Responsive Dynamic Graph Disentanglement for Metro Flow Forecasting

Qiang Gao[1,2], Zizheng Wang[1], Li Huang[1,2*], Goce Trajcevski[3], Guisong Liu[1,2,4], Xueqin Chen[4*]

[1]School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, China
[2]Engineering Research Center of Intelligent Finance, Ministry of Education, Chengdu, China
[3]Iowa State University, Iowa, USA
[4]Kash Institute of Electronics and Information Industry, Kashgar, China
qianggao@swufe.edu.cn, 223081200032@smail.swufe.edu.cn, lihuang@swufe.edu.cn, gocet25@iastate.edu, gliu@swufe.edu.cn, nedchen0728@gmail.com

Abstract

The metro flow in Urban Rail Transit Systems (URTS) differs from other urban traffic flows because it is characterized by: (1) highly predetermined scheduling; and (2) interactively dynamic dependencies over the fixed physical infrastructure that vary with spatiotemporal and environmental factors. Notwithstanding the advances in graph neural networks, existing efforts fail to fully capture the characteristics and complex spatiotemporal dynamics specific to metro flow, as the innate graph-aware interactions underlying a metro flow are frequently affected by an amalgamation of: intrinsic connectivity, environmental associations, and flow-activated correlation, which usually dynamically evolve over time while containing redundant signals. We propose **ReDyNet**, a novel **Re**sponsive **Dy**namic Graph Neural **Net**work to accurately understand the spatiotemporal dynamics of metro flow and external factors. Specifically, it employs a responsive mechanism that adapts to variations in metro flow and external influences, ensuring the construction of an appropriate dynamic graph. In addition, ReDyNet follows the merits of information bottleneck (IB) theory with redundancy disentanglement to enhance the clarity and precision of contextual spatial signals. Our experiments conducted on three real-world metro passenger flow datasets demonstrate that the proposed ReDyNet outperforms several representative baselines.

1 Introduction

Urban Rail Transit Systems (URTS) (Zhu et al. 2023) such as metro networks have become indispensable means of transport for residents in metropolitan areas. In this context, the extensive use of sensing technologies and passenger information systems yields vast amounts of passenger flow data, which spurs unprecedented opportunities to constantly improve the service quality of URTS. As a vital element in the development of smart cities, accurate metro flow forecasting is essential for effective transportation management, operational efficiency, and passenger convenience (Xiong et al. 2019; Xie et al. 2023). Specifically, forecasting metro flow requires estimating the number of passengers who enter and leave metro stations at different time intervals, which is not trivial due to the intricate spatiotemporal dynamics inherent in metro systems.

*Corresponding authors: Li Huang and Xueqin Chen

Due to intrinsic periodicity, recent advances in the traffic flow forecasting domains provide a straightforward paradigm to handle the metro flow forecasting problem; that is, we can model higher-order temporal dependencies from historical flow observations, while considering spatial constraints, to discover the periodicity for future trend forecasting (Liu, Liu, and Jia 2019). For instance, we can leverage popular graph neural networks (GNNs) to model the relational structure of metro stations and their dynamic interactions. The common pipeline is that (dynamic) GNNs attempt to integrate spatial and temporal signals into unified representations, significantly enhancing prediction accuracy and robustness (Li et al. 2017; Yu, Yin, and Zhu 2018). These sophisticated solutions can adapt to the common characteristics of metro and traffic flows – e.g., periodicity and topological structures – providing opportunities to improve transportation management and operational efficiency.

However, metro flow differs from other types of urban traffic flow due to its specific properties: (1) It contains characteristics of highly predetermined scheduling, marked by greater reliability. Each station underlies unique traits, and the flow patterns typically show consistency (e.g., chain reaction) in addition to strong periodicity. For example, an increase in the number of passengers entering one station naturally results in a similar rise in the number of passengers exiting the subsequent stations since passengers do not linger at the station or suddenly disappear from it. (2) The spatially-aware interaction regarding higher-order dependencies is inherently dynamic over the fixed physical infrastructure, with varying flow directions and aggregation patterns that differ significantly between weekdays and weekends, and even fluctuate throughout the day. This dynamic nature requires responsive graph structures and pattern aggregation to capture these intrinsic dynamics.

With these in mind, we raise three key concerns that are crucial for exposing interactive correlations to effectively model metro flow dynamics, as depicted in Fig. 1. *K1-Intrinsic Connectivity*: pertains to the diverse spatially-aware correlations between different stations, such as geographical proximity and similar surrounding environments, yielding analogous flow patterns and dynamic station connections. *K2-Environmental Associations*: encompass external influences such as commuting date, current weather and habitual activities, which exposes the factual diver-

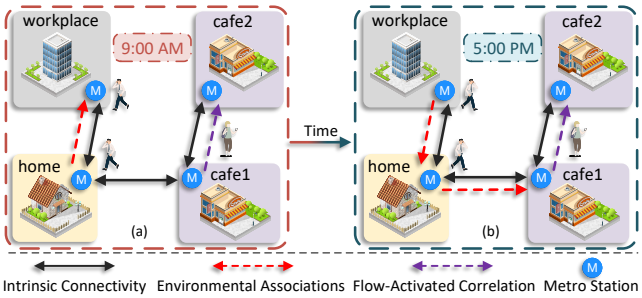


Figure 1: A toy example showing key concerns in metro flow modeling. (a) Morning (9:00 AM) and (b) Evening (5:00 PM) illustrate the dynamic relationships among city regions.

sity of flow evolution. For instance, weekdays typically exhibit stronger connections between residential areas and workplaces, while weekends highlight connections between residential areas and recreational sites. Moreover, adverse weather conditions also affect flow patterns, generally reducing passenger movement. *K3-Flow-Activated Correlation*: involves dependencies influenced by intricate changes in passenger flow. For example, a sudden increase in flow at a particular station prompts passengers to transfer to other stations, thereby reinforcing connections between stations.

To tackle the above concerns, we introduce **ReDyNet**, a novel **Re**sponsive **Dy**namic Graph Neural **Net**work tailored for metro flow forecasting, specifically focusing on constructing a responsive and spatially-aware structure for temporal-sensitive metro flow. By dynamically modulating the graph signals rather than the overuse of predefined/static connections such as geographical proximity, ReDyNet starts with the responsive evolutionary relevance of the different station flows in latent space to facilitate the discernment of nuanced and contextually relevant station dependencies and interrelationships (*K1*), along with considering the environmental context (*K2*) and flow pattern correlation (*K3*) of each station. To underscore the spatially-aware dynamics of metro systems, ReDyNet leverages the regular characteristics and higher-order dependencies of metro flow to avoid extensive transformations that could result in oversmoothing and distortion issues, rather than maintaining predefined factual relationships between different stations such as unchangeable geographical connectivity. More importantly, motivated by recent Information Bottleneck (IB) principles we devise a *Redundancy Context Disentanglement* to effectively disentangle graph signals into *redundancy* and *essentiality*, primarily seeking to mitigate the impact of redundant information introduced by the integration of excessive factors. This disentanglement compresses essential representations, clarifies graph signals, and enhances forecast performance. Our contributions are summarized as follows:

- We introduce ReDyNet, a novel forecasting solution that constructs a spatially responsive structure for temporal-sensitive metro flow without using a predefined graph structure. By dynamically modulating, it effectively captures nuanced and contextually relevant station dependencies, overcoming the limits of static relation learning.
- We propose to disentangle redundant and essential components from contextual spatial signals, which enhances the learning of dynamic interactions tailored to each flow observation from the majority of task-relative semantics.
- Experiments on three real-world metro passenger flow datasets demonstrate that ReDyNet consistently outperforms multiple baselines, confirming its robustness and effectiveness in accurately forecasting metro passenger flow.

2 Problem and Methodology

We now provide the basic terminology and formally define the problem, followed by an overview of ReDyNet and a detailed discussion of its main components.

Definition 1 (Metro Network). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a metro network in a city, where $\mathcal{V}$ represents a set of $n = |\mathcal{V}|$ stations and $\mathcal{E}$ is a set of edges that depict spatial connectivity between different stations. Usually, we use an adjacent matrix $\mathcal{A} \in [0,1]^{n\times n}$ to describe station connectivity.*

Definition 2 (Metro Flow). *Let $\boldsymbol{\mathcal{X}}^{t-\omega+1:t} \in \mathbb{R}^{n\times c\times\omega}$ represent historical metro flows with the observation window ω, where any $\boldsymbol{\mathcal{X}}_\tau \in \mathbb{R}^{n\times c}$ is a graph signal depicting the metro observations of n stations with c situations at time step τ. Herein, c refers to the types of metro situations/conditions, e.g., entry and exit flows. Note that, in most cases, we will omit the superscript of $\boldsymbol{\mathcal{X}}^{t-\omega+1:t}$ for simplicity.*

Definition 3 (Problem: Metro Flow Forecasting). *By only giving metro flow $\boldsymbol{\mathcal{X}}^{t-\omega+1:t} \in \mathbb{R}^{n\times c\times\omega}$, instead of relying on $\mathcal{G}$, we aim to learn a model $\mathcal{M}$ to forecast metro trends (i.e., volumes) of next ω time steps $\widehat{\boldsymbol{Y}}^{t+1:t+\omega} \in \mathbb{R}^{n\times c\times\omega}$ by incorporating the context of date $\boldsymbol{\mathcal{D}}$ and weather $\boldsymbol{\mathcal{W}}$:*

$$\widehat{\boldsymbol{Y}}^{t+1:t+\omega} = \mathcal{M}(\boldsymbol{\mathcal{X}}^{t-\omega+1:t} | (\boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{W}})). \quad (1)$$

Fig. 2 illustrates the framework skeleton of ReDyNet, which contains four modules: Contextual Spatial Embedding, Redundancy Context Disentanglement, Responsive Dynamic Learning and Task Adaption, discussed next.

2.1 Contextual Spatial Embedding (CSE)

CSE aims to distill spatial-inherent contexts (e.g., historical temporal dynamics and commute date) into a unified latent space, primarily seeking to offer a dynamic spatial-aware signal in the temporal domain rather than relying on a static and predefined spatial design (i.e., graph $\mathcal{G}$). Intuitively, we usually use an adjacent matrix $\mathcal{A}$ to describe spatial correlations between different stations. In contrast, we attempt to use embeddings (e.g., $\boldsymbol{F}_o \in \mathbb{R}^{n\times d_e}$) that account for temporal dynamics to specify the spatial interactions between different stations. As revealed by previous efforts (Bai et al. 2020; Lan et al. 2022), the spatial interaction in the latent space between two stations can be described by multiplying $\boldsymbol{F}_o$ and $\boldsymbol{F}_o^\top$, which can generally be summarized as:

$$\tilde{\mathcal{A}} = \boldsymbol{D}^{-\frac{1}{2}}\mathcal{A}\boldsymbol{D}^{-\frac{1}{2}} = \text{Softmax}(\text{ReLU}(\boldsymbol{F_o} \cdot \boldsymbol{F_o}^\top)), \quad (2)$$

where $\tilde{\mathcal{A}} \in \mathbb{R}^{n\times n}$ is the normalized adjacency matrix constrained by the degree matrix $\boldsymbol{D}$. With this basis in mind, it is essential to well prepare the spatial-inherent embeddings as

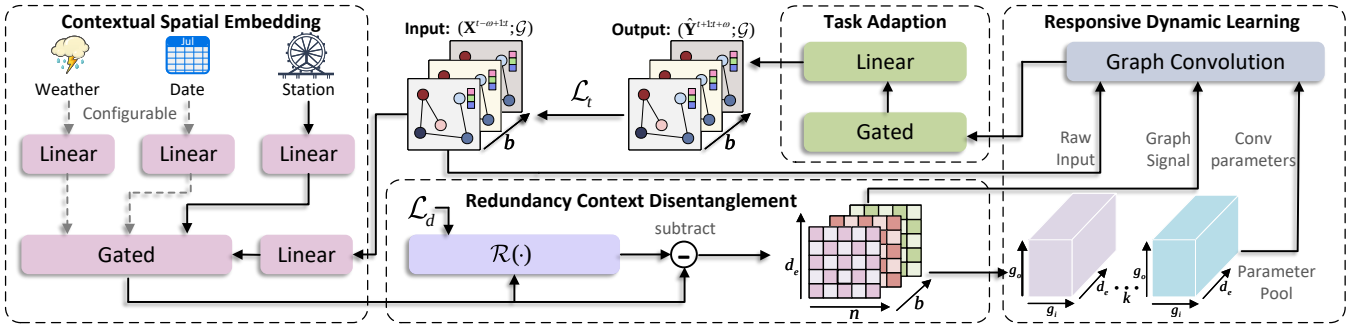


Figure 2: The network skeleton of our proposed ReDyNet.

a prerequisite for downstream dynamic interaction learning, along with considering the environmental associations.

In this study, in addition to fusing station-based flow dynamics (i.e., *Temporal-aware Spatial Embedding*), we devise two additional blocks, i.e., *Date Embedding* and *Weather Embedding*, to address distinct environmental associations corresponding to station identity flexibly.

Temporal-aware Spatial Embedding. Given the historical metro flow $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n\times c\times \omega}$, it seeks to operate the past flow dynamics as an intrinsic characteristic of each station for the purpose of distilling the spatial dependency between different stations. To this end, we first diffuse the dynamics of past flow into each station's characteristics by:

$$\boldsymbol{E}_s = \text{Linear}(\text{broad}(\boldsymbol{X}, \boldsymbol{S})), \tag{3}$$

where $\boldsymbol{E}_s \in \mathbb{R}^{n\times d_e}$, $\boldsymbol{S} \in \mathbb{R}^n$ is trainable station embedding with random initialization for identity clarity purpose, and broad() is the Broadcast operation. We then use a gated operation (Chung et al. 2014; Yao, Mao, and Luo 2019) to adaptively scale the contribution of temporal-aware spatial signals. This gated operation, generally represented as $\text{Gated}(\boldsymbol{X}, \boldsymbol{E})$, can be described in the following manner:

$$\text{Gated}(\boldsymbol{X}, \boldsymbol{E}) = \psi(\boldsymbol{X}) \otimes \sigma(\boldsymbol{E}) + (1 - \sigma(\boldsymbol{E})) \otimes \boldsymbol{E}, \tag{4}$$

where $\otimes$ is the element-wise product, ψ is an activation function (herein it is *Tanh* for the purpose of retaining nonlinearity) and σ is the gated function, i.e., *Sigmoid*. Hence, we handle the temporal-ware spatial signals $\boldsymbol{E}_s$ as follows:

$$\boldsymbol{F}_s = \text{Gated}(\boldsymbol{X}, \boldsymbol{E}_s), \tag{5}$$

where $\boldsymbol{F}_s \in \mathbb{R}^{n\times d_e}$ represents the embeddings of different stations exposing the temporal-aware spatial context.

Date Embedding. We consider two granularities of the commute date, i.e., the hour-of-day and the day-of-week aspects. Akin to the above spatial embedding, the date context is determined across the metro flow as follows:

$$\boldsymbol{E}_d = \text{Linear}(\text{broad}(\boldsymbol{X}, \text{Linear}(\text{Reshape}(\boldsymbol{\mathcal{D}})))), \tag{6}$$

where $\boldsymbol{E}_d \in \mathbb{R}^{n\times d_e}$. $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{\omega\times 2}$ represent hour-of-day and day-of-week date. Then, we operate a gated function for $\boldsymbol{E}_d$:

$$\boldsymbol{F}_d = \text{Gated}(\boldsymbol{X}, \boldsymbol{E}_d). \tag{7}$$

Weather Embedding. For the weather context $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{n\times c_w\times \omega}$ (c_w refers to different weather conditions), we feed it into the linear layer for a dimension alignment purpose, which can be denoted as follows:

$$\boldsymbol{E}_w = \text{Linear}(\text{Reshape}(\boldsymbol{\mathcal{W}})), \tag{8}$$

where $\boldsymbol{E}_w \in \mathbb{R}^{n\times d_e}$. The weather embeddings $\boldsymbol{F}_w \in \mathbb{R}^{n\times d_e}$ can be obtained by a similar gated operation, denoted as:

$$\boldsymbol{F}_w = \text{Gated}(\boldsymbol{X}, \boldsymbol{E}_w). \tag{9}$$

Finally, we fuse the embeddings extracted by the above three context-aware blocks and obtain the final output $\boldsymbol{F}_o$:

$$\boldsymbol{F}_o = \text{LayerNorm}(\boldsymbol{F}_s + \boldsymbol{F}_d + \boldsymbol{F}_w). \tag{10}$$

2.2 Redundancy Context Disentanglement (RCD)

In practical terms, contextual spatial signals $\boldsymbol{F}_o$ absorbing multiple contexts could contain redundancy that is not relevant to our forecasting task (Shwartz-Ziv and Tishby 2017). Thus, we devise this module to separate redundancy from the essentiality (denoted as $\boldsymbol{F}$) underlying $\boldsymbol{F}_o$.

Technical Derivation. Inspired by (Yang et al. 2021), we introduce a redundancy function $\mathcal{R}(\cdot)$ to learn and extract spatially irrelevant information, i.e., redundancy, from the original signal, aiming at enhancing the clarity of the relevant spatial context. This process is defined as follows:

$$\boldsymbol{F} = \boldsymbol{F}_o - \mathcal{R}(\boldsymbol{F}_o) \in \mathbb{R}^{n\times d_e}, \tag{11}$$

Let $\boldsymbol{F}^*$ denote the optimal context in our task. According to Information Bottleneck (IB) theory (Tishby, Pereira, and Bialek 2000), our objective in this module is to minimize:

$$\mathcal{L}[p(\boldsymbol{F}|\boldsymbol{F}_o)] = \boldsymbol{I}(\boldsymbol{F}; \boldsymbol{F}_o) - \beta \boldsymbol{I}(\boldsymbol{F}; \boldsymbol{F}^*), \tag{12}$$

where $\boldsymbol{I}$ denotes mutual information (MI). In Eq. (12), the goal is to balance two competing objectives, that is, while the forecasting task $\mathcal{M}(\cdot)$ seeks to maximize $\boldsymbol{I}(\boldsymbol{F}; \boldsymbol{F}^*)$, ensuring that the representation $\boldsymbol{F}$ retains as much relevant information as possible about the optimal signal $\boldsymbol{F}^*$, the regularization task $\mathcal{R}(\cdot)$ aims to minimize $\boldsymbol{I}(\boldsymbol{F}; \boldsymbol{F}_o)$, thereby discarding redundant information from the input $\boldsymbol{F}_o$. More specifically, we start with the entropy of $\boldsymbol{F}_o$, defined as:

$$\boldsymbol{H}(\boldsymbol{F}_o) = \boldsymbol{I}(\boldsymbol{F}_o; \boldsymbol{F}_o) = \boldsymbol{I}(\boldsymbol{F}_o - \mathcal{R}(\boldsymbol{F}_o) + \mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o). \tag{13}$$

We can decompose $\boldsymbol{I}(\boldsymbol{F}_o; \boldsymbol{F}_o)$ using the chain rule for MI:

$$\begin{aligned} \boldsymbol{I}(\boldsymbol{F}_o; \boldsymbol{F}_o) &= \boldsymbol{I}(\boldsymbol{F} + \mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o) \\ &= \boldsymbol{I}(\boldsymbol{F}, \mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o) \\ &= \boldsymbol{I}(\boldsymbol{F}; \boldsymbol{F}_o) + \boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o | \boldsymbol{F}). \end{aligned} \tag{14}$$

We then consider the non-negativity of MI as:

$$\boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o) \geq \boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o|\boldsymbol{F}) \geq 0. \tag{15}$$

This inequality relies on two MI properties: $\boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o)$ is non-negative (≥ 0), and conditional mutual information $\boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o \mid \boldsymbol{F})$ is non-negative and less than or equal to $\boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o)$, i.e., $\boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o \mid \boldsymbol{F}) \leq \boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o)$. Upon these, we derive the following inequality:

$$\boldsymbol{I}(\boldsymbol{F}; \boldsymbol{F}_o) \geq \boldsymbol{H}(\boldsymbol{F}_o) - \boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o). \tag{16}$$

Applying the principles of MI and entropy, we can obtain:

$$\boldsymbol{I}(\mathcal{R}(\boldsymbol{F}_o); \boldsymbol{F}_o) = \boldsymbol{H}(\boldsymbol{F}_o) - \boldsymbol{H}(\boldsymbol{F}_o \mid \mathcal{R}(\boldsymbol{F}_o)). \tag{17}$$

Redundancy Bound. By combining the above Eq. (16) and Eq. (17), we can produce a lower bound for $\boldsymbol{I}(\boldsymbol{F}; \boldsymbol{F}_o)$, which can expressed as follows:

$$\boldsymbol{I}(\boldsymbol{F}; \boldsymbol{F}_o) \geq \boldsymbol{H}(\boldsymbol{F}_o \mid \mathcal{R}(\boldsymbol{F}_o)) \approx \boldsymbol{H}(\boldsymbol{F}_o \mid \boldsymbol{Z}), \tag{18}$$

where $\boldsymbol{Z} \in \mathbb{R}^{n \times d_r}$ denotes the latent variables in $\mathcal{R}(\cdot)$. Conditional entropy is defined as the expected negative log-likelihood of the data given the latent variables, denoted as:

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{F}_o \mid \boldsymbol{Z}) &= -\mathbb{E}_{p(\boldsymbol{F}_o, \boldsymbol{Z})}[\log p(\boldsymbol{F}_o|\boldsymbol{Z})] \\ &\approx \mathbb{E}_{q(\boldsymbol{Z}|\boldsymbol{F}_o)}[-\log p(\boldsymbol{F}_o|\boldsymbol{Z})]. \end{aligned} \tag{19}$$

Given the sparsity of observed spatial dependencies, we introduce the Kullback-Leibler (KL) divergence (Kingma and Welling 2013; Higgins et al. 2017) to align the learned distribution with the prior under the guidance of variational Bayes, ensuring smooth and structured latent representations. The approximate objective function is expressed as:

$$\mathcal{L}_d = \mathbb{E}_{q(\boldsymbol{Z}|\boldsymbol{F}_o)}[-\log p(\boldsymbol{F}_o|\boldsymbol{Z})] + \beta \mathrm{KL}(q(\boldsymbol{Z}|\boldsymbol{F}_o) \| p(\boldsymbol{Z})). \tag{20}$$

Herein, $q(\boldsymbol{Z}|\boldsymbol{F}_o)$ denotes the approximate posterior, which is to estimate the distribution of the latent variables $\boldsymbol{Z}$ given the input $\boldsymbol{F}_o$. Within our practice, we also operate another gated operation to model the approximate posterior:

$$\boldsymbol{Z} = \mathrm{Linear}(\psi(\mathrm{Linear}(\boldsymbol{F}_o)) \otimes \sigma(\mathrm{Linear}(\boldsymbol{F}_o))). \tag{21}$$

For $\boldsymbol{Z} \in \mathbb{R}^{n \times d_r}$, it is obtained by using the reparameterization trick, and our $\mathcal{R}(\cdot)$, in practice, is simply defined as:

$$\mathcal{R}(\boldsymbol{F}_o) = \mathrm{Linear}(\psi(\mathrm{Linear}(\boldsymbol{Z})) \otimes \sigma(\mathrm{Linear}(\boldsymbol{Z}))) \in \mathbb{R}^{n \times d_e}. \tag{22}$$

2.3 Responsive Dynamic Learning (RDL)

To capture dynamic interactions between stations under the evolution of metro flow, RDL is initially responsible for using the essentiality $\boldsymbol{F}$ to compute the dynamic adjacency matrix (cf. Eq. (2)). However, due to the potential variability of flow patterns across stations over time, we collect a set of flow pieces using different sampling dates to accommodate possible pattern variations. Thus, Eq. (2) can be revised as:

$$\tilde{\mathcal{A}} = \boldsymbol{D}^{-\frac{1}{2}} \mathcal{A} \boldsymbol{D}^{-\frac{1}{2}} = \mathrm{Softmax}(\mathrm{ReLU}(\boldsymbol{F} \cdot \boldsymbol{F}^\top)), \tag{23}$$

where $\tilde{\mathcal{A}} \in \mathbb{R}^{b \times n \times n}$ and b refers to the collection size. Like most prior efforts using graph convolution, we implement the graph convolution operation using Chebyshev polynomials (Defferrard, Bresson, and Vandergheynst 2016) to approximate the eigenvalues of the graph Laplacian which eliminates the need to directly compute the eigendecomposition of the graph Laplacian, summarized as follows:

$$\boldsymbol{T}_0 = I,\ \boldsymbol{T}_1 = \tilde{\mathcal{A}},\ \boldsymbol{T}_k = 2\tilde{\mathcal{A}}\boldsymbol{T}_{k-1} - \boldsymbol{T}_{k-2},\ k \geq 2. \tag{24}$$

Herein, k is the order of the Chebyshev polynomials. However, the current solutions using Chebyshev polynomials usually rely on building a shared convolution (parameters) for all flow observations, narrowing the diversity in pattern capture. Thus, we devise a dynamic convolution parameter selection strategy to accommodate different aggregation patterns for different flow observations. Specifically, the convolutional weight $\boldsymbol{W}$ is selected from a learnable pool $\boldsymbol{W}_p$ based on the $\boldsymbol{F}$, which can be obtained using *einsum* as:

$$\boldsymbol{W} = \sum\nolimits_{e=1}^{d_e} (\boldsymbol{F}_e \cdot \boldsymbol{W}_{p,e}) \in \mathbb{R}^{b \times n \times k \times g_i \times g_o}. \tag{25}$$

Herein, $\boldsymbol{W}_p \in \mathbb{R}^{d_e \times k \times g_i \times g_o}$ refers to the parameter pool of graph convolution and $g_i \times g_o$ scales the kernel size. Similarly, the bias $\boldsymbol{b}$ in graph convolution can be obtained by:

$$\boldsymbol{b} = \boldsymbol{F} \cdot \boldsymbol{b}_p \in \mathbb{R}^{b \times n \times g_o}, \tag{26}$$

where $\boldsymbol{b}_p \in \mathbb{R}^{g_i \times g_o}$ is a trainable parameter pool of biases.

Due to the regularity of metro flow and the dynamic nature of spatial-aware structure, we use raw $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{b \times n \times c \times \omega}$ as input to the graph convolution. The convolution parameters and the structure signals are highly dynamic, allowing us to capture the intrinsic interactions of the underlying changes from the raw data. We define our convolution as:

$$\boldsymbol{C} = \pi(\sum_{k=0}^{k} \boldsymbol{W}_k \boldsymbol{T}_k(\tilde{\mathcal{A}}) \mathrm{Reshape}(\boldsymbol{\mathcal{X}}) + \boldsymbol{b}) \in \mathbb{R}^{b \times n \times g_o}, \tag{27}$$

where $\boldsymbol{W}_k \in \mathbb{R}^{b \times n \times g_i \times g_o}$ and here π empirically determined to be a *GELU* activation function.

2.4 Task Learning

Task Adaption. To accommodate the downstream forecast of future flow, we first operate the above latent results $\boldsymbol{C}$ derived from RDL with the gated mechanism for the purpose of modulating the output non-linearity. Specifically, the gate settings can be summarized as follows:

$$\boldsymbol{p} = \sigma(\mathrm{Linear}(\boldsymbol{C})), \quad \boldsymbol{q} = \psi(\mathrm{Linear}(\boldsymbol{C})), \tag{28}$$

where $\boldsymbol{p}$ controls the gating level and $\boldsymbol{q}$ adjusts the non-linearity. Subsequently, the final forecast $\widehat{\boldsymbol{Y}} \in \mathbb{R}^{n \times c \times \omega}$ is:

$$\widehat{\boldsymbol{Y}} = \mathrm{Reshape}(\mathrm{Linear}(\boldsymbol{p} \otimes \boldsymbol{q} + (\boldsymbol{1} - \boldsymbol{p}) \otimes \boldsymbol{C})). \tag{29}$$

Final Objective. Like common routines in flow data optimization (Song et al. 2020; Lan et al. 2022; Xie et al. 2023), we employ the Huber loss as the task objective, which is less sensitive to outliers than the squared error loss, denoted as:

$$\mathcal{L}_t = \begin{cases} \frac{1}{2}(\boldsymbol{Y} - \widehat{\boldsymbol{Y}})^2 & \text{for } |\boldsymbol{Y} - \widehat{\boldsymbol{Y}}| \leq \delta, \\ \delta(|\boldsymbol{Y} - \widehat{\boldsymbol{Y}}| - \frac{1}{2}\delta & \text{otherwise.} \end{cases} \tag{30}$$

Herein, δ is a threshold used to control the transition point between square loss and absolute loss. Meanwhile, we integrate the disentangle objective (cf. Eq. (20)) and treat this objective as a weak signal to enhance task learning. In sum, our final objective is denoted as: $\mathcal{L} = \mathcal{L}_t + \alpha \mathcal{L}_d$, where hyperparameter α is a trade-off between the two losses.

	Model	15min			30min			45min			60min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Beijing Metro	DCRNN	32.45	67.23	28.61%	38.54	81.80	37.25%	47.07	103.12	55.01%	55.40	125.22	86.58%
	STGCN	32.16	62.62	33.66%	37.85	71.94	46.29%	44.96	84.52	79.80%	50.89	96.74	158.07%
	AGCRN	25.17	47.87	23.97%	25.32	47.22	26.69%	26.29	48.95	35.99%	26.93	50.88	53.62%
	STTN	35.61	78.42	36.47%	32.74	63.38	32.84%	33.20	62.40	44.69%	35.81	68.61	91.78%
	FGNN	27.34	49.22	27.50%	28.30	51.32	29.45%	29.78	54.55	31.58%	31.62	58.51	34.57%
	ASTGCN	26.43	48.91	28.17%	26.28	48.76	29.03%	26.27	48.71	29.24%	26.92	49.73	30.81%
	GMAN	24.27	40.10	28.91%	23.75	40.07	26.88%	24.04	40.89	27.21%	24.77	42.30	30.70%
	GWNet	22.45	39.37	21.29%	22.81	40.26	22.28%	23.31	41.46	23.04%	23.87	42.77	23.70%
	STDGRL	21.85	41.23	20.15%	22.34	42.35	21.67%	22.81	43.18	29.08%	22.89	43.37	43.93%
	PDFormer	20.96	36.12	20.50%	21.29	37.10	21.58%	21.69	38.16	21.99%	22.12	39.25	22.75%
	ReDyNet	**19.36**	**34.05**	**18.93%**	**19.57**	**34.53**	**19.33%**	**19.81**	**35.08**	**19.54%**	**20.07**	**35.66**	**19.88%**
Shanghai Metro	DCRNN	27.94	54.24	26.33%	31.92	63.95	29.37%	37.22	79.20	31.57%	42.07	93.81	34.35%
	STGCN	28.27	52.26	31.36%	31.87	59.38	35.27%	36.92	70.13	40.05%	42.04	81.21	44.31%
	AGCRN	24.01	47.11	23.16%	25.46	50.96	24.70%	27.04	55.57	26.47%	28.41	59.61	26.96%
	STTN	29.03	56.20	26.61%	29.30	57.85	25.78%	30.21	60.49	26.29%	30.97	60.73	26.69%
	FGNN	28.21	54.06	26.57%	29.71	57.56	27.87%	31.49	61.99	29.48%	33.42	66.61	31.12%
	ASTGCN	26.26	50.05	26.90%	26.29	50.23	26.54%	26.70	51.37	26.20%	27.61	53.30	26.85%
	GMAN	25.70	48.11	32.27%	25.68	48.97	30.75%	26.11	50.08	30.55%	26.82	51.36	31.83%
	GWNet	22.24	41.98	21.17%	22.76	43.83	21.09%	23.34	45.66	21.29%	23.99	47.59	21.77%
	STDGRL	23.72	46.87	21.43%	24.38	49.29	21.66%	25.42	52.90	22.48%	26.58	57.40	23.41%
	PDFormer	22.07	43.24	19.03%	22.56	44.56	19.26%	23.19	46.52	19.72%	23.88	48.60	20.18%
	ReDyNet	**21.15**	**40.54**	**17.73%**	**21.66**	**42.16**	**17.90%**	**22.23**	**43.97**	**18.14%**	**22.86**	**45.83**	**18.53%**
Hangzhou Metro	DCRNN	27.11	49.52	22.80%	31.23	58.23	26.16%	36.90	70.97	28.55%	42.75	85.05	32.43%
	STGCN	28.24	49.05	30.32%	32.23	56.21	35.48%	37.76	65.94	42.39%	44.58	77.80	61.17%
	AGCRN	23.62	40.35	23.35%	24.94	43.19	26.47%	25.91	45.28	25.44%	27.40	46.78	31.34%
	STTN	28.12	48.47	24.08%	28.81	49.05	27.53%	28.62	49.60	25.27%	30.63	52.40	35.37%
	FGNN	25.73	43.88	23.41%	26.83	45.91	24.09%	28.31	48.81	24.83%	30.04	52.35	26.25%
	ASTGCN	26.22	45.29	24.65%	27.25	47.09	25.47%	28.71	49.92	26.27%	30.45	53.52	27.44%
	GMAN	24.15	39.32	23.43%	24.62	40.90	21.47%	25.42	42.80	21.43%	26.14	44.20	21.98%
	GWNet	22.28	37.53	20.77%	22.75	38.36	20.79%	23.55	39.99	21.30%	24.20	41.32	21.40%
	STDGRL	23.27	39.55	20.91%	23.77	40.43	21.41%	24.89	42.88	22.30%	25.83	45.18	25.70%
	PDFormer	22.47	37.85	19.96%	22.79	38.67	20.49%	23.39	39.76	20.70%	24.08	41.19	21.02%
	ReDyNet	**21.30**	**36.05**	**19.18%**	**21.69**	**36.83**	**19.81%**	**22.18**	**37.97**	**20.38%**	**22.73**	**39.04**	**20.45%**

Table 1: Performance comparison of ReDyNet and baselines on Beijing, Shanghai, and Hangzhou Metro.

3 Experiments

Datasets. We select three real-world metro flow datasets: *Beijing Metro* (Zhang et al. 2020), *Shanghai Metro* (Liu et al. 2020), and *Hangzhou Metro* (Liu et al. 2020). Like common flow preprocessing ways, the original metro flow data has been aggregated into 15-minute intervals and normalized to *zero mean* (Xie et al. 2023), yielding 4 time steps for each hour. In forecasting, we use metro flow observation from the past hour to forecast the flow for the next hour, i.e., $\omega = 4$ in alignment with previous studies (Xie et al. 2023). We follow the standard dataset split manner by dividing the original into training, validation, and testing sets with a ratio of 7:1:2. The statistics of datasets are summarized in Table 2.

Dataset	#Node	#Edge	#Time step	#Interval
Beijing Metro	276	630	1800	15min
Shanghai Metro	288	958	6716	15min
Hangzhou Metro	80	248	1825	15min

Table 2: The statistics of used datasets.

Baselines. We compare ReDyNet with ten representative baselines ranging from popular traffic flow and recent metro flow models. They are: *DCRNN* (Li et al. 2018), *STGCN* (Yu, Yin, and Zhu 2018), *AGCRN* (Bai et al. 2020), *STTNs* (Xu et al. 2020), *FourierGNN (FGNN)* (Yi et al. 2024), *ASTGCN* (Guo et al. 2019), *GMAN* (Zheng et al. 2020), *Graph WaveNet (GWNet)* (Wu et al. 2019), *STDGRL* (Xie et al. 2023), *PDFormer* (Jiang et al. 2023).

Implementations. Our ReDyNet is implemented with PyTorch, accelerated by an NVIDIA RTX 4090. d_e is set to 128, d_r is set to 32 and c_w is 6. g_o in RDL is set to 256, and the order of the Chebyshev polynomials k is 4. We choose Adam as our optimizer with up to 300 epochs. b is set to 16, α is 0.01 and the initial learning rate is 0.003. All parameters were determined through grid search to ensure optimal performance. For reproducibility, the source codes are available at https://github.com/wangzz-yyzz/ReDyNet.

Metrics. We use three commonly used evaluation protocols, including mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE).

Overall Performance. Table 1 reports the results of all methods on three datasets, where the best gain is stressed in **bold** while the second best is underlined. We have the following findings: Baseline models like STDGRL show increasing forecasting error, highlighting their limitations in capturing long-term dependencies. Dynamic graph-based models like PDFormer and GWNet, outperform others by leveraging graph structures. Adaptive graph-aware methods like AGCRN and STDGRL, update graph structure during training, perform better than models relying on predefined graphs but still fall short of dynamic graph models. Predefined graph models, including ASTGCN and STGCN, utilize a fixed graph structure, which limits their ability to capture dynamic interactions underlying temporal-ware flows. In contrast, our proposed ReDyNet consistently outperforms all baselines, demonstrating the superiority of dynamic interactive signal capture in our responsive graph learning without predefined structure constraints, along with the re-

dundancy disentanglement under the guide of IB principles. We will conduct a more in-depth investigation of ReDyNet.

Ablation Study. We now investigate the impact of each module design in ReDyNet. Correspondingly, we yield five variants of ReDyNet, including: *w/o RCD* removes the RCD; *w/o TA* removes the gated operation in Task Adaption; *w/o Dy-W* removes the parameter pool for graph convolution, i.e., use static graph convolution parameters; *w/o Dy-S* removes the dynamic adjacency matrix in latent representation learning; and *w/o Dy-WS* removes both the parameter pool and dynamic adjacency matrix. As shown in Fig. 3, we find that: (1) removing any modules does degrade model performance. (2) While removing the RCD has a slight impact on short-term forecasting, the error increases with longer periods, exposing the significance of RCD in long-term accuracy. (3) The Task Adaption setting improves accuracy through its gating structure. (4) The dynamic adjacency matrix is effective in graph learning, and the convolution parameter pool is crucial for accurately aggregating dynamic flow, as evidenced by the largest error in the *w/o Dy-S* curve.

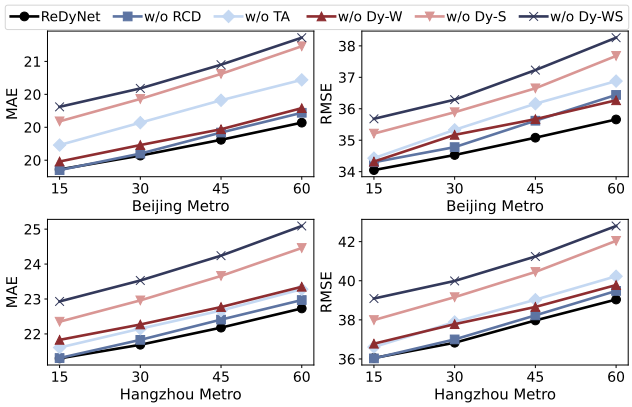
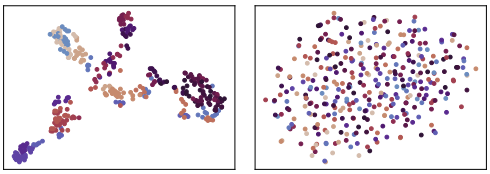


Figure 3: Ablation Study on Beijing and Hangzhou Metro.

Context Impact. To investigate the impact of embedding in CSE, we produce four variants: *RD-S* removes both Data Embedding and Weather Embedding, *RD-SD* removes the Weather Embedding, *RD-SW* removes the Date Embedding, and *RD-DW* removes the Temporal-aware Spatial Embedding. The results in Table 3 indicate that each of them contributes to metro flow learning. However, removing the

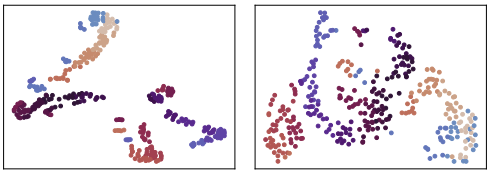
	Metric	RD-S	RD-SD	RD-SW	RD-DW	**ReDyNet**
15min	MAE	19.53	19.42	19.81	24.80	**19.36**
	RMSE	34.28	34.17	34.98	44.16	**34.05**
	MAPE	18.89%	**18.58%**	18.92%	20.18%	18.93%
30min	MAE	19.86	19.67	20.12	26.55	**19.57**
	RMSE	35.34	34.75	36.28	49.09	**34.53**
	MAPE	19.47%	19.51%	19.36%	22.49%	**19.33%**
45min	MAE	20.27	20.00	20.50	28.74	**19.81**
	RMSE	36.65	35.29	37.41	55.65	**35.08**
	MAPE	19.89%	19.78%	19.76%	24.51%	**19.54%**
60min	MAE	20.66	20.09	20.87	31.07	**20.07**
	RMSE	37.79	35.89	38.49	63.33	**35.66**
	MAPE	20.30%	20.13%	20.25%	25.97%	**19.88%**

Table 3: Performance comparison on Beijing Metro.



(a) ReDyNet w/o RCD. (b) ReDyNet.

Figure 4: t-SNE visualization of $\mathcal{R}(\boldsymbol{F}_o)$.



(a) ReDyNet w/o RCD. (b) ReDyNet.

Figure 5: t-SNE visualization of $\boldsymbol{F}$.

Weather Embedding has the least impact, likely due to the coarse-grained weather description. And removing the Temporal-aware Spatial Embedding significantly worsens the performance, highlighting its critical importance.

Disentanglement Interpretability. As $\mathcal{R}(\boldsymbol{F}_o)$ indicates the irrelevant information disentangled from $\boldsymbol{F}_o$ during task learning, we use t-SNE to visualize $\mathcal{R}(\boldsymbol{F}_o)$ for ReDyNet and $\mathcal{R}(\boldsymbol{F}_o)$ after removing RCD, as shown in Fig. 4. In detail, we used distinct colors to mark flow samples according to their time periods of the day (0-23 hours). Fig. 4a reveals that $\mathcal{R}(\boldsymbol{F}_o)$ without RCD has the leakage of useful semantics (the context of commuting date) from the raw input. In contrast, $\mathcal{R}(\boldsymbol{F}_o)$ produced by ReDyNet in Fig. 4b exhibits a date-independent distribution, which, in turn, underscores the effectiveness of RCD in disentangling redundancy and preserving the essential characteristics of raw input. Besides, Fig. 5 presents visualizations of $\boldsymbol{F}$ for both ReDyNet and $\boldsymbol{F}$ after the removal of RCD. In Fig. 5a, flow samples are tightly clustered and less dispersed, indicating an over-reliance on explicit categorical input (i.e., date) and a lack of representation of the diversity of samples. Conversely, Fig. 5b illustrates that ReDyNet produces a more dispersed yet still date-clustered distribution of samples. This broader dispersion suggests better integration of surrounding context (i.e., emphasize the diversity) and strong robustness.

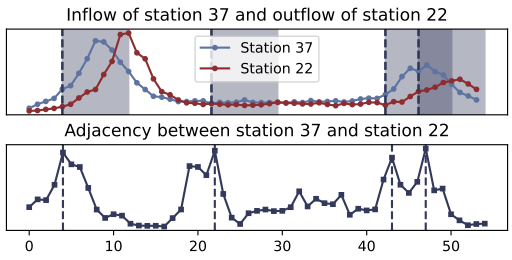


Figure 6: Dynamic connectivity between two stations.

Responsive Dynamic Interpretability. Fig. 6 presents a dynamic connectivity analysis between two stations. The upper plot shows the inflow at station 37 and outflow at station 22, while the lower plot illustrates their dynamic connectivity through adjacency changes. This adjacency measure fluctuates over time, indicating varying connectivity strength. ReDyNet accurately captures the delayed relationship in passenger flow between the two stations (i.e., an increased inflow leads to a corresponding outflow at the other station), enhancing connectivity during peak hours. Fig. 7 shows the transference in connectivity between station 22 and two other stations. The red and blue shaded areas represent the two stations most strongly connected to station 22 during different time periods, showing how ReDyNet captures the transferable connectivity as flow patterns shift.

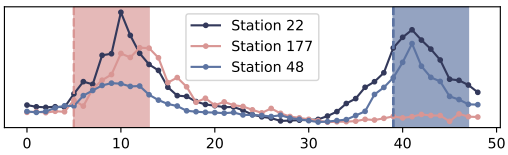
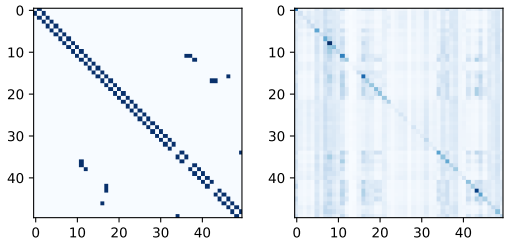


Figure 7: Connectivity transference of multiple stations.

Adjacency Interpretability. Prior efforts usually rely on a predefined adjacency matrix to collaborate with graph learning, while ReDyNet aims to construct a spatially responsive structure over metro flow. Fig. 8a shows geographical connectivity derived from the first 50 stations in *Beijing Metro*, uncovering the unique properties in contrast to typical road networks. Fig. 8b shows a dynamic adjacency matrix learned by our method, illustrating that ReDyNet can capture more complex, data-driven relations based on flow dynamics.



(a) predefined. (b) dynamic.

Figure 8: Adjacency matrix visualization.

Efficiency. Fig. 9 shows the efficiency comparison on *Beijing Metro*. We can observe that our ReDyNet achieves competitive training and inference costs.

4 Related Work

Traffic vs. Metro Flow Forecasting. Recent advancements in flow forecasting primarily seek to explore and interpret complex patterns. Specifically, prior efforts focused on forecasting vehicle flow across road networks by capturing spatiotemporal correlations (Wu et al. 2020) with various sequential modeling approaches such as RNNs (Li et al. 2017)

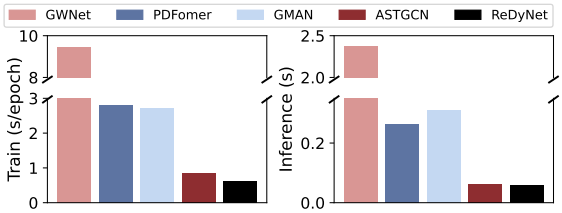


Figure 9: Efficiency evaluation on Beijing Metro.

and attentions (Guo et al. 2019; Jiang et al. 2023). Some efforts improve model capabilities by incorporating feature transformations in the frequency domain (Wu et al. 2019; Yi et al. 2024). Metros play a crucial role in urban traffic, leading to increased research on their flow distribution. However, due to unique characteristics such as station-specific patterns and dynamic changes, traditional traffic prediction methods face challenges when applied to metro flow (Ye et al. 2020; Ou et al. 2020). Recently, Xie et al. (Xie et al. 2023) introduced STDGRL, a bespoke dynamic graph model specifically designed to learn dynamic spatial relationships between metro stations, thereby capturing the unique flow patterns of individual stations. While STDGRL has improved accuracy compared to other traffic flow forecasting methods, it has not fully handled unique properties of metro flow.

Graph-inspired Spatiotemporal Learning. Lately, graph learning has become a mainstream technique for extracting spatial or higher-order dependencies from spatiotemporal data. Early studies focused on applying various GNNs, such as graph convolutional network (GCN) (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2016) and graph attention network (GAT) (Veličković et al. 2017; Fang et al. 2021), to directly capture spatial dependencies within fixed transportation networks (Zhao et al. 2019). These models represent road segments and stations as graph nodes, with temporal dependencies learned independently. Subsequent research concentrated on simultaneously capturing dynamic changes in both spatial and temporal patterns. Researchers achieved this by integrating GNNs into state transition processes within sequential modeling methods, and utilizing multiple time-evolving graphs to replace the fixed transportation network (Li et al. 2017; Yu, Yin, and Zhu 2018). Recent work has proposed building virtual graphs in the latent space (Bai et al. 2020), leading to a remarkable improvement in model performance.

5 Conclusion

We introduced ReDyNet, a novel dynamic responsive graph network for metro flow forecasting. It adapts to variations in metro flow and external factors by constructing responsive graph signals that effectively capture spatiotemporal dynamics. By incorporating IB theory with redundancy disentanglement, it enhances the clarity of spatial signals. Experiments on real-world metro datasets show that ReDyNet significantly outperforms baselines. In the future, we will refine the responsive mechanism and explore more external factors, e.g., socioeconomic variables and real-time events.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No.62102326 and No.62376228, Sichuan Science and Technology Program under Grant No.2023ZYD0145, the Natural Science Foundation of Sichuan Province under Grant No.2023NSFSC1411 and No.25QNJJ0627.

References

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33: 17804–17815.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29.

Fang, Z.; Long, Q.; Song, G.; and Xie, K. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 364–373.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 922–929.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C. P.; Glorot, X.; Botvinick, M. M.; Mohamed, S.; and Lerchner, A. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3.

Jiang, J.; Han, C.; Zhao, W. X.; and Wang, J. 2023. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 4365–4373.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; and Li, P. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*, 11906–11917. PMLR.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations (ICLR '18)*.

Liu, L.; Chen, J.; Wu, H.; Zhen, J.; Li, G.; and Lin, L. 2020. Physical-Virtual Collaboration Modeling for Intra-and Inter-Station Metro Ridership Prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Liu, Y.; Liu, Z.; and Jia, R. 2019. DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transportation Research Part C: Emerging Technologies*, 101: 18–34.

Ou, J.; Sun, J.; Zhu, Y.; Jin, H.; Liu, Y.; Zhang, F.; Huang, J.; and Wang, X. 2020. STP-TrellisNets: Spatial-Temporal Parallel TrellisNets for Metro Station Passenger Flow Prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 1185–1194.

Shwartz-Ziv, R.; and Tishby, N. 2017. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

Song, C.; Lin, Y.; Guo, S.; and Wan, H. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 914–921.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

Xie, P.; Ma, M.; Li, T.; Ji, S.; Du, S.; Yu, Z.; and Zhang, J. 2023. Spatio-Temporal Dynamic Graph Relation Learning for Urban Metro Flow Prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(10): 9973–9984.

Xiong, Z.; Zheng, J.; Song, D.; Zhong, S.; and Huang, Q. 2019. Passenger flow prediction of urban rail transit based on deep learning methods. *Smart Cities*, 2(3): 371–387.

Xu, M.; Dai, W.; Liu, C.; Gao, X.; Lin, W.; Qi, G.-J.; and Xiong, H. 2020. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*.

Yang, K.; Zhou, T.; Zhang, Y.; Tian, X.; and Tao, D. 2021. Class-disentanglement and applications in adversarial detection and defense. *Advances in Neural Information Processing Systems*, 34: 16051–16063.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 7370–7377.

Ye, J.; Zhao, J.; Ye, K.; and Xu, C. 2020. Multi-stgcnet: A graph convolution based spatial-temporal framework for subway passenger flow forecasting. In *2020 International joint conference on neural networks (IJCNN)*, 1–8. IEEE.

Yi, K.; Zhang, Q.; Fan, W.; He, H.; Hu, L.; Wang, P.; An, N.; Cao, L.; and Niu, Z. 2024. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in Neural Information Processing Systems*, 36.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*.

Zhang, J.; Chen, F.; Cui, Z.; Guo, Y.; and Zhu, Y. 2020. Deep learning architecture for short-term passenger flow forecasting in urban rail transit. *IEEE Transactions on Intelligent Transportation Systems*, 22(11): 7004–7014.

Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; and Li, H. 2019. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems*, 21(9): 3848–3858.

Zheng, C.; Fan, X.; Wang, C.; and Qi, J. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 1234–1241.

Zhu, L.; Chen, C.; Wang, H.; Yu, F. R.; and Tang, T. 2023. Machine learning in urban rail transit systems: a survey. *IEEE Transactions on Intelligent Transportation Systems*.