing the foundation for our methodology.

Urban Flow Map. Following standard practice, a city or a region of interest is partitioned into a uniform $I \times J$ grid map based on longitude and latitude. At a given time interval, the traffic flow across the whole area is represented by an urban flow map, denoted as $X \in \mathbb{R}^{I \times J}$, where each element $x_{i,j} \in \mathbb{R}_+$ indicates the total flow volume (e.g., vehicles, pedestrians) within the corresponding grid cell (i, j).

Coarse-grained and Fine-grained Urban Flow. Urban flow maps can be constructed at varying spatial granularities depending on the scale of observation. A fine-grained flow map $X_f \in \mathbb{R}^{(N_f \times N_f)}$ provides detailed mobility information, while a coarse-grained flow map $X_c \in \mathbb{R}^{(N_c \times N_c)}$ offers an aggregated perspective. Each cell in the coarse-grained map corresponds to a non-overlapping $N \times N$ block of fine-grained cells, where $N \in \mathbb{Z}+$ is the upscaling factor. This hierarchical aggregation imposes a crucial spatial constraint: the total flow in any coarse cell equals the sum of flows in its constituent fine-grained cells. Formally:

$$x^c_{i,j} = \sum_{i'=N_i}^{N_{(i+1)}-1} \sum_{j'=N_j}^{N_{(j+1)}-1} x^f_{i',j'}, \tag{1}$$

where $x^c_{i,j}$ is the flow in coarse cell (i, j), and $x^f_{i',j'}$ are the flows in the corresponding fine-grained cells.

Fine-Grained Urban Flow Inference. Given an observed coarse-grained urban flow map $X_c \in \mathbb{R}^{(I \times J)}$ and an integer upscaling factor N, the goal of FUFI task is to learn an inference function F_θ that can accurately reconstruct the corresponding high-resolution, fine-grained flow map $X_f \in \mathbb{R}^{(NI \times NJ)}$. The predicted map $\hat{X}_f = F_\theta(X_c)$ should

closely approximate the true fine-grained map while strictly satisfying the spatial constraint above. Optionally, the inference can incorporate external factors E (such as time of day, weather), yielding the formulation $\hat{X}_f = F_\theta(X_c, E)$.

4 Methodology

As illustrated in Figure 2, our proposed PLGF framework is a conditional deep network designed to infer fine-grained urban flow maps from coarse-grained inputs and external factors. The entire design is driven by two principles carefully chosen to address the core challenges of efficiency and optimization: a lightweight architecture built on progressive learning and a strategy for adaptive optimization via a novel loss function. The progressive learning principle tackles the challenge of high-ratio N-times upsampling by decomposing it into $\log_2(N)$ sequential 2-times upsampling stages. This approach is inherently more parameter-efficient and stable than a single-step upsampling process. At each stage, context-aware modulation ensures the model dynamically adapts to external conditions. The entire framework is optimized end-to-end with our proposed DualFocal Loss, which forces the model to adaptively focus on the most challenging aspects of the urban flow data distribution.

PLGF: A Lightweight Fusion Architecture

The PLGF architecture materializes our design principles into a three-part pipeline as shown in Figure 3: (1) an *Environment Context Embedding* block to process external factors, (2) a series of stacked *Progressive Upscaling Blocks (PUBs)* to iteratively refine features and increase resolution, and (3) a final *Density-based Recovery block* to ensure spatial constraint, fine-grained output.

Environment Context Embedding. Firstly, urban flow is not static, which is heavily influenced by a myriad of external factors such as time, weather, and public events. A robust FUFI model must be able to adapt its predictions to these changing conditions. To achieve this, we need a mechanism to encode these heterogeneous external factors into a unified and powerful conditioning signal.

The Environment Context Embedding module transforms raw external factors $E \in \mathbb{R}^7$ into a unified, high-dimensional conditional vector $\boldsymbol{e}_{\text{cond}}$. The categorical features (e.g., day of the week, hour) are mapped to dense embeddings $\{\boldsymbol{v}_{\text{day}}, \boldsymbol{v}_{\text{hour}}, \boldsymbol{v}_{\text{weather}}\}$, while continuous features (e.g., temperature) are processed by a multi-layer perceptron (MLP) to form $\boldsymbol{v}_{\text{cont}}$. To capture interdependencies adaptively, these embeddings are passed through a multi-head self-attention (MHA) layer. The output is then aggregated and projected to produce the final conditional vector:

$$\boldsymbol{e}_{\text{cond}} = \text{Linear}(\text{MHA}([\boldsymbol{v}_{\text{day}}, \boldsymbol{v}_{\text{hour}}, \boldsymbol{v}_{\text{weather}}, \boldsymbol{v}_{\text{cont}}])), \quad (2)$$

which $\boldsymbol{e}_{\text{cond}} \in \mathbb{R}^d$ serves as a dynamic conditioning signal throughout the subsequent progressive upsampling stages.

Progressive Upscaling Block. As illustrated in Figure 3, the core of the PLGF framework is the Progressive Upscaling Block (PUB). Given initial features $\boldsymbol{F}_0$ extracted from the input convolutional layer processing the coarse-grained flow X_c, the PUB iteratively refines and upsamples these features, dynamically conditioned on the context vector $\boldsymbol{e}_{\text{cond}}$:

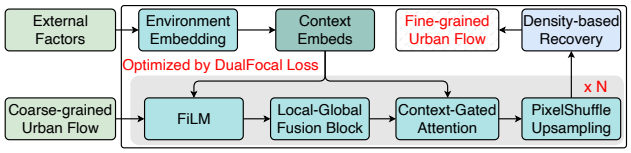

Figure 2: The pipeline of PLGF architecture.

$$\boldsymbol{F}_s = \text{PUB}_s(\boldsymbol{F}_{s-1},\ \boldsymbol{e}_{\text{cond}}), \quad s = 1, \ldots, S, \quad (3)$$

where $\boldsymbol{F}_s \in \mathbb{R}^{C \times (2^s H) \times (2^s W)}$, and C, H, W denote the channel, height, and width of the initial feature map, respectively. Each PUB is composed of four main components: a Feature-wise Linear Modulation (FiLM) for context-aware feature adaptation, a Local-Global Fusion Block for comprehensive feature extraction, a Context-Gated Attention Block for feature refinement, and finally a PixelShuffle Block for $2\times$ spatial upsampling.

FiLM for Context-Aware Modulation. At the start of each PUB, a FiLM layer (Perez et al. 2018) injects contextual information into the feature maps. Given the feature map $\boldsymbol{F}_s$ from the previous stage and the conditional vector $\boldsymbol{e}_{\text{cond}}$, the FiLM layer generates channel-wise scaling γ_s and shifting parameters β_s via linear projections:

$$\boldsymbol{F}'_{s-1} = (1 + \gamma_s) \odot \boldsymbol{F}_{s-1} + \beta_s \quad (4)$$

$$\gamma_s = f_s(\boldsymbol{e}_{\text{cond}}), \quad \beta_s = h_s(\boldsymbol{e}_{\text{cond}}), \quad (5)$$

where $\odot$ denotes element-wise multiplication. This dynamic modulation enables the network to adapt feature responses at each stage to the specific environmental context.

Local-Global Fusion Block. Urban flow patterns exhibit complex spatial dependencies. Predicting the flow in one grid cell requires understanding both its immediate surroundings (e.g., local traffic) and its role within the larger city-wide traffic network (e.g., its position). Capturing these multi-scale spatial features simultaneously is a major challenge. Therefore, we design a dual-path block to explicitly capture both local and global features.

Local Path: Efficiently Capturing Fine-Grained Details. The local path is engineered to capture detailed, hierarchical features within local neighborhoods. At its core is a carefully optimized Residual Dense Block (RDB), a choice motivated by its high parameter efficiency. The architecture of our RDB is motivated by several key principles aimed at maximizing both performance and efficiency. The efficiency of the RDB stems from its use of dense connectivity. As shown in Figure 3, the output of each convolutional layer is concatenated with the inputs of all subsequent layers within the block. This mechanism encourages feature reuse, allowing the network to build highly discriminative local representations without needing to learn redundant feature maps, which is key to its lightweight nature. For an input $\boldsymbol{x}_0$, the operation of the l-th layer H_l is:

$$\boldsymbol{x}_l = H_l([\boldsymbol{x}_0 \oplus \boldsymbol{x}_1 \oplus \cdots \oplus \boldsymbol{x}_{l-1}]), \quad (6)$$

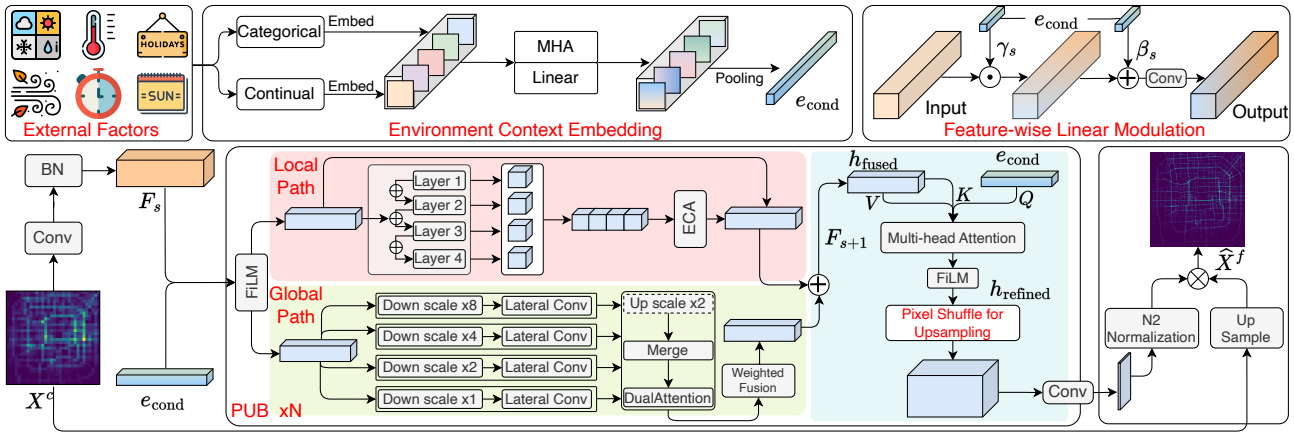

Figure 3: The overall details of PLGF architecture, which consists of four main components, extract spatio-temporal features using a progressive and context-aware manner.

where $[\oplus]$ denotes the concatenation. To enhance training stability and performance, we adopt Group Normalization and the GELU activation function.

After the dense feature extraction, the collection of generated feature maps is fused via a 1×1 convolution. To further enhance discriminative power, we then apply an Efficient Channel Attention (ECA) layer (Wang et al. 2020). This lightweight attention mechanism adaptively recalibrates channel-wise feature responses, amplifying informative features while suppressing less useful ones. Finally, a residual connection adds the refined features back to the original input $\boldsymbol{x}_0$. This strategy stabilizes training and focuses the block on learning only the essential residual information. The entire RDB operation is summarized as:

$$\text{RDB}(\boldsymbol{x}_0) = \boldsymbol{x}_0 + \alpha \cdot \text{ECA}(f_{\text{fusion}}([\boldsymbol{x}_0 \oplus \boldsymbol{x}_1 \oplus \cdots \oplus \boldsymbol{x}_l])), \quad (7)$$

where α is a small residual scaling factor (initialized with 0.1) to stabilize the training process.

Global Path: Adaptively Aggregating Multi-Scale Context. The global path is designed to capture long-range dependencies by aggregating context from multiple spatial scales. We employ an Enhanced Feature Pyramid Network (EnhancedFPN), which augments the standard FPN (Lin et al. 2017a) with adaptive attention mechanisms. Given the input feature map $\boldsymbol{x}_0$, EnhancedFPN first constructs a feature pyramid $\{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{s-1}\}$ using adaptive average pooling at progressively larger scales $\boldsymbol{p}_i = \textbf{AvgPool}(\boldsymbol{x}_0, 2^i)$. This provides multiple views of the input at different levels of granularity. Following this, a top-down fusion pathway systematically enriches finer-grained feature maps (e.g., $\boldsymbol{p}_i$) by upsampling and merging them with semantic context from coarser feature maps (e.g., $\boldsymbol{p}_{i+1}$). This process systematically enriches the finer-grained features with a higher level of semantic context than the coarser-grained ones, creating a semantically rich feature hierarchy.

Crucially, to make this fusion process adaptive, a Dual Attention module is applied at each fusion level. It uses channel and spatial attention to refine salient flow patterns and spatial structures, focusing on the most relevant information at each scale $\{\boldsymbol{p}'_0, \boldsymbol{p}'_1, \ldots, \boldsymbol{p}'_{s-1}\}$. After refining all levels, the final output is produced via an adaptive weighted fusion of all pyramid feature maps:

$$\text{EnhancedFPN}(\mathbf{x}_0) = \sum_{i=0}^{s-1} \sigma(\beta_i) \cdot \boldsymbol{p}'_i, \quad (8)$$

where $\sigma(\cdot)$ is the sigmoid function and the learnable weights β_i allow the network to adaptively emphasize the most informative scales for a given input. The outputs from both local and global paths are concatenated and fused to integrate fine-grained details with multi-scale global context.

Context-Gated Attention. After fusing local and global features, a final, precise refinement is needed. Traditional methods often use separate, heavy convolutional layers for each time period to integrate external information (Liang et al. 2019; Gao et al. 2024). This reduces computational efficiency and significantly increases model parameters. We employ a Context-Gated Attention block for this final, adaptive refinement. This block uses the powerful mechanism of multi-head attention to dynamically gate the fused features $\boldsymbol{h}_{\text{fused}}$ based on the context vector $\boldsymbol{e}_{\text{cond}}$. Specifically, the context vector serves as the "query", while the feature map acts as the "key" and "value":

$$\boldsymbol{v}_{\text{agg}} = \text{MHA}(\boldsymbol{e}_{\text{cond}}, \boldsymbol{h}_{\text{fused}}, \boldsymbol{h}_{\text{fused}}). \quad (9)$$

The resulting vector, $\boldsymbol{v}_{\text{agg}}$, represents a contextually-weighted aggregation of the most salient spatial information. We then use this aggregated vector to generate a new set of FiLM-style parameters (γ, β). This performs a second, deeper modulation on the original fused features: $\boldsymbol{h}_{\text{refined}} = \gamma \odot \boldsymbol{h}_{\text{fused}} + \beta$. This two-step process ensures that the features are not just broadly conditioned, but are precisely and adaptively refined based on a deep, contextual understanding of the fused local-global information.

Spatial Upsampling with PixelShuffle. The final component of the PUB is responsible for upsampling the spatial resolution by a factor of 2. We adopt the PixelShuffle layer (Shi et al. 2016), as it is computationally efficient and effectively mitigates checkerboard artifacts. Specifically, the

PixelShuffle operation first increases the number of feature channels through a standard convolution, then rearranges channel elements into the spatial dimensions, effectively doubling both height and width:

$$\boldsymbol{F}_s = \text{PixelShuffle}(\boldsymbol{h}_{\text{refined}}). \quad (10)$$

The resulting feature map $\boldsymbol{F}_s$ serves as the input for the next progressive stage, or, in the final stage, is passed to the output generation module.

Density-based Recovery. A key physical constraint in FUFI is that the sum of flows in the fine-grained sub-regions must equal the observed flow in the corresponding coarse-grained region. Any valid model must strictly adhere to this conservation law. Following the final PUB, a density-based recovery module ensures this constraint is met. It first uses a convolutional layer with a ReLU activation to produce a non-negative raw density map. Then, the parameter-free N2-Normalization layer (Liang et al. 2019) normalizes the densities within each super-region to sum to one, creating a valid probability distribution. The final fine-grained flow map $\hat{X}_f$ is obtained by element-wise multiplication of this distribution with the upsampled coarse-grained flow X_c. This process guarantees both physical consistency and fine-grained expressiveness.

$$\hat{X}_f = \text{Norm}(\text{ReLU}(\text{Conv}(\boldsymbol{F}_s))) \odot \text{Upsample}(X_c). \quad (11)$$

Focalized Optimization Strategy

A fundamental challenge in urban flow modeling is the extreme data imbalance. A few central areas exhibit extremely high flow, while the vast majority of regions have very low or near-zero flow. Standard regression losses like Mean Squared Error (MSE) are dominated by the large absolute errors in high-flow regions. This biases the model towards fitting these few areas, while neglecting the subtle but critical patterns in the more prevalent low-flow regions. Consequently, the model's performance on relative metrics (like MAPE) and its overall robustness suffer. To address this, we propose **DualFocal Loss**, a novel loss function tailored for such skewed regression tasks. It integrates two key ideas: dual-scale supervision and hard sample mining.

Dual-Scale Supervision. To ensure the model remains sensitive to errors across the entire spectrum of flow magnitudes, we integrate supervision at two complementary scales: the linear scale and the logarithmic scale. (1) *Linear scale*: We compute a standard L1 loss between the predicted flow $\hat{X}_f$ and the target X_f,i.e., $\mathcal{L}_{\text{l1}} = |\hat{X}_f - X_f|$. This term emphasizes accuracy in absolute flow values, particularly benefiting high-flow regions. (2) *Logarithmic scale*: To enhance sensitivity in low-flow regions, we introduce a second component, $\mathcal{L}_{\log} = |\log(1+\hat{X}_f) - \log(1+X_f)|$, which applies the L1 norm after a $\log(1+x)$ (ensures numerical stability for zero flows) transformation. The logarithmic compression of the large dynamic range of urban flow amplifies the relative importance of errors in low-flow areas. This encourages the model to achieve high fidelity in both bustling centers and quiet residential zones. The combined dual-scale loss is formulated as a weighted sum $\mathcal{L}_{\text{ds}} = \mathcal{L}_{\text{l1}} + \lambda \cdot \mathcal{L}_{\log}$, where λ balances the contribution of the two scales.

Focalized for Hard Sample Mining. While dual-scale supervision balances the learning across regions with varying flow magnitudes, it treats all predictions within those regions equally. To compel the model to focus on more difficult predictions, we introduce a focal mechanism inspired by Focal Loss (Lin et al. 2017b). We dynamically re-weight the contribution of each sample to the loss based on its current prediction difficulty (i.e., its $\mathcal{L}_{\text{ds}}$ value).

$$\mathcal{L}_{\text{Dual-Focal}} = (f(\beta \cdot \mathcal{L}_{\text{ds}}))^{\gamma} \cdot \mathcal{L}_{\text{ds}}, \quad (12)$$

where the term $(f(\beta \cdot \mathcal{L}_{\text{ds}}))^{\gamma}$ is a modulating factor that increases the weight of hard-to-predict samples (those with large $\mathcal{L}_{\text{ds}}$). This forces the model to allocate more of its capacity to overcome challenging cases, improving overall accuracy and robustness. By jointly leveraging dual-scale supervision and focalized re-weighting, the proposed Dual-Focal Loss provides a comprehensive optimization strategy tailored to the unique challenges of FUFI task.

5 Experiments

In this section, we conduct extensive experiments to validate our proposed framework. We first benchmark PLGF against state-of-the-art methods to demonstrate its superiority. Subsequently, we present extensive ablation studies to dissect the contribution of each key architectural and optimization component, and analyze the model's overall efficiency.

Experimental Setup

Datasets. To comprehensively assess the effectiveness of our proposed framework, we conduct experiments on the widely used public TaxiBJ benchmark dataset (Liang et al. 2019). This large-scale dataset records city-wide taxi flows in Beijing over four distinct periods, each often considered as an independent real-world scenario, we denote as **Task-1** to **Task-4**. Each data sample consists of a coarse-grained flow map (32×32) and its corresponding fine-grained ground truth of (128×128), and an external feature vector such as weather, time, and other contextual information. We follow the standard data partitioning protocol: the temporally ordered data is split into training, validation, and test sets with a ratio of 7:1:2, ensuring reproducibility.

Baselines. We benchmark our model against a broad range of representative methods. First, we include several classic methods from the computer vision domain to establish a foundational baseline, including SRCNN (Dong et al. 2015), ESPCN (Shi et al. 2016), VDSR (Kim, Lee, and Lee 2016), DeepSD (Vandal et al. 2017), and SRResNet (Ledig et al. 2017). Also, we compare against a set of SOTA models designed specifically for the FUFI task, including UrbanFM (Liang et al. 2019), UrbanPy (Ouyang et al. 2020), DeepLGR (Liang et al. 2021), FODE (Zhou et al. 2020), UrbanODE (Zhou et al. 2021), CUFAR (Yu et al. 2023), and UNO (Gao et al. 2024).

Metrics. In line with standard practice, we evaluate model performance using three widely adopted metrics, which together capture both absolute and relative prediction accuracy: Mean Square Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), where lower values indicate better performance.

	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
PLGF	**14.408**	**1.890**	**0.281**	**17.364**	**2.098**	**0.274**	**18.765**	**2.179**	**0.268**	**11.327**	**1.716**	**0.270**
Δ (%)	↓1.93	↓1.92	↓5.39	↓2.02	↓2.33	↓5.52	↓1.61	↓1.71	↓3.94	↓1.62	↓1.15	↓2.17

Table 1: Overall performance comparison with baseline methods.

Implementation. All experiments are conducted on a server equipped with NVIDIA 4090 GPUs, and the results are reported as the average of five runs.

Overall Performance

Table 1 reports the comprehensive performance comparison of our proposed PLGF framework against a suite of baselines. The results clearly demonstrate the superiority of our approach. As expected, FUFI-specific models consistently outperform general-purpose super-resolution methods, underscoring the necessity of domain-specific architectural designs. Among the FUFI-specific baselines, recent competitive models such as CUFAR and UNO demonstrate the most competitive performance, setting a high benchmark for accuracy. Notably, compared to the best prior baseline (UNO), PLGF yields relative improvements of 1.6%–2.0% in MSE, 1.2%–2.3% in MAE, and up to 2.1%–5.5% in MAPE across different tasks. This pronounced reduction in MAPE is particularly significant, as it demonstrates PLGF's superior ability to capture subtle, low-activity flow patterns, directly addressing the core challenge of the highly imbalanced, non-uniform data distributions motivating our work. Furthermore, the consistent improvements across all temporal splits highlight the robustness and generalization capability of our approach, confirming that the architectural innovations and the Focalized optimization strategy not only enhance overall accuracy but also ensure reliable performance in both high-flow and low-flow regions. Collectively, by explicitly addressing the imbalance and context sensitivity of urban flow data, PLGF achieves new SOTA and sets a strong foundation for future urban computing research.

Efficiency Evaluation

Figure 4 provides a comprehensive efficiency comparison among representative FUFI models, where the horizontal axis indicates the average training time per epoch, the vertical axis reflects MAE performance, and the bubble size

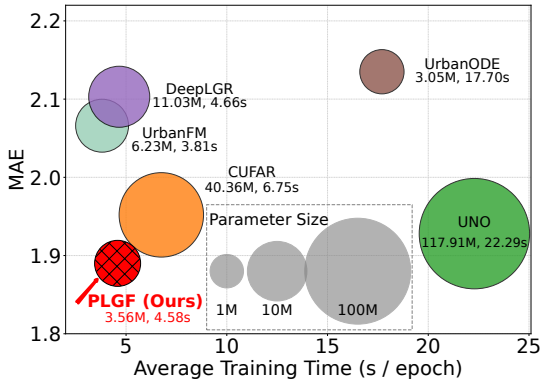

Figure 4: Model efficiency comparison.

denotes the number of model parameters. As shown, our proposed PLGF achieves state-of-the-art accuracy (lowest MAE) while maintaining remarkable efficiency. Specifically, PLGF requires only 3.56M parameters and attains a 10% lower MAE than UrbanFM (2.06) and DeepLGR (2.10) under a comparable parameter scale, while requiring a similar or shorter training time per epoch. Furthermore, when compared to UNO, which achieves a similar MAE but relies on a model that is over 33 times larger (117.91M parameters) and requires nearly 5 times longer training per epoch (22.29s vs. 4.58s), PLGF demonstrates an outstanding improvement in both computational and memory efficiency. This result highlights the superior efficiency of PLGF: it delivers advanced performance with a dramatic reduction in computational cost and model size (approximately 97%).

Generalizability Analysis of DualFocal Loss

To further demonstrate the universality and effectiveness of our DualFocal loss, we conduct plug-and-play experiments in Task 2 by integrating it into several representative FUFI models, including UrbanFM, CUFAR, and UNO. As illus-

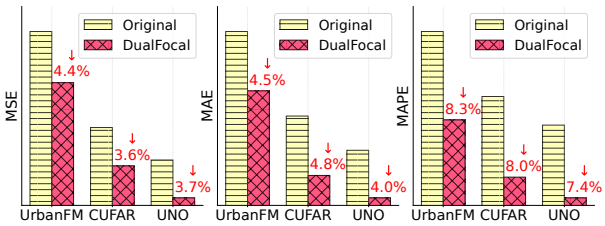

Figure 5: Plug-and-play effectiveness of DualFocal loss.

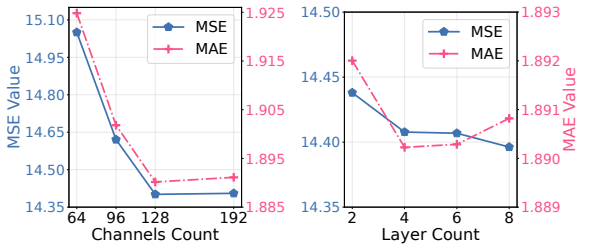

Figure 6: Parameters setting and analysis.

trated in Figure 5, DualFocal consistently delivers substantial performance improvements across all evaluation metrics. Specifically, DualFocal reduces MSE by 4.4%, 3.6%, and 3.7% on UrbanFM, CUFAR, and UNO, respectively, while also lowering MAE by up to 4.8%. Most notably, the gains in MAPE are particularly pronounced: DualFocal achieves relative reductions of 8.3% (UrbanFM), 8.0% (CUFAR), and 7.4% (UNO). The result underscores the core design of our loss function; by focusing the optimization on difficult, high-variance samples, it successfully addresses the challenge of skewed data distribution and forces the models to pay closer attention to relative errors, which is particularly critical in low-flow regions. In addition, it validates DualFocal loss as a general, effective, and easily adoptable optimization strategy that can seamlessly enhance a wide range of existing models, making it a promising choice for future FUFI research and applications.

Paramerters Studies

To identify the optimal balance between performance and complexity, we analyzed the impact of model width (number of channels) and depth (number of local-global feature extraction layers). The results, summarized in Figure 6, reveal a clear trade-off for both dimensions. In the left panel, we observe that increasing the channel count from 64 to 128 leads to a substantial reduction in both MSE and MAE, indicating that a richer feature representation significantly enhances model accuracy. However, further increasing the channel number to 192 yields only marginal gains, suggesting diminishing returns beyond a certain embedding size. The right panel examines the effect of altering the number of feature extraction layers. Both MSE and MAE decrease as the layer count increases from 2 to 4, reflecting the benefit of deeper hierarchical feature extraction. However, as the depth continues to grow, the improvements plateau and may even slightly fluctuate, implying that excessive depth does not necessarily translate into better performance and may introduce unnecessary computational overhead. Overall, these results highlight the importance of carefully balancing width (e.g., 128) and depth (e.g., 4): sufficient channel capacity and moderate network depth are both crucial for achieving optimal performance without excessive complexity.

Method	MSE	MAE	MAPE
PLGF	14.408	**1.890**	**0.281**
wo/ FiLM	14.481	1.894	0.282
wo/ Local-Global Fusion	17.745	2.049	0.288
wo/ Context-Gated Attention	18.638	2.095	0.294
wo/ DualFocal	14.574	1.926	0.304
wo/ Logarithmic	**14.349**	1.902	0.301

Table 2: Ablation study for core component in PLGF.

Ablation Studies

To systematically evaluate the contribution of each component within PLGF, we conduct ablation experiments by incrementally removing key modules and reporting the resulting performance (Table 2). The findings provide clear empirical support for the core design motivations of our framework. First, the Local-Global Feature Fusion and Context-Gated Attention modules are both crucial for accurate urban flow inference, as evidenced by the significant performance drop when either is removed. Local-Global Feature Fusion enables the model to jointly capture fine-grained spatial details and broad contextual patterns, while Context-Gated Attention ensures adaptive recalibration based on dynamic external and contextual signals. Together, they validate our motivation that effective urban flow modeling must jointly leverage multi-scale spatial information and dynamically adapt to contextual variations. The DualFocal Loss is also essential, notably improving MAPE and effectively addressing the imbalance and long-tail distribution of urban flows, thus ensuring accurate predictions for both high-flow and low-flow regions. Other components, such as FiLM-based feature modulation and logarithmic transformation, offer additional but smaller improvements. FiLM promotes flexible feature adaptation, while the logarithmic transformation enhances robustness to outliers, as reflected by slightly higher MSE and consistent gains in MAE and MAPE.

In summary, these ablation results strongly confirm the necessity and effectiveness of our architectural choices and optimization strategies, each of which is closely rooted in the unique challenges and dynamics of the fine-grained urban flow inference task.

6 Conclusion

In this work, we proposed a novel lightweight framework for fine-grained urban flow inference that systematically tackles two critical barriers to practical deployment: model bloat and unfocused optimization. Specifically, our PLGF model introduces a Progressive Local-Global Fusion strategy, and the DualFocal Loss provides a powerful, adaptive learning mechanism that is sensitive to the skewed data distributions.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 62506097, No.62502404, and No. U25A20530, U25B2045. This work is also supported by Shenzhen Basic Research Program Natural Science Foundation under Grant No. JCYJ20250604145542055, Hong Kong Research Grants Council (Research Impact Fund No.R1015-23, Collaborative Research Fund No.C1043-24GF, General Research Fund No.11218325), Institute of Digital Medicine of City University of Hong Kong (No.9229503), Huawei (Huawei Innovation Research Program), Tencent (CCF-Tencent Open Fund, Tencent Rhino-Bird Focused Research Program), Alibaba (CCF-Alimama Tech Kangaroo Fund No. 2024002), Didi (CCF-Didi Gaia Scholars Research Fund), Kuaishou, and Bytedance.

References

Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2015. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2): 295–307.

Gao, Q.; Song, X.; Huang, L.; Trajcevski, G.; Zhou, F.; and Chen, X. 2024. Enhancing Fine-Grained Urban Flow Inference via Incremental Neural Operator. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 5826–5834. International Joint Conferences on Artificial Intelligence Organization.

Kim, J.; Lee, J. K.; and Lee, K. M. 2016. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1646–1654.

Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681–4690.

Li, J.; Wang, S.; Zhang, J.; Miao, H.; Zhang, J.; and Yu, P. S. 2022. Fine-grained urban flow inference with incomplete data. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5851–5864.

Liang, Y.; Ouyang, K.; Jing, L.; Ruan, S.; Liu, Y.; Zhang, J.; Rosenblum, D. S.; and Zheng, Y. 2019. Urbanfm: Inferring fine-grained urban flows. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 3132–3142.

Liang, Y.; Ouyang, K.; Wang, Y.; Liu, Y.; Zhang, J.; Zheng, Y.; and Rosenblum, D. S. 2021. Revisiting convolutional neural networks for citywide crowd flow analytics. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, 578–594. Springer.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017a. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017b. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*.

Ouyang, K.; Liang, Y.; Liu, Y.; Tong, Z.; Ruan, S.; Zheng, Y.; and Rosenblum, D. S. 2020. Fine-grained urban flow inference. *IEEE transactions on knowledge and data engineering*, 34(6): 2755–2770.

Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Qu, H.; Gong, Y.; Chen, M.; Zhang, J.; Zheng, Y.; and Yin, Y. 2022. Forecasting fine-grained urban flows via spatio-temporal contrastive self-supervision. *IEEE Transactions on Knowledge and Data Engineering*, 35(8): 8008–8023.

Shi, W.; Caballero, J.; Huszar, F.; Totz, J.; Aitken, A. P.; Bishop, R.; Rueckert, D.; and Wang, Z. 2016. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network . In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1874–1883. Los Alamitos, CA, USA.

Vandal, T.; Kodra, E.; Ganguly, S.; Michaelis, A.; Nemani, R.; and Ganguly, A. R. 2017. 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*, 1663–1672.

Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; and Hu, Q. 2020. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11531–11539.

Wang, R.; Liu, Y.; Gong, Y.; Liu, W.; Chen, M.; Yin, Y.; and Zheng, Y. 2023. Fine-grained urban flow inference with unobservable data via space-time attraction learning. In *2023 IEEE International Conference on Data Mining (ICDM)*, 1367–1372. IEEE.

Wang, S.; Cao, J.; and Philip, S. Y. 2020. Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, 34(8): 3681–3700.

Xie, P.; Li, T.; Liu, J.; Du, S.; Yang, X.; and Zhang, J. 2020. Urban flow prediction from spatiotemporal data using machine learning: A survey. *Information Fusion*, 59: 1–12.

Yu, H.; Xu, X.; Zhong, T.; and Zhou, F. 2023. Overcoming forgetting in fine-grained urban flow inference via adaptive knowledge replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5393–5401.

Zheng, Y.; Cai, Y.; Cai, Z.; Fan, C.; Wang, S.; and Wang, J. 2024a. FGITrans: Cross-City Transformer for Fine-grained Urban Flow Inference. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 3415–3423.

Zheng, Y.; Wu, J.; Cai, Z.; Wang, S.; and Wang, J. 2024b. AdaTM: Fine-grained Urban Flow Inference with Adaptive

Knowledge Transfer across Multiple Cities. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 3424–3432.

Zheng, Y.; Zhong, L.; Wang, S.; Yang, Y.; Gu, W.; Zhang, J.; and Wang, J. 2023. Diffuflow: Robust fine-grained urban flow inference with denoising diffusion model. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 3505–3513.

Zhou, F.; Jing, X.; Li, L.; and Zhong, T. 2021. Inferring high-resolutional urban flow with internet of mobile things. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7948–7952. IEEE.

Zhou, F.; Li, L.; Zhong, T.; Trajcevski, G.; Zhang, K.; and Wang, J. 2020. Enhancing urban flow maps via neural odes. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,{IJCAI} 2020*.