WHEN EVENTS RESHAPE TRAFFIC GRAPHS: EVENT-TRIGGERED DYNAMIC GRAPH LEARNING FOR TRAFFIC FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

With the increasing complexity of urban traffic systems and the growing availability of spatio-temporal observations, accurate traffic forecasting under dynamic and event-driven conditions has become a critical challenge. Most existing dynamic graph forecasting methods update spatial dependencies according to fixed time intervals or historical traffic correlations, allowing them to capture that traffic relationships have changed but making it difficult to identify whether such changes are triggered by accidents, road construction, severe weather, or large-scale events. Moreover, event impacts are rarely isolated to individual locations, but instead propagate through upstream and downstream road segments and may further induce traffic redistribution over alternative routes. Existing methods also tend to model long-term traffic regularities and short-term event disturbances within a unified representation, causing transient anomalies to interfere with stable trends and periodic patterns. To overcome these limitations, we propose ETDGN, an **E**vent-**T**riggered **D**isentangled **G**raph **N**etwork for traffic forecasting. ETDGN introduces an event-triggering mechanism that jointly exploits explicit contextual information and implicit traffic anomalies to selectively activate graph updates when event-related changes occur. An event-conditioned dynamic graph is then constructed to capture event-induced spatial dependency shifts and model the propagation of disturbances over the traffic network. To prevent transient events from contaminating stable traffic dynamics, ETDGN further disentangles long-term trend and periodic components from short-term event residuals, and incorporates event-aware residual correction into the routine forecast. Extensive experiments demonstrate that ETDGN outperforms state-of-the-art baselines in both forecasting accuracy and scalability, validating its effectiveness for continual spatio-temporal forecasting. Code is available at https://anonymous.4open.science/r/ETDGN-ICLR-2027-C367.

1 INTRODUCTION

With the rapid development of intelligent transportation systems and urban sensing infrastructures (Shao et al., 2024; Zhang et al., 2025), large-scale traffic observations have become increasingly available from road sensors (Shi et al., 2025), cameras (Fan et al., 2025), and connected vehicles (Jin et al., 2024). Accurate traffic forecasting is essential for urban transportation management. However, real-world traffic systems are frequently affected by accidents, road construction, severe weather, and large-scale activities, which can abruptly alter local traffic states and reshape spatial dependencies among road segments.

Spatio-temporal graph neural networks (Kong et al., 2024; Dong et al., 2024; Cao et al., 2025) have been widely used to model traffic dependencies. Early methods rely on predefined road graphs, while subsequent studies introduce adaptive or data-driven graph learning (Wu et al., 2019). More recent dynamic graph models (Hu et al., 2025; Cao et al., 2025; Liu & Zhang, 2026) allow spatial correlations to evolve over time. Nevertheless, graph evolution is still mainly driven by temporal variation or traffic correlations (Chen et al., 2026; Yu et al., 2025; Liu & Zhang, 2026), providing limited understanding of *why* spatial dependencies change and *how* event-induced disturbances subsequently propagate through the traffic network.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Challenge 1: Event-Unaware Dynamic Graph Adaptation. As illustrated in Figure 1(a)–(b), most dynamic graph models update graph structures at predefined time intervals or continuously reconstruct spatial correlations according to traffic observations. Although such strategies are capable of capturing time-varying dependencies, the resulting graph changes are implicitly coupled with time progression rather than explicitly linked to the events that cause them. This event-agnostic updating paradigm not only introduces unnecessary graph variations during regular traffic conditions, but also makes it difficult to establish an interpretable connection among event occurrence, graph structural change, and subsequent traffic-state evolution.

Challenge 2: Limited Modeling of Event-Induced Impact Propagation. A traffic event rarely affects only the road segment where it occurs. As shown in Figure 1(c), congestion caused by an accident or severe weather may gradually propagate toward upstream and downstream roads, while traffic diversion can simultaneously alter the states of alternative routes. Such event-induced dependencies are directional, time-varying, and highly dependent on the current traffic context. Moreover, traffic observations simultaneously contain stable long-term trends, recurrent periodic patterns, and short-lived event disturbances. Modeling these heterogeneous components within a unified representation may incorrectly absorb transient event-induced deviations into regular traffic dynamics, thereby contaminating stable patterns and degrading forecasting reliability, particularly over longer prediction horizons.

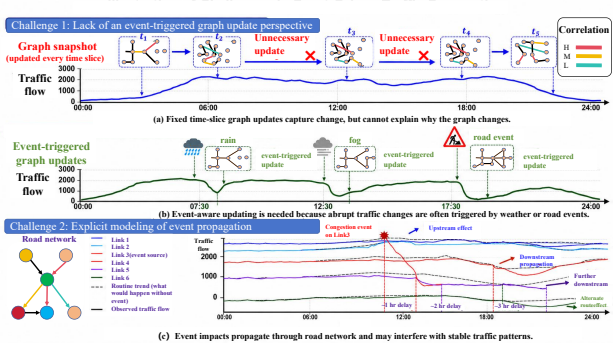

Figure 1: Illustration of the two key challenges addressed by ETDGN.

To address these challenges, we propose ETDGN, an Event-Triggered Disentangled Graph Network for traffic forecasting. ETDGN selectively activates graph adaptation according to event-related evidence. Specifically, an event-triggering mechanism jointly exploits available exogenous contexts and implicit abnormalities inferred from traffic observations to determine whether event-specific graph adaptation is necessary. Once activated, an event-conditioned graph reorganizes spatial dependencies around the detected disturbance and explicitly models how its impact propagates across related road segments. To prevent short-term event effects from interfering with stable traffic dynamics, ETDGN further disentangles routine trend and periodic components from event-induced residuals. The routine branch models regular traffic evolution, while the event branch provides adaptive residual corrections according to the detected event state and its propagated influence. Together, these designs establish an explicit modeling chain from *event triggering*, to *graph reorganization*, to *impact propagation*, and finally to *traffic forecasting*.

Our main contributions are summarized as follows: **1)** We introduce an event-triggered perspective into dynamic graph traffic forecasting, shifting graph adaptation from conventional time-driven updating toward event-aware selective updating and explicitly connecting traffic abnormalities with graph structural changes. **2)** We develop an event-conditioned dynamic graph and propagation mechanism to capture the direction, range, and temporal evolution of event impacts, together with a disentangled forecasting strategy that prevents transient disturbances from contaminating stable traffic trends and periodic patterns. **3)** Extensive experiments on multiple real-world datasets demonstrate that ETDGN consistently outperforms state-of-the-art baselines in terms of forecasting accuracy, adaptability, and scalability.

2 Related Work

Dynamic Graph Learning for Traffic Forecasting. Early methods such as STGCN (Yu et al., 2018) and DCRNN (Li et al., 2018) rely on fixed road graphs, whereas Graph WaveNet (Wu et al., 2019) and AGCRN (Bai et al., 2020) learn adaptive spatial dependencies. Continual forecasting methods such as TrafficStream (Chen et al., 2021), STKEC (Wang et al., 2023a), and PECPM (Wang et al., 2023b) adapt to expanding graphs and distribution shifts through replay, knowledge consolidation, or pattern matching. STID (Shao et al., 2022), STAEformer (Liu et al., 2023a), and iTrans-

former (Liu et al., 2024) provide alternative representations for spatio-temporal or multivariate dependencies. More recent methods, including STRAP (Zhang et al., 2026a), EAC (Chen & Liang, 2025), and STBP (Liu & Zhang, 2026), further improve adaptation to evolving spatio-temporal patterns through retrieval, prompt tuning, and dynamic correlation modeling. Meanwhile, DGCRN (Li et al., 2023), MTSF-DG (Zhao et al., 2023), STDN (Cao et al., 2025), MetaDG (Zou et al., 2026), and FedSTGD (Wang et al., 2026a) explicitly model time-varying graph structures or spatial dependencies.

Event-Aware Traffic Modeling and Impact Propagation. Recent studies have increasingly considered abnormal traffic variations and external contexts. TESTAM (Lee & Ko, 2024), HyperD (Shao et al., 2026), and ConFormer (Wang et al., 2026b) model non-recurring traffic through expert routing, periodicity decoupling, and accident-informed graph propagation, respectively; PDFormer (Jiang et al., 2023) and CaST (Xia et al., 2023) further model delayed or causal spatial influence. Other approaches explore physical interactions, multi-source dependencies, and disentangled traffic dynamics (Xiang et al., 2026; Zhang et al., 2026b; Xia et al., 2023; 2026). Overall, ETDGN differs from existing methods by explicitly linking event occurrence to graph adaptation and modeling the direction, range, and temporal evolution of event-induced propagation.

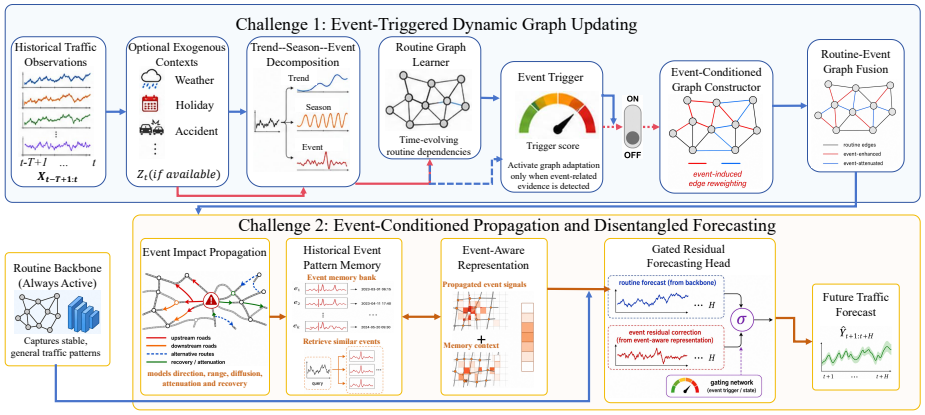

Figure 2: The overall workflow and architecture of ETDGN

3 Overview of ETDGN

The workflow and architecture of ETDGN are shown in Figure 2. ETDGN consists of two core components: an event-triggered dynamic graph updating module and an event-aware disentangled forecasting module. The former determines *when and why* the traffic graph should be adjusted by jointly exploiting explicit exogenous contexts, implicit traffic residuals, and spatial propagation evidence. Instead of updating the graph at every time step, it constructs an event-conditioned graph only when event-related changes are detected. The latter focuses on *how* the detected event influence evolves after graph reorganization. It explicitly propagates the event effect over the event-conditioned graph and generates an event-aware residual correction over the routine forecast.

3.1 Problem Formulation

Let $G = (V, E, \mathbf{A}^0)$ denote a traffic spatio-temporal graph, where V is the set of sensors, road segments, or spatial grids, and $\mathbf{A}^0 \in \mathbb{R}^{N\times N}$ is the base spatial relation matrix. Given historical traffic observations and optional exogenous contexts, the forecasting task is defined as:

$$\begin{aligned} &\mathbf{X}_{t-T+1:t} \in \mathbb{R}^{T\times N\times C}, \qquad \mathbf{Z}_{t-T+1:t} \in \mathbb{R}^{T\times D_z}, \\ &\boldsymbol{\tau}_{t-T+1:t} \in \mathbb{R}^{T\times D_\tau}, \qquad \hat{\mathbf{Y}}_{t+1:t+H} \in \mathbb{R}^{H\times N\times C_y}, \\ &\hat{\mathbf{Y}}_{t+1:t+H} = f_\Theta\left(\mathbf{X}_{t-T+1:t}, \mathbf{A}^0, \mathbf{Z}_{t-T+1:t}, \boldsymbol{\tau}_{t-T+1:t}\right). \end{aligned} \tag{1}$$

Here, $\mathbf{X}$ denotes traffic observations, $\mathbf{Z}$ denotes optional exogenous variables such as weather or holidays, and $\boldsymbol{\tau}$ denotes temporal context. If explicit exogenous variables are unavailable, we set $\mathbf{Z} = \emptyset$, and the model infers latent event effects from traffic residuals and spatial propagation evidence.

3.2 EVENT-TRIGGERED DYNAMIC GRAPH UPDATING

Dynamic graph changes should be related to event effects instead of being updated only according to time. If the graph is reconstructed at every time step, routine fluctuations and truly event-induced structural changes are treated in the same way, making it difficult to explain why spatial dependencies change. ETDGN therefore first establishes a routine traffic reference, then evaluates event-related evidence, and finally activates an event-conditioned graph only when such evidence becomes sufficiently strong. Traffic observations contain routine trends, recurrent seasonal patterns, and abnormal residuals. ETDGN decomposes the input sequence and builds the initial representation as:

$$\begin{aligned} &\mathbf{X}^{tr} = MA(\mathbf{X}), \qquad \mathbf{X}^{se} = F_{se}\left(\mathbf{X} - \mathbf{X}^{tr}, \boldsymbol{\tau}\right), \qquad \mathbf{X}^{ev} = \mathbf{X} - \mathbf{X}^{tr} - \mathbf{X}^{se}, \\ &\mathbf{H}_t^0 = \phi_x(\mathbf{X}_t) + \phi_{tr}(\mathbf{X}_t^{tr}) + \phi_{se}(\mathbf{X}_t^{se}) + \phi_{ev}(\mathbf{X}_t^{ev}) + \phi_\tau(\boldsymbol{\tau}_t) + \phi_z(\mathbf{Z}_t). \end{aligned} \tag{2}$$

The trend component $\mathbf{X}^{tr}$ captures slowly varying traffic levels, the seasonal component $\mathbf{X}^{se}$ captures recurrent temporal patterns, and the residual component $\mathbf{X}^{ev}$ captures deviations that cannot be explained by routine patterns. This decomposition is necessary because not every large traffic change should be treated as an event. For example, morning and evening peaks may produce substantial variations but still belong to normal periodic evolution. By separating predictable routine components from unexplained residuals, $\mathbf{X}^{ev}$ provides cleaner evidence for detecting event-related abnormalities. Meanwhile, the encoded representation $\mathbf{H}_t^0$ preserves all components for subsequent spatio-temporal modeling, avoiding information loss caused by using only residual signals. The routine backbone models traffic evolution under normal conditions. Given $\mathbf{H}_t^0$, it constructs a routine dynamic graph and produces routine traffic representations:

$$\begin{aligned} &\mathbf{Q}_t = \mathbf{H}_t^0\mathbf{W}_q, \quad \mathbf{K}_t = \mathbf{H}_t^0\mathbf{W}_k, \qquad \mathbf{S}_t^r = \frac{\mathbf{Q}_t\mathbf{K}_t^\top}{\sqrt{d}} + \mathbf{B}(\mathbf{A}^0), \\ &\mathbf{A}_t^r = TopK(Softmax(\mathbf{S}_t^r)), \qquad \mathbf{H}_t^r = F_r\left(\mathbf{H}_t^0, \mathbf{A}^0, \mathbf{A}_t^r\right). \end{aligned} \tag{3}$$

The base graph bias $\mathbf{B}(\mathbf{A}^0)$ preserves prior road-network relations, while the learned routine graph $\mathbf{A}_t^r$ captures time-varying dependencies produced by normal traffic evolution. Importantly, this routine graph is not regarded as the final event-aware structure. Its role is to provide a stable reference describing how spatial dependencies are expected to evolve without strong event intervention. The corresponding representation $\mathbf{H}_t^r$ is therefore used both for base forecasting and for evaluating whether the current observation deviates from routine evolution. This separation makes it possible to distinguish ordinary temporal variation from event-triggered graph changes, which is central to Challenge 1. The event trigger estimates node-level event intensity from three sources: explicit exogenous context, implicit traffic abnormality, and spatial propagation evidence. We use multiple evidence sources because relying on any single signal may be unreliable. Explicit events may be unavailable, traffic residuals may contain noise, and isolated local anomalies do not necessarily indicate a network-level event. Their joint use provides a more robust criterion for determining whether graph adaptation should be activated:

$$\begin{aligned} &s_{t,i}^z = \sigma(MLP_z[\mathbf{Z}_t, \Delta\mathbf{Z}_t, \boldsymbol{\tau}_t]), && \bar{s}_{t,i}^x = \sum_j \mathbf{A}_{ij}^0 s_{t,j}^x, \\ &\tilde{\mathbf{x}}_{t,i}^r = Head_r(\mathbf{H}_{t,i}^r), \quad \mathbf{r}_{t,i} = \left|\mathbf{x}_{t,i} - \tilde{\mathbf{x}}_{t,i}^r\right|, && s_{t,i}^x = \sigma\left(MLP_x\left[\mathbf{X}_{t,i}^{ev}, \Delta\mathbf{x}_{t,i}, \mathbf{r}_{t,i}\right]\right), \\ &s_{t,i}^p = \sigma\left(MLP_p\left[s_{t,i}^x, \bar{s}_{t,i}^x, \Delta\mathbf{x}_{t,i}, \sum_j \mathbf{A}_{ij}^0 \Delta\mathbf{x}_{t,j}\right]\right), \\ &g_{t,i} = \sigma\left(MLP_g\left[s_{t,i}^z, s_{t,i}^x, s_{t,i}^p, \mathbf{H}_{t,i}^r\right]\right). \end{aligned} \tag{4}$$

Here, $s_{t,i}^z$ captures observable event evidence from exogenous contexts. If such contexts are unavailable, this term is set to zero. The score $s_{t,i}^x$ measures whether the current traffic state deviates from routine evolution, while $s_{t,i}^p$ examines whether similar abnormalities are also observed in spatially related nodes. The latter is important because an actual traffic disturbance often produces coordinated changes across several connected road segments, whereas measurement noise is more likely to remain isolated. The final trigger score $g_{t,i}$ therefore represents a combined estimate of whether the current graph change can reasonably be attributed to an event. In this way, ETDGN moves dynamic graph updating from a purely time-driven mechanism toward an event-evidence-driven mechanism. To describe event persistence and recovery, ETDGN maintains an event memory:

$$\begin{aligned} \rho_{t,i} &= \sigma\left(MLP_\rho\left[g_{t,i}, g_{t-1,i}, \Delta\mathbf{x}_{t,i}\right]\right), \\ m_{t,i} &= g_{t,i} + (1 - g_{t,i})\rho_{t,i} m_{t-1,i}. \end{aligned} \tag{5}$$

The trigger intensity $g_{t,i}$ only describes the evidence observed at the current time step, whereas real traffic events usually persist for a period of time and gradually recover rather than disappearing

immediately. Therefore, directly using $g_{t,i}$ would make the event state overly sensitive to short-term fluctuations. The memory state $m_{t,i}$ preserves the influence of previously detected events and allows ETDGN to distinguish event emergence, continuation, attenuation, and recovery. This temporal persistence is also important for Challenge 2, because the propagation range of an event may continue to evolve even after the strongest local abnormality has already passed. To translate the continuous event state into a computation-level activation decision, we define

$$a_t = \mathbb{I}\left(\max_i m_{t,i} > \tau_e\right), \tag{6}$$

where τ_e is the event activation threshold selected on the validation set. When $a_t = 1$, the event-conditioned graph and propagation branch are activated; otherwise, event-specific computation is skipped and prediction relies on the routine branch. When activated, the event-conditioned graph is constructed from the routine representation and event state as:

$$\begin{aligned} S^e_{ij,t} &= \frac{(\mathbf{H}^r_{t,i}\mathbf{W}^e_q)(\mathbf{H}^r_{t,j}\mathbf{W}^e_k)^\top}{\sqrt{d}} + \psi_{geo}(i,j) + \psi_{state}(i,j,t) + \psi_{event}(i,j,t), \\ \psi_{event}(i,j,t) &= MLP_e[m_{t,i}, m_{t,j}, g_{t,i}, g_{t,j}], \qquad G^e_{ij,t} = 1-(1-m_{t,i})(1-m_{t,j}), \\ A^e_{ij,t} &= TopK\big(Softmax_j(S^e_{ij,t})\big)\cdot G^e_{ij,t}. \end{aligned} \tag{7}$$

The event relation score combines spatial proximity, traffic-state consistency, and event co-activation. This formulation is motivated by the fact that event-induced dependencies are not arbitrary: affected roads are usually constrained by topology, current traffic states, and whether both nodes participate in the same disturbance. The region gate $G^e_{ij,t}$ further prevents the event graph from being activated in normal regions. Therefore, the event-conditioned graph does not globally replace the routine graph; instead, it selectively reorganizes spatial dependencies around event-affected regions. This design establishes an explicit connection between event evidence and graph structural changes, directly addressing the core requirement of Challenge 1. Finally, ETDGN fuses the routine graph and event graph:

$$\begin{aligned} \lambda_t &= \sigma\left(MLP_\lambda\left[Mean(\mathbf{m}_t), Max(\mathbf{m}_t), \boldsymbol{\tau}_t\right]\right), \\ \mathbf{A}_t &= (1-\lambda_t)\mathbf{A}^r_t + \lambda_t\mathbf{A}^e_t + \eta\mathbf{A}^0. \end{aligned} \tag{8}$$

The fusion coefficient λ_t determines how much the current spatial structure should deviate from routine evolution. When event intensity is low, λ_t remains small and the model mainly relies on the routine graph, avoiding unnecessary graph reconstruction. When an event becomes evident, the event-conditioned graph receives a larger contribution and temporarily reorganizes the spatial dependencies. The base topology $\mathbf{A}^0$ is retained as a structural prior to avoid unconstrained graph changes. Therefore, the fused graph provides a continuous transition between routine and event conditions instead of switching abruptly between two independent graph structures.

3.3 Event-Conditioned Propagation and Disentangled Forecasting

Although the event-conditioned graph identifies event-related spatial dependencies, identifying affected edges alone does not describe how an event develops after it occurs. This directly motivates the second challenge. Event impacts may spread toward upstream roads, downstream roads, and alternative routes, and their affected range may gradually expand and then shrink during recovery. Therefore, ETDGN explicitly propagates event states over the event-conditioned graph to characterize the range, direction, and temporal evolution of the disturbance. Meanwhile, event-induced changes are typically short-lived, while traffic series also contain stable trend and periodic components. If these signals are directly mixed during prediction, transient abnormalities may be absorbed into long-term traffic representations and reduce long-horizon forecasting stability. For this reason, ETDGN keeps routine and event representations separate and uses the event branch only to model the deviation from routine evolution. ETDGN propagates event influence over the event-conditioned graph:

$$\begin{aligned} \mathbf{P}^0_t &= \mathbf{m}_t, \qquad \mathbf{P}^{l+1}_t = \alpha\mathbf{A}^e_t\mathbf{P}^l_t + (1-\alpha)\mathbf{P}^l_t, \\ \mathbf{P}_t &= \mathbf{P}^L_t, \qquad \mathbf{H}^e_t = F_e(\mathbf{H}^r_t, \mathbf{A}_t, \mathbf{P}_t). \end{aligned} \tag{9}$$

The propagated event influence $\mathbf{P}_t$ describes how disturbances expand from initially affected nodes to other related regions. Each propagation step enlarges the effective receptive field, allowing the model to capture progressively wider influence ranges. Because propagation is performed over $\mathbf{A}^e_t$ rather than only the static road graph, the propagation pathway is conditioned on the current event state and can vary across different events. The residual connection controlled by α retains local event information during multi-step propagation and prevents the event signal from being excessively

diluted. Consequently, $\mathbf{P}_t$ provides an explicit representation of event impact evolution, while $\mathbf{H}_t^e$ encodes the corresponding event-induced traffic deviation. This design directly supports Challenge 2 by modeling not only whether an influence exists, but also how far it spreads and how its spatial pattern changes over time.

Because event samples are often sparse, ETDGN introduces a lightweight memory to retrieve reusable patterns from historical contexts:

$$\begin{aligned} &\mathbf{K}^{mem} \in \mathbb{R}^{M\times d}, \quad \mathbf{V}^{mem} \in \mathbb{R}^{M\times d}, \quad \mathbf{q}_t = Pool[\mathbf{H}_t^r, \mathbf{H}_t^e, \mathbf{m}_t, \boldsymbol{\tau}_t]\,, \\ &\boldsymbol{\alpha}_t^{mem} = Softmax\left(\frac{\mathbf{q}_t(\mathbf{K}^{mem})^\top}{\sqrt{d}}\right), \quad \mathbf{C}_t^{mem} = \boldsymbol{\alpha}_t^{mem}\mathbf{V}^{mem}. \end{aligned} \tag{10}$$

If exogenous variables are available, $\mathbf{Z}_t$ can be included in the query. The historical memory is introduced because strong events are usually much less frequent than routine traffic patterns, making it difficult to learn reliable event responses from the current sample alone. The retrieved context $\mathbf{C}_t^{mem}$ allows the model to reuse similar historical event responses, such as rainy peaks, holiday transitions, abnormal congestion drops, or recovery stages. Importantly, the memory does not determine whether an event occurs; this decision is still made by the event trigger. Instead, it provides additional context after activation, helping the model estimate the likely evolution of rare but recurring disturbances.

To avoid redundant expert stacking, ETDGN adopts a residual forecasting strategy rather than parallel predictors. The routine representation is first decoded to obtain the base forecast, while the event-aware representation and retrieved event context are jointly used to estimate only the event-induced deviation:

$$\begin{aligned} &\hat{\mathbf{Y}}^r = \mathcal{D}_r(\mathbf{H}_t^r)\,, \qquad \Delta\hat{\mathbf{Y}}^e = \mathcal{D}_e(\mathbf{H}_t^e, \mathbf{P}_t, \mathbf{C}_t^{\mathrm{mem}})\,, \\ &\gamma_t = \sigma(\mathrm{MLP}_\gamma[\mathrm{Mean}(\mathbf{m}_t), \mathrm{Max}(\mathbf{m}_t), \lambda_t, \mathbf{C}_t^{\mathrm{mem}}, \boldsymbol{\tau}_t])\,, \\ &\hat{\mathbf{Y}} = \hat{\mathbf{Y}}^r + \gamma_t \odot \Delta\hat{\mathbf{Y}}^e. \end{aligned} \tag{11}$$

Here, $\hat{\mathbf{Y}}^r$ represents the routine forecast determined by stable traffic evolution, while $\Delta\hat{\mathbf{Y}}^e$ models the deviation caused by the detected event and its propagated influence. We adopt residual correction instead of directly mixing routine and event representations because the latter may cause short-term event information to overwrite stable long-term patterns. The gate γ_t controls how strongly the event correction contributes to the final prediction. During routine periods, γ_t approaches zero and ETDGN mainly relies on the routine forecast. When an event occurs, the residual branch is gradually activated according to the event intensity, graph adaptation degree, propagation state, and historical context. Therefore, event-induced disturbances are modeled only as deviations from routine evolution, which reduces their interference with trend and periodic patterns and improves the stability of long-horizon forecasting.

4 Experiments

Datasets. We evaluate ETDGN on three public traffic forecasting datasets, including PEMS-Stream (Chen et al., 2001), CA-Stream (Liu et al., 2023b), and PEMS08 (Song et al., 2020), and further use TaxiBJ (Zhang et al., 2017) for the case study; detailed dataset descriptions and experimental settings are provided in Appendix A.

Baselines and Metrics. We compare ETDGN with representative methods from three categories: ▷ *conventional traffic forecasting models*, including DCRNN (Li et al., 2017) and iTransformer (Liu et al., 2024); ▷ *continual spatio-temporal forecasting models*, including TrafficStream (Chen et al., 2021), STKEC (Wang et al., 2023a), PECPM (Wang et al., 2023b), STRAP (Zhang et al., 2026a), EAC (Chen & Liang, 2025), and STBP (Liu & Zhang, 2026); and ▷ *event-aware traffic forecasting models*, represented by ConFormer (Wang et al., 2026b). The performance of all methods is evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). We report the results at 3-, 6-, and 12-step forecasting horizons as well as the average performance over the entire 12-step prediction window.

4.1 Main Results

The main experimental results are summarized in Table 1, where we report the performance at forecasting horizons of 3, 6, and 12 steps, together with the average results. Overall, ETDGN

Table 1: Main experimental results. **Bold**: best, underline: second best.

Dataset	Metric	Horizon	DCRNN 2018	TrafficStream 2021	STKEC 2023	PECPM 2023	iTransformer 2024	STRAP 2025	EAC 2025	STBP 2026	ConFormer 2026	ETDGN
PEMS-Stream	MAE	3	20.43±0.13	14.29±0.11	14.31±0.14	14.35±0.11	17.62±0.59	14.41±0.11	13.86±0.20	11.68±0.07	11.19±0.07	**10.54±0.07**
		6	20.47±0.20	16.43±0.04	16.41±0.12	16.38±0.14	20.90±0.95	16.19±0.11	15.33±0.16	12.26±0.10	11.83±0.08	**11.12±0.08**
		12	21.46±0.10	21.66±0.05	21.53±0.13	21.55±0.21	28.20±0.91	21.71±0.19	19.01±0.29	13.43±0.06	13.52±0.07	**12.22±0.07**
		Avg.	20.79±0.12	16.93±0.03	16.75±0.10	16.87±0.14	21.46±0.77	16.80±0.11	15.63±0.25	12.27±0.08	11.88±0.08	**11.17±0.06**
	RMSE	3	33.49±0.18	23.04±0.11	23.08±0.15	23.05±0.09	28.17±1.00	23.28±0.13	22.22±0.17	19.23±0.10	20.30±0.10	**18.05±0.11**
		6	33.63±0.33	27.04±0.03	26.81±0.13	26.77±0.25	34.10±1.45	26.54±0.15	25.16±0.29	20.43±0.14	21.02±0.14	**19.28±0.12**
		12	35.08±0.10	35.30±0.14	35.29±0.13	34.59±0.28	46.24±1.59	34.64±0.15	30.63±0.45	22.65±0.10	23.30±0.26	**21.31±0.11**
		Avg.	34.07±0.19	27.61±0.06	27.39±0.12	27.36±0.17	34.66±0.83	27.45±0.15	25.42±0.26	20.54±0.10	21.21±0.14	**19.29±0.10**
	MAPE(%)	3	28.57±0.72	18.35±0.67	18.64±0.60	18.28±0.78	32.43±2.89	18.76±0.42	18.39±0.40	15.00±0.22	15.55±0.30	**14.10±0.20**
		6	28.31±0.60	20.68±0.64	20.55±0.37	20.81±0.55	36.83±3.39	21.27±0.40	20.05±0.33	15.52±0.20	15.50±0.30	**14.62±0.21**
		12	30.56±1.22	28.05±0.21	27.19±0.60	28.48±0.62	54.68±4.24	28.11±1.15	24.15±0.65	16.78±0.27	16.86±0.39	**16.73±0.19**
		Avg.	29.15±0.79	21.73±0.41	21.39±0.51	21.82±0.40	39.82±4.00	22.04±0.51	20.53±0.45	15.58±0.16	15.47±0.36	**14.71±0.18**
CA-Stream	MAE	3	24.43±0.83	17.79±0.18	17.79±0.20	18.04±0.09	20.05±0.05	23.62±0.44	17.69±0.34	14.99±0.18	15.26±0.12	**14.11±0.08**
		6	24.24±0.72	20.27±0.21	20.40±0.04	20.32±0.07	24.46±0.06	25.52±0.55	19.67±0.66	15.85±0.08	15.68±0.21	**14.83±0.07**
		12	25.77±0.89	26.78±0.54	27.04±0.15	26.60±0.23	34.26±0.07	30.81±0.85	25.03±1.48	17.24±0.11	17.54±0.23	**16.16±0.08**
		Avg.	24.81±0.78	21.21±0.21	21.04±0.12	21.00±0.08	25.45±0.05	26.06±0.45	20.23±0.71	15.84±0.09	16.09±0.24	**14.82±0.07**
	RMSE	3	37.30±1.02	28.00±0.22	27.87±0.20	27.93±0.15	31.56±0.07	34.66±0.66	27.34±0.34	24.37±0.33	23.36±0.25	**22.11±0.13**
		6	37.11±0.91	32.25±0.16	32.56±0.05	31.75±0.07	37.84±0.11	37.68±0.68	30.66±0.61	25.55±0.19	25.65±0.25	**23.32±0.12**
		12	39.07±1.15	41.75±0.51	42.30±0.16	41.19±0.39	50.81±0.12	46.74±0.42	37.63±2.43	28.19±0.11	26.89±0.11	**25.47±0.14**
		Avg.	37.83±0.99	32.84±0.43	33.09±0.12	32.64±0.21	39.31±0.12	39.05±0.83	31.26±1.26	25.69±0.14	24.73±0.19	**23.31±0.12**
	MAPE(%)	3	25.59±0.99	17.12±0.40	16.64±0.24	17.54±1.04	21.64±0.21	18.95±0.46	18.38±1.53	14.29±0.04	13.63±0.24	**12.90±0.18**
		6	25.42±0.83	19.33±0.22	18.88±0.19	19.73±0.71	26.52±0.23	20.49±0.33	19.47±0.88	14.88±0.09	14.30±0.20	**13.47±0.19**
		12	26.74±0.84	25.52±0.49	24.90±0.36	25.90±0.90	40.12±0.38	24.79±0.63	24.35±1.42	**16.23±0.08**	16.66±0.28	16.59±0.20
		Avg.	25.92±0.88	19.89±0.37	19.64±0.20	20.46±1.14	28.44±0.20	21.21±0.44	20.19±1.52	14.93±0.04	14.39±0.20	**13.55±0.17**
PEMS08	MAE	3	14.60±0.12	13.78±0.07	13.71±0.08	13.58±0.07	14.17±0.12	13.52±0.06	13.42±0.05	13.14±0.06	13.25±0.05	**12.31±0.04**
		6	15.10±0.14	14.26±0.04	14.18±0.05	14.06±0.06	14.67±0.14	13.90±0.04	13.81±0.07	13.55±0.05	13.48±0.05	**12.69±0.05**
		12	15.65±0.16	14.87±0.06	14.65±0.08	14.56±0.06	15.32±0.10	14.38±0.06	14.26±0.08	13.96±0.04	14.03±0.06	**13.11±0.05**
		Avg.	15.12±0.13	14.31±0.04	14.14±0.05	14.15±0.04	14.74±0.08	13.91±0.05	13.77±0.07	13.48±0.04	13.69±0.06	**12.71±0.04**
	RMSE	3	23.60±0.20	22.32±0.08	22.31±0.07	22.28±0.08	23.00±0.16	22.04±0.10	21.76±0.12	21.70±0.08	21.52±0.07	**20.21±0.08**
		6	24.20±0.22	23.03±0.08	22.83±0.10	22.70±0.09	23.51±0.19	22.64±0.12	22.29±0.12	22.15±0.10	22.36±0.08	**20.70±0.08**
		12	24.85±0.24	23.65±0.11	23.37±0.15	23.45±0.11	24.33±0.21	23.04±0.12	22.82±0.08	22.85±0.09	22.66±0.07	**21.20±0.09**
		Avg.	24.22±0.21	23.02±0.09	22.84±0.09	22.83±0.10	23.55±0.19	22.64±0.09	22.45±0.11	22.16±0.06	22.05±0.09	**20.70±0.08**
	MAPE(%)	3	9.80±0.18	9.22±0.09	9.06±0.11	9.08±0.08	9.49±0.14	8.59±0.06	8.76±0.08	8.60±0.05	8.75±0.08	**8.08±0.06**
		6	10.20±0.20	9.55±0.09	9.48±0.10	9.42±0.11	9.90±0.12	9.26±0.08	9.08±0.06	8.88±0.09	9.05±0.08	**8.37±0.07**
		12	10.65±0.22	9.92±0.08	9.84±0.12	9.80±0.10	10.37±0.18	9.63±0.09	9.44±0.11	9.29±0.08	9.15±0.10	**8.70±0.07**
		Avg.	10.22±0.19	9.57±0.09	9.48±0.08	9.40±0.12	9.99±0.14	9.26±0.08	9.15±0.08	8.91±0.05	9.01±0.07	**8.38±0.06**

achieves the best performance in nearly all settings across the three datasets and different forecasting horizons.

Results of existing forecasting methods. Conventional and continual traffic forecasting models show different behaviors under evolving traffic conditions. DCRNN performs reasonably on the standard PEMS08 benchmark but suffers clear degradation on PEMS-Stream and CA-Stream, indicating the limitation of predefined spatial dependencies under continuously changing traffic conditions. TrafficStream, STKEC, and PECPM achieve substantially better results by continuously adapting previously learned traffic knowledge. STRAP and EAC further improve forecasting accuracy through lightweight adaptation and pattern reuse, while STBP demonstrates strong performance by dynamically modeling spatial correlations in streaming environments. ConFormer benefits from explicitly incorporating accident information and achieves competitive results, especially under abnormal traffic conditions.

Results of ETDGN. ETDGN consistently outperforms all competing methods on PEMS-Stream, CA-Stream, and PEMS08. Compared with the strongest baseline for each metric, ETDGN reduces the average MAE by **5.98%**, **6.44%**, and **5.71%** on PEMS-Stream, CA-Stream, and PEMS08, respectively. Similar improvements are observed for RMSE and MAPE. The advantage remains evident at longer forecasting horizons; for example, ETDGN reduces the 12-step MAE on PEMS-Stream from 13.43 to 12.22. This suggests that simply adapting spatial correlations over time is insufficient when traffic dependencies are abruptly reorganized by events. By selectively activating event-conditioned graph updates and explicitly propagating event impacts, ETDGN better captures temporary spatial dependency shifts. Meanwhile, separating event-induced deviations from routine traffic evolution prevents short-term disturbances from contaminating stable temporal patterns, leading to more consistent long-horizon forecasting.

4.2 Ablation Study & Parameter Sensitivity Analysis

Ablation settings. To validate the key designs of ETDGN, we construct four variants targeting the two challenges discussed in Section 1. *Time Update* replaces event-triggered graph adaptation with graph reconstruction at every time step, directly evaluating whether graph changes should be driven by events rather than temporal progression. *w/o E-Graph* removes the event-conditioned graph while retaining event detection. *w/o Prop.* disables graph-based event propagation by directly using the detected event state, and *w/o Dis.* removes trend–season–event disentanglement and models the mixed traffic signal directly.

Ablation findings. As shown in Table 2, all variants lead to consistent degradation. The *Time Update* variant causes the largest drop, increasing MAE by 8.2% and 7.6% on PEMS-Stream and CA-Stream, respectively. This result directly supports our first motivation: continuously updating spatial dependencies according to time does not necessarily reflect the actual cause of graph changes. Removing the event-conditioned graph further confirms that identifying abnormal states alone is insufficient without reorganizing their spatial dependencies. The *w/o Prop.* variant also shows clear degradation, demonstrating that event effects should not be treated as isolated node-level abnormalities but need to be propagated over affected traffic relations. Finally, the decline of *w/o Dis.* confirms the benefit of separating transient event deviations from routine traffic dynamics.

Table 2: Ablation study. PS, CA, and P8 denote PEMS-Stream, CA-Stream, and PEMS08, respectively.

Variant	PS		CA		P8	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Time Update	12.09	20.62	15.94	24.95	13.22	21.46
w/o E-Graph	11.82	20.15	15.58	24.44	13.08	21.22
w/o Prop.	11.70	19.96	15.49	24.30	13.02	21.16
w/o Dis.	11.53	19.78	15.25	23.95	12.95	21.04
ETDGN	**11.17**	**19.29**	**14.82**	**23.31**	**12.71**	**20.70**

Parameter sensitivity analysis. We further investigate two challenge-specific hyperparameters: the event detection threshold $\delta = \kappa\sigma_r$ and the propagation depth L. As shown in Table 3, ETDGN remains stable over moderate parameter ranges. A small δ incorrectly activates event-aware graph updates for routine fluctuations, whereas an overly large value overlooks weak but meaningful disturbances. The best performance is obtained around $\delta = 1.5\sigma_r$. Increasing L from 1 to 3 consistently improves performance, indicating that local one-hop propagation is insufficient to capture the spatial extent of event impacts. Further increasing the depth slightly degrades accuracy, likely because excessive propagation diffuses localized event information to irrelevant regions. Therefore, we use $\delta = 1.5\sigma_r$ and $L = 3$ by default.

Table 3: Sensitivity to event threshold δ and propagation depth L. Values are MAE.

δ/σ_r	0.5	1.0	**1.5**	2.0	2.5
PS	11.58	11.31	**11.17**	11.25	11.41
CA	15.21	14.98	**14.82**	14.91	15.08
P8	12.99	12.80	**12.71**	12.77	12.90
L	1	2	**3**	4	5
PS	11.46	11.29	**11.17**	11.22	11.33
CA	15.14	14.95	**14.82**	14.87	14.99
P8	12.93	12.79	**12.71**	12.74	12.83

4.3 Efficiency Study

An effective event-aware forecasting model should improve adaptability without introducing excessive computational overhead. We therefore compare ETDGN with representative baselines under the same hardware and batch-size settings.

As shown in Table 4, ETDGN achieves the best overall efficiency among all compared methods. Compared with STBP, the strongest efficiency-oriented baseline, ETDGN reduces MACs by 4.8%, training time by 4.9%, median inference latency by 5.0%, and peak GPU memory by 4.2%, while improving throughput from 0.65k to 0.69k samples per second. The advantage becomes more pronounced compared with ConFormer, indicating that explicit event modeling does not necessarily introduce substantial computational overhead.

Table 4: Efficiency comparison on CA-Stream. Lower is better except throughput. **Bold**: best, underline: second best.

Model	Param. (M)	MACs (G)	Train (s/ep.)	Med. (ms)	P95 (ms)	Thru.↑ (k/s)	Mem. (MB)
DCRNN	0.37	9.84	113.2	11.10	14.40	0.43	1130
TrafficStream	0.41	8.95	101.6	10.40	13.50	0.47	1064
iTransformer	2.08	18.72	145.8	15.20	19.70	0.33	1576
EAC	0.43	8.51	92.7	9.78	12.84	0.50	1008
STBP	0.36	7.12	73.4	7.62	9.93	0.65	848
ConFormer	0.38	10.96	100.1	9.50	12.29	0.52	1032
ETDGN	**0.35**	**6.78**	**69.8**	**7.24**	**9.41**	**0.69**	**812**

The efficiency gain mainly comes from the event-triggered computation paradigm. Conventional dynamic graph models repeatedly reconstruct spatial dependencies as traffic observations evolve, whereas ETDGN activates the event-conditioned graph and propagation process only when event-related evidence is detected. During routine periods, prediction mainly relies on the lightweight routine branch, thereby avoiding unnecessary graph reconstruction and event propagation. Consequently, event-triggered adaptation improves not only forecasting accuracy but also computational efficiency, enabling ETDGN to achieve a favorable accuracy–efficiency trade-off.

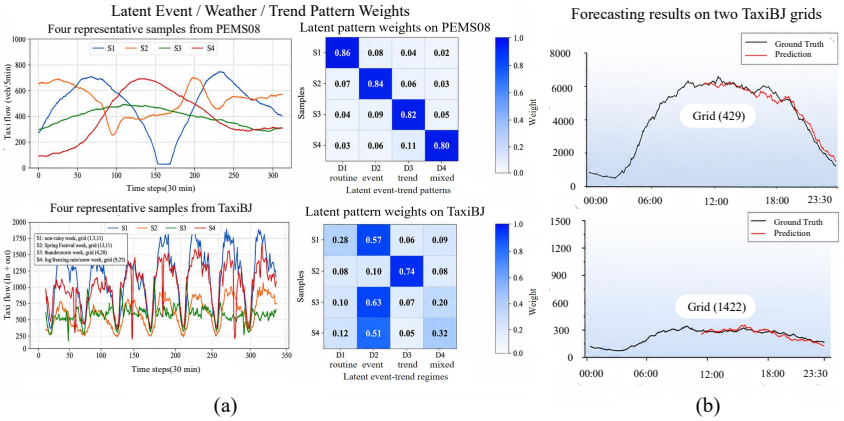

Figure 3: Case study of ETDGN. (a) Latent event–trend pattern analysis on PEMS08 and TaxiBJ. (b) Forecasting visualization on two representative TaxiBJ grids.

4.4 Case Study

To provide an intuitive analysis of how ETDGN responds to explicit external events, we further conduct a case study on TaxiBJ (Zhang et al., 2017). Detailed dataset information is provided in Appendix A.

Event-pattern distinction. As shown in Figure 3(a), ETDGN exhibits clearly differentiated latent responses under different traffic contexts. On PEMS08, where explicit external variables are unavailable, representative samples are mainly assigned to distinct latent routine, event, trend, or mixed patterns, indicating that the model can infer heterogeneous traffic regimes from traffic observations alone. More importantly, on TaxiBJ, the learned weights are strongly aligned with the observed external conditions. The non-rainy sample mainly activates the weather-related latent pattern only weakly, whereas the Spring Festival sample places substantially larger weight on the holiday-related pattern. Similarly, the thunderstorm sample and the sample containing fog, freezing rain, and snow exhibit strong responses to weather-dominated or mixed latent patterns. These results indicate that ETDGN does not simply treat all abrupt variations as generic temporal changes, but can associate different traffic deviations with their underlying contextual conditions.

Forecasting under event-affected conditions. Figure 3(b) further presents representative predictions at two spatially distinct grids. Despite their substantially different traffic magnitudes and temporal profiles, ETDGN closely follows the ground-truth evolution after the forecasting point. In the high-flow region, the model captures both the local fluctuations around the afternoon peak and the subsequent evening decline, while in the low-flow region it preserves the smoother traffic evolution without introducing excessive event corrections. This behavior is consistent with the event-triggered design: the model strengthens event-aware adaptation only when abnormal contextual or traffic evidence is present, while relying primarily on routine dynamics during stable periods.

Overall, the case study provides qualitative evidence for the two key designs of ETDGN. The latent pattern responses demonstrate that graph adaptation can be associated with explicit or implicitly inferred event evidence, while the forecasting examples show that event-aware corrections can be introduced without destroying stable traffic evolution. This supports the proposed event-triggering–graph-reorganization–impact-propagation modeling paradigm.

5 Conclusion

In this work, we propose ETDGN, an event-triggered dynamic graph framework for traffic forecasting. ETDGN selectively adapts spatial dependencies according to explicit or inferred event evidence, models event-impact propagation, and separates transient disturbances from routine traffic evolution through gated residual forecasting. Experiments on multiple real-world datasets demonstrate consistent improvements in forecasting accuracy and efficiency. Future work will explore richer event modalities and finer-grained supervision to further improve generalization and interpretability.

AI USE STATEMENT

In this work, we did not use generative AI tools for tasks with required disclosure. We have not used generative AI tools for research-related tasks requiring disclosure, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for language polishing, grammatical correction, formatting assistance, and minor stylistic improvements of the manuscript. We have reviewed all AI-assisted work and manually verified that these edits did not alter the technical content, experimental results, claims, or conclusions. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide an anonymized implementation of ETDGN at `https://anonymous.4open.science/r/ETDGN-ICLR-2027-C367`. Dataset descriptions, preprocessing, and evaluation settings are provided in Appendix A; additional experimental analyses are reported in Appendix B; the inference procedure is summarized in Appendix C; and the complete optimization objectives and implementation details are provided in Appendix D. The released code includes the configurations and scripts required to reproduce the main experiments.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

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, pp. 11463–11471, 2025.

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.

Jinwen Chen, Hao Miao, Chenxi Liu, Yan Zhao, and Kai Zheng. Visionst: Coordinating cross-modal traffic prediction with interactive geo-image encoding. In *Proceedings of the ACM Web Conference 2026*, pp. 7307–7317, 2026.

Wei Chen and Yuxuan Liang. Expand and compress: Exploring tuning principles for continual spatio-temporal graph forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 81631–81656, 2025.

Xu Chen, Junshan Wang, and Kunqing Xie. Trafficstream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 3620–3626. International Joint Conferences on Artificial Intelligence Organization, 8 2021.

Zheng Dong, Renhe Jiang, Haotian Gao, Hangchen Liu, Jinliang Deng, Qingsong Wen, and Xuan Song. Heterogeneity-informed meta-parameter learning for spatiotemporal time series forecasting. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 631–641, 2024.

Jin Fan, Wenchao Weng, Qikai Chen, Huifeng Wu, and Jia Wu. Pdg2seq: Periodic dynamic graph to sequence model for traffic flow prediction. *Neural Networks*, 183:106941, 2025.

Jie Hu, Taichuan Zheng, Lilan Peng, Fei Teng, Shengdong Du, and Tianrui Li. Lightst: A simplifying spatio-temporal graph neural network for traffic flow forecasting. *IEEE Transactions on Big Data*, 11(5):2517–2528, 2025. doi: 10.1109/TBDATA.2025.3544131.

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, pp. 4365–4373, 2023.

Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Shao, Jincai Huang, Junbo Zhang, and Yu Zheng. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5388–5408, 2024.

Weiyang Kong, Ziyu Guo, and Yubao Liu. Spatio-temporal pivotal graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 8627–8635, 2024.

Hyunwook Lee and Sungahn Ko. Testam: A time-enhanced spatio-temporal attention model with mixture of experts. In *International Conference on Learning Representations*, volume 2024, pp. 3011–3029, 2024.

Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data*, 17(1):1–21, 2023.

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

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*, 2018. URL https://mlanthology.org/iclr/2018/li2018iclr-diffusion/.

Aoyu Liu and Yaying Zhang. A general spatio-temporal backbone with scalable contextual pattern bank for urban continual forecasting. In *International Conference on Learning Representations*, volume 2026, pp. 72080–72103, 2026.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Staeformer: spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. *arXiv preprint arXiv:2308.10425*, 2023a.

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. *Advances in neural information processing systems*, 36:75354–75371, 2023b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Minlan Shao, Zijian Zhang, Yili Wang, Yiwei Dai, Xu Shen, and Xin Wang. Hyperd: Hybrid periodicity decoupling framework for traffic forecasting. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence*, 2026.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 4454–4458, 2022.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):291–305, 2024.

Ying Shi, Wentian Cui, Ruiqin Wang, Jungang Lou, and Qing Shen. Stadgcn: spatial–temporal adaptive dynamic graph convolutional network for traffic flow prediction. *Neural Computing and Applications*, pp. 1–14, 2025.

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, pp. 914–921, 2020.

Binwu Wang, Yudong Zhang, Jiahao Shi, Pengkun Wang, Xu Wang, Lei Bai, and Yang Wang. Knowledge expansion and consolidation for continual traffic prediction with expanding graphs. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):7190–7201, 2023a.

Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2223–2232, 2023b.

Feng Wang, Tianxiang Chen, Shuyue Wei, Qian Chu, Yi Zhang, Yifan Sun, and Zhiming Zheng. Unlocking dynamic inter-client spatial dependencies: A federated spatio-temporal graph learning method for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1123–1131, 2026a.

Hongjun Wang, Jiawei Yong, Jiawei Wang, Shintaro Fukushima, and Renhe Jiang. Towards resilient transportation: A conditional transformer for accident-informed traffic forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026b. doi: 10.1145/3770854.3780312.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019.

Mingyuan Xia, Chunxu Zhang, Zijian Zhang, Hao Miao, Qidong Liu, Yuanshao Zhu, and Bo Yang. Timeemb: A lightweight static-dynamic disentanglement framework for time series forecasting. *Advances in Neural Information Processing Systems*, 38:76227–76258, 2026.

Yutong Xia, Yuxuan Liang, Haomin Wen, Xu Liu, Kun Wang, Zhengyang Zhou, and Roger Zimmermann. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. *Advances in Neural Information Processing Systems*, 36:37068–37088, 2023.

Bairan Xiang, Chenguang Zhao, and Huan Yu. Micro-macro coupled koopman modeling on graph for traffic flow prediction. In *International Conference on Learning Representations*, volume 2026, pp. 7365–7380, 2026.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 3634–3640. International Joint Conferences on Artificial Intelligence Organization, 2018.

Chengqing Yu, Fei Wang, Chuanguang Yang, Zezhi Shao, Tao Sun, Tangwen Qian, Wei Wei, Zhulin An, and Yongjun Xu. Merlin: Multi-view representation learning for robust multivariate time series forecasting with unfixed missing rates. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3633–3644, 2025.

Haoyu Zhang, Hao Miao, Xinke Jiang, Yuchen Fang, and Yifan Zhang. Strap: Spatio-temporal pattern retrieval for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 38:118006–118041, 2026a.

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

Ping Zhang, Jiayu Leng, Liang Yang, Anchen Li, Xiaochun Cao, and Riting Xia. Multi-source information driven spatio-temporal hypergraph learning for traffic forecasting. In *Proceedings of the ACM Web Conference 2026*, pp. 580–591, 2026b.

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, pp. 1093–1101, 2025.

Kai Zhao, Chenjuan Guo, Yunyao Cheng, Peng Han, Miao Zhang, and Bin Yang. Multiple time series forecasting with dynamic graph modeling. *Proceedings of the VLDB Endowment*, 17(4): 753–765, 2023.

Yiqing Zou, Hanning Yuan, Qianyu Yang, Ziqiang Yuan, Shuliang Wang, and Sijie Ruan. Meta dynamic graph for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16584–16592, 2026.

A Dataset Details

Table 5: External contextual variables used in the TaxiBJ case study.

Context	Availability	Representation
Temperature	Yes	Continuous value
Wind speed	Yes	Continuous value
Weather	Yes	17-category one-hot vector
Holiday	Yes	Binary/calendar indicator

We evaluate ETDGN on three real-world traffic forecasting datasets, including PEMS-Stream (Chen et al., 2001), CA-Stream (Liu et al., 2023b), and PEMS08 (Song et al., 2020). PEMS-Stream and CA-Stream are streaming traffic datasets constructed from the California Performance Measurement System (PeMS), with a sampling interval of 5 minutes. PEMS-Stream spans multiple years and contains a gradually evolving sensor network, while CA-Stream exhibits more rapid structural variation and large-scale node expansion. PEMS08 is a widely used traffic forecasting benchmark consisting of 170 traffic sensors with 5-minute sampling intervals. Following common practice, all datasets are split chronologically into training, validation, and test sets with a ratio of 6:2:2. We use the previous 12 time steps to predict the next 12 steps, and report the forecasting performance at horizons 3, 6, and 12, corresponding to 15, 30, and 60 minutes, respectively.

As shown in Table 5, TaxiBJ is a citywide crowd-flow dataset constructed from taxi trajectories in Beijing (Zhang et al., 2017). We use it as an auxiliary case-study dataset to investigate the behavior of ETDGN when explicit event-related contextual variables are available. The traffic observations are organized over a 32×32 spatial grid, where each grid contains two channels representing taxi inflow and outflow. The sampling interval is 30 minutes.

The dataset contains traffic observations from multiple years. In our implementation, the traffic tensors have shapes $4888\times2\times32\times32$, $4780\times2\times32\times32$, $5596\times2\times32\times32$, and $7220\times2\times32\times32$ for the four yearly subsets, respectively. Each spatial grid is treated as a traffic node when constructing the input representation.

In addition to traffic flow, TaxiBJ provides several explicit contextual variables that can be used by the event-triggering module. These include temperature, wind speed, weather conditions, and holiday indicators. Temperature and wind speed are continuous variables, while weather is encoded as a 17-dimensional one-hot vector. Holiday information is obtained from the accompanying holiday calendar. These variables are aligned with the traffic timestamps and incorporated into the explicit event evidence $\mathbf{Z}_t$ in ETDGN.

The 17 weather categories include Sunny, Cloudy, Overcast, Rainy, Sprinkle, Moderate Rain, Heavy Rain, Rainstorm, Thunderstorm, Freezing Rain, Snowy, Light Snow, Moderate Snow, Heavy Snow, Foggy, Sandstorm, and Dusty. These categories cover both routine weather conditions and severe weather events, making TaxiBJ particularly suitable for analyzing whether the event trigger can distinguish normal traffic evolution from externally induced disturbances.

For the latent-pattern case study, we select representative samples from different contextual conditions, including a non-rainy week, the Spring Festival period, a thunderstorm period, and a period containing fog, freezing rain, and snow.

B Additional Experimental Results

In this section, we provide additional experiments to further evaluate the two key designs of ETDGN. Specifically, we examine whether the event trigger is aligned with observable external events, visualize how event influence propagates over the traffic graph, evaluate forecasting performance over different incremental periods, and investigate scalability under increasing graph sizes.

Table 6: Event-trigger alignment results on TaxiBJ. Higher values are better for Precision, Recall, and F1, while lower values are better for Normal Update.

Method	Precision (%)	Recall (%)	F1 (%)	Normal Update (%)↓
Time Update	21.6	**100.0**	35.5	100.0
Residual Only	69.8	82.5	75.6	12.8
Context Only	84.6	73.1	78.4	**2.8**
ETDGN	**88.4**	89.7	**89.0**	3.9

B.1 Event Trigger Effectiveness

The first challenge addressed by ETDGN is whether dynamic graph adaptation can be associated with event-related changes rather than temporal progression alone. To provide a direct evaluation, we conduct an event-alignment analysis on TaxiBJ, where explicit weather and holiday contexts are available. Severe-weather and holiday periods are treated as observable event windows. Since these contexts cannot cover all latent traffic disturbances, this experiment evaluates alignment with observable events rather than defining a complete event-detection benchmark.

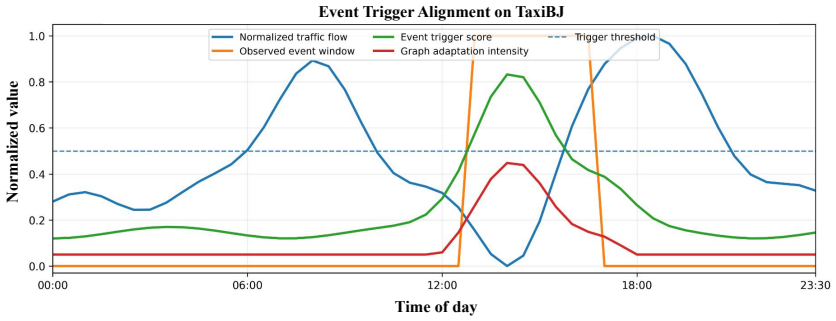

Figure 4: Alignment between traffic evolution, observable event contexts, event-trigger scores, and graph-adaptation intensity on TaxiBJ. The trigger score remains low during routine periods and rises around the event-affected interval, followed by stronger graph adaptation.

As shown in Table 6, the time-driven strategy trivially achieves full event recall because the graph is updated at every time step. However, it also produces graph updates throughout normal periods, resulting in poor event precision and a Normal Update rate of 100%. Using only traffic residuals substantially reduces unnecessary updates but remains sensitive to routine traffic fluctuations. In contrast, relying only on explicit contexts achieves a low Normal Update rate but misses traffic abnormalities that are not fully explained by observed external variables.

ETDGN obtains the best F1 score by jointly exploiting explicit contextual information, implicit traffic abnormalities, and spatially coherent responses. Meanwhile, event-specific graph adaptation is suppressed during 96.1% of normal periods. These results demonstrate that ETDGN does not simply reconstruct the graph as time advances, but selectively activates graph adaptation when event-related evidence is observed.

Figure 4 provides a representative example. The event trigger remains inactive during routine traffic evolution and increases rapidly when the observed event occurs. The graph-adaptation intensity follows the trigger response rather than changing continuously over time. This observation further supports the event-triggered perspective underlying Challenge 1.

B.2 Spatio-temporal Event Impact Evolution

To further examine Challenge 2, we analyze how the impact of a detected event evolves after activation. Rather than visualizing the complete road graph, we align roads according to their relative position to the event source and visualize the event propagation state $P_{t,i}$ over time. The event

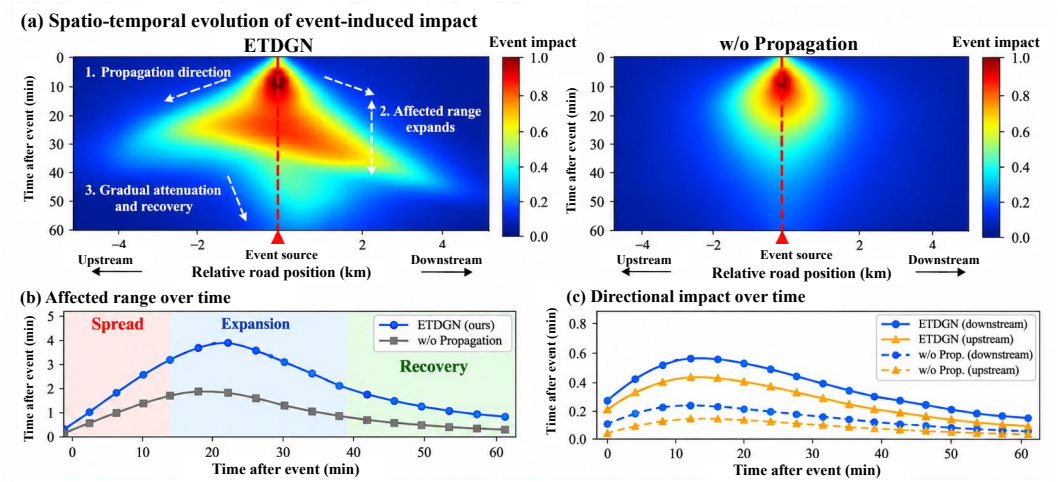

Figure 5: Spatio-temporal evolution of event-induced traffic impact. The event source is located at position zero. The heatmap shows the propagation state $P_{t,i}$ over relative road position and time, while the curves summarize the affected range and directional impact.

source is set to position zero, with upstream and downstream roads placed on the two sides. This representation allows us to directly inspect the propagation direction, affected range, and temporal evolution of event-induced disturbances.

As shown in Figure 5, the impact is initially concentrated around the event source and gradually spreads to surrounding roads as time progresses. The affected range first expands and then decreases as the disturbance attenuates, revealing a complete emergence–propagation–recovery process. The asymmetric responses on the upstream and downstream sides further indicate that event influence is not treated as an isotropic spatial effect.

For comparison, removing the propagation module substantially weakens this coherent spatio-temporal pattern. Its event responses remain more localized and fail to exhibit a clear expansion and directional propagation process. This result complements the quantitative *w/o Prop.* ablation in the main text and demonstrates that ETDGN explicitly models how event-induced disturbances propagate across the traffic network, rather than only learning generic spatial correlations. Therefore, the proposed propagation mechanism provides a direct solution to Challenge 2 by characterizing the direction, affected range, and temporal evolution of event-induced traffic impacts.

B.3 Event Impact Evolution Case Study

To provide an intuitive interpretation of the learned event-aware representation, we conduct case studies on representative traffic events and visualize their evolution from event onset to recovery. Figure 6 presents two representative cases with different spatial impact patterns: a localized accident event and a wider-area weather event. Each case contains three complementary views, including latent event-state evolution, spatial impact evolution, and traffic responses at representative locations.

In the latent space shown in Figure 6(a), routine traffic states are separated from event-related states after the event occurs. More importantly, the event states exhibit a continuous transition across the onset, propagation, and recovery stages rather than forming an isolated response. This indicates that the learned representation preserves the temporal evolution of event-induced disturbances and distinguishes transient event dynamics from routine traffic patterns.

Figure 6(b) further visualizes the spatial evolution of event impact. For the accident case, the impact is initially concentrated around the event source and gradually extends to downstream roads and connected alternative routes. The affected region then shrinks during recovery. In contrast, the weather case exhibits a broader spatial response, with multiple connected roads becoming affected as the event develops. These different propagation patterns suggest that ETDGN does not impose a fixed propagation range, but adapts the spatial extent of event influence according to the observed event conditions.

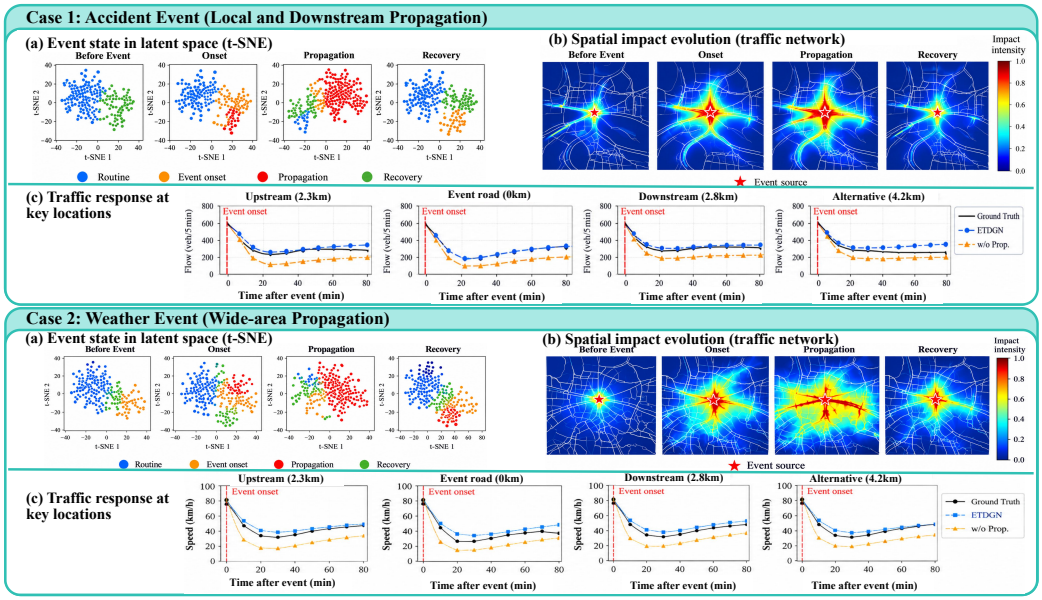

Figure 6: Event impact evolution under representative accident and weather events. Each case shows (a) latent event-state evolution, (b) spatial impact evolution over the traffic network, and (c) traffic responses at representative locations from event onset to recovery.

The traffic responses in Figure 6(c) provide a direct connection between the learned event representation and observable traffic dynamics. Following the event onset, the traffic flow at the event source changes sharply, followed by delayed responses at upstream, downstream, and alternative-route locations. ETDGN consistently follows the temporal variation of the ground-truth traffic response more closely than the model without the propagation module, particularly after the initial disturbance when the impact has spread beyond the event source. This difference is more evident for the weather case, where the disturbance affects a wider spatial region.

Overall, these case studies reveal a complete event-induced evolution process: *emergence* $\rightarrow$ *propagation* $\rightarrow$ *attenuation* $\rightarrow$ *recovery*. The results provide qualitative evidence that ETDGN explicitly captures not only whether an event occurs, but also where its impact propagates, how far the disturbance spreads, and how its effect evolves over time. This complements the quantitative ablation results and directly supports the motivation of Challenge 2.

Table 7: Period-wise MAE on PEMS-Stream and CA-Stream. Lower values are better.

	PEMS-Stream							CA-Stream			
Method	2011	2012	2013	2014	2015	2016	2017	Jan.	Feb.	Mar.	Apr.
TrafficStream	18.50	17.80	17.20	16.80	16.50	15.90	16.00	22.40	21.50	20.70	19.76
EAC	16.80	16.30	15.90	15.50	15.40	14.90	14.90	21.40	20.60	19.80	19.00
STBP	13.20	12.80	12.40	12.10	12.00	11.80	11.90	16.50	15.90	15.40	15.28
ConFormer	12.90	12.40	12.00	11.60	11.70	11.30	11.30	16.80	16.20	15.90	15.70
ETDGN	**12.10**	**11.70**	**11.20**	**10.80**	**10.90**	**10.50**	**11.00**	**15.50**	**14.90**	**14.50**	**14.38**

B.4 Period-wise Continual Forecasting Performance

The main results report the average forecasting performance over the complete streaming datasets. To further examine whether the performance gain is preserved throughout different incremental stages, we report period-wise results on PEMS-Stream and CA-Stream.

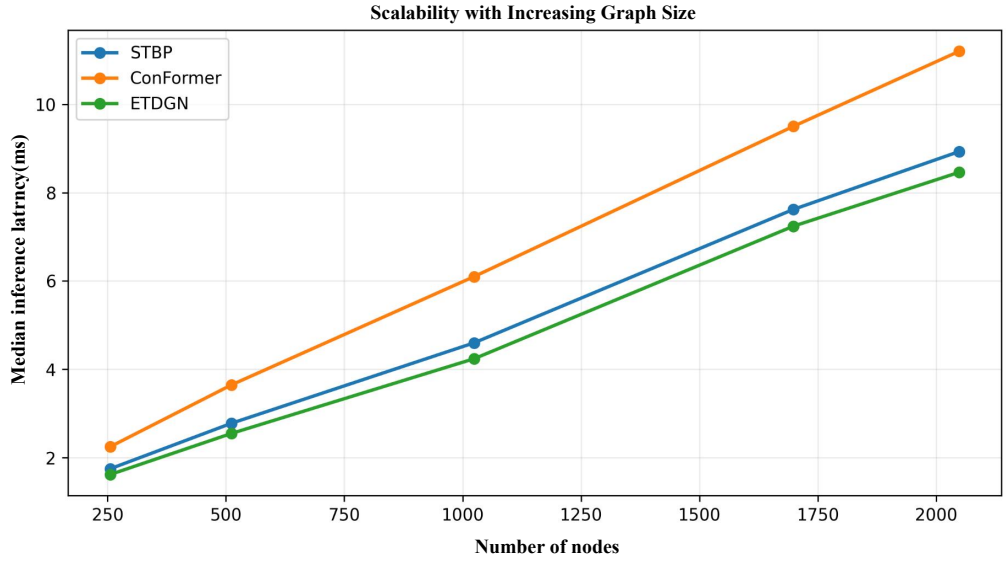

Figure 9: Inference latency with increasing graph sizes.

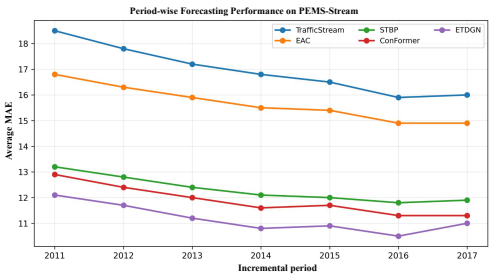

Figure 7: Period-wise MAE on PEMS-Stream from 2011 to 2017.

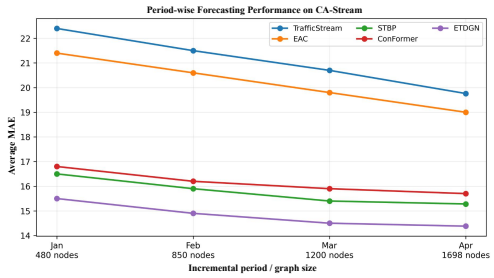

Figure 8: Period-wise MAE on CA-Stream over incremental graph-expansion stages.

As shown in Table 7 and Figures 7–8, ETDGN maintains the best forecasting performance throughout all incremental stages. On PEMS-Stream, the advantage is consistently preserved from 2011 to 2017, indicating that the improvement is not dominated by a small subset of periods. On CA-Stream, ETDGN also remains superior as the traffic graph gradually expands and the underlying spatial dependencies become increasingly complex.

These results indicate that event-triggered graph adaptation remains effective under both temporal distribution evolution and structural graph expansion. In particular, ETDGN preserves routine knowledge during stable periods while introducing event-specific structural changes only when necessary, which reduces unnecessary interference with previously learned traffic patterns.

B.5 Scalability under Graph Expansion

The efficiency experiment in the main text evaluates computational cost on CA-Stream. We further investigate scalability by progressively increasing the number of graph nodes while keeping the batch size and implementation environment unchanged.

As shown in Table 8, the computational cost of all methods increases as the graph becomes larger. Nevertheless, ETDGN consistently achieves lower inference latency and memory consumption than the compared strong baselines across different graph sizes. At 1,698 nodes, ETDGN requires only 7.24 ms inference latency and 812 MB peak memory, which is consistent with the CA-Stream efficiency results reported in the main text.

More importantly, the efficiency advantage is maintained as the graph expands. This suggests that explicit event modeling does not introduce persistent graph-level computation. Instead, the event-conditioned branch is selectively activated only when event-related evidence becomes sufficiently strong, while routine periods mainly rely on the lightweight routine branch.

Figure 9 further illustrates the scalability trend. Although inference latency gradually increases with the number of nodes, ETDGN exhibits a slower growth rate than ConFormer and remains consistently more efficient than STBP. The advantage becomes increasingly important for large traffic graphs, where repeatedly reconstructing event-specific dependencies would otherwise introduce substantial computation.

The favorable scalability mainly results from the selective event-triggered computation paradigm. ETDGN does not continuously activate event-conditioned graph construction and propagation. During routine periods, forecasting primarily relies on stable routine dependencies, whereas additional event-specific computation is introduced only when necessary. Consequently, the proposed event-aware modeling capability can be maintained without sacrificing scalability under expanding traffic networks.

Table 8: Scalability comparison under increasing graph sizes. Lower values are better.

Nodes	STBP		ConFormer		ETDGN	
	Lat. (ms)	Mem. (MB)	Lat. (ms)	Mem. (MB)	Lat. (ms)	Mem. (MB)
256	1.75	340	2.25	390	**1.62**	**325**
512	2.78	420	3.65	500	**2.55**	**400**
1024	4.60	610	6.10	720	**4.24**	**575**
1698	7.62	848	9.50	1032	**7.24**	**812**
2048	8.93	990	11.20	1220	**8.46**	**945**

C ALGORITHM

Algorithm 1 summarizes the complete inference procedure of ETDGN. Different from conventional dynamic graph forecasting models that update graph structures at every time step, ETDGN performs event-conditioned graph adaptation only when abnormal traffic variations are detected. The overall procedure consists of four stages: routine representation construction, event state inference, event-aware propagation, and gated residual forecasting.

In the first stage, ETDGN decomposes the observed traffic sequence into routine trend, periodic component, and event-related residual. These components are jointly encoded with temporal and optional exogenous contexts to obtain the initial traffic representation. Based on this representation, a routine dynamic graph is constructed to capture regular traffic dependencies. The routine branch provides a stable reference for identifying deviations caused by abnormal events.

In the second stage, ETDGN estimates event evidence from three complementary sources: available external contexts, traffic residuals, and spatially coherent abnormal patterns. These signals are aggregated into node-level event scores and further accumulated through the event persistence mechanism. This design prevents the model from interpreting normal temporal variations as structural changes and enables selective graph adaptation only under event-related disturbances.

In the third stage, when the accumulated event state exceeds the activation threshold, ETDGN constructs an event-conditioned graph and propagates event influence through the learned spatial dependencies. The propagation module explicitly models how an event spreads across connected road segments, including its affected range, directional influence, and temporal attenuation process. Historical event patterns stored in the memory module are additionally retrieved to provide prior knowledge for similar disturbance scenarios.

Finally, ETDGN separates routine forecasting and event-induced correction. The routine branch predicts stable traffic evolution, while the event branch estimates only the additional deviation caused by the detected event. A dynamic gate controls the contribution of the event correction according to the current event intensity, allowing the model to rapidly respond to abnormal situations without disturbing normal traffic prediction.

Algorithm 1 Inference Procedure of ETDGN

Require: Historical traffic observations $\mathbf{X}_{t-T+1:t}$, optional exogenous contexts $\mathbf{Z}_{t-T+1:t}$, temporal contexts $\boldsymbol{\tau}_{t-T+1:t}$, base graph $\mathbf{A}^0$, propagation depth L, event activation threshold τ_e.

Ensure: Future traffic prediction $\widehat{\mathbf{Y}}_{t+1:t+H}$.

Decompose the traffic sequence:

$$\mathbf{X}^{tr} = MA(\mathbf{X}), \quad \mathbf{X}^{se} = F_{se}(\mathbf{X} - \mathbf{X}^{tr}, \boldsymbol{\tau}), \quad \mathbf{X}^{ev} = \mathbf{X} - \mathbf{X}^{tr} - \mathbf{X}^{se}.$$

Encode the current traffic representation $\mathbf{H}_t^0$ from $\mathbf{X}_t$, $\mathbf{X}_t^{tr}$, $\mathbf{X}_t^{se}$, $\mathbf{X}_t^{ev}$, $\boldsymbol{\tau}_t$, and $\mathbf{Z}_t$.

Construct the routine dynamic graph $\mathbf{A}_t^r$ from $\mathbf{H}_t^0$ and obtain the routine representation $\mathbf{H}_t^r$.

Obtain the routine traffic estimate $\widetilde{\mathbf{x}}_t^r \leftarrow Head_r(\mathbf{H}_t^r)$ and residual $\mathbf{r}_t \leftarrow |\mathbf{x}_t - \widetilde{\mathbf{x}}_t^r|$.

Compute explicit event evidence $\mathbf{s}_t^z$, traffic abnormality evidence $\mathbf{s}_t^x$, and spatial propagation evidence $\mathbf{s}_t^p$.

Fuse the evidence into node-level trigger scores $\mathbf{g}_t$.

Compute persistence coefficients $\boldsymbol{\rho}_t$ and update event memory:

$$\mathbf{m}_t \leftarrow \mathbf{g}_t + (1 - \mathbf{g}_t) \odot \boldsymbol{\rho}_t \odot \mathbf{m}_{t-1}.$$

Obtain the event activation state $a_t \leftarrow \mathbb{I}(\max_i m_{t,i} > \tau_e)$.

Decode the routine forecast $\widehat{\mathbf{Y}}^r \leftarrow \mathcal{D}_r(\mathbf{H}_t^r)$.

if $a_t = 1$ **then**

  Construct the event-conditioned graph $\mathbf{A}_t^e$ from $\mathbf{H}_t^r$, $\mathbf{m}_t$, $\mathbf{g}_t$, and $\mathbf{A}^0$.

  Compute the graph adaptation coefficient

$$\lambda_t \leftarrow \sigma(MLP_\lambda[Mean(\mathbf{m}_t), Max(\mathbf{m}_t), \boldsymbol{\tau}_t])\,.$$

  Fuse routine, event-conditioned, and base graphs:

$$\mathbf{A}_t \leftarrow (1 - \lambda_t)\mathbf{A}_t^r + \lambda_t \mathbf{A}_t^e + \eta \mathbf{A}^0.$$

  Initialize event propagation $\mathbf{P}_t^0 \leftarrow \mathbf{m}_t$.

  for $l = 0, \ldots, L-1$ **do**

    Propagate the event impact:

$$\mathbf{P}_t^{l+1} \leftarrow \alpha \mathbf{A}_t^e \mathbf{P}_t^l + (1 - \alpha)\mathbf{P}_t^l.$$

  end for

  Set $\mathbf{P}_t \leftarrow \mathbf{P}_t^L$ and obtain the event-aware representation

$$\mathbf{H}_t^e \leftarrow F_e(\mathbf{H}_t^r, \mathbf{A}_t, \mathbf{P}_t).$$

  Query the historical event pattern memory:

$$\mathbf{q}_t \leftarrow Pool[\mathbf{H}_t^r, \mathbf{H}_t^e, \mathbf{m}_t, \boldsymbol{\tau}_t],$$

$$\boldsymbol{\alpha}_t^{mem} \leftarrow Softmax\left(\frac{\mathbf{q}_t(\mathbf{K}^{mem})^\top}{\sqrt{d}}\right), \quad \mathbf{C}_t^{mem} \leftarrow \boldsymbol{\alpha}_t^{mem}\mathbf{V}^{mem}.$$

  Decode the event-induced residual:

$$\Delta\widehat{\mathbf{Y}}^e \leftarrow \mathcal{D}_e(\mathbf{H}_t^e, \mathbf{P}_t, \mathbf{C}_t^{mem}).$$

  Compute the residual correction gate:

$$\gamma_t \leftarrow \sigma(MLP_\gamma[Mean(\mathbf{m}_t), Max(\mathbf{m}_t), \lambda_t, \mathbf{C}_t^{mem}, \boldsymbol{\tau}_t])\,.$$

  Generate the final forecast:

$$\widehat{\mathbf{Y}}_{t+1:t+H} \leftarrow \widehat{\mathbf{Y}}^r + \gamma_t \odot \Delta\widehat{\mathbf{Y}}^e.$$

else

  Skip event-conditioned graph construction, event propagation, and memory retrieval.

  Set $\lambda_t \leftarrow 0$, $\gamma_t \leftarrow 0$, and

$$\widehat{\mathbf{Y}}_{t+1:t+H} \leftarrow \widehat{\mathbf{Y}}^r.$$

end if

return $\widehat{\mathbf{Y}}_{t+1:t+H}$.

D Optimization Details

ETDGN is trained to preserve accurate routine forecasting while learning event-specific graph adaptation and residual correction. Since the event trigger relies on deviations from routine traffic evolution, directly optimizing all components from random initialization may produce unreliable event evidence before the routine reference is sufficiently learned. We therefore adopt a two-stage optimization strategy consisting of routine-reference warm-up and joint event-aware training.

D.1 Routine Reference Warm-up

The routine branch provides the reference against which abnormal traffic variations are identified. We define the routine component of the observed traffic as

$$\mathbf{X}^{rt} = \mathbf{X}^{tr} + \mathbf{X}^{se}, \tag{12}$$

where $\mathbf{X}^{tr}$ and $\mathbf{X}^{se}$ denote the trend and seasonal components, respectively.

The routine estimation head introduced in the event trigger predicts

$$\widetilde{\mathbf{X}}_t^r = Head_r\left(\mathbf{H}_t^r\right). \tag{13}$$

To ensure that the residual used for event detection mainly reflects deviations from routine evolution, we introduce a routine reconstruction objective:

$$\mathcal{L}_{rec} = \frac{1}{TNC}\sum_{t=1}^{T}\left\|\widetilde{\mathbf{X}}_t^r - \left(\mathbf{X}_t^{tr} + \mathbf{X}_t^{se}\right)\right\|_1. \tag{14}$$

The routine forecasting branch is simultaneously optimized using

$$\mathcal{L}_{pred}^r = \frac{1}{HNC_y}\sum_{h=1}^{H}\sum_{i=1}^{N}\left\|\widehat{\mathbf{Y}}_{t+h,i}^r - \mathbf{Y}_{t+h,i}\right\|_1. \tag{15}$$

During the warm-up stage, the optimization objective is

$$\mathcal{L}_{warm} = \mathcal{L}_{pred}^r + \lambda_{rec}\mathcal{L}_{rec}. \tag{16}$$

This stage provides a stable routine reference before the event-conditioned branch is jointly optimized.

D.2 Forecasting Objective

After routine warm-up, the complete ETDGN model is trained end-to-end. The primary objective minimizes the forecasting error of the final prediction:

$$\mathcal{L}_{pred} = \frac{1}{HNC_y}\sum_{h=1}^{H}\sum_{i=1}^{N}\left\|\widehat{\mathbf{Y}}_{t+h,i} - \mathbf{Y}_{t+h,i}\right\|_1. \tag{17}$$

The forecasting loss is applied to all prediction horizons and remains the dominant optimization objective.

D.3 Event Trigger Supervision

The event trigger should respond to event-related abnormalities while remaining inactive during routine traffic evolution. Let $e_{t,i} \in \{0,1\}$ denote the event indicator for node i at time t.

When explicit event-related annotations or aligned contextual event indicators are available, they are directly used to construct $e_{t,i}$. For datasets without explicit event labels, pseudo event indicators are generated from the routine residual:

$$r_{t,i} = \left|x_{t,i} - \widetilde{x}_{t,i}^r\right|, \qquad e_{t,i} = \mathbb{I}\left(r_{t,i} > \delta\right), \tag{18}$$

where

$$\delta = \kappa \sigma_r, \tag{19}$$

and σ_r denotes the standard deviation of routine residuals estimated from the training data. Following the sensitivity analysis in the main text, we use $\kappa = 1.5$ by default.

The event consistency loss is defined as binary cross-entropy:

$$\mathcal{L}_{cons} = -\frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \Big[e_{t,i} \log g_{t,i} + (1 - e_{t,i}) \log(1 - g_{t,i}) \Big]. \tag{20}$$

Since event-related periods are substantially less frequent than routine traffic states, we additionally penalize unnecessary event activation:

$$\mathcal{L}_{sparse} = \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} g_{t,i}. \tag{21}$$

The complete trigger objective is

$$\mathcal{L}_{trig} = \mathcal{L}_{cons} + \lambda_{sp} \mathcal{L}_{sparse}. \tag{22}$$

The consistency term prevents trivial event suppression, while the sparsity term discourages the model from activating event-specific computation for routine fluctuations.

D.4 Event Graph Regularization

The event-conditioned graph is intended to capture temporary spatial dependency changes only around event-affected regions. Although the $TopK$ operator and event-region gate already restrict the candidate connections, we further regularize the magnitude of the event-specific graph:

$$\mathcal{L}_{graph} = \frac{1}{TN^2} \sum_{t=1}^{T} \left\| \mathbf{A}_t^e \right\|_1. \tag{23}$$

This regularization suppresses unnecessarily dense event-conditioned dependencies and encourages the graph adaptation to remain localized around event-related regions.

D.5 Routine–Event Disentanglement

ETDGN keeps routine traffic representations and event-induced deviations in separate branches. To reduce redundant information shared by the two representations, we minimize their element-wise overlap:

$$\mathcal{L}_{dis} = \frac{1}{TNd} \sum_{t=1}^{T} \left\| \mathbf{H}_t^r \odot \mathbf{H}_t^e \right\|_1. \tag{24}$$

Unlike the previous formulation based on undefined intermediate representations, $\mathcal{L}_{dis}$ directly constrains the routine representation $\mathbf{H}_t^r$ and event-aware representation $\mathbf{H}_t^e$ used by the forecasting model. A smaller overlap encourages the event branch to focus on deviations that cannot be adequately represented by routine traffic evolution.

D.6 Historical Memory Consistency

The historical event memory retrieves reusable event-response patterns according to the current query $\mathbf{q}_t$. Since memory retrieval is mainly required when event evidence is present, we weight the consistency objective by the average event state

$$\bar{m}_t = \frac{1}{N} \sum_{i=1}^{N} m_{t,i}. \tag{25}$$

The memory consistency loss is defined as

$$\mathcal{L}_{mem} = \frac{\sum_{t=1}^{T} \bar{m}_t \left(1 - \frac{\mathbf{q}_t^\top \mathbf{C}_t^{mem}}{\|\mathbf{q}_t\|_2 \|\mathbf{C}_t^{mem}\|_2 + \epsilon}\right)}{\sum_{t=1}^{T} \bar{m}_t + \epsilon}. \tag{26}$$

This objective encourages the retrieved memory context to remain semantically consistent with the current event state while avoiding unnecessary constraints during routine periods.

D.7 Joint Objective and Training Strategy

After routine-reference warm-up, all ETDGN components are optimized jointly using

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{pred} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{trig}\mathcal{L}_{trig} \\ + \lambda_{graph}\mathcal{L}_{graph} + \lambda_{dis}\mathcal{L}_{dis} + \lambda_{mem}\mathcal{L}_{mem},\end{aligned} \tag{27}$$

where λ_{rec}, λ_{trig}, λ_{graph}, λ_{dis}, and λ_{mem} control the relative strengths of the auxiliary objectives.

To prevent auxiliary regularization from dominating the forecasting objective at the beginning of joint training, the regularization terms can be gradually introduced using

$$\xi_e = \min\left(1, \frac{e}{E_{ramp}}\right), \tag{28}$$

where e denotes the current joint-training epoch and E_{ramp} denotes the regularization warm-up period. The practical joint objective is therefore

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{pred} + \lambda_{rec}\mathcal{L}_{rec} + \xi_e\Big(\lambda_{trig}\mathcal{L}_{trig} + \lambda_{graph}\mathcal{L}_{graph} \\ + \lambda_{dis}\mathcal{L}_{dis} + \lambda_{mem}\mathcal{L}_{mem}\Big).\end{aligned} \tag{29}$$

This training strategy first establishes reliable routine dynamics and subsequently introduces event-specific constraints, reducing the risk that inaccurate early residuals cause excessive event activation.

D.8 Implementation Details

For datasets without explicit event annotations, event supervision is derived from routine forecasting residuals according to Eq. 18. For datasets containing event-related contextual variables, such as weather or holiday information, the aligned context indicators are used as additional supervision for the event trigger.

During inference, the continuous trigger score $\mathbf{g}_t$ and event memory $\mathbf{m}_t$ are retained for event-state modeling. A hard activation decision is used only to determine whether event-conditioned graph construction and propagation need to be executed. The activation threshold τ_e is selected using the validation set and is fixed during testing.

The propagation depth is set to $L = 3$, and the residual-based event threshold is set to $\delta = 1.5\sigma_r$, following the parameter-sensitivity results in the main text. All normalization statistics are estimated using the training split only. Model selection is performed according to validation forecasting performance, and the same evaluation protocol is used for all reported test results.