Learning Continuously Evolving Spatio-Temporal Explanations for Traffic Flow Forecasting

Cuiying Huo[1] **Baoxu Wang**[1] **Lin Wu**[1] **Yu Mei**[1] **Dongxiao He**[1] **Yawen Li**[2,*] **Di Jin**[1]
[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]School of Economics and Management,
Beijing University of Posts and Telecommunications, Beijing, China
{huocuiying,wangbaoxu,sky_wulin,yumei,hedongxiao,jindi}@tju.edu.cn,
warmly0716@126.com

Abstract

Traffic flow forecasting plays a pivotal role in intelligent transportation systems. However, most existing methods adopt a black-box learning paradigm, resulting in a lack of interpretability in their decision-making processes. Existing explainable methods mostly follow the feature attribution paradigm, leaving explanations at the level of static or discrete feature evidence and making it difficult to reveal the continuous evolution mechanism of spatio-temporal dependencies during dynamic traffic propagation. Meanwhile, prior methods often decouple spatial and temporal explanations, treating spatial structures and temporal dynamics independently and thus failing to capture the intrinsic coupling between spatial dependencies and temporal contexts in model decision-making. To this end, we propose STGIB, a new spatio-temporal graph information bottleneck theory for characterizing continuously evolving explanations in traffic forecasting. STGIB redefines traffic explanation as a time-indexed spatio-temporal explanation trajectory, and jointly learns dynamic key structures and their temporal representations under predictive sufficiency and information compression constraints. Furthermore, we derive tractable recursive variational bounds to make the theoretical objective optimizable, and instantiate a new explainable traffic flow forecasting model via spatio-temporal explanation trajectory generation. Experiments on real-world traffic datasets demonstrate that STGIB maintains competitive predictive performance while generating more faithful and spatio-temporally consistent explanations.

1 Introduction

Intelligent Transportation Systems (ITS) [1, 2] have become a critical infrastructure for mitigating traffic congestion and improving urban operational efficiency in the process of modern urbanization. As a fundamental task in urban traffic management, traffic flow forecasting [3] plays a crucial role in supporting dynamic traffic resource allocation and congestion alleviation, while also providing important prior information for autonomous driving and public transportation planning [4, 5].

The central objective of traffic flow forecasting is to model complex spatio-temporal dependencies. Early studies employed Convolutional Neural Networks (CNNs) to extract spatial features and combined them with Recurrent Neural Networks (RNNs) or Temporal Convolutional Networks (TCNs) to capture temporal patterns [6, 7]. More recently, Graph Neural Network (GNN)-based methods [8, 9] have become the dominant paradigm. Spatio-Temporal Graph Neural Networks (STGNNs) exploit traffic network topology and historical observations to capture spatial dependencies among locations and temporal dependencies across time spans [10, 11, 12], and have been widely

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

applied to traffic flow and speed forecasting [13, 14]. However, despite their superior performance, the black-box nature of these models often leads to limited interpretability, obscuring the underlying decision-making mechanisms and limiting their trustworthiness in real-world ITS applications.

Existing studies have proposed various explainable methods and achieved certain progress in graph learning[16, 17, 18, 19], deep neural networks[20, 21, 22], and time-series forecasting[23, 24, 25]. Although these methods can provide useful insights for understanding traffic forecasting models, the dynamically evolving and coupled spatio-temporal dependencies in traffic scenarios still pose numerous non-trivial challenges to existing explanation paradigms. First, existing methods mainly follow the feature attribution paradigm, explaining predictions by identifying important nodes or salient temporal segments [26, 27]. Such methods can only provide static or discrete decision evidence, making it difficult to characterize the dependencies among different pieces of evidence and their evolution as traffic states change. As shown in Fig. 1, within the same local road network, the influence between upstream and downstream road segments may strengthen, weaken, or even shift to different propagation pathways as traffic states evolve. Therefore, feature attribution methods are insufficient to explain how coupled spatio-temporal predictive dependencies are dynamically formed and continuously evolve during traffic propagation. Second, although existing methods attempt to explain both spatial and temporal factors, they typically identify important spatial structures and salient temporal segments in isolation [28]. Such decoupled explanations overlook the fact that spatial dependencies are dynamically formed and changed along temporal evolution, making it difficult to characterize the intrinsic coupling between spatial and temporal factors. As a result, the generated explanations fail to faithfully reflect the model's genuine decision-making rationale. Finally, these methods lack explicit temporal constraints when generating structural explanations. In traffic forecasting, key spatial dependencies are shaped by the historical evolution of traffic states rather than solely by network topology. Without such constraints, explanatory subgraphs may capture spurious or redundant connections from noisy temporal observations, leading to biased and unstable explanations that deviate from actual traffic propagation patterns.

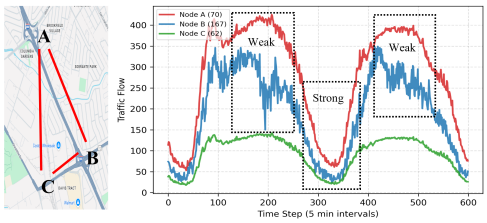


Figure 1: Local subgraph and traffic series from PEMS08 [15], showing time-varying spatial dependencies: correlations strengthen during valleys and weaken during peaks.

To address above challenges, we propose a new **S**patio-**T**emporal **G**raph **I**nformation **B**ottleneck (**STGIB**) theory for characterizing continuous spatio-temporal evolutionary explanations in traffic forecasting. STGIB redefines traffic explanations as spatio-temporal explanation trajectories, shifting the explanatory object from conventional static subgraphs or decoupled spatio-temporal attribution results to predictive dependency processes that evolve with traffic states. Specifically, STGIB learns a set of time-indexed explanatory variables and organizes them into a spatio-temporal evolution sequence under a Markovian dependence assumption. By maximizing the mutual information between explanatory variables and prediction targets, i.e., predictive sufficiency, while minimizing the mutual information between explanatory variables and the original spatio-temporal graph, i.e., compression, STGIB filters task-irrelevant information while preserving predictive sufficiency, thereby learning evolving key structures and their temporal representations. We further derive tractable recurrent variational bounds. Under the chain rule and the Markovian assumption, the compression objective is transformed into time-step-wise variational upper bounds, which identify key structural dependencies within historical spatio-temporal contexts and extract task-relevant temporal representations. Meanwhile, the predictive sufficiency objective is transformed into a variational lower bound to maximize the prediction-relevant information encoded in the explanatory variables.

Based on the above theory, we further instantiate an interpretable traffic forecasting model for generating spatio-temporal explanation trajectories, learning the continuous evolution of traffic predictive dependencies under a unified optimization objective. We adopt a recurrent modeling strategy. At each time step, a Bernoulli prior characterizes the sparse distribution of dynamic explanatory subgraphs to identify key structural dependencies for prediction. Meanwhile, a Gaussian mixture prior regularizes latent temporal representations to compress redundancy and noise in time series while preserving key temporal patterns associated with traffic state evolution. The model ultimately generates an explana-

tion trajectory that evolves synchronously with traffic observations, rather than a decoupled attribution result at a single time step, enabling explicit modeling of continuously evolving spatio-temporal predictive dependencies. Systematic experiments on real-world traffic datasets demonstrate that STGIB maintains competitive predictive performance while producing spatio-temporally consistent explanations, and achieves superior results on explainability metrics such as fidelity and sparsity.

In summary, our main contributions are as follows: **1)** We redefine traffic forecasting explanations as time-indexed spatio-temporal explanation trajectories, shifting the explanatory object from conventional static subgraphs or decoupled attribution results to predictive dependency processes that continuously evolve with traffic states. Based on this definition, we propose STGIB, a new theory for interpretable traffic forecasting, which provides a unified information-theoretic framework for learning continuously evolving spatio-temporal explanation trajectories. **2)** We derive tractable recurrent variational bounds for STGIB, making the theoretical objective optimizable at the model level. And we further instantiate a traffic forecasting model based on spatio-temporal explanation trajectories, enabling interpretable modeling of continuously evolving spatio-temporal processes. **3)** Experiments on real-world traffic datasets show that STGIB can generate more faithful and spatio-temporally consistent explanations while maintaining competitive predictive performance.

2 Preliminaries and Notations

Notations. We represent the road network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of N sensors and $\mathcal{E}$ represents the edges. The network structure is encoded by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. At each time step t, the traffic flow is denoted by a matrix $\mathbf{X}_t \in \mathbb{R}^{N \times C}$, where C represents the feature dimension (e.g., flow volume) for each sensor.

Traffic Flow Forecasting. The spatial-temporal traffic forecasting problem aims to predict future traffic flows based on historical observations and traffic network topology. Formally, given a historical sequence $\mathcal{X}_{hist} = (\mathbf{X}_1, \ldots, \mathbf{X}_T) \in \mathbb{R}^{T \times N \times C}$, we define the forecasting target as $\mathbf{Y} = (\mathbf{X}_{T+1}, \ldots, \mathbf{X}_{T+H})$ and learn a mapping function f_θ to predict:

$$\hat{\mathbf{Y}} = (\hat{\mathbf{X}}_{T+1}, \ldots, \hat{\mathbf{X}}_{T+H}) = f_\theta(\mathcal{X}_{hist}; \mathcal{G}), \tag{1}$$

where $\hat{\mathbf{X}}_{T+h} \in \mathbb{R}^{N \times C}$ represents the predicted traffic flow matrix at future time step $T + h$.

3 Spatio-Temporal Graph Information Bottleneck Principle

In this section, we formalize STGIB as an information-theoretic principle for interpretable traffic forecasting. STGIB redefines traffic explanation as a continuous spatio-temporal explanation trajectory that evolves recursively with traffic states and dynamically captures prediction-relevant structural dependencies and temporal representations. We then define a unified information bottleneck objective that preserves predictive sufficiency while compressing task-irrelevant spatio-temporal information.

3.1 Theoretical Formulation

Formally, given the historical traffic observations and road network structure $\mathcal{D} = (\mathbf{X}_{1:T}, \mathbf{A})$, we define the explanation for traffic forecasting as a time-indexed trajectory,

$$\mathbf{E}_{1:T} = \{(\mathcal{S}_1, \mathbf{Z}_1), ..., (\mathcal{S}_t, \mathbf{Z}_t), ..., (\mathcal{S}_T, \mathbf{Z}_T)\}. \tag{2}$$

where $\mathcal{S}_t$ denotes the decisive subgraph at time step t, capturing key spatial dependencies supporting the current prediction, while $\mathbf{Z}_t$ denotes the compact temporal representation formed under the corresponding structural dependency. Unlike a static explanatory subgraph or isolated spatio-temporal attribution, $\mathbf{E}_{1:T}$ characterizes a dynamic explanation process in which predictive dependencies are continuously updated with traffic state evolution.

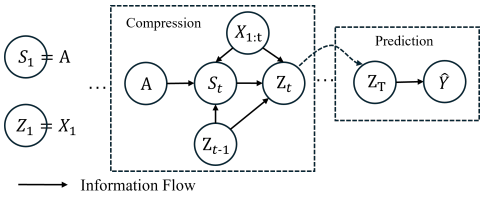


Figure 2: Markovian dependence in STGIB.

As illustrated in Fig 2, STGIB organizes $\mathcal{S}_t$ and $\mathbf{Z}_t$ as a recurrent spatio-temporal evolution process under

a conditional Markovian assumption. The current explanatory state is generated from the previous compressed state and the current observation context through a structure-to-representation transition. Specifically, $\mathbf{Z}_{t-1}$ summarizes the historical evolution of traffic states and explanatory structures, while $\mathbf{X}_{1:t}$ provides state-dependent evidence for the current update. STGIB first infers the explanatory subgraph by

$$p(\mathcal{S}_t \mid \mathcal{S}_{<t}, \mathbf{Z}_{<t}, \mathbf{X}_{1:t}, \mathbf{A}) = p(\mathcal{S}_t \mid \mathbf{Z}_{t-1}, \mathbf{X}_{1:t}, \mathbf{A}), \tag{3}$$

and then updates the temporal representation under the inferred structure:

$$p(\mathbf{Z}_t \mid \mathcal{S}_{\leq t}, \mathbf{Z}_{<t}, \mathbf{X}_{1:t}, \mathbf{A}) = p(\mathbf{Z}_t \mid \mathcal{S}_t, \mathbf{Z}_{t-1}, \mathbf{X}_{1:t}). \tag{4}$$

Through this recurrent organization, the explanation trajectory records the time-varying structural dependencies selected at each step and the corresponding temporal representations formed under these structures, thereby characterizing the continuous evolution of spatio-temporal predictive evidence.

Following the information bottleneck principle [29], STGIB seeks an explanation trajectory that is both predictive and compact. Specifically, the trajectory should preserve target-relevant information for forecasting while filtering redundant structures, transient perturbations, and noisy temporal signals from the original spatio-temporal input. This leads to the following objective:

$$\mathcal{L}_{\text{General}} \triangleq I(\mathbf{E}_{1:T}; \mathcal{D}) - \beta I(\mathbf{Z}_T; \mathbf{Y}), \tag{5}$$

where $I(\mathbf{E}_{1:T}; \mathcal{D})$ measures the information retained from the original spatio-temporal input, $I(\mathbf{Z}_T; \mathbf{Y})$ measures the predictive information preserved in the final representation, and β controls the trade-off between compactness and predictive sufficiency.

However, directly optimizing Eq. (5) is intractable because it involves high-dimensional dependencies between the entire explanation trajectory and the original spatio-temporal graph input. Moreover, traffic data are graph-structured and non-i.i.d., where node states and spatial dependencies are coupled through the road network. To obtain a parameterizable and optimizable objective, we use the spatial local dependency to characterize the feasible interpretation space Ω.

Spatial Local Dependence. Each node representation and explanation is parameterized within its k-hop receptive field, restricting the search for explanatory subgraphs to local spatial dependencies. This makes explanation learning tractable, while higher-order effects can still be propagated through recurrent temporal evolution.

Under the feasible space Ω characterized by the above dependencies, STGIB defines a minimal sufficient spatio-temporal explanation trajectory by optimizing the following constrained objective:

$$\min_{\mathbb{P}(\mathbf{E}_{1:T} \mid \mathcal{D}) \in \Omega} \mathcal{L}_{\text{STGIB}} \triangleq I(\mathbf{E}_{1:T}; \mathcal{D}) - \beta I(\mathbf{Z}_T; \mathbf{Y}). \tag{6}$$

3.2 Variational Bounds

However, directly optimizing the mutual information terms in Eq. 6 is intractable. We therefore derive tractable variational bounds for the predictive and compression terms, transforming the STGIB objective into a recurrent optimization form.

We first consider the predictive sufficiency term $I(\mathbf{Z}_T; \mathbf{Y})$. Since the true conditional distribution $p(\mathbf{Y} \mid \mathbf{Z}_T)$ is generally unavailable, we introduce a variational distribution $q_\psi(\mathbf{Y} \mid \mathbf{Z}_T)$ to approximate it and obtain the following lower bound.

Proposition 1. *The mutual information $I(\mathbf{Z}_T; \mathbf{Y})$ is lower-bounded by*

$$I(\mathbf{Z}_T; \mathbf{Y}) \geq \mathbb{E}_{p(\mathbf{Z}_T, \mathbf{Y})} \left[\log q_\psi(\mathbf{Y} \mid \mathbf{Z}_T)\right] + H(\mathbf{Y}), \tag{7}$$

where $H(\mathbf{Y})$ denotes the entropy of the target variable and remains constant.

This bound provides an optimizable surrogate for the predictive sufficiency term by encouraging the final explanatory representation $\mathbf{Z}_T$ to preserve target-relevant information. The proof follows the standard variational lower bound of Barber and Agakov [30] and is provided in Appendix E.

The compression term $I(\mathbf{E}_{1:T}; \mathcal{D})$ involves the joint dependency between the input history and the entire explanation trajectory. We follow the order of spatio-temporal evolution and define the corresponding ordering-induced histories as:

$$\mathbf{H}_s^{(t)} = \{\mathcal{S}_k, \mathbf{Z}_\tau \mid 1 \leq k < t,\ 1 \leq \tau < t\}, \mathbf{H}_z^{(t)} = \{\mathcal{S}_k, \mathbf{Z}_\tau \mid 1 \leq k \leq t,\ 1 \leq \tau < t\}. \tag{8}$$

Here, $\mathbf{H}_s^{(t)}$ contains all explanatory variables generated before $\mathcal{S}_t$, while $\mathbf{H}_z^{(t)}$ additionally includes $\mathcal{S}_t$, since $\mathbf{Z}_t$ is generated after the current explanatory subgraph. These definitions decompose the compression term into time-step-wise conditional information terms.

Proposition 2. *Following the chain rule of mutual information, the compression term can be decomposed into time-step-wise conditional information terms:*

$$I(\mathbf{E}_{1:T}; \mathcal{D}) = \sum_{t=1}^{T} \left[I(\mathcal{S}_t; \mathcal{D} \mid \mathbf{H}_s^{(t)}) + I(\mathbf{Z}_t; \mathcal{D} \mid \mathbf{H}_z^{(t)}) \right]. \tag{9}$$

The proof is provided in Appendix F.

To obtain computable bounds, we introduce variational priors $\mathbb{Q}(\mathcal{S}_t)$ and $\mathbb{Q}(\mathbf{Z}_t)$ for the explanatory subgraph and latent temporal representation, respectively, allowing the decomposed compression terms to be upper-bounded in a tractable form.

Proposition 3. *Under the spatial locality and conditional Markovian assumptions, the conditional compression terms admit the following upper bounds:*

$$I(\mathcal{S}_t; \mathcal{D} \mid \mathbf{H}_s^{(t)}) = I(\mathcal{S}_t; \mathbf{X}_{1:t}, \mathbf{A} \mid \mathbf{H}_s^{(t)}) \leq \mathbb{E}\left[\log \frac{\mathbb{P}(\mathcal{S}_t \mid \mathbf{A}, \mathbf{X}_{1:t}, \mathbf{Z}_{t-1})}{\mathbb{Q}(\mathcal{S}_t)} \right], \tag{10}$$

$$I(\mathbf{Z}_t; \mathcal{D} \mid \mathbf{H}_z^{(t)}) = I(\mathbf{Z}_t; \mathbf{X}_{1:t}, \mathbf{A} \mid \mathbf{H}_z^{(t)}) \leq \mathbb{E}\left[\log \frac{\mathbb{P}(\mathbf{Z}_t \mid \mathcal{S}_t, \mathbf{X}_{1:t}, \mathbf{Z}_{t-1})}{\mathbb{Q}(\mathbf{Z}_t)} \right]. \tag{11}$$

The proof is provided in Appendix G.

The two upper bounds provide a recurrent variational characterization of the compression term. The first bound corresponds to the information required to generate the current explanatory subgraph from the historical spatio-temporal context, while the second bound corresponds to the information required to obtain the current temporal representation under the inferred structure. Together with the predictive lower bound in Proposition 1, these bounds make the STGIB objective tractable and provide the theoretical basis for the parameterized model.

4 STGIB Instantiation

This section instantiates STGIB as an interpretable traffic forecasting model for generating spatio-temporal explanation trajectories. At each step, it uses the previous compressed state to infer the current explanatory subgraph, updates the temporal representation under this structure, and finally predicts future traffic states from the last representation. Accordingly, the Markovian evolution implements the explanation trajectory, the dynamic subgraph generator realizes the structural posterior $\mathbb{P}_\phi(\mathcal{S}_t \mid \mathbf{A}, \mathbf{X}_{1:t}, \mathbf{Z}_{t-1})$, the structure-guided representation learner realizes the temporal posterior $\mathbb{P}_\phi(\mathbf{Z}_t \mid \mathcal{S}_t, \mathbf{X}_{1:t}, \mathbf{Z}_{t-1})$, and the prediction head implements $q_\psi(\mathbf{Y} \mid \mathbf{Z}_T)$.

4.1 Spatio-Temporal Input Representation

Given the traffic flow sequence $\mathbf{X}_{flow} \in \mathbb{R}^{T \times N \times 1}$, we project each scalar observation $x_{t,v}$ into a d-dimensional flow embedding:

$$\mathbf{e}_{t,v}^{flow} = \mathbf{W}_f x_{t,v} + \mathbf{b}_f. \tag{12}$$

We further introduce learnable embeddings for time-of-day, day-of-week, and node identity, denoted by $\mathbf{E}_{day}$, $\mathbf{E}_{week}$, and $\mathbf{E}_{node}$. The spatio-temporal representation is obtained by summation:

$$\mathbf{h}_{t,v} = \mathbf{e}_{t,v}^{flow} + \mathbf{E}_{day}[t_d] + \mathbf{E}_{week}[t_w] + \mathbf{E}_{node}[v], \tag{13}$$

where t_d and t_w denote the time-of-day and day-of-week indices. We denote the encoded spatio-temporal representation at time step t as $\mathbf{H}_t = \{\mathbf{h}_{t,v}\}_{v=1}^{N}$.

STGIB can operate on the step-level sequence $\mathbf{H}_{1:T}$. For efficiency, we optionally apply temporal patching [31] by grouping adjacent observations into non-overlapping patches:

$$\mathbf{P}_i = [\mathbf{H}_{t_s} \parallel \cdots \parallel \mathbf{H}_{t_e}]\mathbf{W}_p + \mathbf{b}_p, \tag{14}$$

where $t_s = (i-1)P + 1$ and $t_e = iP$. Patching reduces the recurrent length from T to $T' = \lfloor T/P \rfloor$ without changing the STGIB formulation. For notation, we use $\mathbf{U}_{1:L}$ as the recurrent input sequence, where $\mathbf{U}_{1:L} = \mathbf{H}_{1:T}, L = T$ without patching, and $\mathbf{U}_{1:L} = \mathbf{P}_{1:T'}, L = T'$ with patching.

4.2 Markovian Evolution of Explanation Trajectory

The recurrent trajectory starts from an initial compressed state $\mathbf{Z}_0$, which is obtained by projecting the first recurrent input $\mathbf{Z}_0 = \mathbf{U}_1\mathbf{W}_0 + \mathbf{b}_0$. At step t, $\mathbf{Z}_{t-1}$ summarizes the previously generated explanation trajectory and serves as the compressed historical state. Since the previous latent state may lose detailed historical evidence over long sequences, we retrieve context from $\mathbf{U}_{1:t}$ using $\mathbf{Z}_{t-1}$ as the query before generating the current explanatory variables:

$$\mathbf{C}_t = \mathrm{softmax}\left(\frac{(\mathbf{Z}_{t-1}\mathbf{W}_Q)(\mathbf{U}_{1:t}\mathbf{W}_K)^\top}{\sqrt{d}}\right)(\mathbf{U}_{1:t}\mathbf{W}_V). \tag{15}$$

The current input, retrieved context, and previous state are then fused by a GRU [32]:

$$\widetilde{\mathbf{Z}}_t = \mathrm{GRU}(\mathbf{U}_t + \mathbf{C}_t, \mathbf{Z}_{t-1}). \tag{16}$$

Here, $\widetilde{\mathbf{Z}}_t$ is an intermediate transition state. It is used to generate the current subgraph $\mathcal{S}_t$ and the posterior of $\mathbf{Z}_t$, while the sampled $\mathbf{Z}_t$ becomes the compressed state for the next recurrent step.

4.3 Structural Variational Inference for Explanation Subgraph

The structural posterior in Eq. 10 is instantiated as an edge-wise Bernoulli distribution. For each candidate edge $(i, j) \in \mathcal{E}$, we compute its logit and retention probability by

$$\alpha_{i,j}^{(t)} = \mathrm{MLP}\left(\widetilde{\mathbf{z}}_{t,i} \,\|\, \widetilde{\mathbf{z}}_{t,j}\right), \qquad \pi_{i,j}^{(t)} = \sigma(\alpha_{i,j}^{(t)}). \tag{17}$$

Here, $\pi_{i,j}^{(t)}$ parameterizes the posterior probability that edge (i, j) is retained in the explanatory subgraph, while $s_{i,j}^{(t)}$ denotes the corresponding sampled edge indicator. Thus,

$$\mathbb{P}_\phi(\mathcal{S}_t \mid \mathbf{A}, \mathbf{U}_{1:t}, \mathbf{Z}_{t-1}) = \prod_{(i,j)\in\mathcal{E}} \mathrm{Bern}\left(s_{i,j}^{(t)}; \pi_{i,j}^{(t)}\right). \tag{18}$$

For differentiable sampling, we use the Gumbel-Sigmoid relaxation:

$$s_{i,j}^{(t)} = \sigma\left(\frac{\alpha_{i,j}^{(t)} + g_{i,j}^{(1)} - g_{i,j}^{(0)}}{\tau}\right), \quad g_{i,j}^{(1)}, g_{i,j}^{(0)} \sim \mathrm{Gumbel}(0, 1), \tag{19}$$

where τ denotes the temperature.

To implement the structural compression upper bound, we use an uninformative Bernoulli prior

$$\mathbb{Q}(\mathcal{S}_t) = \prod_{(i,j)\in\mathcal{E}} \mathrm{Bern}(s_{i,j}^{(t)}; \gamma), \tag{20}$$

where γ controls the prior edge-retention probability. Minimizing the expectation in Eq. 10 is implemented by minimizing the KL divergence between the structural posterior and prior, yielding

$$\mathcal{L}_S^{(t)} = \sum_{(i,j)\in\mathcal{E}} \left[\pi_{i,j}^{(t)} \log \frac{\pi_{i,j}^{(t)}}{\gamma} + (1 - \pi_{i,j}^{(t)}) \log \frac{1 - \pi_{i,j}^{(t)}}{1 - \gamma}\right]. \tag{21}$$

This term encourages the model to retain only prediction-relevant edges in $\mathcal{S}_t$.

4.4 Temporal Variational Inference under Explanatory Structures

Given the inferred explanatory subgraph $\mathcal{S}_t$, we update the temporal representation by structure-aware message passing:

$$\mathbf{r}_{t,i} = \sum_{j\in\mathcal{N}_i} a_{i,j}^{(t)} s_{i,j}^{(t)} \mathbf{W}\widetilde{\mathbf{z}}_{t,j}, \tag{22}$$

where $a_{i,j}^{(t)}$ is the attention coefficient, and $s_{i,j}^{(t)}$ weights the contribution of edge (i, j) selected in $\mathcal{S}_t$. The aggregated representation $\mathbf{R}_t = \{\mathbf{r}_{t,i}\}_{i=1}^N$ parameterizes the temporal posterior:

$$\mathbb{P}_\phi(\mathbf{Z}_t \mid \mathcal{S}_t, \mathbf{U}_{1:t}, \mathbf{Z}_{t-1}) = \mathcal{N}\left(\mathbf{Z}_t; \boldsymbol{\mu}_t, \mathrm{diag}(\boldsymbol{\sigma}_t^2)\right), \tag{23}$$

with

$$\boldsymbol{\mu}_t = \mathbf{R}_t \mathbf{W}_\mu + \mathbf{b}_\mu, \qquad \log \boldsymbol{\sigma}_t^2 = \mathbf{R}_t \mathbf{W}_\sigma + \mathbf{b}_\sigma. \tag{24}$$

The latent state is sampled by reparameterization:

$$\mathbf{Z}_t = \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{25}$$

Traffic sequences usually contain multiple temporal modes induced by daily periodicity, peak-hour transitions, and irregular disturbances. To match the multi-modal nature of traffic dynamics, we define the variational prior as a learnable Gaussian mixture:

$$\mathbb{Q}(\mathbf{Z}_t) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\mathbf{Z}_t; \boldsymbol{\mu}_k, \mathrm{diag}(\boldsymbol{\sigma}_k^2)\right). \tag{26}$$

Following Eq. 11, the temporal compression is estimated with M reparameterized samples $\mathbf{z}_t^{(m)}$:

$$\mathcal{L}_Z^{(t)} \approx \frac{1}{M} \sum_{m=1}^{M} \left[\log \mathcal{N}\left(\mathbf{z}_t^{(m)}; \boldsymbol{\mu}_t, \mathrm{diag}(\boldsymbol{\sigma}_t^2)\right) - \log \sum_{k=1}^{K} w_k \mathcal{N}\left(\mathbf{z}_t^{(m)}; \boldsymbol{\mu}_k, \mathrm{diag}(\boldsymbol{\sigma}_k^2)\right) \right]. \tag{27}$$

This term compresses redundant temporal information and guides $\mathbf{Z}_t$ to preserve prediction-relevant traffic evolution patterns.

4.5 Predictive Sufficiency and Unified Optimization

Following Eq. 7, predictive sufficiency requires maximizing the variational log-likelihood $\log q_\psi(\mathbf{Y} \mid \mathbf{Z}_L)$. We instantiate q_ψ with a regression decoder $\widehat{\mathbf{Y}} = f_\psi(\mathbf{Z}_L)$, where $\mathbf{Z}_L$ is the final state in the generated explanation trajectory. In practice, minimizing the negative log-likelihood is implemented with Huber loss, a robust surrogate for continuous traffic forecasting:

$$\mathcal{L}_{pred} = \begin{cases} \frac{1}{2}(\mathbf{Y} - \widehat{\mathbf{Y}})^2, & |\mathbf{Y} - \widehat{\mathbf{Y}}| \le \delta, \\ \delta\left(|\mathbf{Y} - \widehat{\mathbf{Y}}| - \frac{1}{2}\delta\right), & \text{otherwise.} \end{cases} \tag{28}$$

The final objective combines predictive sufficiency with the two recurrent compression terms:

$$\mathcal{L} = \sum_{t=1}^{L} \left(\mathcal{L}_S^{(t)} + \mathcal{L}_Z^{(t)}\right) + \beta \mathcal{L}_{pred}, \tag{29}$$

where β balances forecasting accuracy and information compression. The learned trajectory $\{(\mathcal{S}_t, \mathbf{Z}_t)\}_{t=1}^{L}$ provides a time-indexed explanation of how prediction-relevant spatial dependencies and temporal representations evolve along the traffic sequence.

5 Experiments

5.1 Experimental Settings

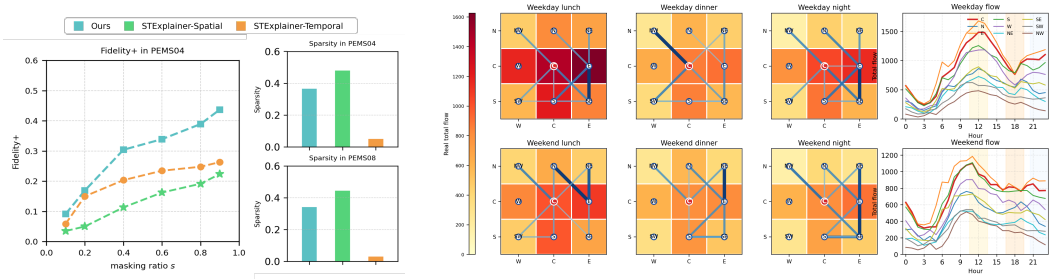
Datasets and Preprocessing. We evaluate STGIB on one grid-based dataset, TDrive [33], and three graph-based datasets, PeMS04, PeMS07, and PeMS08 [15]. Detailed dataset statistics are provided in Appendix I.1. For PeMS datasets, we follow standard protocols and split the data into training, validation, and testing sets with a ratio of 6:2:2, using 12 historical steps to predict the next 12 steps. For TDrive, we use a 7:1:2 split and predict the next-step inflow and outflow from the past 6 steps.

Baselines. We compare STGIB with 19 state-of-the-art baselines from four categories: (1) GNN-based models, including DCRNN [9], STGCN [8], GWNET [10], MTGNN [34], STSGCN [35], STFGNN [36], STGODE [37], STGNCDE [38], and STDN [39]; (2) self-attention-based models, including STTN [40], GMAN [41], ASTGNN [42], TFormer [43], and PDFormer [33]; (3) self-supervised and efficient models, including ST-SSL [44], GraphST [45], and LightST [46]; and (4) explainable and IB-driven models, including STExplainer [28] and RSTIB-MLP [47].

Evaluation Metrics. To evaluate our model, we consider two categories of metrics. For predictive performance, we use MAE, RMSE, and MAPE, excluding missing values and zeros following prior work. For interpretability, we adopt *Sparsity* and *Fidelity* to assess the compactness and sufficiency of the learned spatio-temporal explanations, where *Fidelity* measures prediction shifts under counterfactual masking. Detailed definitions are provided in Appendix I.3.

Table 1: Performance on real-world traffic datasets. The best results are highlighted in **bold**, and the second-best results are underlined.

MODEL	PeMS04			PeMS07			PeMS08			TDrive		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DCRNN	22.54	36.81	14.62%	23.79	36.26	12.41%	18.02	28.33	11.09%	21.86	38.85	18.12%
STGCN	21.92	34.51	13.95%	22.71	35.62	11.84%	17.96	26.94	11.36%	21.48	37.36	17.18%
GWNET	19.18	31.89	13.12%	21.36	33.87	9.19%	15.21	24.67	9.62%	19.53	36.19	16.17%
MTGNN	19.29	31.34	13.08%	20.63	34.29	8.92%	15.25	25.08	10.05%	18.69	35.18	16.49%
STSGCN	21.02	33.83	13.71%	24.41	38.74	10.32%	17.29	26.61	11.08%	24.02	41.85	18.76%
STFGNN	19.97	31.65	13.18%	21.88	35.96	9.10%	16.47	26.38	10.41%	22.36	40.54	18.68%
STGODE	20.66	33.05	13.64%	23.15	35.95	10.26%	16.95	26.09	10.74%	21.78	39.64	18.04%
STGNCDE	19.37	30.92	12.90%	20.48	34.21	8.73%	15.33	24.96	9.78%	19.24	36.11	16.24%
STDN	23.60	35.32	20.22%	23.32	36.42	10.41%	20.63	30.26	18.85%	24.49	38.62	22.62%
STTN	19.61	31.73	13.51%	21.18	34.77	9.82%	15.34	25.12	10.22%	20.31	37.14	16.52%
GMAN	18.97	31.79	13.07%	21.12	33.91	9.16%	15.45	24.76	10.25%	19.17	36.03	16.37%
TFormer	19.02	31.16	12.84%	20.58	34.23	8.86%	15.05	25.03	9.81%	18.82	34.69	16.84%
ASTGNN	18.64	31.21	12.51%	20.76	33.84	8.97%	15.11	24.56	9.59%	18.75	33.58	15.71%
PDFormer	18.56	30.45	12.37%	19.98	33.37	8.57%	13.78	23.88	9.45%	17.59	31.87	15.17%
ST-SSL	18.71	31.29	12.61%	20.52	33.81	9.76%	14.89	24.32	10.18%	16.78	31.15	**13.48%**
GraphST	19.26	31.39	13.04%	20.71	34.14	10.19%	14.97	25.00	10.49%	15.96	31.81	14.67%
LightST	19.21	31.31	12.70%	20.78	33.95	8.98%	15.43	24.52	9.84%	16.49	33.47	16.87%
STExplainer	18.86	30.45	12.23%	20.03	33.44	8.49%	14.72	23.87	9.71%	16.47	31.16	15.78%
RSTIB-MLP	18.64	30.34	12.41%	19.81	33.85	8.42%	14.55	24.19	9.42%	16.85	32.14	16.78%
STGIB	**18.48**	**30.17**	**12.07%**	**19.54**	**32.89**	**8.18%**	**13.61**	**23.52**	**9.01%**	**14.84**	**28.42**	14.52%



(a) (b)

Figure 3: (a) Comparison with STExplainer. (b) Traffic patterns and learned explanatory trajectories around Wangfujing in TDrive. Blue lines show evolutionary explanatory subgraphs identified by STGIB on weekdays and weekends, with shades denoting connection strength.

5.2 Performance Comparison

Table 1 reports results on four real-world traffic datasets. On the three graph-based PeMS datasets, STGIB achieves the best performance across all metrics, demonstrating its effectiveness in modeling road-network traffic dynamics. Compared with advanced GNN-based, self-attention-based, and interpretable baselines, STGIB learns compact prediction-relevant dependencies and benefits from modeling explanations as continuously evolving spatio-temporal trajectories. On the grid-based TDrive dataset, STGIB obtains the best MAE and RMSE, while its MAPE is not optimal, likely due to the sensitivity of MAPE to low-volume grids where small ground-truth values amplify relative errors. The superior MAE and RMSE still indicate more accurate absolute inflow and outflow predictions, showing that STGIB maintains strong predictive performance across both graph-based and grid-based traffic scenarios. Additional experiments, including efficiency and hyperparameter analyses, are provided in Appendix H.

5.3 Explainability Evaluation

We evaluate the compactness and faithfulness of STGIB explanations by comparing with STExplainer in Fig. 3a, where its spatial and temporal components are reported as STE-S and STE-T. For *Sparsity*, STE-S is highly sparse due to its static structure, while STE-T retains most temporal dependencies and

shows low sparsity. STGIB produces compact explanations by filtering redundant connections along the evolving trajectory. For *Fidelity* on PeMS04, STGIB consistently outperforms both baselines across masking ratios. STE-S loses critical evidence despite high sparsity, whereas STE-T improves fidelity by retaining excessive dependencies. In contrast, STGIB achieves higher fidelity with compact explanatory subgraphs, indicating that its trajectory captures essential spatio-temporal predictive evidence beyond isolated spatial or temporal attribution.

5.4 Explainability Case Study

We conduct a case study on the TDrive dataset in the Wangfujing area to examine whether STGIB captures geographically meaningful explanation trajectories. As shown in Fig. 3b, we visualize dependencies from the central region during representative weekday and weekend periods. STGIB produces time-varying explanations. On weekdays, dependencies shift across lunch, dinner, and night with real traffic flow, reflecting commuting and evening mobility patterns. On weekends, dependency directions and strengths differ, showing smoother and delayed traffic dynamics. These patterns align with the hourly flow curves. The results show that STGIB generates an explanation trajectory evolving with traffic states, rather than a static subgraph or isolated temporal attribution, and captures the dynamic formation and transfer of traffic predictive dependencies.

5.5 Ablation Study

To assess the effectiveness of STGIB, we conduct ablation studies with five variants: w/o Temp removes temporal representation compression, w/o Struct replaces dynamic explanatory subgraphs with the predefined graph, Drop-0.5 randomly drops 50% of edges, w/o Patch performs recurrent evolution at the original step-level granularity, and w/o Mem removes historical context retrieval.

Table 2: Ablation study on PEMS04, PEMS08 and TDrive datasets.

Variant	PEMS04			PEMS08			TDrive		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
w/o Temp	18.69	30.34	12.29%	13.87	23.94	9.36%	15.47	30.04	16.17%
w/o Struct	18.79	30.53	12.38%	13.98	23.97	9.38%	15.96	30.27	16.54%
Drop-0.5	19.87	31.42	13.03%	14.78	24.22	9.67%	16.75	31.56	17.15%
w/o patch	18.55	30.24	12.25%	13.76	23.76	9.24%	15.02	28.67	15.58%
w/o Mem	18.96	30.97	12.42%	14.23	24.12	9.42%	15.34	29.87	15.26%
STGIB	**18.48**	**30.17**	**12.07%**	**13.61**	**23.52**	**9.01%**	**14.84**	**28.42**	**14.52%**

As shown in Table 2, removing structural inference or temporal compression consistently degrades performance, confirming the importance of learning dynamic explanatory subgraphs and filtering redundant temporal information. The large gap between STGIB and Drop-0.5 further shows that the learned subgraphs are not random sparse structures, but prediction-relevant dependencies guided by variational compression. Removing historical context retrieval also leads to clear performance drops, indicating its role in preserving useful evolutionary evidence. Notably, w/o Patch achieves performance close to STGIB, suggesting that patching is not the main source of predictive gains. Instead, it mainly changes the computational granularity of recurrent evolution.

6 Conclusion

In this paper, we proposed STGIB, a spatio-temporal graph information bottleneck theory for interpretable traffic forecasting. STGIB redefines traffic explanation as a time-indexed spatio-temporal trajectory, characterizing how prediction-relevant dependencies evolve with traffic states. We derived tractable recurrent variational bounds and instantiated STGIB as an interpretable forecasting model that learns dynamic explanatory subgraphs and compact temporal representations. Experiments on traffic datasets show that STGIB achieves competitive predictive performance while producing more faithful and spatio-temporally consistent explanations. The Wangfujing case study further demonstrates its ability to reveal geographically meaningful traffic evolution patterns.

7 Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62422210, No. 92370111, No. 62276187, No. 62272340 and No. 62532002), the Postdoctoral Fellowship Program of CPSF under Grant No.GZC20251059, and the China Postdoctoral Science Foundation under Grant No.2026M791665.

References

[1] Sheng-hai An, Byung-Hyug Lee, and Dong-Ryeol Shin. A survey of intelligent transportation systems. In *2011 third international conference on computational intelligence, communication systems and networks*, pages 332–337. IEEE, 2011.

[2] Kashif Naseer Qureshi and Abdul Hanan Abdullah. A survey on intelligent transportation systems. *Middle-East Journal of Scientific Research*, 15(5):629–642, 2013.

[3] Anirudh Ameya Kashyap, Shravan Raviraj, Ananya Devarakonda, Shamanth R Nayak K, Santhosh KV, and Soumya J Bhat. Traffic flow prediction models–a review of deep learning techniques. *Cogent Engineering*, 9(1):2010510, 2022.

[4] Boris Medina-Salgado, Eddy Sánchez-DelaCruz, Pilar Pozos-Parra, and Javier E Sierra. Urban traffic flow prediction techniques: A review. *Sustainable Computing: Informatics and Systems*, 35:100739, 2022.

[5] Chen Chen, Ziye Liu, Shaohua Wan, Jintai Luan, and Qingqi Pei. Traffic flow prediction based on deep learning in internet of vehicles. *IEEE transactions on intelligent transportation systems*, 22(6):3776–3789, 2020.

[6] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, pages 802–810, 2015.

[7] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[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.

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

[10] 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.

[11] Qi Zhang, Jianlong Chang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Spatio-temporal graph structure learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1177–1185, 2020.

[12] Xuxiang Ta, Zihan Liu, Xiao Hu, Le Yu, Leilei Sun, and Bowen Du. Adaptive spatio-temporal graph neural network for traffic forecasting. *Knowledge-based systems*, 242:108199, 2022.

[13] Hong Zhang, Linlong Chen, Jie Cao, Xijun Zhang, Sunan Kan, and Tianxin Zhao. Traffic flow forecasting of graph convolutional network based on spatio-temporal attention mechanism. *International Journal of Automotive Technology*, 24(4):1013–1023, 2023.

[14] Junho Song, Jiwon Son, Dong-hyuk Seo, Kyungsik Han, Namhyuk Kim, and Sang-Wook Kim. St-gat: A spatio-temporal graph attention network for accurate traffic speed prediction. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 4500–4504, 2022.

[15] 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.

[16] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*, 2020.

[17] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[18] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys*, 55(13s):1–42, 2023.

[19] Ninghao Liu, Qizhang Feng, and Xia Hu. Interpretability in graph neural networks. In *Graph neural networks: foundations, frontiers, and applications*, pages 121–147. Springer, 2022.

[20] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[21] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, 32(11):4793–4813, 2020.

[22] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10797–10806. IEEE, 2020.

[23] Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33:6441–6452, 2020.

[24] Ozan Ozyegen, Igor Ilic, and Mucahit Cevik. Evaluation of interpretability methods for multivariate time series forecasting. *Applied Intelligence*, 52(5):4727–4743, 2022.

[25] Youngjin Park, Anh Tong, Sehyun Lee, Jihyeon Seong, Qin Xie, and Jaesik Choi. Towards transparent time series analysis: Exploring methods and enhancing interpretability. *ACM Computing Surveys*, 2026.

[26] Kin Kwan Leung, Clayton Rooke, Jonathan Smith, Saba Zuberi, and Maksims Volkovs. Temporal dependencies in feature importance for time series predictions. *arXiv preprint arXiv:2107.14317*, 2021.

[27] Kesen Zhao and Liang Zhang. Causality-inspired spatial-temporal explanations for dynamic graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[28] Jiabin Tang, Lianghao Xia, and Chao Huang. Explainable spatio-temporal graph neural networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2432–2441, 2023.

[29] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[30] David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

[31] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

[32] Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1724–1734, 2014.

[33] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4365–4373, 2023.

[34] 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.

[35] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 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, pages 914–921, 2020.

[36] Mengzhang Li and Zhanxing Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4189–4196, 2021.

[37] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 364–373, 2021.

[38] 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.

[39] Lingxiao Cao, Bin Wang, Guiyuan Jiang, Yanwei Yu, and Junyu Dong. Spatiotemporal-aware trend-seasonality decomposition network for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11463–11471, 2025.

[40] 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.

[41] 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.

[42] 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.

[43] Haoyang Yan, Xiaolei Ma, and Ziyuan Pu. Learning dynamic and hierarchical traffic spatiotemporal features with transformer. *IEEE Transactions on Intelligent Transportation Systems*, 23 (11):22386–22399, 2021.

[44] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4356–4364, 2023.

[45] Qianru Zhang, Chao Huang, Lianghao Xia, Zheng Wang, Siu Ming Yiu, and Ruihua Han. Spatial-temporal graph learning with adversarial contrastive adaptation. In *International Conference on Machine Learning*, pages 41151–41163. PMLR, 2023.

[46] Qianru Zhang, Xinyi Gao, Haixin Wang, Siu Ming Yiu, and Hongzhi Yin. Efficient traffic prediction through spatio-temporal distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1093–1101, 2025.

[47] Min Chen, Guansong Pang, Wenjun Wang, and Cheng Yan. Information bottleneck-guided MLPs for robust spatial-temporal forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 8821–8855. PMLR, 13–19 Jul 2025.

[48] Yuankai Wu, Huachun Tan, Lingqiao Qin, Bin Ran, and Zhuxi Jiang. A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, 90:166–180, 2018.

[49] Nicholas G Polson and Vadim O Sokolov. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 79:1–17, 2017.

[50] Sayed A Sayed, Yasser Abdel-Hamid, and Hesham Ahmed Hefny. Artificial intelligence-based traffic flow prediction: a comprehensive review. *Journal of Electrical Systems and Information Technology*, 10(1):13, 2023.

[51] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 922–929, 2019.

[52] Anton Kuznietsov, Balint Gyevnar, Cheng Wang, Steven Peters, and Stefano V Albrecht. Explainable ai for safe and trustworthy autonomous driving: A systematic review. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[53] Jiangnan Liu and Shipeng Xu. Advancements of graph neural networks in urban traffic prediction. In *Proceedings of the 1st International Conference on Engineering Management, Information Technology and Intelligence*, pages 62–66, 2024.

[54] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[55] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. *Advances in Neural Information Processing Systems*, 33:20437–20448, 2020.

A Broader impacts

This paper aims to advance the field of interpretable spatiotemporal modeling, with a particular focus on Intelligent Transportation Systems. By introducing a unified spatiotemporal graph information bottleneck theory to identify minimum sufficient dependencies, our framework improves the transparency and reliability of traffic flow forecasting, which is crucial for high-risk urban management and traffic congestion mitigation. While advances in machine learning may entail societal implications, we do not foresee any concrete ethical issues or harmful consequences arising from this study.

B Limitation.

The current framework does not explicitly incorporate rich physical-world priors, such as road semantics, traffic rules, or domain-specific propagation constraints. Integrating such external priors into the explanation trajectory could further enhance the physical plausibility and interpretability of STGIB, which we leave for future work.

C Related Work

Traffic Flow Forecasting. Spatio-temporal modeling is central to traffic flow forecasting, where the goal is to capture both spatial interactions among regions or sensors and temporal evolution across historical observations[48, 49, 50]. Early studies mainly combined Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs) or Temporal Convolutional Networks (TCNs) to model traffic dynamics in grid-based scenarios [6, 7]. While these methods are effective for Euclidean spatial layouts, they are less suitable for road networks with irregular topologies. To address this issue, Spatio-Temporal Graph Neural Networks (STGNNs) have become the dominant paradigm, using graph structures to represent road networks and message passing to model spatial dependencies [51, 10, 11, 12]. Subsequent studies further improve forecasting accuracy by learning adaptive graph structures, modeling long-range spatial correlations, or incorporating attention mechanisms to capture dynamic temporal dependencies [10, 33]. Despite their strong predictive performance [13, 14], these models are often criticized for their opaque decision-making processes. This limits their deployment in high-stakes Intelligent Transportation Systems (ITS), where understanding whether a prediction is driven by recurrent patterns, congestion propagation, or sudden incidents is crucial for reliable traffic management [52, 53].

Explainable Traffic Flow Forecasting. Interpretability for traffic forecasting remains less explored than predictive modeling, and most existing attempts follow the feature-attribution paradigm transferred from graph learning and time-series analysis. Attention-based forecasting models, such as ASTGCN [51], GMAN [41], and PDFormer [33], assign importance weights to road segments, time steps, or propagation patterns, providing preliminary clues for model behavior. However, such attribution scores mainly indicate which spatial or temporal components are important, and do not necessarily constitute faithful explanations of how prediction evidence is dynamically formed. More recent works explicitly study explanations for spatio-temporal forecasting. For example, STExplainer [28] identifies influential spatial and temporal components to explain STGNN predictions, while other diagnostic frameworks trace information propagation for predicted congestion. Nevertheless, these methods typically rely on static subgraphs, isolated temporal attribution, or decoupled spatial-temporal importance analysis, making it difficult to characterize how prediction-relevant dependencies continuously evolve with traffic states. Learning compact and faithful explanation trajectories for traffic forecasting therefore remains an open challenge.

D Additional Preliminaries

D.1 Mathematical Preliminaries

In this section, we outline the fundamental definitions and theorems from information theory that serve as the cornerstone of our derivations [54]. For the sake of generality and clarity, we employ three arbitrary discrete random variables X, Y, and Z (with values in sets $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ respectively) to illustrate these concepts. It is important to note that these variables are used here solely for

illustrative purposes and do not correspond to specific components of our model unless explicitly stated otherwise.

Entropy and Conditional Entropy. The entropy $H(X)$ measures the uncertainty associated with a random variable X. It is defined as the expected negative log-likelihood:

$$H(X) = \mathbb{E}_{x\sim p(x)}[-\log p(x)] = -\sum_{x\in\mathcal{X}} p(x)\log p(x). \tag{30}$$

The conditional entropy $H(X|Y)$ measures the remaining uncertainty of X given that Y is known. It can be expressed as the expected negative log-likelihood of the conditional probability over the joint distribution:

$$\begin{aligned} H(X|Y) &= -\sum_{y\in\mathcal{Y}} p(y) \sum_{x\in\mathcal{X}} p(x|y)\log p(x|y) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(x|y) \\ &= \mathbb{E}_{(x,y)\sim p(x,y)}\big[-\log p(x|y)\big]. \end{aligned}$$

Mutual Information (MI). Mutual information $I(X;Y)$ measures the dependence between two random variables. It quantifies the amount of information obtained about one random variable through observing the other. Formally, it can be expressed in terms of entropy difference, joint distribution summation, or as an expectation over the joint distribution:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(X) + H(Y) - H(X,Y) \\ &= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\ &= \mathbb{E}_{(x,y)\sim p(x,y)}\left[\log\frac{p(x,y)}{p(x)p(y)}\right]. \end{aligned} \tag{31}$$

Kullback-Leibler Divergence. The Kullback-Leibler (KL) divergence, often referred to as relative entropy, is a non-symmetric measure of the difference between two probability distributions P and Q. For discrete probability distributions $p(x)$ and $q(x)$ defined on the same probability space $\mathcal{X}$, the KL divergence is defined as:

$$D_{KL}(p(x)\|q(x)) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}. \tag{32}$$

By comparing the definition of Mutual Information with that of KL divergence, we can reframe Mutual Information as the KL divergence between the joint distribution $p(x,y)$ and the product of the marginal distributions $p(x)p(y)$. This relationship indicates how far the joint distribution deviates from the assumption of independence:

$$\begin{aligned} I(X;Y) &= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \\ &= \mathbb{E}_{(x,y)\sim p(x,y)}\left[\log\frac{p(x,y)}{p(x)p(y)}\right] \\ &= D_{KL}(p(x,y)\|p(x)p(y)). \end{aligned} \tag{33}$$

Conditional Mutual Information (CMI). Conditional mutual information $I(X;Y|Z)$ measures the expected mutual information between X and Y given the realization of a third variable Z. This metric is crucial for isolating the dependency between two variables when a confounding variable is present. It is defined as:

$$\begin{aligned} I(X;Y|Z) &= H(X|Z) - H(X|Y,Z) \\ &= \sum_{z\in\mathcal{Z}} p(z) \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y|z)\log\frac{p(x,y|z)}{p(x|z)p(y|z)} \\ &= \sum_{x\in\mathcal{X},y\in\mathcal{Y},z\in\mathcal{Z}} p(x,y,z)\log\left(\frac{p(x,y|z)}{p(x|z)p(y|z)}\right). \end{aligned} \tag{34}$$

Chain Rule of Mutual Information. The chain rule provides a mechanism to decompose the mutual information between a random variable X and a set of variables into a sum of conditional mutual information terms. Consider three random variables X, Y, and Z; the chain rule states that the joint mutual information can be expanded as:

$$I(X; Y, Z) = I(X; Z) + I(X; Y|Z). \tag{35}$$

This theorem can be generalized to the multivariate setting. For a random variable X and a set of n variables $\{Y_1, Y_2, \ldots, Y_n\}$, the chain rule allows us to express the mutual information as:

$$I(X; Y_1, Y_2, \ldots, Y_n) = \sum_{i=1}^{n} I(X; Y_i \mid Y_1, \ldots, Y_{i-1}), \tag{36}$$

where for the first term ($i = 1$), the conditioning set $\{Y_1, \ldots, Y_{i-1}\}$ is empty, and the term reduces to the standard mutual information $I(X; Y_1)$.

Proposition 4. ***(Vanishing Conditional Mutual Information under Independence).*** *Let X, Y, and Z be random variables. If X and Y are conditionally independent given Z, denoted as $X \perp Y \mid Z$, then the conditional mutual information between X and Y given Z is zero, i.e., $I(X; Y \mid Z) = 0$.*

Proof. Recall the definition of conditional mutual information expressed in terms of the Kullback-Leibler divergence between the conditional joint distribution and the product of conditional marginals:

$$I(X; Y \mid Z) = \sum_{z \in \mathcal{Z}} p(z) \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y \mid z) \log \frac{p(x, y \mid z)}{p(x \mid z) p(y \mid z)}. \tag{37}$$

By the definition of conditional independence, the condition $X \perp Y \mid Z$ implies that for all realizations x, y, z, the joint conditional probability factorizes as:

$$p(x, y \mid z) = p(x \mid z) p(y \mid z). \tag{38}$$

Substituting this factorization into the logarithmic term of the conditional mutual information equation yields:

$$\begin{aligned} I(X; Y \mid Z) &= \sum_{x,y,z} p(x, y, z) \log \left(\frac{p(x \mid z) p(y \mid z)}{p(x \mid z) p(y \mid z)} \right) \\ &= \sum_{x,y,z} p(x, y, z) \log(1) \\ &= \sum_{x,y,z} p(x, y, z) \cdot 0 \\ &= 0. \end{aligned} \tag{39}$$

Thus, when X and Y contain no shared information other than what is already explained by Z, their conditional mutual information vanishes. □

D.2 Preliminaries for Graph Information Bottleneck

The Graph Information Bottleneck (GIB) principle extends the classical Information Bottleneck (IB) to graph-structured data, aiming to learn a minimal sufficient representation for downstream tasks [55]. Formally, given the input graph data $\mathcal{D} = (\mathbf{A}, \mathbf{X})$ consisting of the adjacency matrix $\mathbf{A}$ and node features $\mathbf{X}$, and a target task $\mathbf{Y}$ (e.g., node labels), GIB seeks to learn a latent representation $\mathbf{Z}$ that balances expressiveness and robustness. The optimal representation is obtained by minimizing the following objective:

$$\min_{\mathbb{P}(\mathbf{Z}|\mathcal{D})} \mathcal{L}_{\text{GIB}} = -I(\mathbf{Y}; \mathbf{Z}) + \beta I(\mathcal{D}; \mathbf{Z}), \tag{40}$$

where $I(\cdot; \cdot)$ denotes the mutual information, and β is a Lagrange multiplier controlling the trade-off between prediction sufficiency (maximizing $I(\mathbf{Y}; \mathbf{Z})$) and information compression (minimizing $I(\mathcal{D}; \mathbf{Z})$).

A key challenge in applying the IB principle to graph data is the non-i.i.d. nature of nodes and the intractability of the mutual information terms. To address this, GIB introduces a local-dependence

assumption, positing that the optimal representation of a node depends primarily on its local neighborhood rather than the entire graph. Based on this, tractable variational upper bounds are derived for the compression term. Specifically, the mutual information $I(\mathcal{D}; \mathbf{Z})$ is upper-bounded by decomposing the global dependency into local structural and feature regularization terms:

$$I(\mathcal{D}; \mathbf{Z}) \leq \sum_{l \in S_A} \text{AIB}^{(l)} + \sum_{l \in S_X} \text{XIB}^{(l)}, \tag{41}$$

where AIB and XIB represent the information bottleneck terms for the adjacency structure and node features, respectively, calculated via Markovian steps within a static graph.

However, the standard GIB framework is inherently designed for static graphs. The formulation of $\mathcal{D} = (\mathbf{A}, \mathbf{X})$ assumes a fixed topology $\mathbf{A}$ and time-invariant features $\mathbf{X}$. Consequently, the variational bounds rely solely on spatial Markov chains (aggregating information from k-hop neighbors) and do not account for temporal dependencies. In spatio-temporal forecasting, the minimal sufficient statistics must account not only for spatial redundancy but also for the temporal evolution of traffic dynamics, which standard GIB cannot capture.

E Proof of Proposition 1

We first restate the proposition regarding the variational lower bound of the predictive mutual information.

Proposition 1. *Let $q_\psi(\mathbf{Y} \mid \mathbf{Z}_T)$ be a variational approximation of the true posterior $p(\mathbf{Y} \mid \mathbf{Z}_T)$, parameterized by a decoder network. The mutual information is lower bounded by the expected log-likelihood:*

$$I(\mathbf{Z}_T; \mathbf{Y}) \geq \mathbb{E}_{\mathbf{Z}_T, \mathbf{Y}} \left[\log q_\psi(\mathbf{Y} \mid \mathbf{Z}_T) \right] + H(\mathbf{Y}), \tag{42}$$

where $H(\mathbf{Y})$ is the constant entropy of the target.

Proof. The mutual information $I(\mathbf{Z}_T; \mathbf{Y})$ quantifies the reduction in uncertainty of $\mathbf{Y}$ given $\mathbf{Z}_T$. By definition, it can be decomposed into the difference between the marginal entropy $H(\mathbf{Y})$ and the conditional entropy $H(\mathbf{Y} \mid \mathbf{Z}_T)$. To derive the lower bound, we utilize the non-negativity of the Kullback-Leibler (KL) divergence between the true posterior $p(\mathbf{Y} \mid \mathbf{Z}_T)$ and the variational approximation $q_\psi(\mathbf{Y} \mid \mathbf{Z}_T)$.

The derivation proceeds as follows:

$$\begin{aligned}
I(\mathbf{Z}_T; \mathbf{Y}) &= H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{Z}_T) \\
&= H(\mathbf{Y}) + \mathbb{E}_{(\mathbf{Z}_T, \mathbf{Y}) \sim p} \left[\log p(\mathbf{Y} \mid \mathbf{Z}_T) \right] \\
&= H(\mathbf{Y}) + \mathbb{E}_{(\mathbf{Z}_T, \mathbf{Y}) \sim p} \left[\log \left(q_\psi(\mathbf{Y} \mid \mathbf{Z}_T) \cdot \frac{p(\mathbf{Y} \mid \mathbf{Z}_T)}{q_\psi(\mathbf{Y} \mid \mathbf{Z}_T)} \right) \right] \\
&= H(\mathbf{Y}) + \mathbb{E}_{(\mathbf{Z}_T, \mathbf{Y}) \sim p} \left[\log q_\psi(\mathbf{Y} \mid \mathbf{Z}_T) \right] + \mathbb{E}_{\mathbf{Z}_T \sim p} \left[\mathbb{E}_{\mathbf{Y} \sim p(\cdot \mid \mathbf{Z}_T)} \left[\log \frac{p(\mathbf{Y} \mid \mathbf{Z}_T)}{q_\psi(\mathbf{Y} \mid \mathbf{Z}_T)} \right] \right] \\
&= H(\mathbf{Y}) + \mathbb{E}_{(\mathbf{Z}_T, \mathbf{Y})} \left[\log q_\psi(\mathbf{Y} \mid \mathbf{Z}_T) \right] + \mathbb{E}_{\mathbf{Z}_T} \left[D_{KL}(p(\mathbf{Y} \mid \mathbf{Z}_T) \| q_\psi(\mathbf{Y} \mid \mathbf{Z}_T)) \right] \\
&\geq H(\mathbf{Y}) + \mathbb{E}_{(\mathbf{Z}_T, \mathbf{Y})} \left[\log q_\psi(\mathbf{Y} \mid \mathbf{Z}_T) \right].
\end{aligned} \tag{43}$$

The inequality holds because the KL divergence satisfies $D_{KL}(\cdot \| \cdot) \geq 0$. Consequently, maximizing the expected log-likelihood of the variational decoder $\mathbb{E}[\log q_\psi(\mathbf{Y} \mid \mathbf{Z}_T)]$ effectively maximizes the lower bound of the mutual information $I(\mathbf{Z}_T; \mathbf{Y})$. □

F Proof of Proposition 2

We first restate the proposition regarding the decomposition of the mutual information between the input data and the joint explanation trajectory.

Proposition 2. *Based on the **Chain Rule of Mutual Information**, the mutual information between the input $\mathcal{D}$ and the explanation trajectory $\mathbf{E}_{1:T}$ can be decomposed into the sum of conditional*

mutual information at each time step:

$$I(\mathbf{E}_{1:T};\mathcal{D}) = \sum_{t=1}^{T}\left[I(\mathcal{S}_t;\mathcal{D} \mid \mathbf{H}_s^{(t)}) + I(\mathbf{Z}_t;\mathcal{D} \mid \mathbf{H}_z^{(t)})\right], \tag{44}$$

where $\mathbf{H}_s^{(t)}$ and $\mathbf{H}_z^{(t)}$ denote the historical context variables for the structure and state at time t, respectively.

Proof. To prove this decomposition, we first establish a canonical ordering for the random variables within the explanation trajectory $\mathbf{E}_{1:T} = \{(\mathcal{S}_t, \mathbf{Z}_t)\}_{t=1}^{T}$. Based on the sequential generation process of the proposed model, the variables are generated in the following chronological order:

$$\mathcal{S}_1, \mathbf{Z}_1, \mathcal{S}_2, \mathbf{Z}_2, \ldots, \mathcal{S}_t, \mathbf{Z}_t, \ldots, \mathcal{S}_T, \mathbf{Z}_T. \tag{45}$$

Let $\mathbf{V}_{1:2T}$ denote this ordered sequence of $2T$ variables, where the odd-indexed elements correspond to structures (i.e., $\mathbf{V}_{2t-1} = \mathcal{S}_t$) and the even-indexed elements correspond to latent states (i.e., $\mathbf{V}_{2t} = \mathbf{Z}_t$).

According to the general **Chain Rule of Mutual Information**, the mutual information between a set of variables $\mathbf{V}_{1:K}$ and a target $\mathcal{D}$ can be expanded as the sum of conditional mutual information terms:

$$I(\mathbf{V}_{1:K};\mathcal{D}) = \sum_{k=1}^{K} I(\mathbf{V}_k;\mathcal{D} \mid \mathbf{V}_{1:k-1}), \tag{46}$$

where $\mathbf{V}_{1:0} = \emptyset$. Applying this rule to our sequence $\mathbf{E}_{1:T}$, we sum over $t = 1$ to T for both variable types:

$$I(\mathbf{E}_{1:T};\mathcal{D}) = \sum_{t=1}^{T}\left(I(\mathcal{S}_t;\mathcal{D} \mid \text{History}(\mathcal{S}_t)) + I(\mathbf{Z}_t;\mathcal{D} \mid \text{History}(\mathbf{Z}_t))\right). \tag{47}$$

Here, $\text{History}(\mathcal{S}_t)$ represents the set of all variables generated prior to $\mathcal{S}_t$, and similarly for $\mathbf{Z}_t$. Specifically:

- For the structure $\mathcal{S}_t$, the preceding variables are all structures and states from time steps 1 to $t-1$. This set aligns exactly with our definition of $\mathbf{H}_s^{(t)}$:

$$\text{History}(\mathcal{S}_t) = \{\mathcal{S}_1, \mathbf{Z}_1, \ldots, \mathcal{S}_{t-1}, \mathbf{Z}_{t-1}\} \equiv \mathbf{H}_s^{(t)}. \tag{48}$$

- For the latent state $\mathbf{Z}_t$, the preceding variables include the history of $\mathcal{S}_t$ plus the structure $\mathcal{S}_t$ itself (since $\mathcal{S}_t$ is generated before $\mathbf{Z}_t$ within the same step). This matches our definition of $\mathbf{H}_z^{(t)}$:

$$\text{History}(\mathbf{Z}_t) = \{\mathcal{S}_1, \mathbf{Z}_1, \ldots, \mathcal{S}_{t-1}, \mathbf{Z}_{t-1}, \mathcal{S}_t\} \equiv \mathbf{H}_z^{(t)}. \tag{49}$$

Substituting these history sets back into Eq. (47), we obtain:

$$I(\mathbf{E}_{1:T};\mathcal{D}) = \sum_{t=1}^{T}\left[I(\mathcal{S}_t;\mathcal{D} \mid \mathbf{H}_s^{(t)}) + I(\mathbf{Z}_t;\mathcal{D} \mid \mathbf{H}_z^{(t)})\right]. \tag{50}$$

This completes the proof. □

G Proof of Proposition 3

We provide the detailed derivation for the variational upper bounds of the conditional mutual information terms in the STGIB objective. The derivation relies on the Data Processing Inequality, the Markovian dependence assumption, and the non-negativity of the Kullback-Leibler (KL) divergence.

G.1 Bound for Structure Information Bottleneck (S)

Consider the first term $I(\mathcal{S}_t; \mathcal{D} \mid \mathbf{H}_s^{(t)})$. We derive the upper bound $\mathcal{L}_{\mathrm{S}}^{(t)}$ as follows.

Proof. First, based on the causality of $\mathcal{S}_t$ (i.e., it depends only on past information), we consider the relevant data subset $\mathcal{D}_{1:t} = (\mathbf{A}, \mathbf{X}_{1:t})$.

$$\begin{aligned} I(\mathcal{S}_t; \mathcal{D} \mid \mathbf{H}_s^{(t)}) &= I(\mathcal{S}_t; \mathbf{A}, \mathbf{X}_{1:t} \mid \mathbf{H}_s^{(t)}) \\ &\leq I(\mathcal{S}_t; \mathbf{Z}_{t-1}, \mathbf{A}, \mathbf{X}_{1:t} \mid \mathbf{H}_s^{(t)}). \end{aligned} \tag{51}$$

The inequality (51) holds because adding the variable $\mathbf{Z}_{t-1}$ to the conditioning set of the mutual information target (second argument) does not decrease the mutual information, i.e., $I(X;Y) \leq I(X;Y,Z)$.

Next, we remove the conditioning on the history $\mathbf{H}_s^{(t)}$. Using the definition of mutual information $I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y,Z)$ and the property that conditioning reduces entropy ($H(X \mid Z) \leq H(X)$), combined with the assumption that $\mathbf{Z}_{t-1}$ contains the sufficient statistics from $\mathbf{H}_s^{(t)}$ for generating $\mathcal{S}_t$ (implying $H(\mathcal{S}_t \mid \mathbf{Z}_{t-1}, \ldots, \mathbf{H}_s^{(t)}) = H(\mathcal{S}_t \mid \mathbf{Z}_{t-1}, \ldots)$), we obtain:

$$I(\mathcal{S}_t; \mathbf{Z}_{t-1}, \mathbf{A}, \mathbf{X}_{1:t} \mid \mathbf{H}_s^{(t)}) \leq I(\mathcal{S}_t; \mathbf{Z}_{t-1}, \mathbf{A}, \mathbf{X}_{1:t}). \tag{52}$$

Let $\mathbf{V}_t = (\mathbf{A}, \mathbf{X}_{1:t}, \mathbf{Z}_{t-1})$. We expand the mutual information term by introducing a variational prior $\mathbb{Q}(\mathcal{S}_t)$ to approximate the intractable marginal $\mathbb{P}(\mathcal{S}_t)$.

$$\begin{aligned} I(\mathcal{S}_t; \mathbf{V}_t) &= \mathbb{E}_{\mathbb{P}(\mathcal{S}_t, \mathbf{V}_t)} \left[\log \frac{\mathbb{P}(\mathcal{S}_t \mid \mathbf{V}_t)}{\mathbb{P}(\mathcal{S}_t)} \right] \\ &= \mathbb{E}_{\mathbb{P}(\mathcal{S}_t, \mathbf{V}_t)} \left[\log \left(\frac{\mathbb{P}(\mathcal{S}_t \mid \mathbf{V}_t)}{\mathbb{Q}(\mathcal{S}_t)} \cdot \frac{\mathbb{Q}(\mathcal{S}_t)}{\mathbb{P}(\mathcal{S}_t)} \right) \right] \\ &= \mathbb{E}_{\mathbb{P}(\mathcal{S}_t, \mathbf{V}_t)} \left[\log \frac{\mathbb{P}(\mathcal{S}_t \mid \mathbf{V}_t)}{\mathbb{Q}(\mathcal{S}_t)} \right] + \mathbb{E}_{\mathbb{P}(\mathcal{S}_t)} \left[\log \frac{\mathbb{Q}(\mathcal{S}_t)}{\mathbb{P}(\mathcal{S}_t)} \right] \\ &= \mathbb{E}_{\mathbb{P}} \left[\log \frac{\mathbb{P}(\mathcal{S}_t \mid \mathbf{A}, \mathbf{X}_{1:t}, \mathbf{Z}_{t-1})}{\mathbb{Q}(\mathcal{S}_t)} \right] - D_{KL}(\mathbb{P}(\mathcal{S}_t) \| \mathbb{Q}(\mathcal{S}_t)) \\ &\leq \mathbb{E}_{\mathbb{P}} \left[\log \frac{\mathbb{P}(\mathcal{S}_t \mid \mathbf{A}, \mathbf{X}_{1:t}, \mathbf{Z}_{t-1})}{\mathbb{Q}(\mathcal{S}_t)} \right] \qquad (53) \\ &\triangleq \mathcal{L}_{\mathrm{S}}^{(t)}. \qquad (54) \end{aligned}$$

The final step uses the definition of KL divergence, $D_{KL}(P\|Q) = \mathbb{E}_P[\log \frac{P}{Q}] = -\mathbb{E}_P[\log \frac{Q}{P}]$. □

G.2 Bound for Temporal Information Bottleneck (Z)

Consider the second term $I(\mathbf{Z}_t; \mathcal{D} \mid \mathbf{H}_z^{(t)})$. We derive the upper bound $\mathcal{L}_{\mathrm{Z}}^{(t)}$ as follows.

Proof. Similar to the S derivation, we consider the relevant input subset $(\mathbf{A}, \mathbf{X}_{1:t})$. We explicitly introduce the current structure $\mathcal{S}_t$ and the previous state $\mathbf{Z}_{t-1}$ into the conditioning set using the property that adding variables increases mutual information:

$$\begin{aligned} I(\mathbf{Z}_t; \mathcal{D} \mid \mathbf{H}_z^{(t)}) &= I(\mathbf{Z}_t; \mathbf{A}, \mathbf{X}_{1:t} \mid \mathbf{H}_z^{(t)}) \\ &\leq I(\mathbf{Z}_t; \mathbf{A}, \mathbf{X}_{1:t}, \mathcal{S}_t, \mathbf{Z}_{t-1} \mid \mathbf{H}_z^{(t)}). \end{aligned} \tag{55}$$

Using the Chain Rule, we decompose this term with respect to the static graph structure $\mathbf{A}$:

$$\begin{aligned} &I(\mathbf{Z}_t; \mathbf{A}, \mathbf{X}_{1:t}, \mathcal{S}_t, \mathbf{Z}_{t-1} \mid \mathbf{H}_z^{(t)}) \\ &= I(\mathbf{Z}_t; \mathbf{X}_{1:t}, \mathcal{S}_t, \mathbf{Z}_{t-1} \mid \mathbf{H}_z^{(t)}) + \underbrace{I(\mathbf{Z}_t; \mathbf{A} \mid \mathbf{H}_z^{(t)}, \mathbf{X}_{1:t}, \mathcal{S}_t, \mathbf{Z}_{t-1})}_{\text{Term A}}. \end{aligned} \tag{56}$$

Analyzing **Term A**: In our framework, the latent state $\mathbf{Z}_t$ is computed by aggregating information over the dynamic graph structure $\mathcal{S}_t$. Given the dynamic structure $\mathcal{S}_t$, the previous state $\mathbf{Z}_{t-1}$, and

features $\mathbf{X}_{1:t}$, the original static structure $\mathbf{A}$ provides no additional information for generating $\mathbf{Z}_t$. In other words, $\mathbf{Z}_t \perp \mathbf{A} \mid \mathcal{S}_t, \mathbf{X}_{1:t}, \mathbf{Z}_{t-1}$. Therefore, **Term A** $= 0$.

Now we focus on the remaining term. We first remove the conditioning on the history $\mathbf{H}_z^{(t)}$ to obtain an upper bound, relying on the property that conditioning reduces entropy (i.e., $H(\mathbf{Z}_t \mid \mathbf{H}_z^{(t)}) \leq H(\mathbf{Z}_t)$):

$$I(\mathbf{Z}_t; \mathbf{X}_{1:t}, \mathcal{S}_t, \mathbf{Z}_{t-1} \mid \mathbf{H}_z^{(t)}) \leq I(\mathbf{Z}_t; \mathbf{X}_{1:t}, \mathcal{S}_t, \mathbf{Z}_{t-1}). \tag{57}$$

Let $\mathbf{V}'_t = (\mathbf{X}_{1:t}, \mathcal{S}_t, \mathbf{Z}_{t-1})$. We expand the mutual information by introducing a variational prior $\mathbb{Q}(\mathbf{Z}_t)$ to approximate the marginal $\mathbb{P}(\mathbf{Z}_t)$:

$$\begin{aligned}
I(\mathbf{Z}_t; \mathbf{V}'_t) &= \mathbb{E}_{\mathbb{P}(\mathbf{Z}_t, \mathbf{V}'_t)} \left[\log \frac{\mathbb{P}(\mathbf{Z}_t \mid \mathbf{V}'_t)}{\mathbb{P}(\mathbf{Z}_t)} \right] \\
&= \mathbb{E}_{\mathbb{P}(\mathbf{Z}_t, \mathbf{V}'_t)} \left[\log \left(\frac{\mathbb{P}(\mathbf{Z}_t \mid \mathbf{V}'_t)}{\mathbb{Q}(\mathbf{Z}_t)} \cdot \frac{\mathbb{Q}(\mathbf{Z}_t)}{\mathbb{P}(\mathbf{Z}_t)} \right) \right] \\
&= \mathbb{E}_{\mathbb{P}(\mathbf{Z}_t, \mathbf{V}'_t)} \left[\log \frac{\mathbb{P}(\mathbf{Z}_t \mid \mathbf{V}'_t)}{\mathbb{Q}(\mathbf{Z}_t)} \right] + \mathbb{E}_{\mathbb{P}(\mathbf{Z}_t)} \left[\log \frac{\mathbb{Q}(\mathbf{Z}_t)}{\mathbb{P}(\mathbf{Z}_t)} \right] \\
&= \mathbb{E}_{\mathbb{P}} \left[\log \frac{\mathbb{P}(\mathbf{Z}_t \mid \mathbf{X}_{1:t}, \mathcal{S}_t, \mathbf{Z}_{t-1})}{\mathbb{Q}(\mathbf{Z}_t)} \right] - D_{KL}(\mathbb{P}(\mathbf{Z}_t) \| \mathbb{Q}(\mathbf{Z}_t)) \\
&\leq \mathbb{E}_{\mathbb{P}} \left[\log \frac{\mathbb{P}(\mathbf{Z}_t \mid \mathbf{X}_{1:t}, \mathcal{S}_t, \mathbf{Z}_{t-1})}{\mathbb{Q}(\mathbf{Z}_t)} \right] \qquad (58) \\
&\triangleq \mathcal{L}_{\mathrm{Z}}^{(t)}. \qquad (59)
\end{aligned}$$

□

H Additional Experiments

H.1 Statistical Significance of Main Results

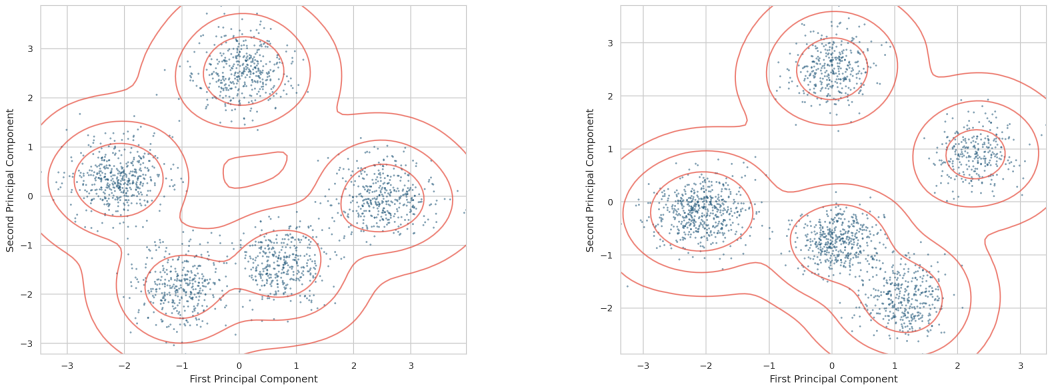
Table 3: Detailed performance on real-world traffic datasets. Results are presented in the format of "mean ± standard deviation" evaluated over N independent runs with different random seeds. The best mean results are highlighted in **bold**, and the second-best are underlined.

MODEL	PeMS04			PeMS07			PeMS08			TDrive		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DCRNN	22.54 ± 0.20	36.81 ± 0.35	14.62% ± 0.06%	23.79 ± 0.19	36.26 ± 0.33	12.41% ± 0.05%	18.02 ± 0.18	28.33 ± 0.31	11.09% ± 0.05%	21.86 ± 0.20	38.85 ± 0.34	18.12% ± 0.06%
STGCN	21.92 ± 0.18	34.51 ± 0.32	13.95% ± 0.05%	22.71 ± 0.17	35.62 ± 0.30	11.84% ± 0.04%	17.96 ± 0.16	26.94 ± 0.29	11.36% ± 0.04%	21.48 ± 0.18	37.36 ± 0.31	17.18% ± 0.05%
GWNET	19.18 ± 0.14	31.89 ± 0.26	13.12% ± 0.04%	21.36 ± 0.13	33.87 ± 0.25	9.19% ± 0.03%	15.21 ± 0.12	24.67 ± 0.23	9.62% ± 0.03%	19.53 ± 0.14	36.19 ± 0.27	16.17% ± 0.04%
MTGNN	19.29 ± 0.13	31.34 ± 0.24	13.08% ± 0.04%	20.63 ± 0.12	34.29 ± 0.22	8.92% ± 0.03%	15.25 ± 0.11	25.08 ± 0.21	10.05% ± 0.03%	18.69 ± 0.13	35.18 ± 0.25	16.49% ± 0.04%
STSGCN	21.02 ± 0.16	33.83 ± 0.28	13.71% ± 0.05%	24.41 ± 0.15	38.74 ± 0.30	10.32% ± 0.04%	17.29 ± 0.14	26.61 ± 0.26	11.08% ± 0.04%	24.02 ± 0.17	41.85 ± 0.29	18.76% ± 0.05%
STFGNN	19.97 ± 0.15	31.65 ± 0.25	13.18% ± 0.04%	21.88 ± 0.14	35.96 ± 0.24	9.10% ± 0.03%	16.47 ± 0.13	26.38 ± 0.22	10.41% ± 0.03%	22.36 ± 0.15	40.54 ± 0.26	18.68% ± 0.04%
STGODE	20.66 ± 0.14	33.05 ± 0.27	13.64% ± 0.04%	23.15 ± 0.13	35.95 ± 0.25	10.26% ± 0.03%	16.95 ± 0.12	26.09 ± 0.23	10.74% ± 0.03%	21.78 ± 0.14	39.64 ± 0.28	18.04% ± 0.04%
STGNCDE	19.37 ± 0.12	30.92 ± 0.22	12.90% ± 0.03%	20.48 ± 0.11	34.21 ± 0.21	8.73% ± 0.02%	15.33 ± 0.10	24.96 ± 0.20	9.78% ± 0.02%	19.24 ± 0.12	36.11 ± 0.23	16.24% ± 0.03%
STDN	23.60 ± 0.20	35.32 ± 0.34	20.22% ± 0.06%	23.32 ± 0.19	36.42 ± 0.32	10.41% ± 0.05%	20.63 ± 0.19	30.26 ± 0.33	18.85% ± 0.06%	24.49 ± 0.20	38.62 ± 0.35	22.62% ± 0.06%
STTN	19.61 ± 0.13	31.73 ± 0.23	13.51% ± 0.03%	21.18 ± 0.12	34.77 ± 0.22	9.82% ± 0.02%	15.34 ± 0.11	25.12 ± 0.21	10.22% ± 0.02%	20.31 ± 0.13	37.14 ± 0.24	16.52% ± 0.03%
GMAN	18.97 ± 0.11	31.79 ± 0.21	13.07% ± 0.03%	21.12 ± 0.10	33.91 ± 0.20	9.16% ± 0.02%	15.45 ± 0.09	24.76 ± 0.19	10.25% ± 0.02%	19.17 ± 0.11	36.03 ± 0.22	16.37% ± 0.03%
TFormer	19.02 ± 0.10	31.16 ± 0.20	12.84% ± 0.02%	20.58 ± 0.09	34.23 ± 0.19	8.86% ± 0.02%	15.05 ± 0.08	25.03 ± 0.18	9.81% ± 0.02%	18.82 ± 0.10	34.69 ± 0.20	16.84% ± 0.02%
ASTGNN	18.64 ± 0.09	31.21 ± 0.19	12.51% ± 0.02%	20.76 ± 0.08	33.84 ± 0.18	8.97% ± 0.02%	15.11 ± 0.07	24.56 ± 0.17	9.59% ± 0.02%	18.75 ± 0.09	33.58 ± 0.19	15.71% ± 0.02%
PDFormer	18.56 ± 0.09	30.45 ± 0.18	12.37% ± 0.02%	19.98 ± 0.08	33.37 ± 0.17	8.57% ± 0.02%	13.78 ± 0.07	23.88 ± 0.16	9.45% ± 0.02%	17.59 ± 0.09	31.87 ± 0.18	15.17% ± 0.02%
ST-SSL	18.71 ± 0.08	31.29 ± 0.17	12.61% ± 0.02%	20.52 ± 0.07	33.81 ± 0.16	9.76% ± 0.02%	14.89 ± 0.06	24.32 ± 0.15	10.18% ± 0.02%	16.78 ± 0.08	31.15 ± 0.17	**13.48%** ± 0.01%
GraphST	19.26 ± 0.10	31.39 ± 0.19	13.04% ± 0.02%	20.71 ± 0.09	34.14 ± 0.18	10.19% ± 0.02%	14.97 ± 0.08	25.00 ± 0.17	10.49% ± 0.02%	15.96 ± 0.07	31.81 ± 0.16	14.67% ± 0.02%
LightST	19.21 ± 0.09	31.31 ± 0.18	12.70% ± 0.02%	20.78 ± 0.08	33.95 ± 0.17	8.98% ± 0.02%	15.43 ± 0.07	24.52 ± 0.16	9.84% ± 0.02%	16.49 ± 0.08	33.47 ± 0.18	16.87% ± 0.02%
STExplainer	18.86 ± 0.08	30.45 ± 0.16	12.23% ± 0.01%	20.03 ± 0.07	33.44 ± 0.15	8.49% ± 0.01%	14.72 ± 0.06	23.87 ± 0.14	9.71% ± 0.01%	16.47 ± 0.07	31.16 ± 0.15	15.78% ± 0.03%
RSTIB-MLP	18.64 ± 0.07	30.34 ± 0.15	12.41% ± 0.01%	19.81 ± 0.06	33.85 ± 0.14	8.42% ± 0.01%	14.55 ± 0.05	24.19 ± 0.13	9.42% ± 0.01%	16.85 ± 0.06	32.14 ± 0.14	16.78% ± 0.03%
STGIB	**18.48** ± 0.06	**30.17** ± 0.13	**12.07%** ± 0.01%	**19.54** ± 0.05	**32.89** ± 0.12	**8.18%** ± 0.01%	**13.61** ± 0.04	**23.52** ± 0.11	**9.01%** ± 0.01%	**14.84** ± 0.05	**28.42** ± 0.12	14.52% ± 0.02%

We report the standard deviation of STGIB performance under five runs with different random seeds. The result in Table 3 that the performance of STGIB is stable.

H.2 Analysis of Learned GMM Prior

To examine whether the Gaussian mixture prior captures meaningful latent structures, we visualize samples drawn from the learned prior and project them into a two-dimensional space using PCA. As shown in Fig. 4, the learned prior forms several distinguishable density regions on both PeMS04 and TDrive. The mixture components are relatively separated from each other, while samples within each component are densely concentrated. This indicates that the prior does not collapse into a single unimodal distribution, but instead captures multiple latent traffic patterns. Such a multi-modal



(a) GMM Prior Distribution in PeMS04

(b) GMM Prior Distribution in TDrive

Figure 4: Visualization of the learned GMM prior distributions on PeMS04 and TDrive. Samples drawn from the learned GMM prior are projected into a 2D space by PCA, where contours indicate the density of different mixture components.

structure is consistent with the heterogeneous dynamics of traffic sequences, where different temporal states, such as peak, valley, and transition periods, may correspond to different latent modes. By regularizing temporal representations with this learned prior, STGIB can better compress redundant fluctuations while preserving diverse traffic evolution patterns that are useful for prediction.

H.3 Efficiency Study

Table 4: Efficiency-performance trade-off on PeMS04 and PeMS08. Runtime is reported as seconds per epoch.

Type	**Model**	**PeMS04**					**PeMS08**				
		MAE	RMSE	MAPE	Train	Infer	MAE	RMSE	MAPE	Train	Infer
Baseline	STExplainer	18.86	30.45	12.23%	192.09	29.34	14.72	23.87	9.71%	204.06	30.23
	ASTGNN	18.64	31.21	12.51%	217.56	72.19	15.11	24.56	9.59%	145.79	8.67
	PDFormer	18.56	30.45	12.37%	136.47	8.47	13.78	23.88	9.45%	109.40	7.48
STGIB	w/o Patch	18.55	30.24	12.25%	117.48	13.94	13.76	23.76	9.24%	90.39	10.25
	$P = 6$	18.95	31.08	13.24%	25.49	2.15	14.25	24.75	9.89%	20.62	1.26
	$P = 4$	18.76	30.57	12.78%	35.25	3.58	13.89	24.24	9.64%	28.57	2.45
	$P = 3$	18.61	30.34	12.38%	44.91	4.64	13.75	23.86	9.26%	36.37	3.48
	$P = 2$	18.48	30.17	12.07%	67.36	7.28	13.61	23.52	9.01%	51.04	5.18

We evaluate the efficiency-performance trade-off of STGIB from two perspectives: comparison with representative baselines and the effect of temporal patching. As shown in Table 4, STGIB with $P = 2$ achieves the best predictive performance while requiring substantially less training and inference time than STExplainer, ASTGNN, and PDFormer on both datasets. Compared with PDFormer, STGIB reduces training time by about 50.6% on PeMS04 and 53.3% on PeMS08, while also achieving lower inference latency.

We further vary the patch size to examine patching as an efficiency-oriented temporal coarsening strategy. The w/o Patch variant performs recurrent evolution at the original step-level granularity and obtains results close to STGIB, indicating that patching is not the main source of predictive gains. However, it incurs noticeably higher computational cost due to the longer recurrent trajectory. Larger patch sizes such as $P = 4$ and $P = 6$ further reduce runtime, but their prediction errors increase, suggesting that overly coarse temporal granularity may obscure fine-grained traffic evolution. Overall, $P = 2$ provides the best balance between forecasting accuracy and computational efficiency, supporting our use of patching as an acceleration strategy rather than an additional explanatory mechanism.

H.4 Hyperparameter Analysis

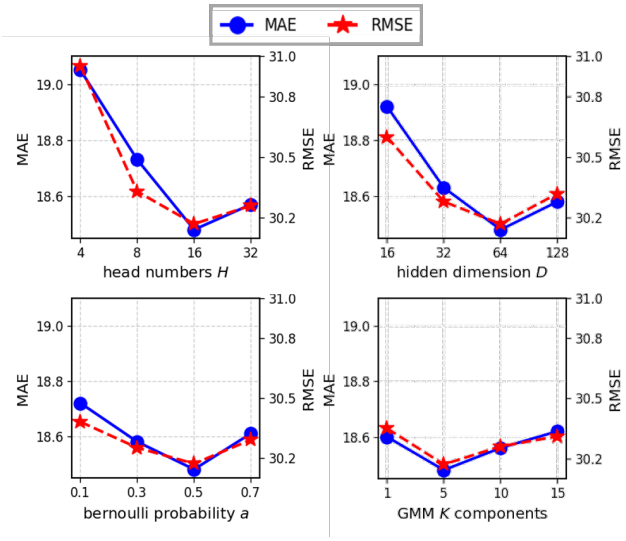


Figure 5: Hyperparameter analysis of the proposed STGIB.

We conduct a sensitivity analysis on PeMS04 to examine the effects of four key hyperparameters: attention heads H, hidden dimension D, structural sparsity α, and the number of GMM components K, as shown in Fig. 5. Regarding model capacity, performance improves as H increases to 16 and D to 64, indicating that sufficient parameterization is necessary to capture complex spatio-temporal dynamics. Further increasing model complexity (e.g., $H = 32$ or $D = 128$), however, degrades performance, suggesting that the model begins to overfit. For structure learning, varying the Bernoulli prior α exhibits a clear U-shaped trend, with $\alpha = 0.5$ achieving the best trade-off between connectivity preservation and noise suppression. Finally, the analysis of latent components highlights the importance of multi-modality: performance is optimized at $K = 5$, substantially outperforming the unimodal setting ($K = 1$), while larger values introduce redundancy without further gains.

I Experimental details

I.1 Detailed Dataset Description

The experiments are conducted on four real-world traffic datasets, namely PeMS04, PeMS07, PeMS08 [15], and TDrive [33]. The three PeMS datasets are collected from the California Transportation Agencies (Caltrans) Performance Measurement System, where traffic flow is recorded by sensors deployed on highway networks. TDrive is a grid-based urban mobility dataset constructed from taxi trajectories, where each grid cell is treated as a spatial unit and its inflow and outflow are used as traffic observations. The detailed statistics are summarized in Table 5.

- **PeMS04**: This dataset contains traffic flow data from the San Francisco Bay Area (District 4), with 307 sensors from January 1, 2018, to February 28, 2018.
- **PeMS07**: This dataset is collected from the Los Angeles Area (District 7), with 883 sensors from May 1, 2017, to August 31, 2017.
- **PeMS08**: This dataset corresponds to the San Bernardino Area (District 8), with 170 sensors from July 1, 2016, to August 31, 2016.
- **TDrive**: This dataset is constructed from Beijing urban taxi trajectory records. We divide the target area into grid cells and model each grid as a node, using hourly inflow and outflow as traffic features.

For PeMS datasets, we construct the adjacency matrix $\mathbf{A}$ based on road-network distances between sensors. The edge weights are calculated by a thresholded Gaussian kernel:

$$\mathbf{A}_{ij} = \begin{cases} \exp\left(-\frac{\text{dist}(v_i,v_j)^2}{\sigma^2}\right), & \text{if dist}(v_i, v_j) \leq \kappa, \\ 0, & \text{otherwise}, \end{cases} \tag{60}$$

where $\text{dist}(v_i, v_j)$ denotes the road-network distance between sensors v_i and v_j, σ is the standard deviation of distances, and κ controls graph sparsity. For TDrive, we construct the spatial graph according to grid adjacency, where neighboring grid cells are connected to reflect local urban mobility interactions.

Table 5: Statistics of the real-world traffic datasets.

DATASET	#NODES	#INTERVAL	#FEATURES	#TIME RANGE
PEMS04	307	5MIN	1	01/2018–02/2018
PEMS07	883	5MIN	1	05/2017–08/2017
PEMS08	170	5MIN	1	07/2016–08/2016
TDRIVE	1024	60MIN	2	01/2015–06/2015

I.2 Model Settings

All experiments are implemented in PyTorch 1.9.1 and executed on a workstation powered by an NVIDIA RTX 3090 GPU. Regarding the model architecture, we initialize the embedding dimensions for nodes, traffic flow, and temporal features to 8, while the hidden dimension of STGIB is projected to 64. We adopt a patch size of 2 and utilize 16 attention heads to capture complex spatio-temporal dependencies. For the variational constraint, we instantiate the latent state prior as a Gaussian Mixture Model with 5 components. Additionally, the sparsity parameter γ of the Bernoulli prior in the Structure Information Bottleneck is set to 0.5 to regulate the graph density. The balancing coefficient β for the IB regularization terms is set to 1. We optimize the model using Adam with an initial learning rate of 0.001, employing a multi-step scheduler that decays the rate by 0.5 at epochs 50, 100, and 150. The batch size is set to 32, which is adjusted to 16 in cases of limited GPU memory, with a maximum of 200 training epochs.

I.3 Calculation method of evaluation indicators

In this section, we provide the detailed definitions and calculation methods for both predictive performance metrics and explainability metrics used in our experiments.

Predictive Performance Metrics Let $Y = \{y_1, y_2, \ldots, y_n\}$ denote the ground truth values and $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$ denote the predicted values, where n is the total number of samples. The three metrics are defined as follows:

- **Mean Absolute Error (MAE)** measures the average absolute difference between predicted and observed values:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{61}$$

- **Root Mean Squared Error (RMSE)** penalizes larger errors more heavily and is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{62}$$

- **Mean Absolute Percentage Error (MAPE)** measures the relative error as a percentage:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{63}$$

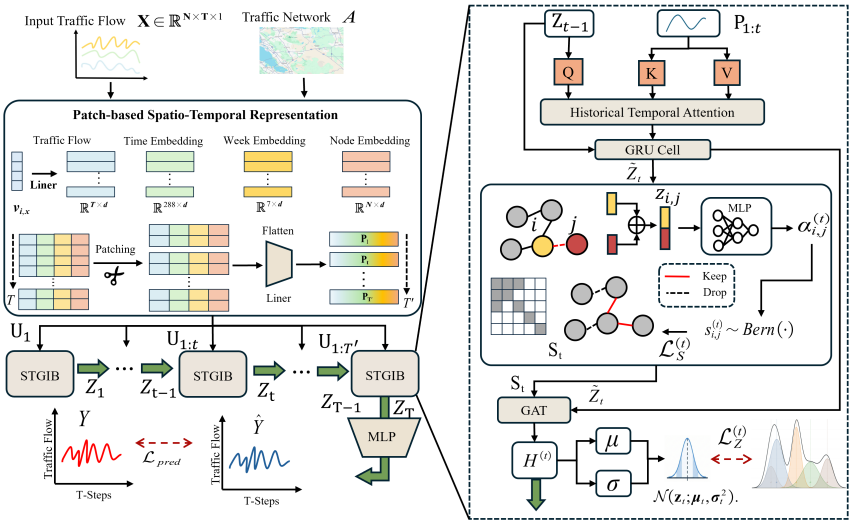


Figure 6: The overall architecture of STGIB.

Note that missing values and zeros in the ground truth are excluded from the calculation to prevent biased evaluations.

Explainability Metrics Due to the lack of ground-truth labels specifically designed for spatio-temporal explainability, we adopt the quantitative evaluation method established in STExplainer. Let f be the trained predictive function and $\mathcal{G}_i$ represent the input full graph for the i-th sample. We focus on the explainable subgraph generated at the last time step, denoted as m_i, which preserves the most critical spatial-temporal dependencies identified by the model.

Sparsity. This metric evaluates the compactness of the explanation. Since the model is designed to filter out redundant information, sparsity is calculated as the proportion of edges discarded in the generated subgraph. Formally:

$$\text{Sparsity} = \frac{1}{Q}\sum_{i=1}^{Q}\left(1 - \frac{|m_i|}{|\mathcal{E}_i|}\right), \tag{64}$$

where Q is the number of test samples, $|m_i|$ is the number of edges in the explainable subgraph, and $|\mathcal{E}_i|$ is the total number of edges in the original graph. Higher sparsity indicates a more compact explanation.

Fidelity. This metric assesses the sufficiency of the generated subgraph. Following the counterfactual evaluation method, we measure the prediction shift incurred when the information within m_i is removed. Let $s \in [0, 1]$ be the masking ratio. The fidelity at ratio s is defined as:

$$\text{Fidelity}(s) = \frac{1}{Q}\sum_{i=1}^{Q}\left\| f(\mathcal{G}_i) - f(\mathcal{G}_i^{\setminus \text{mask}(m_i,s)}) \right\|, \tag{65}$$

where $\mathcal{G}_i^{\setminus \text{mask}(m_i,s)}$ represents the input graph where a proportion s of the edges in the explainable subgraph m_i are masked out. Specifically, $s = 0$ corresponds to the original graph $\mathcal{G}_i$, while $s = 1$ indicates that the entire explainable subgraph m_i is removed from the input. A higher fidelity score implies that the masked subgraph contained crucial information for the model's prediction.

J Framework

The framework of STGIB is shown in Figure 6.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Abstract and Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: See Appendix E, F and G.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 5.1 and Appendix I.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code and data was submitted along with the supplementary materials.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 5.1 and Appendix I.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Appendix H.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix I.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We ensured that the study complied with the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix A.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not suffer from this risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were used only for language polishing and do not impact the scientific contribution.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.