Enhancing Topological Dependencies in Spatio-Temporal Graphs with Cycle Message Passing Blocks

Minho Lee*
AI Aided Engineering
minho.lee@aiae.co

Yun Young Choi†*
AI Aided Engineering
young@aiae.co

Sun Woo Park
Max Planck Institute for Mathematics
s.park@mpim-bonn.mpg.de

Seunghwan Lee
AI Aided Engineering
seunghwan.lee@aiae.co

Joohwan Ko
University of Massachusetts Amherst
joohwanko@cs.umass.edu

Jaeyoung Hong
AI Aided Engineering
jaeyoung.hong@aiae.co

Abstract

Graph Neural Networks (GNNs) and Transformer-based models have been increasingly adopted to learn the complex vector representations of spatio-temporal graphs, capturing intricate spatio-temporal dependencies crucial for applications such as traffic datasets. Although many existing methods utilize multi-head attention mechanisms and message-passing neural networks (MPNNs) to capture both spatial and temporal relations, these approaches encode temporal and spatial relations independently, and reflect the graph's topological characteristics in a limited manner. In this work, we introduce the Cycle to Mixer (Cy2Mixer), a novel spatio-temporal GNN based on topological non-trivial invariants of spatio-temporal graphs with gated multi-layer perceptrons (gMLP). The Cy2Mixer is composed of three blocks based on MLPs: A temporal block for capturing temporal properties, a message-passing block for encapsulating spatial information, and a cycle message-passing block for enriching topological information through cyclic subgraphs. We bolster the effectiveness of Cy2Mixer with mathematical evidence emphasizing that our cycle message-passing block is capable of offering differentiated information to the deep learning model compared to the message-passing block. Furthermore, empirical evaluations substantiate the efficacy of the Cy2Mixer, demonstrating state-of-the-art performances across various spatio-temporal benchmark datasets. The source code is available at https://github.com/leemingo/cy2mixer.

1 Introduction

Spatio-temporal forecasting predicts future events or states involving both spatial and temporal components, with traffic forecasting being one of the most representative examples among these problems. Traffic data is frequently conceptualized as a spatio-temporal graph, where road connections and traffic flows gleaned from sensors are represented as edges and nodes [1, 2], and traffic forecasting aims to predict future traffic in road networks based on preceding traffic data [3–7]. Given this, many turn to Graph Neural Networks (GNNs), especially those using Message Passing Neural Networks (MPNNs), to study this data. GNNs based on MPNNs have been increasingly adopted to learn the complex vector representations of spatio-temporal graphs, capturing intricate dependencies in

*Equal contribution.

M. Lee et al., Enhancing Topological Dependencies in Spatio-Temporal Graphs with Cycle Message Passing Blocks. *Proceedings of the Third Learning on Graphs Conference (LoG 2024)*, PMLR 269, Virtual Event, November 26–29, 2024.

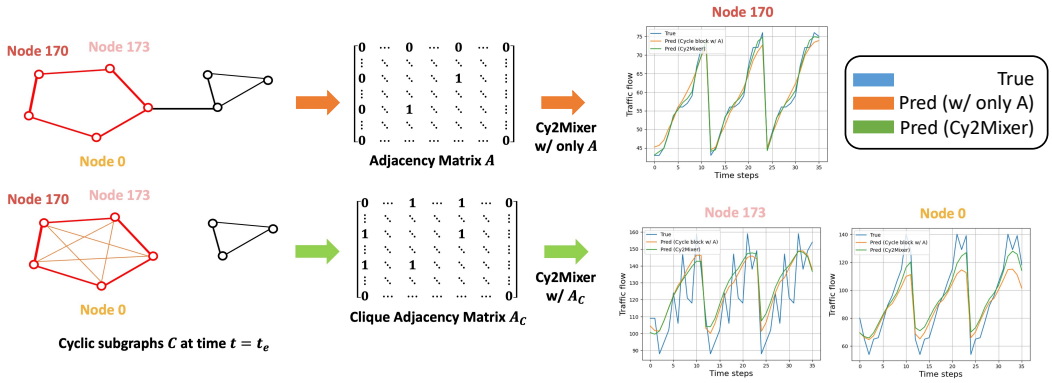

Figure 1: Prediction example of Cy2Mixer in the PEMS04 dataset. While Cy2Mixer shows similar performance between Node 170 and Node 173, which are connected via the adjacency matrix A, Cy2Mixer exhibits superior performance at Node 0 when utilizing the clique adjacency matrix A_C, indicating the effectiveness of the cycle block in capturing cyclic subgraph relationships.

traffic datasets [8–10]. However, traditional message passing-based methodologies have exhibited limitations, including the inability to adequately account for temporal variations in traffic data and over-smoothing problems [11]. Following the success of Transformers in various domains, they have been employed either independently or synergetically with MPNNs to address these challenges in traffic forecasting [11–13], some implementations of which are coupled with preprocessing techniques like Dynamic Time Warping (DTW) [14], showcasing competitive performance outcomes [11].

Previous research efforts have predominantly focused on employing additional algorithms or incorporating intricate structures to capture the complex patterns inherent in spatio-temporal graphs. This often results in heuristics that inflate the complexity of structure and computational costs. Additionally, many of these studies have leaned heavily on experimental results to justify their performance improvements, lacking comprehensive explanations or theoretical grounding. To address these challenges, we start with a mathematical hypothesis that leverages the topological non-trivial invariants of a spatio-temporal graph to enhance the predictive performances of GNNs. This hypothesis aims to discern the spatio-temporal relationships among nodes within the graph. We demonstrate that topological features shed light on facets of temporal traffic data that might be overlooked when focusing solely on the pre-existing edges in a traffic network.

Building on this mathematical foundation, we propose a simple yet unique model, Cycle to Mixer (Cy2Mixer), that integrates gated Multi-Layer Perceptron (gMLP) [15] and MPNNs supplemented with topological non-trivial invariants in graphs, to enhance predictive accuracy. Our Cy2Mixer layer comprises of three key components: (a) Temporal block: Seizes the temporal characteristics of all nodes. (b) Spatial message-passing block: Encompasses spatial relationships along with local neighborhoods. (c) Cycle message-passing block: Captures supplementary information among nodes in cyclic subgraphs. Notably, our cycle message-passing block aims to encapsulate topological features, enabling a finer comprehension of the intricate connectivity patterns within the network inspired by Cy2C-GNNs [16, 17]. We validate our methodology using traffic datasets, showcasing the competitive edge of our proposed models. Furthermore, a qualitative assessment elucidates the efficacy of topological invariants in enhancing the Cy2Mixer's predictive performance as demonstrated in Figure 1, showing improved accuracy for nodes connected through cyclic subgraphs when using the clique adjacency matrix. While our experiments focus on traffic datasets as representative spatio-temporal datasets, we also conducted experiments to demonstrate that our methodology performs well on other spatio-temporal datasets. Our work demonstrates that our approach is not limited to a single task but performs effectively across various tasks, while also evaluating the efficiency of preprocessing time.

The contributions of this work are as follows: (1) We use homotopy invariance of fundamental groups and homology groups to deduce that topological non-trivial invariants of a spatio-temporal network become a contributing factor for influencing the spatio-temporal forecast. (2) We propose a simple yet novel network, Cy2Mixer, based on the gMLP and Cy2C-GNNs inspired from the theory of covering spaces. (3) We show that the proposed model not only demonstrates efficiency in preprocessing time but also consistently delivers superior performance across a variety of spatio-temporal datasets.

2 Related Works

In recent years, deep learning approaches have emerged as powerful tools in the spatio-temporal modeling by leveraging their ability to automatically learn complex, nonlinear spatio-temporal patterns, and traffic forecasting is one of the primary applications of spatio-temporal modeling. Convolutional Neural Networks (CNNs) have been effectively applied to traffic forecasting to capture spatio-temporal patterns [3, 18, 19]. With the success of Transformers in fields like natural language processing and computer vision, owing to their ability to model complex relationships over long sequences without relying on recurrent structures, they have been effectively adapted to traffic forecasting by leveraging self-attention mechanisms to capture long-range spatio-temporal dependencies [11, 13, 20]. Meanwhile, to capture the dynamic spatial correlations within traffic networks, Graph Neural Networks (GNNs) have become a prominent approach for traffic forecasting due to their ability to model complex dependencies [8, 10, 21, 22], and some studies have integrated self-attention mechanisms with GNNs to further enhance the modeling of spatial relationships within traffic networks [12, 23]. Some recent studies adapt traffic graph structures through learnable adjacency matrices and probabilistic modeling [24, 25] and integrate hierarchical topological views to capture multi-level spatial patterns [26, 27]. However, many existing approaches that incorporate spatial information in traffic networks have relied on either limited methods or overly heuristic strategies with auxilary information from road networks, which may be challenging to apply to other spatio-temporal tasks. These methods often struggle to comprehensively capture the full complexity of spatial dependencies, leading to suboptimal modeling of traffic dynamics.

3 Preliminaries

In this section, we provide explanations for some notations and define the traffic prediction problem.

3.1 Message passing neural networks (MPNNs)

Each m-th layer $H^{(m)}$ of MPNNs and hidden node attributes $h_v^{(m)}$ with dimension k_m are given by:

$$\begin{cases} h_v^{(m)} := \text{COMBINE}^{(m)}\left(h_v^{(m-1)}, h'_u\right) \\ h_v^{(0)} := X_v \end{cases} \tag{1}$$

$$\text{where } h'_u = \text{AGGREGATE}_v^{(m)}\left(\left\{\!\left\{ h_u^{(m-1)} \mid u \in N(v) \right\}\!\right\}\right). \tag{2}$$

X_v is the initial node attribute at v and $N(v)$ is the set of neighborhood nodes. Note that $\text{AGGREGATE}_v^{(m)}$ is a function that aggregates features of nodes adjacent to v, and $\text{COMBINE}^{(m)}$ is a function which combines features of the node v with those of nodes adjacent to v.

3.2 gated Multi-Layer Perceptron (gMLP)

gMLP has been shown to achieve comparable performance to Transformer models across diverse domains, including computer vision and natural language processing, with improved efficiency [15]. With the given input $X \in \mathbb{R}^{n \times d}$, where n denotes sequence length and d denotes dimension, it can be defined as:

$$Z = \sigma(XU), \quad \tilde{Z} = s(Z), \quad Y = \tilde{Z}V, \tag{3}$$

where σ represents the activation function, which is GeLU [28] in this context, and U and V correspond to linear projections based on the channel (feature) dimensions, serving roles similar to the feed-forward network in Transformer. The layer $s(\cdot)$ corresponds to the Spatial Gating Unit, which is responsible for capturing cross-token interactions. We construct the aforementioned layer as:

$$s(Z) = Z_1 \odot f_{W,b}(Z_2), \tag{4}$$

where $f_{W,b}$ denotes a linear projection $f_{W,b}(Z) = WZ + b$. Here, W and b refer to the weight matrix and bias, respectively. Z_1 and Z_2 denote two independent components split from Z along the channel dimension, and $\odot$ denotes element-wise multiplication.

3.3 Traffic Prediction Problem

Traffic sensor. Traffic sensors are deployed within the traffic system to record essential information, such as the flow of vehicles on roads and the speeds of these vehicles.

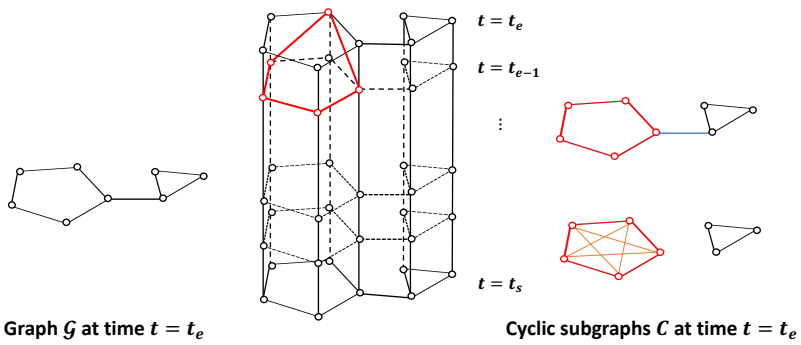

Figure 2: An illustration of lifting a cyclic subgraph of the traffic network $\mathcal{G}$ to a temporal cyclic subgraph of the topological space $\mathcal{G} \times I$ representing the traffic dataset. Every temporal cyclic subgraph of the traffic dataset $\mathcal{G} \times I$ can be obtained from cyclic subgraph of the underlying traffic network $\mathcal{G}$ by using Theorem 4.1. By transforming cyclic subgraphs into cliques by adding and deleting edges suitably, Cy2Mixer effectively models how traffic signals measured in distinct nodes and time can affect each other.

Traffic network. Traffic network can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A, A_C)$, where $\mathcal{V} = \{v_1, \cdots, v_N\}$ denotes the set of N nodes representing sensors within the traffic network ($|V| = N$). Next, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges, and the adjacency matrix A of the network $\mathcal{G}$ can be obtained based on the distances between nodes. Additionally, A_C is the clique adjacency matrix of the network, which is incorporated into the GNN architecture. The detailed procedure for obtaining the clique adjacency matrix is provided in Appendix A.7. The matrices $A \in \mathbb{R}^{N \times N}$ and $A_C \in \mathbb{R}^{N \times N}$ are time-independent input variables since the structure remains unchanged over time. As will be noted in Section 4 and Appendix A.6, cyclic substructures of traffic network are crucial in understanding additional conditions imposed in accurately predicting traffic signals. As demonstrated in previous studies [16], utilizing clique adjacency matrices A_C can effectively boost capabilities of conventional deep learning architectures in incorporating cyclic substructures of traffic networks. As such, we use the clique adjacency matrix A_C as additional input to our traffic network data.

Traffic signal. The traffic signal $X_t \in \mathbb{R}^{N \times C}$ represents the data measured at time t across N nodes in the network. In this context, C denotes the number of features being recorded by the sensors, and in this study, it represents the flow of the road network.

Problem formalization. Our objective is to train a mapping function f to predict future traffic signals by utilizing the data observed in the previous T steps, which can be illustrated as follows:

$$\left[X_{(t-T+1)}, \cdots, X_t; \mathcal{G}\right] \xrightarrow{f} \left[X_{(t+1)}, \cdots, X_{(t+T')}\right]. \tag{5}$$

4 Mathematical Backgrounds

One of the prominent ways to mathematically quantify the effectiveness and discerning capabilities of GNNs is to interpret the graph dataset as a collection of 1-dimensional topological spaces $\mathcal{G} := \{G_i\}_i$, and identify the given GNNs as a function $GNN : \mathcal{G} \to \mathbb{R}^k$ which represents graphs in a given dataset as real k-dimensional vectors. These approaches were carefully executed in previous literature, which focused on pinpointing the discernability of various architectural designs for improving GNNs, as seen in [16, 29–32]. Throughout these references, the characterizations of topological invariants of such 1-dimensional topological spaces provided grounds for verifying whether such GNNs can effectively capture geometric non-trivial properties of graph datasets, such as cyclic substructures or connected components of graphs. In light of these previous studies, it is hence of paramount interest to reinterpret temporal graph datasets as a collection of higher dimensional topological spaces and understand what non-trivial properties of such topological spaces the novel deep learning techniques processing temporal graph datasets should encapsulate.

Topological Space. We interpret the traffic dataset as a 2-dimensional topological space constructed from the traffic network $\mathcal{G}$, and traffic signals as a function defined over the 2-dimensional topological space. To elaborate, the previous section indicates that we can identify a traffic network with a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A, A_C)$, and a traffic signal at time t, denoted by $X_t \in \mathbb{R}^{N\times C}$, as a collection of functions $\{f_t : \mathcal{G} \to \mathbb{R}^{N\times C}\}_t$, each of which represents measurements taken over the graph $\mathcal{G}$ at time t. Notice that the continuous time variable t parametrizes a closed interval $I := [t_s, t_e]$, where t_s and t_e denote the start and end time of the traffic dataset. This observation allows us to identify a temporal traffic network as a topological space $\mathcal{G} \times I$, and the collection of traffic signals varying with respect to a time variable t as a function over $\mathcal{G} \times I$ satisfying the following condition:

$$\begin{aligned} X :&\mathcal{G} \times I \to \mathbb{R}^{N\times C} \\ X(g, t) &= X_t(g) \text{ for all } t \in I. \end{aligned} \tag{6}$$

Temporal cyclic structures. One of the prominent topological properties of a graph $\mathcal{G}$ is its cyclic subgraphs. Cyclic substructures of traffic networks are crucial in determining additional obstructions imposed on forecasting traffic signals. To see this, the existence of such cycles indicates that there exists a pair of nodes $v, w \in \mathcal{V}$ which are connected by at least two distinct paths, obtained from moving along the edges of the graph. In terms of traffic dataset, the existence of a cyclic subgraph indicates that there are at least two paths that a flow of traffic can move along from one point to another. Hence, it is crucial to understand how cyclic subgraphs of a given traffic network $\mathcal{G}$ affect future traffic signals. This can be achieved by understanding possible cyclic substructures inherent in a traffic signal $X : \mathcal{G} \times I \to \mathbb{R}^{N\times C}$. The following result shows that cyclic substructures of a traffic signal can be fully understood from the cycle bases of a traffic network $\mathcal{G}$. We leave the proof of the theorem as well as remarks on mathematical relations between cyclic subgraphs and temporal traffic data in Appendix A.6.

Theorem 4.1. *Given any choice of $t_0 \in I$, let $\pi_{t_0} : \mathcal{G} \times I \to \mathcal{G} \times \{t_0\} \cong \mathcal{G}$ be the projection map which sends the interval I to a singleton set $\{t_0\}$. Let $\mathcal{C}_{\mathcal{G}\times I} := \{C_1, C_2, \cdots, C_n\}$ be a cycle basis of the topological space $\mathcal{G} \times I$. Then the set*

$$\pi_{t_0}(\mathcal{C}_{\mathcal{G}\times I}) := \{\pi_{t_0}(C_1), \cdots, \pi_{t_0}(C_n)\} \tag{7}$$

is a cycle basis of the traffic network $\mathcal{G}$.

Temporal cliques. Theorem 4.1 demonstrates that every cyclic structure of temporal graph datasets $\mathcal{G} \times I$, originating from lifting the cyclic structures of the baseline traffic network $\mathcal{G}$ to any time period, can be captured using the projection map π_{t_0}. In fact, the lifting procedure does not require that the cyclic structure over $\mathcal{G} \times I$ has to be defined over a fixed time t. It can be lifted to a cyclic subgraph of $\mathcal{G} \times I$ such that its nodes represent traffic signals measured at different moments of time.

One of the topological invariants that conventional GNNs fail to incorporate effectively is the cyclic substructures of graphs [29, 33]. This is because GNNs can only distinguish any pairs of two graphs G and H up to isomorphism of their universal covers or unfolding trees. To overcome this issue, one can substitute cyclic subgraphs with cliques. An approach inspired by the theory of covering spaces, the strategy alters the geometry of universal covers to allow GNNs to encapsulate cyclic structures effectively [16]. We implement the application of the analogous operation to temporal traffic data by utilizing the clique adjacency matrix A_C.The clique adjacency matrix A_C allows room for GNNs to determine whether additional interactions among nodes of $\mathcal{G}$, whose measurements are taken at varying moments of time, are relevant components for forecasting traffic signals. This allows us to capture potential effects of topological invariants of traffic networks on temporal traffic measurements, as suggested from Theorem 4.1. Figure 2 illustrates how lifting a cyclic subgraph of $\mathcal{G}$ to a new cyclic subgraph of $\mathcal{G} \times I$ of varying temporal instances can enrich our prediction of future traffic signals. The two cycles are subgraphs colored in red. The temporal cyclic subgraph spans over time $t = t_0$ and the end time $t = t_e$ of the dataset. By changing the colored cyclic subgraph into cliques, we add edges to nodes on the cyclic subgraph lying in two different instances of time. These edges establish relations among measurements taken at different nodes and time periods.

5 Methodology

In this section, we present the architecture of Cy2Mixer and elaborate on how it distinguishes itself from other models. Figure 3 provides a comprehensive visualization of our model's framework. This architecture is primarily composed of three main components: the embedding layer, a stack of L Cy2Mixer encoder layers, and the output layer.

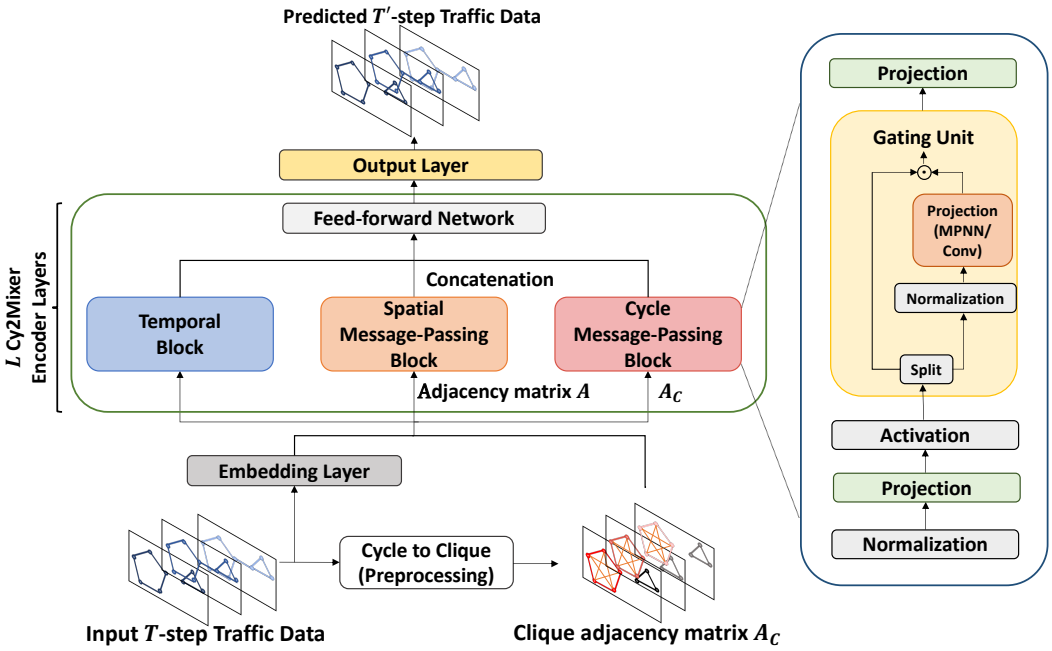

Figure 3: The overall framework of Cy2Mixer. Layers within the green box indicate the Cy2Mixer encoder layer, which comprises a temporal block, spatial message-passing block, and cycle message-passing block to ensure a comprehensive understanding of both temporal and spatial aspects.

5.1 Embedding Layer

Our model follows the structure of the embedding layer from Liu et al. [20]. Given the time-series data $X_{t-T+1:t}$, the feature embedding is defined as $X_{feat} = \text{FC}(X_{t-T+1:t})$, where FC represents a fully connected layer and $X_{feat} \in \mathbb{R}^{T\times N\times d_f}$. Additionally, to account for both weekly and daily periodicities, the temporal embedding $X_{temp} \in \mathbb{R}^{T\times N\times d_t}$ is produced. This is achieved by referencing the day-of-week and timestamp-of-day data to extract two distinct embeddings, which are subsequently concatenated and broadcasted to generate the final temporal embedding with dimension d_t. Furthermore, to address diverse temporal patterns specific to each node, a learnable parameter is introduced in the form of an adaptive embedding of dimension d_a, represented as $X_{ast} \in \mathbb{R}^{T\times N\times d_a}$, to capture and reflect latent, unobserved information within the temporal and spatial patterns of each node. These three embeddings are concatenated, forming the output $H \in \mathbb{R}^{T\times N\times d_h}$ of the embedding layer, where the dimension d_h is equal to $d_f + d_t + d_a$.

$$H = X_{feat}||X_{temp}||X_{ast}. \tag{8}$$

5.2 Cy2Mixer Encoder Layer

The core architecture of the Cy2Mixer encoder layer consists of three distinct blocks, each of which calculates the projection values of Z based on the output H from the embedding layer. In this context, Z corresponds to the output of the projection layer applied to H, analogous to how X in Section 2.2 undergoes projection in the gMLP, and $Z \in \mathbb{R}^{T\times N\times 2d_h}$ is split into $Z_1 \in \mathbb{R}^{T\times N\times d_h}$ and $Z_2 \in \mathbb{R}^{T\times N\times d_h}$ in each blocks. However, unlike the single application of the Spatial Gating Unit in the gMLP, each block in Cy2Mixer employs a specialized gating mechanism designed to capture temporal, spatial, or topological dependencies, utilizing convolution or message-passing networks with different adjacency matrices in each block.

Temporal Block. This block employs a 3×3 convolution network to get the projection values of Z_2. This is subsequently element-wise multiplied with Z_1, generating the output of the Gating Unit:

$$\tilde{Z}_{temporal} = Z_{temporal,1} \odot \text{Conv}(Z_{temporal,2}), \tag{9}$$

Table 1: Traffic flow prediction results for PEMS04, PEMS07, PEMS08, NYTaxi, TDrive, and CHBike. All prediction results in gray are cited from available results obtained from pre-existing publications [20]. Highlighted are the top **first** and second results.

Datasets	Metric	GWNET	STGNCDE	GMAN	ASTGNN	PDFormer	STAEFormer	w/o Cycle block	Cy2Mixer
PEMS04	MAE	19.36	19.21	19.14	18.60	18.36	18.22	18.81	**18.14**
	RMSE	31.72	31.09	31.60	31.03	30.03	30.18	30.65	**30.02**
	MAPE	13.30%	12.77%	13.19%	12.63%	12.00%	11.98%	12.86%	**11.93%**
PEMS07	MAE	21.22	20.62	20.97	20.62	19.97	**19.14**	19.51	19.45
	RMSE	34.12	34.04	34.10	34.02	32.95	**32.60**	33.02	32.89
	MAPE	9.08%	8.86%	9.05%	8.86%	8.55%	**8.01%**	8.16%	8.11%
PEMS08	MAE	15.06	15.46	15.31	14.97	13.58	**13.46**	13.71	13.53
	RMSE	24.86	24.81	24.92	24.71	23.41	23.25	23.63	**23.22**
	MAPE	9.51%	9.92%	10.13%	9.49%	9.05%	8.88%	9.01%	**8.86%**
NYTaxi	MAE	13.30	13.28	13.27	12.98	**12.36**	12.61	12.61	12.59
	RMSE	21.71	21.68	21.66	21.19	**20.18**	20.53	20.53	20.45
	MAPE	13.94%	13.93%	13.89%	13.65%	**12.79%**	12.96%	13.06%	13.03%
TDrive	MAE	19.55	19.29	19.10	18.79	17.79	**16.97**	17.48	16.99
	RMSE	36.18	36.12	36.05	33.93	31.55	31.02	31.31	**30.82**
	MAPE	16.56%	16.50%	16.45%	15.84%	14.68%	13.81%	13.95%	**13.56%**
CHBike	MAE	4.13	4.11	4.10	4.02	3.89	4.03	3.89	**3.80**
	RMSE	5.81	5.80	5.79	5.71	5.48	5.70	5.46	**5.37**
	MAPE	30.92%	30.87%	30.91%	30.91%	30.06%	31.49%	30.10%	**29.20%**

where $\tilde{Z}_{temporal} \in \mathbb{R}^{T \times N \times d_h}$. Here, $Z_{temporal,1}$ and $Z_{temporal,2}$ represent Z_1 and Z_2 for the Temporal block, respectively.

Spatial Message-Passing & Cycle Message-Passing Blocks. These blocks employ the MPNN for their projection function in the Gating Unit. Notably, the spatial message-passing block uses the standard adjacency matrix, A, whereas the cycle message-passing block operates with the clique adjacency matrix, A_C. The clique adjacency matrix, proposed by [16], represents the bases of cycles (or the first homological invariants of the graph [34]) in a suitable form that enables GNNs to effectively process the desired topological features. Following a process similar to the Temporal block, the $\tilde{Z}$ values for both blocks are expressed as:

$$\tilde{Z}_{spatial} = Z_{spatial,1} \odot \text{MPNN}(Z_{spatial,2}, A), \tag{10}$$

$$\tilde{Z}_{cycle} = Z_{cycle,1} \odot \text{MPNN}(Z_{cycle,2}, A_C), \tag{11}$$

where $\tilde{Z}_{spatial} \in \mathbb{R}^{T \times N \times d_h}$ and $\tilde{Z}_{cycle} \in \mathbb{R}^{T \times N \times d_h}$. The final output of each block, denoted as Y, is calculated by applying a linear projection to the output of Gating Unit $\tilde{Z}$, similar to the process in Section 2.2. Considering the effective performance outcomes of incorporating tiny attention into each block in prior research [15], we have adopted the same structure in our model. Then we concatenate all of three outputs, namely, $Y_{temporal}$, Y_{cycle}, and $Y_{spatial}$, using a feed-forward network as:

$$Y_{out} = \text{FC}\left(Y_{temporal} || Y_{cycle} || Y_{spatial}\right), \quad Y_{out} \in \mathbb{R}^{T \times N \times d_h}. \tag{12}$$

5.3 Output Layer

After progressing through the stacked sequence of L Cy2Mixer encoder layers, the output layer extracts the final predictions from the hidden state Y_{out}. Here, T' represents the number of time steps to be predicted, and d_o signifies the dimension of the output features.

$$\hat{Y} = \text{FC}(Y_{out}), \quad Y \in \mathbb{R}^{T' \times N \times d_o}. \tag{13}$$

6 Experiments

Dataset. We evaluated Cy2Mixer on six public traffic datasets: PEMS04, PEMS07, and PEMS08, which contain only traffic data, and NYTaxi, CHBike, and TDrive, which include inflow and outflow data. For PEMS datasets, the model uses the previous hour's data (12 time steps) to predict the next hour, while for NYTaxi, CHBike, and TDrive, it uses six steps to predict the next step. Additional details are in Appendix A.2.

Table 2: Ablation study on effect of cycle message-passing block for PEMS04, PEMS07, and PEMS08. Note that w/ stands for with and w/o stands for without. Time (s) refers to the preprocessing time taken to create each matrix.

Dataset	PEMS04				PEMS07				PEMS08			
Metric	MAE	RMSE	MAPE	Time (s)	MAE	RMSE	MAPE	Time (s)	MAE	RMSE	MAPE	Time (s)
w/o Cycle block	18.81	30.65	12.86%	-	19.74	33.46	8.19%	-	13.56	23.45	8.97%	-
w/ DTW	18.44	30.66	12.16%	67.9732	19.72	33.35	8.31%	562.4729	13.65	23.50	8.94%	20.9622
w/ RWSE	18.29	30.00	11.98%	0.1839	19.57	34.35	8.07%	11.3208	13.68	24.00	8.95%	0.0187
w/ LapPE	18.22	29.97	11.91%	0.1573	21.45	35.25	11.08%	0.9720	13.57	23.65	8.91%	0.0045
Cy2Mixer	**18.14**	**30.02**	**11.93%**	**0.0006**	**19.50**	**33.28**	**8.19%**	**0.0281**	**13.53**	**23.22**	**8.86%**	**0.0006**

Table 3: Case study on the structure of Cy2Mixer for PeMS04. Temporal Block, Spatial Block, and Cycle Block refer to the temporal message-passing block, spatial message-passing block, and cycle message-passing block, respectively.

Cy2Mixer (PEMS04)						MAE	RMSE	MAPE
Temporal Block	Spatial Block	Cycle Block	A	A_C	Tiny attention			
✓					✓	18.55	30.31	12.20%
	✓		✓		✓	22.53	36.33	15.36%
✓	✓		✓		✓	18.81	30.65	12.86%
✓		✓		✓	✓	18.85	30.68	12.72%
	✓	✓	✓	✓	✓	18.56	30.33	12.32%
✓	✓	✓	✓		✓	18.29	30.05	11.95%
✓	✓	✓	✓	✓		18.36	30.15	12.05%
✓	✓	✓	✓	✓	✓	**18.14**	**30.02**	**11.93%**

Baseline models. In this study, we evaluate our proposed approach against various established baseline methods in traffic forecasting. We include GNN-based methods such as GWNet [22] and STGNCDE [10]. We also include GMAN [12], ASTGNN [23], PDFormer [11] and STAEFormer [20], all of which are self-attention-based models designed for the same task as ours.

Experimental settings. We configured our experiments following standard settings for fair comparisons, with data splits of 6:2:2 for PEMS datasets and 7:1:2 for NYTaxi, TDrive, and CHBike. Conducted on an NVIDIA A100 GPU (80GB), with Python 3.10.4 and PyTorch, we used a hyperparameter search to optimize the model on validation data. Evaluation metrics included MAE, RMSE, and MAPE, averaged over 12 forecasted time steps, as detailed in Appendix A.2.

Results from benchmark datasets. The comparison results between our proposed method and various baselines on traffic datasets can be found in Table 1. Cy2Mixer outperforms the baseline models across the majority of datasets. Comparing the results in PEMS04, it can be observed that Cy2Mixer demonstrates notable improvements, with MAE decreasing from 18.22 to 18.14, RMSE from 30.18 to 30.02, and MAPE from 11.98% to 11.93%. In the case of the PEMS07 dataset, the proposed model shows lower performance than STAEFormer in contrast to other datasets. This is because there are no cyclic subgraphs for the PEMS07 dataset, as observed in the Appendix A.1. This outcome underscores the substantial impact of the cycle message-passing block on the model while demonstrating that the proposed model can compete favorably with other models even without utilizing A_C to cycle message-passing block. The robust performance observed in the results of other datasets also demonstrates that Cy2Mixer performs well regardless of the number of predicted time steps or the number of features.

7 Ablation study

Effectiveness of cycle message-passing block. We conducted an ablation study to evaluate the effect of the cycle message-passing block by testing five model variations: Cy2Mixer with the clique adjacency matrix (A_C), with the DTW matrix from PDFormer [11] for long-range dependencies, with alternative structural encodings, including Random-Walk Structural Encoding (RWSE) [35] and Laplacian Eigenvectors Encoding (LapPE) [36], and without the cycle block. Using datasets PEMS04, PEMS07, and PEMS08, results indicated that Cy2Mixer achieved the best performance

Table 4: Air pollution prediction results for *KnowAir* datasets. All prediction results in gray are cited from available results obtained from pre-existing publications. Highlighted are the top **first** and second results.

Dataset	Sub-dataset 1		Sub-dataset 2		Sub-dataset 3	
Metric	RMSE↓	CSI(%)↑	RMSE↓	CSI(%)↑	RMSE↓	CSI(%)↑
GRU	21.00 ± 0.17	45.38 ± 0.52	32.59 ± 0.16	51.07 ± 0.81	45.25 ± 0.85	59.40 ± 0.01
GC-LSTM	20.84 ± 0.11	45.83 ± 0.43	32.10 ± 0.29	51.24 ± 0.13	45.01 ± 0.81	60.58 ± 0.14
$PM_{2.5}$-GNN	19.93 ± 0.11	48.52 ± 0.48	31.37 ± 0.34	**52.33 ± 1.06**	43.29 ± 0.79	61.91 ± 0.78
$PM_{2.5}$-GNN no PBL	20.46 ± 0.18	47.43 ± 0.37	32.44 ± 0.36	51.05 ± 1.15	44.71 ± 1.02	60.64 ± 0.84
$PM_{2.5}$-GNN no export	20.54 ± 0.16	45.73 ± 0.58	31.91 ± 0.32	51.54 ± 1.27	43.72 ± 1.03	61.52 ± 0.95
w/o Cycle block	19.76 ± 0.13	47.95 ± 0.33	31.31 ± 0.20	52.03 ± 0.95	43.27 ± 0.41	61.65 ± 0.52
Cy2Mixer	**19.34 ± 0.13**	**48.58 ± 0.56**	**31.29 ± 0.19**	51.64 ± 0.90	**43.19 ± 0.98**	**62.06 ± 0.69**

when the cycle block included A_C. Additionally, generating A_C is significantly faster than other methods like the DTW matrix, highlighting Cy2Mixer's efficiency for spatio-temporal tasks.

Case study. We conducted a case study on PEMS04 to evaluate each component's impact in Cy2Mixer by testing variations with only the temporal or spatial message-passing blocks, and comparing models with or without the cycle message-passing block and the tiny attention mechanism. Results in Table 3 show that the combined use of adjacency matrix A and clique adjacency matrix A_C in the spatial and cycle message-passing blocks significantly improves performance, as evidenced by reductions in MAE, RMSE, and MAPE. Additionally, using A_C rather than A in the cycle block further validated its effectiveness in enhancing Cy2Mixer's predictive accuracy. We compared the model's performance using A instead of A_C in the cycle message-passing block to assess if the performance improvement was due to increased parameters or the influence of A_C, and this comparison demonstrated the effectiveness of A_C in Cy2Mixer, with visualized results provided in the Appendix A.4.

Additional experiments for different spatio-temporal learning tasks. We conducted additional experiments to validate the proposed Cy2Mixer not only for traffic forecasting but also for other various spatio-temporal tasks. Among these tasks, we specifically verified the performance of our model in the context of air pollution prediction. A whole 4-year dataset *KnowAir* was used for predicting particles smaller than 2.5μm ($PM_{2.5}$) concentrations [37]. The dataset covers in total 184 cities expressed as nodes, and is split into 3 sub-datasets based on dates. Since the comparison models in this context differ from the models used in the main text, Cy2Mixer was compared with models designed for air pollution prediction; GRU, GC-LSTM [38], and $PM_{2.5}$-GNN [37]. The results of this comparison were evaluated using root mean squre error (RMSE) and critical success index (CSI), which are commonly used meteorological metrics, are presented in Table 4. Cy2Mixer consistently demonstrated the highest predictive performance across most datasets and notably, it achieved this without relying on specific domain knowledge.

8 Conclusion

In this paper, we have mathematically investigated the effects of topological non-trivial invariants on capturing the complex dependencies of spatio-temporal graphs. Through our investigation, we gained insight into how the homotopy invariance of fundamental groups and homology groups can be a contributing factor in influencing the predictive performance of spatio-temporal graphs. We then introduce a simple yet novel model, Cy2Mixer, based on the mathematical background and inspired by gMLP. Cy2Mixer comprises of three major components: a temporal block, a spatial message-passing block, and a cycle message-passing block. Notably, the cycle message-passing block enriches topological information in each of the Cy2Mixer encoder layers, drawn from cyclic subgraphs. Indeed, Cy2Mixer achieves state-of-the-art or second-best performance on traffic forecast benchmark datasets. We further investigate the effects of the cycle message-passing block on benchmark datasets by comparing with the DTW method, which makes a new adjacency matrix based on the time-dependent similarity for each node. Compared to the DTW method, the cycle message-passing block captures the spatio-temporal dependency more effectively with a significantly lower computational cost.

References

[1] Zulong Diao, Xin Wang, Dafang Zhang, Yingru Liu, Kun Xie, and Shaoyao He. Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 890–897, 2019. 1

[2] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*, pages 1082–1092, 2020. 1

[3] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017. 1, 3

[4] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 269–278, 2021.

[5] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4454–4458, 2022.

[6] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *arXiv preprint arXiv:2206.09112*, 2022.

[7] Weiguo Zhu, Yongqi Sun, Xintong Yi, Yan Wang, and Zhen Liu. A correlation information-based spatiotemporal network for traffic flow forecasting. *Neural Computing and Applications*, pages 1–19, 2023. 1

[8] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017. 2, 3

[9] Mengzhang Li and Zhanxing Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting, 2021.

[10] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6367–6374, 2022. 2, 3, 8

[11] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. *arXiv preprint arXiv:2301.07945*, 2023. 2, 3, 8, 13

[12] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1234–1241, 2020. 3, 8

[13] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020. 2, 3

[14] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pages 359–370, 1994. 2

[15] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34:9204–9215, 2021. 2, 3, 7

[16] Yun Young Choi, Sun Woo Park, Youngho Woo, and U Jin Choi. Cycle to clique (cy2c) graph neural network: A sight to see beyond neighborhood aggregation. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 4, 5, 7, 16

[17] Yun Young Choi, Sun Woo Park, Minho Lee, and Youngho Woo. Topology-informed graph transformer. *arXiv preprint arXiv:2402.02005*, 2024. 2

[18] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 3

[19] Haoxing Lin, Rufan Bai, Weijia Jia, Xinyu Yang, and Yongjian You. Preserving dynamic attention for long-term spatial-temporal prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 36–46, 2020. 3

[20] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. *arXiv preprint arXiv:2308.10425*, 2023. 3, 6, 7, 8, 13

[21] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1720–1730, 2019. 3

[22] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019. 3, 8

[23] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5415–5428, 2021. 3, 8

[24] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020. 3, 15, 17

[25] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021. 3

[26] Ning Wu, Xin Wayne Zhao, Jingyuan Wang, and Dayan Pan. Learning effective road network representation with hierarchical graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 6–14, 2020. 3

[27] Qian Ma, Zijian Zhang, Xiangyu Zhao, Haoliang Li, Hongwei Zhao, Yiqi Wang, Zitao Liu, and Wanyu Wang. Rethinking sensors modeling: Hierarchical information enhanced traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1756–1765, 2023. 3, 17

[28] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. 3

[29] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Liò, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2625–2640. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/157792e4abb490f99dbd738483e0d2d4-Paper.pdf. 4, 5

[30] Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks. *arXiv preprint arXiv:2102.07835*, 2021.

[31] Sun Woo Park, Yun Young Choi, Dosang Joe, U Jin Choi, and Youngho Woo. The pwlr graph representation: A persistent weisfeiler-lehman scheme with random walks for graph classification. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pages 287–297. PMLR, 2022.

[32] Guillermo Bernárdez, Lev Telyatnikov, Marco Montagna, Federica Baccini, Mathilde Papillon, Miquel Ferriol-Galmés, Mustafa Hajij, Theodore Papamarkou, Maria Sofia Bucarelli, Olga Zaghen, et al. Icml topological deep learning challenge 2024: Beyond the graph domain. *arXiv preprint arXiv:2409.05211*, 2024. 4

[33] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019. 5

[34] Keith Paton. An algorithm for finding a fundamental set of cycles of a graph. *Communications of the ACM*, 12(9):514–518, 1969. 7

[35] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*, 2021. 8

[36] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022. 8

[37] Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. Pm2. 5-gnn: A domain knowledge enhanced graph neural network for pm2. 5 forecasting. In *Proceedings of the 28th international conference on advances in geographic information systems*, pages 163–166, 2020. 9

[38] Yanlin Qi, Qi Li, Hamed Karimian, and Di Liu. A hybrid model for spatiotemporal forecasting of pm2. 5 based on graph convolutional neural network and long short-term memory. *Science of the Total Environment*, 664:1–10, 2019. 9

[39] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002. 15

[40] Robin Hartshorne. Algebraic geometry. *Graduate Texts in Mathematics*, pages 60–68, 206–224, 1977. 15

[41] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1):96–102, 2001. 17

A Appendix

A.1 Dataset description

Detailed statistical information of six traffic datasets are outlined in Table 5. The first three datasets are graph-based datasets, each with a single feature, while the latter three are grid-based datasets with two features. The node counts for the grid-based datasets are 75 (15×5), 270 (15×18), and 1,024 (32×32), respectively. The term "# Cycles" indicates the number of the cycle bases of graphs, and the term "Average Magnitude # Cycles" denotes the average number of nodes present in a cycle subgraph of each graph.

Table 5: Description of statistical information of six traffic datasets.

Datasets	# Nodes	# Edges	# Timesteps	# Time Interval	Time range	# Features	# Cycles	Average Magnitude # Cycles
PEMS04	307	340	16,992	5 min	01/01/2018-02/28/2018	1	45	4.9111
PEMS07	883	866	28,224	5 min	05/01/2017-08/31/2017	1	0	0
PEMS08	170	295	17,856	5 min	07/01/2016-08/31/2016	1	105	7.8571
NYTaxi	75	484	17,520	30 min	01/01/2014-12/31/2014	2	168	4.8810
TDrive	1,024	7,812	3,600	60 min	02/01/2015-06/30/2015	2	714	7.9174
CHBike	270	1,966	4,416	30 min	07/01/2020-09/30/2020	2	2883	14.4707

A.2 Hyperparameter search

We conducted a hyperparameter search to find the optimal model, and hyperparameters for each datasets are listed in Table 6. Note that the embedding dimensions, d_f, d_t, and d_a, followed the settings of previous research [20]. The search ranges were $\{16, 32, 64, 128\}$ for hidden dimension d_h, $\{2, 3, 4, 5, 6\}$ for number of layers, and $\{0, 0.2, 0.4, 0.6, 0.8\}$ for dropout rate, respectively. Considering previous research has demonstrated the effectiveness of incorporating tiny attention, we conducted experiments in our study to compare the performance when tiny attention is added and when it is not. The selection of the optimal model was based on its performance on the validation set. We performed experiments on STAEFormer [20] framework.

Table 6: Hyperparameters for six traffic datasets.

Datasets	# Layers	d_f	d_t	d_a	d_h	Batch size	Dropout	Weight decay	Learning rate	Learning rate decay	Tiny attention
PEMS04	3	24	24	80	152	16	0.4	0.0005	0.001	0.1	O
PEMS07	4	24	24	80	152	16	0.4	0.001	0.001	0.1	X
PEMS08	3	24	24	80	152	16	0.1	0.0015	0.001	0.1	O
NYTaxi	5	24	24	80	256	16	0.4	0.05	0.001	0.1	X
TDrive	6	24	24	80	256	16	0.4	0.05	0.001	0.1	X
CHBike	3	24	24	80	256	16	0.4	0.05	0.001	0.1	O

A.3 Comparison of the adjacency matrix, clique adjacency matrix, and DTW matrix.

In this section, a detailed comparison is conducted between the adjacency matrix and clique adjacency matrix used in Cy2Mixer, and the DTW matrix employed in PDFormer [11]. The three matrices constructed from the traffic network are visually demonstrated in Fig. 4. DTW algorithm is a similarity measure between two time-series that allows for non-linear alignment of the time series. In the context of PDFormer, DTW is used to compute the similarity of the historical traffic flow between nodes to identify the semantic neighbors of each node by calculating a pairwise distance matrix encompassing all combinations of historical traffic flow sequences. This distance matrix enables the computation of a cumulative distance matrix, representing the minimum distance between two time-series up to a specific point. Finally, the cumulative distance matrix facilitates the determination of the optimal warping path, signifying the most favorable alignment between two time-series. DTW algorithm allows the model to capture complex traffic patterns that are not easily captured by linear alignment methods, but has significant drawbacks: it is computationally time-consuming to compute and often relies on heuristic thresholds and parameters. To overcome these limitations and capture the desired spatio-temporal dependencies, we employ the use of the clique adjacency matrix. The clique adjacency matrix can be computed more efficiently than DTW since it is not dependent on time. Furthermore, it provides the model with richer topological information within the graph. The

figure demonstrates that the clique adjacency matrix we use can capture topological information different from the standard adjacency matrix and DTW matrix.

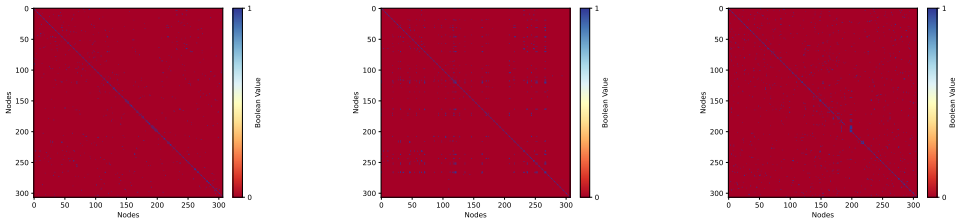

Figure 4: Three matrices of PEMS04: (a) Adjacency matrix A, (b) Clique adjacency matrix A_C, and (c) Matrix constructed by DTW algorithm, respectively. The figure demonstrates that the clique adjacency matrix we use can capture topological information different from the standard adjacency matrix and DTW matrix.

A.4 Visualization of prediction results for each node

In this section, we present the actual prediction results for each node in Cy2Mixer, as shown in Fig 5. To determine whether the cycle block had significant impact on predictions, we compare the results of Cy2Mixer and a model that uses adjacency matrix A instead of clique adjacency matrix A_C in the cycle block. The experiment was conducted on 36 time steps, for Node 170, Node 173 connected to Node 170 through the adjacency matrix, and Node 0 and 169 connected through the clique adjacency matrix. While both models show similar performance for Node 170 and 173, Cy2Mixer demonstrates better prediction accuracy for Node 0 and 169. This indicates that the cycle block effectively incorporates information from the clique adjacency matrix.

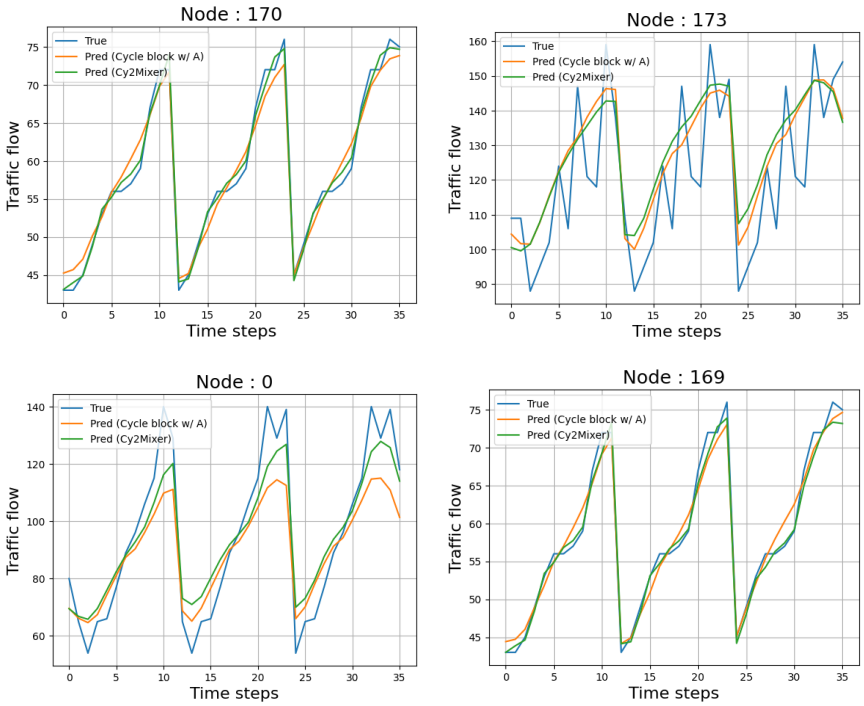

Figure 5: Visualization of prediction results for Node 170, Node 173 (connected to Node 170 in the adjacency matrix), and Nodes 0 and 169 (connected to Node 170 in the clique adjacency matrix) in the PEMS04 dataset.

A.5 Comparison of the influence of the predefined adjacency matrix

In this section, we compared whether a predefined adjacency matrix, calculated based on the distances between nodes, makes a significant difference in the model's performance. Recent studies in traffic prediction problems have explored not using a predefined adjacency matrix, instead learning it alongside the model [24]. To conduct this comparison, we adopted methods from previous studies to create an adjacency matrix and then generated a clique adjacency matrix from it for training in Cy2Mixer. The results of this approach are presented in Table 7. The findings indicate that Cy2Mixer does not show significant differences in performance between using a predefined A and a learned A. Particularly, since learning A and subsequently generating A_C is costly, this study utilized a predefined A.

Table 7: Ablation study on effect of predefined adjacency matrix PEMS04, PEMS07, and PEMS08. Note that w/ stands for with and w/o stands for without.

Dataset	PEMS04			PEMS07			PEMS08		
Metric	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
w/o Cycle block	18.81	30.65	12.86%	19.74	33.46	8.19%	13.56	23.45	8.97%
w/ learned A, A_C	18.81	30.54	12.22%	19.74	33.37	8.33%	13.56	23.46	8.92%
Cy2Mixer	**18.14**	**30.02**	**11.93%**	**19.50**	**33.28**	**8.19%**	**13.53**	**23.22**	**8.86%**

A.6 Proof of Theorem 4.1

The theorem is a corollary of Hatcher [39][Proposition 1.18, Corollary 2.11]. To elaborate, the map $\pi_{t_0} : \mathcal{G} \times I \to \mathcal{G}$ is a homotopy equivalence between the space $\mathcal{G} \times I$ representing the temporal traffic data and the space $\mathcal{G}$ representing the traffic network. The map induces an isomorphism between the fundamental groups and the homology groups of two topological spaces with rational coefficients. That is, these two maps are isomorphisms of groups for all indices $i \geq 0$:

$$(\pi_{t_0})_* : \pi_1(\mathcal{G} \times I) \to \pi_1(\mathcal{G})$$
$$(\pi_{t_0})_* : H^i(\mathcal{G} \times I, \mathbb{Q}) \to H^i(\mathcal{G}, \mathbb{Q})$$

The isomorphisms between the fundamental groups π_1 and the first homology groups H_1 indicate that the set $\pi_{t_0}(\mathcal{C}_{\mathcal{G}\times I})$ has to be a cycle basis of $\mathcal{G}$.

Mathematical relevance of Theorem 4.1 to temporal traffic data. Theorem 4.1 shows in particular that any cyclic subgraph of the traffic signal $\mathcal{G} \times I$ is comprised of nodes of form $(c_1, t_1), (c_2, t_2), \cdots, (c_n, t_n) \in \mathcal{G} \times I$, where the nodes $c_1, c_2, \cdots, c_n$ are on a common cyclic subgraph of $\mathcal{G}$, and the coordinates t_i correspond to different temporal instances. Theorem 4.1 mathematically demonstrates that the topological non-trivial invariants of a traffic network $\mathcal{G}$ can become a contributing factor for influencing the temporal variations measured among the nodes of traffic dataset, a potential aspect of temporal traffic data that may not be fully addressed from solely analyzing the set of pre-existing edges connecting the nodes of a traffic network. To elaborate, topological non-trivial invariants of $\mathcal{G}$ elucidate restrictions associated to constructing a global traffic signal $X : \mathcal{G} \times I \to \mathbb{R}^{N\times C}$ from coherently gluing a series of local temporal traffic signals $\{X_t\}$ measured at each time t. This originates from previously studied mathematical insights that obstructions in gluing continuous functions $f_\alpha : U_\alpha \to \mathbb{R}^k$ defined over collections of open covers $U_\alpha \subset Y$ of a topological space (such that $\cup_\alpha U_\alpha = Y$) to a continuous function $f : Y \to \mathbb{R}^k$ can be detected from topological (or cohomological) invariants of the topological space Y (see for example Chapter 2 or 3 of [40]). Given that traffic forecasting beyond time $t = t_e$ requires a thorough understanding of traffic signals $X : \mathcal{G} \times [0, t_e] \to \mathbb{R}^{N\times C}$, we can hence conclude that a potential candidate to boost performances of traffic forecasting algorithms is to effectively incorporate topological invariants of traffic networks $\mathcal{G}$.

Exemplary Illustration. We provide an exemplary illustration on how cycle structures are closely relevant to determining the range of traffic data (or traffic signals) that can be measured on a given traffic network.

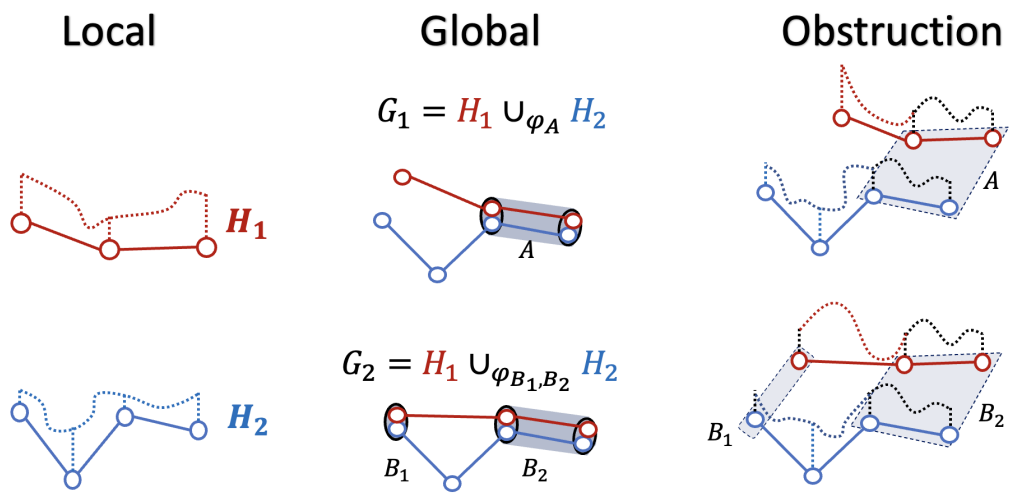

Figure 6: An exemplary illustration of obstructions cycle structures may impose on the range of traffic data or signals that can be measured in a given traffic network.

As shown in Figure 6, suppose we are given with two local traffic networks (graphs) without cycles H_1 (the red graph) and H_2 (the blue graph). The dotted lines above the graphs represent an example of a traffic signal that can be measured over each local traffic network. Let G_1 and G_2 be two global traffic networks (graphs) which can be obtained from gluing the two local traffic networks in a different manner. The traffic network G_1 glues two nodes and one edge of H_1 and H_2 together (along the subnetwork notated as A), which does not add new cyclic subgraph to G_1. On the other hand, the traffic network G_2 glues three nodes and one edge of H_1 and H_2 together (along the subnetworks notated as B_1 and B_2), which adds a new cyclic subgraph to G_2. By gluing the two local traffic networks together, we introduce a new obstruction to traffic signals that can be measured over G_1 and G_2. The signals measured over subnetworks of local traffic network H_1 and H_2 must be identical to one another. Figure 6 clearly shows that G_2 has more obstructions on the traffic signal that can be measured in comparison to G_1. To elaborate, any traffic signal that can be measured on G_2 must satisfy the identical signals over the subnetworks B_1 and B_2, whereas G_1 only requires to satisfy the identical signals over the subnetworks A, which is isomorphic to the subnetwork B_2. This illustration hints that demonstrates that determining cycle structures inherent in a traffic network is crucial to understanding additional obstructions imposed over traffic signals that can be measured or predicted in the given network. Because conventional graph neural networks have limited capabilities in discerning cyclic substructures of graph datasets (as shown in [16]), adding additional architectural component which can detect cycle structures of traffic networks to the algorithm can be of crucial importance in improving accuracies in forecasting traffic signals over any traffic networks.

A.7 Clique adjacency matrix

The motivation for using the clique adjacency matrix in Cy2Mixer stems from recent advances in distinguishing graph structures through the theory of covering spaces [16]. Traditional GNNs represent graphs G and H as identical if their universal covers—essentially infinite graphs created by unfolding the original graphs at each node—are isomorphic, and their node features match upon this unfolding. However, these universal covers inherently lack cyclic subgraphs, which are crucial structures in the original graphs. To improve the ability of GNNs to discern these cyclic features, the clique adjacency matrix (A_C) is used in Cy2Mixer. This matrix effectively transforms the geometry of the universal covers, incorporating cyclic subgraph information and thereby enhancing the model's ability to capture and leverage topological nuances within the graph structure. The clique adjacency matrix A_C is generated as follows: given an undirected graph $G := (V, E)$, we first establish a cycle basis B_G of G, representing a set of cyclic subgraphs that form a basis for the cycle space (or the first homology group) of G. The clique adjacency matrix A_C is then constructed as the adjacency matrix of the union of complete subgraphs formed by each cycle in B_G. This is achieved by adding all

Table 8: Traffic flow prediction results for METR-LA. Highlighted are the top **first** and second results.

Dataset		Metric	STGNCDE	GMAN	PDFormer	HGCN	HIEST	MTGNN	STAEFormer	Cy2Mixer
METR-LA	Horizon 3	MAE	3.35	3.53	2.83	2.80	2.71	**2.66**	2.68	2.70
		RMSE	7.10	6.12	5.55	5.40	5.20	**5.12**	5.22	5.30
		MAPE	8.07%	10.36%	7.46%	7.38%	6.97%	**6.83%**	7.01%	7.13%
	Horizon 6	MAE	4.00	3.95	3.23	3.23	3.13	3.03	**3.00**	3.03
		RMSE	8.59	6.93	6.64	6.48	6.23	**6.09**	6.12	6.22
		MAPE	9.86%	11.79%	9.16%	9.06%	8.62%	**8.26%**	8.34%	8.48%
	Horizon 12	MAE	4.84	4.46	3.64	3.68	3.59	3.45	**3.37**	3.39
		RMSE	10.52	7.81	7.67	7.53	7.37	7.15	**7.09**	7.21
		MAPE	12.34%	13.55%	10.87%	10.84%	10.42%	9.99%	**9.94%**	9.96%

possible edges among the nodes within each cycle basis element $B \in B_G$, enabling A_C to represent cyclic structures comprehensively in Cy2Mixer.

A.8 Performance Analysis of Cy2Mixer on the METR-LA Dataset

We also performed an experiment in the METR-LA dataset [41], which is a traffic speed dataset that records the average vehicle speeds across various sensors in Los Angeles. In our experiments on METR-LA, we evaluated the model's performance at different forecasting horizons, specifically for time steps at horizons of 3, 6, and 12, which correspond to 15, 30, and 60 minutes into the future. The data was split into training, validation, and testing sets with ratios of 7:1:2, respectively. Table 8 presents the comparison results between our proposed method, Cy2Mixer, and various baseline models on the METR-LA dataset. Cy2Mixer demonstrates competitive performance across different forecasting horizons when compared to existing models. While Cy2Mixer does not consistently outperform all baseline models on the METR-LA dataset, it delivers results that are on par with the leading methods. The slight differences in performance metrics could be attributed to the specific characteristics of the METR-LA dataset, including its network structure and traffic patterns. Additionally, Cy2Mixer shows consistent performance across Horizons 3, 6, and 12, unlike MTGNN [24] and HIEST [27], whose performance deteriorates at Horizon 12.