Balancing Timeliness and Accuracy: Adaptive Early Spatiotemporal Forecasting With Multigraph Modeling

Xiao Xiao, *Member, IEEE*, Liwen Wang, Ziyan Peng, Zhisheng Yin, *Member, IEEE*, Yuan Zhang, Xuzhi He, Zhiqiang Li, and Guoqiang Mao, *Fellow, IEEE*

***Abstract*—The time-sensitive IoT applications require forecasts that are both accurate and timely. However, the existing spatiotemporal forecasting methods usually treat prediction time as a fixed observation horizon shared by all nodes, which is fundamentally mismatched with the asynchronous emergence of predictive signals across locations. We formulate adaptive early spatiotemporal forecasting as a coupled nodewise decision problem, where the model must determine when to predict and what to predict from partial observations under evolving spatial dependencies. To this end, we propose early spatiotemporal multigraph convolutional network (ESTMGCN), a unified framework that learns nodewise prediction timing while preserving informative spatiotemporal dependencies and reducing error accumulation in early forecasting. Experiments on four real-world datasets show that the ESTMGCN consistently outperforms strong baselines under different target observation budgets and achieves a better tradeoff between timeliness and accuracy.**

***Index Terms*—Early spatiotemporal forecasting, graph neural networks, Internet of Things, multigraph modeling, prediction-time selection.**

I. Introduction

IOT-DRIVEN sensing infrastructures continuously generate large-scale spatiotemporal data, making forecasting a core capability in many real-world systems, such as traffic management [1], parking guidance [2], [3], wildfire monitoring [4], crime prevention [5], and mobile service recommendation [6]. In these applications, predictive performance is not determined by accuracy alone [7], [8]. A highly accurate forecast that arrives too late may miss the intervention window, whereas an early but unreliable forecast may trigger unnecessary responses, waste public resources, or even cause social panic. Therefore, the practical goal of spatiotemporal event prediction is to achieve a desirable balance between timeliness and accuracy.

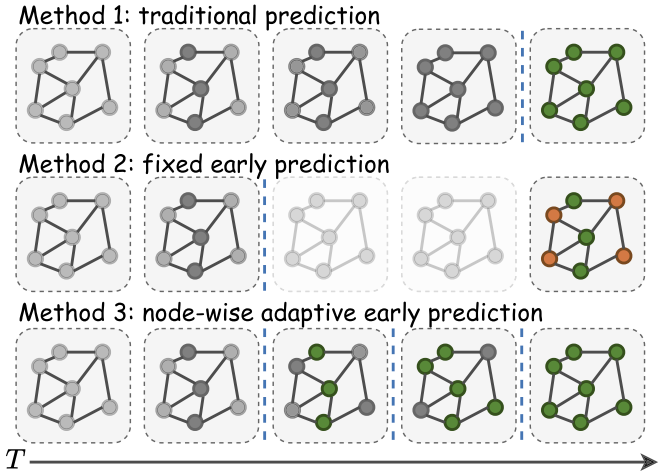


Fig. 1. Comparison of three forecasting paradigms: traditional full-horizon prediction, fixed-time early prediction, and nodewise adaptive early prediction. The blue dashed vertical line indicates the prediction time. The gray shading denotes the amount of observed information, where darker gray means more sufficient information. Green nodes represent correct predictions, while red nodes represent incorrect predictions. Traditional forecasting waits until the end of the observation horizon to predict all nodes; fixed early prediction uses the same predefined prediction time for all nodes; adaptive early prediction determines node-specific prediction times according to the evolving information sufficiency, thereby better balancing timeliness and accuracy.

Fig. 1 contrasts traditional full-horizon prediction, fixed-time early prediction, and nodewise adaptive early prediction.

Most existing spatiotemporal forecasting methods focus primarily on improving predictive accuracy by modeling temporal evolution and spatial dependency with recurrent, convolutional, or graph neural architectures. Although these models have achieved strong performance, they generally operate under a fixed observation horizon: the model waits until a predefined time and then predicts future values for all nodes. Such a strategy is often suboptimal in time-sensitive scenarios, because informative signals do not emerge simultaneously across all locations. Some nodes may become predictable at an early stage, while others may still need additional observations. Treating all nodes with the same prediction time

Received 17 April 2026; revised 20 August 2026; accepted 30 August 2026. Date of publication 4 September 2026; date of current version 26 October 2026. This work was supported in part by the Key Project of the Joint Fund of the National Natural Science Foundation of China under Grant U21A20446 and in part by the Natural Science Basic Research Program of Shaanxi Province under Grant 2025JC-YBMS-716. *(Corresponding author: Liwen Wang.)*

Xiao Xiao, Liwen Wang, Ziyan Peng, and Zhisheng Yin are with Xidian University, Xi'an 710071, China (e-mail: xiaoxiao@xidian.edu.cn; liwenwang@stu.xidian.edu.cn; ziyanpeng@stu.xidian.edu.cn; zsyin@xidian.edu.cn).

Yuan Zhang is with Shanghai Institute of Mechanical and Electrical Engineering, Shanghai 200040, China (e-mail: zhangyuanformal@163.com).

Xuzhi He is with the Department of Biomedical Engineering, University of California, Davis, CA 95616 USA (e-mail: hexuzhi6@gmail.com).

Zhiqiang Li is with the National Key Laboratory of Complex System Control and Intelligent Agent Cooperation, Beijing 100074, China (e-mail: zqli2010@163.com).

Guoqiang Mao is with the School of Transportation, Southeast University, Nanjing 211189, China (e-mail: g.mao@ieee.org).

Digital Object Identifier 10.1109/JIOT.2026.3730882

2327-4662 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

may therefore either delay decisions unnecessarily or force predictions to be made before sufficient evidence has been observed.

Early prediction [9] provides a natural direction for addressing this limitation by making predictions from partial observations instead of waiting for the complete sequence. Traditional full-horizon forecasting [10] usually attains high accuracy but suffers from delayed response. Fixed-time early prediction [11] reduces latency, yet it still requires all nodes to share the same prediction moment, ignoring the heterogeneity of local dynamics. Adaptive early prediction is more flexible because it learns when to stop observing and when to predict based on the incoming data [12]. However, most existing adaptive early prediction methods are designed for generic time-series or classification settings and do not explicitly address the graph-structured and dynamically evolving dependencies that are central to spatiotemporal event forecasting.

Introducing adaptive early prediction into graph-based spatiotemporal forecasting raises several unique challenges.

1) The optimal prediction time is inherently node-dependent. Different regions evolve at different rates and may exhibit very different signal strengths, which means that a unified global stopping time is inadequate.
2) Spatial dependency is not static. In real-world environments, correlations among nodes may change over time due to latent factors, abrupt events, or evolving propagation patterns; thus, relying solely on a fixed graph can limit early-stage prediction quality.
3) Reducing the observation window generally increases uncertainty and makes multistep forecasting more vulnerable to error accumulation, which can severely degrade the reliability of early predictions.

To address these issues, we propose the early spatiotemporal multigraph convolutional network (ESTMGCN), an adaptive early forecasting framework for graph-structured spatiotemporal data. ESTMGCN combines three complementary components. First, a reinforcement-learning-based controller learns how to select an appropriate prediction time for each node and issue its forecast, enabling nodewise adaptive prediction rather than relying on a fixed schedule. Second, a dynamic multigraph convolution module learns what spatiotemporal relations should be emphasized by jointly modeling static topology, dynamic internode dependency, and temporal similarity. Third, a transform-attention module improves how future representations are generated by directly linking encoder states to decoder states, thereby alleviating error accumulation along the forecasting horizon.

The main contributions of this work are summarized as follows.

1) We formulate adaptive early spatiotemporal forecasting as a nodewise decision problem that explicitly balances prediction timeliness and forecasting accuracy.
2) We propose ESTMGCN, an encoder–decoder framework with a reinforcement-learning-based controller that adaptively determines the prediction time for each node instead of relying on a global fixed prediction schedule.
3) We design a dynamic multigraph convolution module and a transform-attention module to capture evolving spatiotemporal dependencies and mitigate error accumulation in early forecasting.
4) We evaluate ESTMGCN on four real-world datasets and demonstrate that it achieves a favorable tradeoff between timeliness and predictive performance across different spatiotemporal forecasting scenarios.

This work extends our KDD'24 paper STEMO [13] by retaining nodewise adaptive early forecasting and multistep temporal similarity modeling, while introducing dynamic multigraph construction, transform-attention, and a redesigned prediction-time controller. It further adds the air quality dataset, stronger baselines, and new ablation, statistical, runtime, stability, and interpretability analyses.

II. Related Work

The spatiotemporal forecasting plays a crucial role in a wide range of applications, including weather forecasting [14], traffic flow prediction [15], and earthquake monitoring [16]. Recent advances in spatiotemporal forecasting have substantially improved predictive accuracy by developing stronger mechanisms for modeling spatial dependency and temporal evolution. DCRNN [11] employs diffusion convolution within an encoder–decoder recurrent architecture to capture directed spatial propagation and temporal dynamics. Graph WaveNet [17] replaces recurrent temporal modeling with dilated causal convolutions and further introduces adaptive graph learning to capture latent dependencies. ASTGCN [18] and GMAN [19] emphasize attention-based spatial–temporal interaction modeling, while AGCRN [20], DGCRN [1], and MTGNN [21] further move toward adaptive or dynamic graph construction for multivariate temporal dependency learning. These studies demonstrate that better graph structures and temporal operators can significantly improve forecasting accuracy under a predefined observation budget. However, their prediction protocols are still largely based on a fixed observation horizon or a predetermined prediction schedule. In other words, they mainly address "how to forecast better" after a given amount of data is observed, rather than "when to forecast" for each node under time-sensitive decision requirements. Recent studies, such as sensor-aware CapsNet [7] and the Exofeature-aware transformer [8], highlight the importance of modeling heterogeneous sensing information for reliable prediction, whereas ESTMGCN focuses on nodewise adaptive prediction timing in spatiotemporal IoT systems.

Early prediction aims to make decisions as early as possible while preserving acceptable reliability. Earlier studies formulate early classification as a nonmyopic or cost-aware sequential decision problem, explicitly optimizing the tradeoff between earliness and accuracy. More recent methods, such as TEASER [22] and EARLIEST [12], introduce adaptive triggering or reinforcement-learning-based halting policies to determine when sufficient evidence has been observed. The related attention-guided models, such as RAIM [23], further improve early prediction by focusing on salient segments in multimodal monitoring streams. More recently, adaptive early prediction has also begun to be explored in graph-structured

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON OF ESTMGCN WITH REPRESENTATIVE BASELINES

Method	Adaptive timing	Node-wise decision	Evolving dependency modeling	Explicit error-control
DCRNN	No	No	No	Yes
Graph WaveNet	No	No	No	No
MTGNN	No	No	No	No
EARLIEST	Yes	No	No	No
RAIM	Yes	No	No	No
STEMO	Yes	Yes	No	No
ASTER	Yes	Yes	Yes	No
ESTMGCN	Yes	Yes	Yes	Yes

TABLE II
MAIN NOTATION USED IN ESTMGCN

Symbol	Definition
$\mathcal{G}=(V,E), N, T$	Graph, number of nodes, and maximum observation length
$\mathbf{X}_t, \mathbf{X}_{0:t}, \hat{X}_T^{(t)}$	Observation at t, observed prefix, and target-step forecast made at t
$\mathbf{A}^{\mathcal{S}}, \mathbf{A}_t^{\mathcal{D}}, \mathbf{A}_{t,t'}^{\mathcal{T}}$	Static, dynamic, and temporal-similarity graphs
$\mathbf{X}_{\mathrm{sta}}, \mathbf{X}_{\mathrm{se}}, \mathbf{X}_{\mathrm{te}}$	Static node embedding and source/target embeddings
$\mathbf{H}_t, \mathbf{H}_{t_2}^{(t)}$	Encoder and decoder hidden representations
$p_t^i, a_t^i, m_t^i, t_i^*$	Decision probability, raw action, first-hit selector, and selected prediction time
L_p, L_c, L_e	Prediction, controller, and earliness losses

spatiotemporal settings, and STEMO [13] addresses adaptive early spatiotemporal forecasting but does not explicitly model dynamic spatial dependency or mitigate multistep forecasting errors. Although these methods provide important insights into timeliness-aware decision making, they are mainly developed for sequence classification or event prediction on individual data streams. Consequently, they usually assume a single decision process for each sequence and do not explicitly model graph-structured internode dependency, nodewise adaptive prediction time in spatiotemporal systems, or the coupling between adaptive timing and multistep forecasting error propagation. Therefore, despite progress in both spatiotemporal forecasting and early prediction, their integration for graph-structured adaptive early forecasting remains limited.

To highlight the key methodological differences, Table I compares ESTMGCN with representative baselines in terms of adaptive timing, nodewise decision, evolving dependency modeling, and explicit error control.

III. PROBLEM FORMULATION

Let $\mathcal{G}=(V,E)$ be a graph with N nodes, where $v_i \in V$ denotes the ith node and $(v_i, v_j) \in E$ denotes an edge between nodes v_i and v_j. Let $A \in \mathbb{R}^{N\times N}$ denote the adjacency matrix of $\mathcal{G}$, which encodes the spatial proximity between nodes and can be constructed according to prior knowledge, such as road-network distance. Let $\mathbf{X}_t \in \mathbb{R}^N$ denote the observed node values at time step t, and let $\mathbf{X}_{0:t}=\{\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_t\}$ denote the partial observation sequence up to time t.

Unlike conventional spatiotemporal forecasting, which uses a fixed observation horizon for all nodes, we consider an adaptive early forecasting setting. Specifically, for each node v_i, the model is expected to determine an appropriate prediction time and output its forecast for the target step T using only partial observations. Let $t_i^* \in \{0, 1, \ldots, T-1\}$ denote the optimal prediction time for node i.

Throughout this article, *adaptive early spatiotemporal forecasting* denotes this forecasting task, whereas *prediction-time selection* denotes the nodewise stopping operation.

For node v_i, let $\hat{X}_{T,i}^{(t)}$ denote the prediction generated at time t based on $X_{0:t}$. The optimal prediction time should balance forecasting accuracy and prediction timeliness, which can be conceptually formulated as

$$t_i^* = \arg\max_{t\in\{0,\ldots,T-1\}} \left[\log P\left(\hat{X}_{T,i}^{(t)} \mid X_{0:t}, G\right) - g(t)\right] \tag{1}$$

where $g(t)=t/(T-1)$ is a normalized monotonically increasing delay penalty. Here, $P(\hat{X}_{T,i}^{(t)} \mid X_{0:t}, G)$ represents the predictive reliability of the candidate forecast. Equation (1) is a conceptual criterion and is not directly optimized. In practice, its reliability–timeliness tradeoff is operationalized by L_p, L_c, and L_e in (24). Specifically, L_p is the MAE prediction loss, L_c uses the negative-absolute-error reward for controller optimization, and L_e encourages earlier predictions.

The main notation used in the model is summarized in Table II.

IV. METHODOLOGY

A. Overview of ESTMGCN

The overall architecture of ESTMGCN is shown in Fig. 2. Given a graph $G=(V,E)$ with N nodes, let $X_t \in \mathbb{R}^N$ denote the observations at time step t, and let $X_{0:t}=\{X_0, \ldots, X_t\}$ denote the partial observation sequence up to time t. Our goal is to perform adaptive early spatiotemporal forecasting, i.e., to determine an appropriate prediction time for each node and to output an accurate forecast of the target horizon using only partial observations.

ESTMGCN adopts an encoder–decoder architecture with three core components: 1) a multigraph structural modeling module; 2) a transform-attention bridge; and 3) an adaptive prediction-time controller. The multigraph module captures complementary relations from static topology, dynamic dependency, and temporal similarity. The encoder summarizes the observed history under the fused graph operator, while the transform-attention bridge generates future-aware decoder representations to alleviate error accumulation. Based on these representations, the controller estimates nodewise decision probabilities for selecting appropriate prediction times.

At each time step t, ESTMGCN first updates the graph structures from the current and historical observations and then computes the encoder hidden state through a graph-aware recurrent unit. The transform-attention bridge subsequently maps encoder states to decoder-side future representations.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

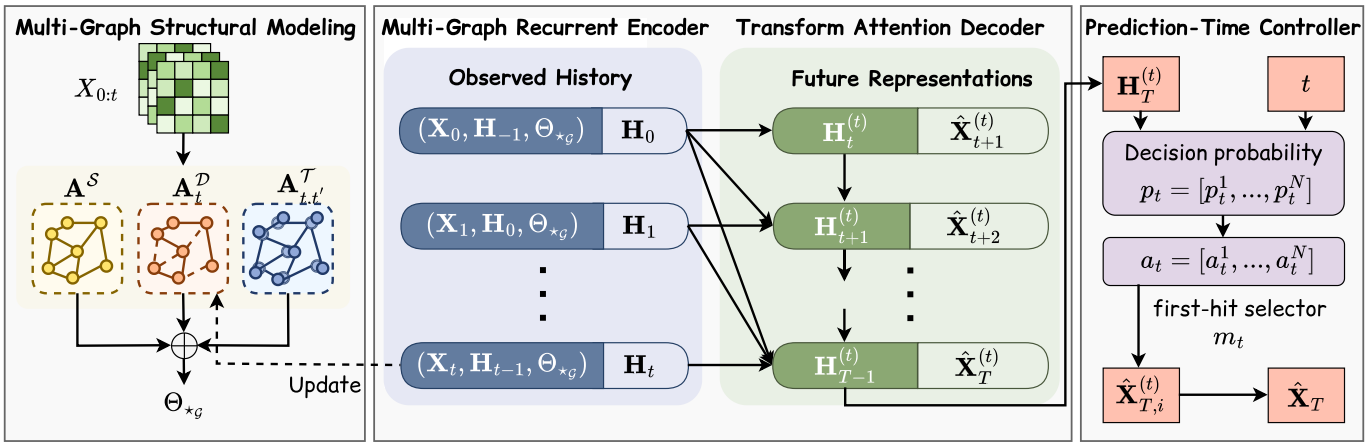


Fig. 2. Architecture of ESTMGCN. The model first updates multigraph structures from the current observation and historical states and then encodes the observed sequence with a multigraph recurrent encoder. A transform-attention bridge maps encoder states to decoder-side future representations for forecasting. Based on the terminal decoder representation $H_T^{(t)}$ and the time-step information, the prediction-time controller estimates nodewise decision probabilities and produces selection actions for stepwise predictions. The final forecast retains the unique first-hit prediction for each node according to (18) and (19).

Finally, the controller produces nodewise decision probabilities, and the first-hit rule assigns one prediction time to each node.

B. Multigraph Structural Modeling

To support adaptive early forecasting under evolving spatiotemporal dependencies, we model three complementary graph structures: a static spatial graph, a dynamic dependency graph, and a multistep temporal similarity graph.

1) Static Spatial Graph: The static graph provides a time-invariant topology prior as a symmetric, undirected weighted graph. Let $\mathbf{A}^{\mathcal{S}} \in \mathbb{R}^{N\times N}$ denote the static adjacency matrix. Following the original setting, its entries are defined from geographical proximity:

$$\mathbf{A}^{\mathcal{S}}_{i,j} = \begin{cases} \exp\left(-\dfrac{d_{ij}^2}{s^2}\right), & i \neq j \\ 0, & i = j \end{cases} \tag{2}$$

where d_{ij} is the distance between nodes v_i and v_j, and s controls the edge-weight distribution. Let $\tilde{\mathbf{D}}$ be the degree matrix of $\mathbf{A}^{\mathcal{S}}$. The row-normalized operator $\tilde{\mathbf{D}}^{-1}\mathbf{A}^{\mathcal{S}}$ stabilizes feature propagation.

2) Dynamic Dependency Graph: A fixed graph cannot characterize rapidly changing internode relations. To model short-term and context-dependent interactions, we construct a dynamic adjacency matrix $\mathbf{A}_t^{\mathcal{D}} \in \mathbb{R}^{N\times N}$ at each time step. Specifically, the current observation $\mathbf{X}_t$ and the previous hidden state H_{t-1} are first projected over the static graph to generate source and target embeddings, denoted by $\mathbf{X}_{\text{se}}$ and $\mathbf{X}_{\text{te}}$, respectively. The computation process for $\mathbf{X}_{\text{se}}$ and $\mathbf{X}_{\text{te}}$ is as follows:

$$\mathbf{X}_{\text{se}} = \text{ReLU}\left(\tilde{\mathbf{D}}^{-1}\mathbf{A}^{\mathcal{S}}\left(\mathbf{W}_l[\mathbf{X}_t;\mathbf{H_{t-1}}]\right)\mathbf{W_{se}}\right)\mathbf{X}_{\text{sta}} \tag{3}$$

$$\mathbf{X}_{\text{te}} = \text{ReLU}\left(\tilde{\mathbf{D}}^{-1}\mathbf{A}^{\mathcal{S}}\left(\mathbf{W}_l[\mathbf{X}_t;\mathbf{H_{t-1}}]\right)\mathbf{W_{te}}\right)\mathbf{X}_{\text{sta}} \tag{4}$$

where $[X_t; H_{t-1}]$ denotes the concatenation of the current observation and the previous hidden state, W_l is a learnable linear projection matrix that maps the concatenated input into an intermediate latent space, and W_{se} and W_{te} are the learnable projection matrices used to produce the source and target embeddings, respectively. The matrix $\mathbf{X}_{\text{sta}}$ is a randomly initialized, learnable static node embedding.

The dynamic adjacency matrix is then defined as

$$\mathbf{A}_t^{\mathcal{D}} = \text{ReLU}\left(\tanh\left(\alpha\left(\mathbf{X}_{se}\mathbf{X}_{te}^T - \mathbf{X}_{te}\mathbf{X}_{se}^T\right)\right)\right) \tag{5}$$

where α controls the saturation rate. The antisymmetric score followed by ReLU produces a directed, nonnegative weighted graph. Self-loops and row normalization are applied before fusion, while the static graph remains in every fused operator as a stable topology prior.

3) Multistep Temporal Similarity Graph: Early forecasting also depends on cross-node temporal similarity and potential lead–lag relations. To model such information, we introduce a multistep temporal similarity tensor $\mathbf{A}_t^{\mathcal{T}} \in \mathbb{R}^{(t+1)\times N\times N}$, in which each slice $\mathbf{A}_{t,t'}^{\mathcal{T}}$ measures the similarity between the current prefix of one node and the historical prefix of another node. For nodes v_i and v_j, we define

$$\mathbf{A}^{\mathcal{T}}_{t,t',i,j} = \begin{cases} \exp\left(-\kappa\cdot\text{DTW}\left(x_{i,0:t}, x_{j,0:t'}\right)\right), & i \neq j \\ 0, & i = j \end{cases} \tag{6}$$

where $\text{DTW}(\cdot,\cdot)$ denotes the dynamic time warping and κ controls the decay of temporal-similarity weights. DTW is computed online by exact dynamic programming, using one $T \times T$ table per node pair and $\mathcal{O}(BN^2T^2)$ time for a batch of size B. Each temporal-similarity slice uses continuous weights and is row-normalized after adding self-loops. The value of κ is selected on the validation set and then fixed for the corresponding dataset.

4) Multigraph Fusion: The three graph structures serve different purposes: $\mathbf{A}^{\mathcal{S}}$ provides stable topology, $\mathbf{A}_t^{\mathcal{D}}$ captures time-varying dependency, and $\mathbf{A}_{t,t'}^{\mathcal{T}}$ models temporal similarity across observation prefixes. The dynamic graph weights are interpreted as data-dependent directional dependency scores rather than causal relations. Although both the dynamic and temporal-similarity graphs are observation dependent, the former models instantaneous latent internode dependency,

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

whereas the latter measures prefix-level temporal-pattern similarity, so they provide complementary information. No explicit top-k pruning is used; sparsity arises implicitly from the ReLU suppression of nonpositive relation scores. We fuse the three graphs into a unified graph operator

$$\mathbf{H}_{\text{out}} = \sum_{k=0}^{1} \mathbf{W}_d^{(k)} \Bigg(\mathbf{H}_{\text{in}}^{(k)} \Bigg(\beta_1 \mathbf{I} + \beta_2 \tilde{\mathbf{D}}^{-1} \mathbf{A}^{\mathcal{S}} + \beta_3 \check{\mathbf{D}}_t^{-1} \left(\mathbf{A}_t^{\mathcal{D}} + \mathbf{I} \right) + \beta_4 \sum_{t'=0}^{t-1} \hat{\mathbf{D}}_{t,t'}^{-1} \left(\mathbf{A}_{t,t'}^{\mathcal{T}} + \mathbf{I} \right) \Bigg) \Bigg) \tag{7}$$

where $\beta_1, \beta_2, \beta_3, \beta_4$ are the fixed scaling coefficients set to $[0.05, 0.95, 0.95, 0.95]$ for all datasets. They scale the four components independently and are not constrained to sum to one. $\check{\mathbf{D}}_t$ and $\hat{\mathbf{D}}_{t,t'}$ are the degree matrices of the dynamic and temporal-similarity graphs, respectively. For simplicity, the fused operation is denoted by

$$\mathbf{H}_{\text{out}} = \Theta_{\star_{\mathcal{G}}} (\mathbf{H}_{\text{in}}) \tag{8}$$

where $\star_{\mathcal{G}}$ denotes the multigraph convolution.

C. Multigraph Recurrent Encoder

After obtaining the fused graph operator, we embed it into a recurrent encoder to jointly model temporal evolution and graph-structured dependency. Instead of using standard linear transformations in GRU, we replace them with multigraph convolution

$$z_t = \sigma \left(\Theta_{z\star_{\mathcal{G}}} (\mathbf{H}_{t-1} \,\|\, \mathbf{X}_t) \right) \tag{9}$$

$$r_t = \sigma \left(\Theta_{r\star_{\mathcal{G}}} (\mathbf{X}_t \,\|\, \mathbf{H}_{t-1}) \right) \tag{10}$$

$$\hat{h}_t = \tanh \left(\Theta_{h\star_{\mathcal{G}}} (r_t \odot \mathbf{H}_{t-1} \,\|\, \mathbf{X}_t) \right) \tag{11}$$

$$\mathbf{H}_t = (1 - z_t) \odot \mathbf{H}_{t-1} + z_t \odot \hat{h}_t \tag{12}$$

where Θ_z, Θ_r, and Θ_h represent multigraph convolution operators applied during the update gate, reset gate, and candidate hidden state computation, respectively. $\sigma(\cdot)$ denotes the sigmoid function, $\odot$ indicates the elementwise multiplication, and $\|$ represents concatenation.

This design preserves the sequential memory mechanism of GRU while allowing each hidden-state update to adapt to the fused spatiotemporal structure inferred from partial observations.

D. Transform-Attention Decoder Bridge

Shortening the observation window generally makes multistep forecasting more vulnerable to error accumulation, as shown in Fig. 3. To alleviate this issue, we introduce a transform-attention bridge that directly maps encoder-side history to decoder-side future representations [19]. We first construct a spatiotemporal embedding that combines temporal position and node-specific signal information. Let $\mathbf{X}_t^{\mathcal{T}} \in \mathbb{R}^D$ denote the temporal embedding of time step t, and let $\mathbf{X}_i^{\mathcal{S}} \in \mathbb{R}^D$ denote the spatial embedding of node v_i, where D is the embedding size. Their fused representation is written as

$$\mathcal{X}_{i,t}^{\mathcal{ST}} = \mathbf{X}_i^{\mathcal{S}} + \mathbf{X}_t^{\mathcal{T}}. \tag{13}$$

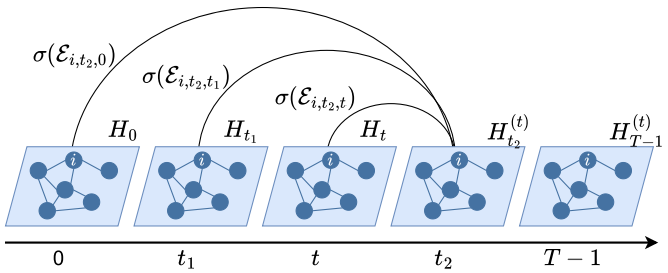


Fig. 3. Structure of the transform-attention module. At time t, the hidden state $\mathbf{H}_{0:t}$ is the encoder output from time 0 to t, the hidden state $\mathbf{H}_{t:T-1}^{(t)}$ is the decoder input, and $\text{softmax}_{t_1}(\mathcal{E}_{i,t}(t_1, t_2))$ represents the attention weight of node v_i between $\mathbf{H}_{t_1}$ and $\mathbf{H}_{t_2}^{(t)}$, where $t_1 = 0, \ldots, t$ and $t_2 = t, \ldots, T-1$.

Based on this embedding, we establish direct dependencies between encoder states and future decoder representations. For node i at observation step t, the attention score between encoder time t_1 and decoder time t_2 is computed by

$$\mathcal{E}_{i,t}(t_1, t_2) = \frac{\langle \mathrm{e}_{i,t_1}, \mathrm{e}_{i,t_2} \rangle}{\sqrt{D}}$$
$$\mathrm{e}_{i,t'} = \text{ReLU} \left(W \mathcal{X}_{i,t'}^{\mathcal{ST}} + b \right) \tag{14}$$

where W and b are the learnable parameters. For a larger value of D, the dot-product magnitude tends to increase, which may push the softmax in (15) into low-gradient regions. To counteract this effect, the dot products are computed and multiplied by the scaling factor $1/\sqrt{D}$. The hidden state of the decoder input is

$$\mathbf{H}_{i,t_2}^{(t)} = \sum_{t_1=0}^{t} \text{softmax}_{t_1} \left(\mathcal{E}_{i,t}(t_1, t_2) \right) \text{ReLU} \left(W_h \mathbf{H}_{i,t_1} + b_h \right) \tag{15}$$

where $t_2 = t, \ldots, T-1$, and the softmax is taken over $t_1 = 0, \ldots, t$.

By directly attending to informative encoder states, the decoder no longer relies solely on recursive propagation across future steps, which helps reduce cumulative prediction errors under early observation.

To interface the transform-attention decoder with the controller, after obtaining the decoder-side states $H_{t:T-1}^{(t)}$, we perform one additional decoder update to produce the terminal decoder representation $H_T^{(t)}$. This representation summarizes the decoder rollout conditioned on the partial observations up to time t and serves as the input to both the prediction-time controller and the stepwise prediction head.

For the controlled comparison in Section V-D4, the standard encoder–decoder attention variant replaces only the transform-attention bridge. The encoder, decoder, multigraph module, controller, hidden dimension $H = 32$, data splits, training hyperparameters, and one-observation input protocol remain unchanged.

E. Adaptive Prediction-Time Controller

We design a controller based on reinforcement learning to adaptively determine the prediction time for each node. To determine the optimal prediction time $t^* = \{t_i^*\}_{i=1}^N$, we introduce a policy π. At time t, the replicated time vector $\boldsymbol{\tau}_t = [t, \ldots, t] \in \mathbb{R}^H$ is concatenated with each node's decoder

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

output $H_{T,i}^{(t)} \in \mathbb{R}^H$, yielding the controller state $S_t \in \mathbb{R}^{N\times 2H}$. A shared linear layer followed by a sigmoid produces the nodewise decision probabilities p_t

$$p_t = \sigma\left(W_c S_t + b_c\right) \tag{16}$$

where W_c and b_c are the learnable parameters, and $\sigma(\cdot)$ denotes the sigmoid activation. With $H = 32$, this shared Linear(64, 1) controller has 65 trainable parameters. The raw action $a_t = \{a_t^i\}_{i=1}^N$ is sampled independently for each sample and node according to $\pi(a_t^i = 1) = p_t^i$.

To avoid excessive exploitation during training, we employ ϵ-greedy action selection

$$a_t^i \sim \begin{cases} \text{Bernoulli}\left(p_t^i\right), & \text{with probability } 1-\epsilon_k \\ \text{Bernoulli}\left(0.5\right), & \text{with probability } \epsilon_k \end{cases} \tag{17}$$

where $\epsilon_k = \exp(-7k/5000)$ for $0 \le k < 5000$ and $\epsilon_k = 0$ thereafter.

The use of ϵ-greedy exploration prevents the controller from collapsing too early to suboptimal decision patterns in the early stage of training. Since the controller must learn both when to predict and how prediction timing affects forecasting quality, pure exploitation can easily bias the policy toward locally favorable but globally suboptimal decision times. Gradually decreasing ϵ allows the model to explore diverse prediction-time selections at the beginning of training and then stabilize the learned policy in later epochs.

Only the first positive action determines the prediction time. The corresponding selector is

$$m_t^i = \begin{cases} a_t^i \prod_{k=0}^{t-1}\left(1-a_k^i\right), & 0 \le t < T-1 \\ \prod_{k=0}^{T-2}\left(1-a_k^i\right), & t = T-1 \end{cases} \tag{18}$$

where the empty product at $t = 0$ equals one. Thus, $\sum_{t=0}^{T-1} m_t^i = 1$ and $t_i^* = \sum_{t=0}^{T-1} t m_t^i$. Later positive actions have no effect, and the final step is selected when no earlier action is positive.

The final prediction is computed using $m_t = \{m_t^i\}_{i=1}^N$

$$\hat{X}_T = \sum_{t=0}^{T-1} m_t \odot \text{ReLU}\left(W_p H_T^{(t)} + b_p\right) \tag{19}$$

where $\odot$ denotes the Hadamard product, and W_p and b_p are the learnable parameters.

F. Training Objective

ESTMGCN is trained end-to-end by jointly optimizing forecasting accuracy, prediction timeliness, and the controller policy.

1) Prediction Loss: The prediction loss is the mean absolute error between the final prediction and the ground truth

$$L_p = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{X}_{T,i} - X_{T,i}\right|. \tag{20}$$

2) Earliness Loss: The earliness loss is defined as

$$L_e = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{t_i^*}\log\pi\left(a_t^i = 1 \mid S_t\right) \tag{21}$$

where t_i^* is the first-hit prediction time defined by (18). Minimizing this term increases the probability of selecting earlier prediction steps. Here, L_e is an auxiliary prefix regularizer on stopping propensity; it does not imply multiple effective predictions because (18) converts the raw action sequence into a unique first-hit event.

3) Controller Loss: Because the controller involves discrete action sampling, it is optimized with a policy-gradient surrogate. For node i, the stepwise reward and the first-hit return are

$$R_t^i = -|\hat{X}_{T,i}^{(t)} - X_{T,i}|, \qquad r^i = \sum_{t=0}^{T-1} m_t^i R_t^i = R_{t_i^*}^i. \tag{22}$$

The reward is computed in the training-data-normalized space with unit scaling, while timeliness is represented separately by L_e. Let $\mathcal{A}$ contain the active sample-node-time tuples up to their first-hit times. The controller loss is

$$L_c = -\frac{1}{|\mathcal{A}|}\sum_{(s,i,t)\in\mathcal{A}}\log\pi\left(a_{s,t}^i \mid S_{s,t}\right)\text{sg}\left(r_s^i - b_t\right) \tag{23}$$

where b_t is a learned scalar baseline shared across samples and nodes at step t. It is updated in every minibatch by minimizing the mean squared error between b_t and the stop-gradient return r_s^i.

4) Joint Objective: The final objective is

$$L = L_p + \omega_1 L_c + \omega_2 L_e \tag{24}$$

where ω_1 controls the contribution of the controller loss L_c, while ω_2 controls the strength of the earliness regularization L_e. A larger ω_2 encourages earlier prediction decisions, whereas a smaller ω_2 places more emphasis on forecasting accuracy.

G. Training Strategy and Practical Inference

The training algorithm in Algorithm 1 updates the learnable parameters of the dynamic graph convolution module $f_d(\cdot)$, transform-attention module $f_a(\cdot)$, and controller $f_c(\cdot)$. These modules and the prediction head are jointly trained from random initialization starting with the first minibatch. A gradient-norm cap of 5 and validation-MAE early stopping are applied for training stability.

During inference, observations arrive sequentially. At each time step t, ESTMGCN updates the fused graph structure and encoder hidden state, constructs decoder-side future representations through the transform-attention bridge, and computes p_t and $\hat{X}_T^{(t)}$. The first step satisfying $p_t^i \ge 0.5$ is selected for node i; if the threshold is never reached, $T-1$ is used. This deterministic rule is fixed across datasets and target observation budgets, and the final prediction follows (19).

By jointly optimizing Θ_d, Θ_a, and Θ_c, ESTMGCN learns the forecasting model and controller policy end-to-end.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

TABLE III
DATASET CONSTRUCTION AND PREDICTION SETTINGS. EACH INPUT CONTAINS $T = 12$ STEPS AND THE TARGET IS THE NEXT STEP

Dataset	Spatial nodes	Interval	Input span	Prediction target
METR-LA	207 highway sensors	5 min	60 min	Traffic speed in the next 5 min
EMS	145 ZIP-code regions	1 h	12 h	Dispatch count in the next hour
NYPD	77 patrol precincts	4 h	48 h	Incident count in the next 4 h
Air Quality	92 monitoring stations	2 h	24 h	$PM_{2.5}$ concentration in the next 2 h

V. EXPERIMENT

This section outlines the experimental settings and findings used to assess the effectiveness of the proposed ESTMGCN model. We conduct comparisons with baseline approaches, evaluate the contributions of individual modules, and investigate how various parameters influence overall model performance.

A. Datasets and Preprocessing

Experiments are conducted on four real-world datasets from diverse application domains: **METR-LA**, **EMS**,[1] **NYPD**, [2] and **Air Quality**[3] [24]. Their node definitions and temporal construction are summarized in Table III.

The EMS hourly counts are obtained by summing two consecutive 30-min dispatch-count bins, while NYPD records are aggregated into 4-h precinct-level counts. The data are divided chronologically into training, validation, and test sets at 70%, 10%, and 20%. Only the primary signal channel is standardized using scalar mean and standard deviation computed from the training split, and predictions are inverse-transformed before evaluation. In METR-LA, zero-valued targets are treated as missing and excluded from the loss and metrics; all finite values are retained for air quality, while zero counts are valid observations for EMS and NYPD. Therefore, the main comparison reports MAE and RMSE, avoiding the instability of MAPE on sparse or zero-inflated counts.

The static graph is constructed by the geographical-distance Gaussian kernel, using sensor or station locations for METR-LA and air quality and region centroids for EMS and NYPD. Dynamic and temporal-similarity graphs use only the current observation prefix, normalization statistics are obtained from the training split, and graph hyperparameters are selected on the validation split.

B. Baselines and Evaluation Protocol

ESTMGCN is compared with representative statistical, recurrent, graph-based, and early prediction baselines, including HA [25], ARIMA [26], LSTM [10], DCRNN [11], Graph WaveNet [17], ASTGCN [18], MTGNN [21], EARLIEST [12], RAIM [23], STEMO [13], ASTER [27], and Graph WaveNet with early exit (GWN-EE). GWN-EE uses Graph WaveNet as the forecasting backbone with a nodewise early-exit head. All methods use the same chronological split and preprocessing pipeline. The dataset-level actual used time is $100\%/MN\sum_{s=1}^{M}\sum_{i=1}^{N}(t_{s,i}^{*}+1)/T$, where M is the number of test samples and N is the number of nodes. At the 100% operating point, all methods use the full T-step input. Fixed-horizon methods use the first nine, six, and three steps at the 75%, 50%, and 25% target observation budgets, respectively. For each adaptive method and target observation budget, the timeliness-control hyperparameter is selected on the validation set to match the target within ± 2 percentage points; among feasible settings, the one with the lowest validation MAE is used for testing. For ESTMGCN, the selected ω_2 is shared by all nine random seeds.

Algorithm 1 Training Algorithm of ESTMGCN

Input: graph G, recorders; dynamic graph convolution module $f_d(\cdot)$, transform-attention module $f_a(\cdot)$, controller $f_c(\cdot)$
Parameter: learnable parameters Θ_d, Θ_a, Θ_c, and $b_{0:T-1}$
Output: learned model

repeat
  Initialize H_{-1} for the encoder, and randomly select a batch ($X \in \mathbb{R}^{B\times T\times N}, X_T \in \mathbb{R}^{B\times N}$) from the recorders
  for $t = 0, \ldots, T-1$ **do**
    $H_t = f_d(X_t, H_{t-1}, G; \Theta_d)$
    $H_{t:T-1}^{(t)} = f_a(X_t, H_{0:t}; \Theta_a)$
    Initialize the decoder input $\hat{X}_t^{(t)}$
    for $t' = t+1, \ldots, T$ **do**
      $\hat{X}_{t'}^{(t)} = \mathrm{ReLU}(W f_d(\hat{X}_{t'-1}^{(t)}, H_{t'-1}^{(t)}, G; \Theta_d) + b)$
    end for
    Obtain the terminal decoder representation $H_T^{(t)}$
    Construct the controller state $S_t = [H_T^{(t)} \| \tau_t]$
    Compute the decision probability $p_t = \sigma(W_c S_t + b_c)$
    Sample the raw action a_t independently with ϵ-greedy exploration
    Update the first-hit selector m_t using Eq. (18)
  end for
  $\hat{X}_T = \sum_{t=0}^{T-1} m_t \odot \hat{X}_T^{(t)}$
  Compute $L = L_p + \omega_1 L_c + \omega_2 L_e$ and the baseline loss L_b
  Compute the stochastic gradients of Θ_d, Θ_a, and Θ_c according to L
  Update Θ_d, Θ_a, and Θ_c; update $b_{0:T-1}$ using L_b
until the stopping criterion is met

[1] Available: https://data.cityofnewyork.us/Public-Safety/EMS-Incident-Dispatch-Data/76xm-jjuj
[2] Available: https://www.kaggle.com/datasets/mrmorj/new-york-city-police-crime-data-historic
[3] Available: https://english.mee.gov.cn/

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

C. Parameter Settings

The model is optimized with the Adam optimizer [28]. We provide the specific parameter settings used in the experiments to ensure reproducibility. These parameters are chosen based on preliminary experiments and existing literature in the field. The initial learning rate is set to 0.001. The layer depth of the dynamic graph convolution module is set to 1, the hidden-state size H is set to 32, the node-embedding dimension D is set to 40, the batch size B is set to 32, and the maximum observation window length T is set to 12. The fixed scaling coefficients β_1, β_2, β_3, and β_4 are set to 0.05, 0.95, 0.95, and 0.95, respectively. The decay hyperparameter κ is selected on the validation set to avoid an overly dense or overly sparse temporal-similarity graph. ω_1 is set to 1, and ω_2 is searched over $\{0.001, 0.005, 0.01, 0.02, 0.05, 0.075\}$. Early stopping is employed to avoid overfitting.

D. Experimental Results

In this section, we present the results of our experiments and discuss the performance of the ESTMGCN model in comparison to other baselines on four datasets. We also analyze the effects of target observation budgets, actual used time, and dynamic spatiotemporal information on prediction accuracy.

1) Forecasting Performance Comparison: Table IV reports the early forecasting results under four target observation budgets, while Table V reports the actual used time achieved by the adaptive methods. ESTMGCN yields the lowest MAE in all 16 dataset and target observation budget combinations and the best or highly competitive RMSE in most settings. This consistent performance on METR-LA, EMS, NYPD, and air quality demonstrates the strong generalization ability of ESTMGCN across heterogeneous spatiotemporal forecasting tasks.

As the target observation budget decreases from 100% to 25%, the performance of all methods degrades because less temporal information is available. However, ESTMGCN exhibits a noticeably more stable degradation trend than the competing methods. For example, on METR-LA, its MAE increases from 2.15 to 3.05, while Graph WaveNet and DCRNN increase from 2.38 to 4.35 and from 2.27 to 3.48, respectively. A similar trend is observed on air quality, where ESTMGCN consistently outperforms strong baselines, such as ASTER and STEMO, under all target observation budgets. These results indicate that ESTMGCN is more robust when only partial observations are available, which is particularly important for real-time early forecasting scenarios.

Compared with conventional forecasting models, including HA, ARIMA, LSTM, DCRNN, Graph WaveNet, ASTGCN, and MTGNN, ESTMGCN achieves consistently better accuracy, especially under limited target observation budgets. It also outperforms the adaptive baselines GWN-EE, EARLIEST, RAIM, STEMO, and ASTER in most cases. For instance, on NYPD with a 25% target observation budget, ESTMGCN attains an MAE of 1.41, which is lower than those of ASTER, STEMO, and MTGNN. These results verify the effectiveness of jointly modeling adaptive observation decisions and dynamic spatiotemporal dependencies.

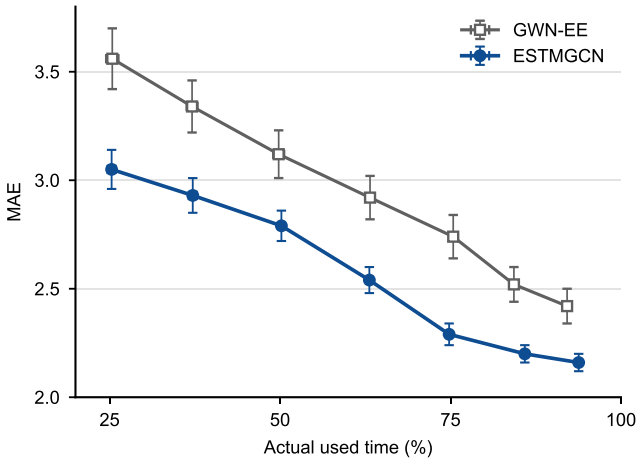


Fig. 4. Accuracy–timeliness operating points on METR-LA. Each point reports mean actual used time and MAE over nine runs, and error bars denote standard deviation. The operating points are obtained by varying ω_2 for ESTMGCN and δ for GWN-EE; they are distinct from the forced full-horizon 100% setting in Table IV.

In summary, the results in Table IV show that ESTMGCN not only delivers the best overall forecasting accuracy but also maintains stronger robustness under constrained target observation budgets, highlighting its practical value for early spatiotemporal forecasting in IoT applications.

2) Timeliness–Accuracy Tradeoff: Fig. 4 compares seven adaptive operating points. ESTMGCN achieves lower MAE than GWN-EE at every matched actual used time, with the gap remaining visible from approximately 25% to 94% observations.

The component-removal results are reported in Table VI. The paired significance tests are reported in Table VII.

3) Statistical Significance Analysis: The tests use paired one-sided t-tests over the same nine seeds. ESTMGCN is significantly better than GWN-EE and Graph WaveNet in all 16 settings after Holm correction. Against STEMO and ASTER, significance varies by dataset and budget, with nonsignificant differences concentrated on several NYPD and full-observation settings.

4) Transform-Attention Error Propagation Analysis: Fig. 5 shows that prediction errors generally increase from X_1 to X_{12} when only X_0 is observed at $t = 0$. Full ESTMGCN remains below both alternatives on METR-LA and NYPD, and its advantage is larger at later horizons. The result indicates that transform-attention reduces error accumulation rather than only improving the first prediction step.

5) Controller Stability and Convergence Analysis: Table VIII summarizes controller stability and convergence.

A run is counted as a failure when policy collapse or an overly early stopping policy prevents a stable validation optimum. The convergence epoch is the epoch of the checkpoint retained by validation-MAE early stopping. The default configuration converges successfully for all nine seeds. Removing the learned baseline produces two premature-stop failures and

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

TABLE IV

RESULTS ARE COMPARED UNDER FOUR TARGET OBSERVATION BUDGETS. THE ACTUAL USED TIME OF ADAPTIVE METHODS IS REPORTED SEPARATELY IN TABLE V. GWN-EE DENOTES GWN-EE. FORECASTING ERRORS FOR LEARNED MODELS ARE REPORTED AS MEAN ± STANDARD DEVIATION OVER NINE RUNS; "–" INDICATES THAT THE AVAILABLE PREFIX IS INSUFFICIENT TO FIT ARIMA

Dataset	Method	100%		75%		50%		25%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
METR-LA	HA	7.22	9.46	7.22	9.46	7.22	9.46	7.22	9.46
	ARIMA	4.12	8.73	8.36	20.88	11.98	16.20	–	–
	LSTM	4.99±0.07	7.48±0.07	7.17±0.11	10.21±0.10	7.50±0.13	10.46±0.11	7.72±0.15	10.68±0.17
	DCRNN	2.27±0.06	3.94±0.07	2.91±0.08	5.76±0.09	3.25±0.10	6.68±0.09	3.48±0.11	7.23±0.10
	Graph WaveNet	2.38±0.15	4.21±0.08	3.22±0.11	6.45±0.09	3.81±0.07	7.70±0.08	4.35±0.16	8.65±0.10
	ASTGCN	3.54±0.16	7.39±0.17	5.05±0.18	7.82±0.19	6.65±0.20	12.70±0.21	7.64±0.23	14.40±0.25
	MTGNN	2.24±0.06	3.90±0.07	2.82±0.08	5.57±0.09	3.15±0.09	6.42±0.11	3.36±0.12	6.93±0.13
	GWN-EE	2.38±0.14	4.21±0.09	2.74±0.10	5.14±0.11	3.12±0.11	6.27±0.13	3.56±0.14	7.02±0.16
	EARLIEST	5.01±0.08	7.46±0.09	5.93±0.10	8.21±0.10	6.14±0.11	9.26±0.10	6.85±0.12	10.06±0.11
	RAIM	4.37±0.11	7.81±0.14	6.12±0.15	8.34±0.17	8.09±0.18	10.17±0.20	9.74±0.22	12.83±0.25
	STEMO	2.23±0.07	3.96±0.06	2.37±0.08	4.45±0.08	3.23±0.08	5.69±0.09	3.49±0.09	6.15±0.09
	ASTER	2.19±0.05	3.88±0.07	2.33±0.06	4.21±0.08	2.92±0.08	5.28±0.09	3.17±0.09	5.94±0.11
	ESTMGCN	**2.15±0.04**	**3.81±0.05**	**2.29±0.05**	**4.05±0.06**	**2.79±0.07**	**5.17±0.08**	**3.05±0.09**	**5.83±0.10**
EMS	HA	1.47	2.41	1.47	2.41	1.47	2.41	1.47	**2.41**
	ARIMA	1.78	3.68	2.39	6.76	–	–	–	–
	LSTM	1.61±0.04	2.63±0.05	1.67±0.04	2.78±0.06	1.67±0.07	2.79±0.08	1.68±0.08	2.81±0.09
	DCRNN	1.45±0.03	2.60±0.03	1.69±0.04	3.19±0.05	1.73±0.06	3.34±0.07	1.78±0.07	3.52±0.09
	Graph WaveNet	1.41±0.04	2.53±0.06	1.47±0.05	2.57±0.06	1.51±0.05	2.68±0.07	1.56±0.06	2.81±0.08
	ASTGCN	1.49±0.04	**1.95±0.03**	1.78±0.04	2.49±0.05	1.85±0.05	2.84±0.06	2.01±0.07	3.12±0.08
	MTGNN	1.19±0.03	2.28±0.04	1.20±0.04	2.29±0.05	1.90±0.07	3.56±0.10	1.93±0.08	3.61±0.11
	GWN-EE	1.41±0.05	2.53±0.07	1.43±0.05	2.55±0.07	1.47±0.06	2.63±0.08	1.52±0.07	2.74±0.09
	EARLIEST	1.62±0.05	2.63±0.06	1.64±0.05	2.67±0.06	1.66±0.06	2.71±0.06	1.68±0.06	2.79±0.07
	RAIM	1.56±0.04	2.45±0.05	1.61±0.05	2.50±0.06	1.64±0.06	2.63±0.07	1.72±0.07	2.71±0.08
	STEMO	1.29±0.02	2.23±0.03	1.32±0.03	2.32±0.04	1.40±0.03	2.41±0.03	1.44±0.03	2.49±0.05
	ASTER	1.20±0.03	2.18±0.04	1.22±0.03	2.26±0.04	1.24±0.04	2.28±0.05	1.26±0.04	2.55±0.06
	ESTMGCN	**1.17±0.02**	2.12±0.03	**1.18±0.02**	**2.19±0.03**	**1.20±0.03**	**2.20±0.04**	**1.20±0.04**	2.47±0.05
NYPD	HA	1.54	2.00	1.54	2.00	1.54	2.00	1.54	2.00
	ARIMA	2.06	3.11	3.64	6.77	–	–	–	–
	LSTM	1.52±0.04	2.16±0.04	1.61±0.05	2.29±0.06	1.73±0.07	2.40±0.07	1.75±0.08	2.40±0.09
	DCRNN	1.58±0.04	2.16±0.04	1.60±0.05	2.21±0.05	1.61±0.05	2.22±0.06	1.66±0.06	2.32±0.07
	Graph WaveNet	1.48±0.08	2.05±0.11	1.50±0.06	2.08±0.11	1.54±0.15	2.14±0.05	1.58±0.08	2.18±0.02
	ASTGCN	2.18±0.06	3.08±0.12	2.88±0.07	3.31±0.13	2.99±0.08	3.42±0.13	3.69±0.09	4.09±0.14
	MTGNN	1.36±0.04	**1.77±0.05**	1.71±0.05	2.40±0.06	1.72±0.06	2.40±0.07	1.72±0.06	2.40±0.08
	GWN-EE	1.48±0.07	2.05±0.09	1.49±0.07	2.07±0.09	1.52±0.08	2.11±0.10	1.56±0.09	2.16±0.11
	EARLIEST	1.47±0.05	2.17±0.05	1.59±0.06	2.23±0.07	1.64±0.07	2.36±0.07	1.65±0.08	2.39±0.09
	RAIM	2.42±0.07	3.35±0.09	3.24±0.09	4.58±0.11	3.66±0.11	5.39±0.13	4.48±0.13	6.03±0.15
	STEMO	1.36±0.03	1.82±0.05	1.38±0.04	1.97±0.06	1.45±0.06	2.12±0.07	1.49±0.07	2.27±0.09
	ASTER	1.37±0.03	1.83±0.04	1.40±0.04	1.87±0.04	1.42±0.04	1.93±0.05	1.46±0.05	1.97±0.06
	ESTMGCN	**1.33±0.02**	1.78±0.03	**1.35±0.03**	**1.82±0.03**	**1.37±0.03**	**1.85±0.04**	**1.41±0.04**	**1.91±0.05**
Air Quality	HA	6.89	8.73	6.89	8.73	6.89	8.73	6.89	8.73
	ARIMA	6.15	8.87	9.41	13.18	–	–	–	–
	LSTM	6.07±0.10	8.33±0.13	6.31±0.12	8.58±0.14	6.62±0.13	9.05±0.15	6.88±0.15	9.41±0.17
	DCRNN	5.49±0.08	7.48±0.10	5.61±0.08	7.73±0.11	5.89±0.10	8.03±0.12	6.03±0.11	8.24±0.13
	Graph WaveNet	5.32±0.09	7.28±0.11	5.44±0.09	7.36±0.10	5.45±0.10	7.41±0.12	5.52±0.11	7.54±0.13
	ASTGCN	5.53±0.10	7.57±0.13	5.63±0.10	7.61±0.12	5.84±0.12	7.76±0.14	5.84±0.13	7.78±0.15
	MTGNN	5.21±0.07	7.11±0.09	5.38±0.08	7.29±0.10	5.56±0.09	7.58±0.11	5.77±0.10	7.87±0.12
	GWN-EE	5.32±0.09	7.28±0.11	5.35±0.09	7.32±0.11	5.40±0.10	7.39±0.12	5.48±0.11	7.49±0.13
	EARLIEST	5.93±0.10	7.99±0.12	6.11±0.11	8.17±0.13	6.35±0.12	8.56±0.14	6.62±0.14	8.88±0.16
	RAIM	6.72±0.09	8.56±0.11	6.74±0.10	8.65±0.12	6.88±0.11	8.74±0.13	6.96±0.12	8.77±0.14
	STEMO	4.91±0.06	6.72±0.08	5.07±0.06	6.83±0.08	5.28±0.07	7.11±0.09	5.41±0.08	7.29±0.10
	ASTER	4.76±0.05	6.52±0.07	4.88±0.06	6.64±0.07	5.03±0.06	6.83±0.08	5.19±0.07	7.11±0.09
	ESTMGCN	**4.57±0.04**	**6.38±0.06**	**4.71±0.05**	**6.57±0.06**	**4.93±0.06**	**6.82±0.07**	**5.14±0.07**	**7.06±0.08**

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

TABLE V

ACTUAL USED TIME OF ADAPTIVE METHODS UNDER FOUR TARGET OBSERVATION BUDGETS. VALUES ARE PERCENTAGES REPORTED AS MEAN ± STANDARD DEVIATION OVER NINE RUNS; 100% DENOTES THE FULL OBSERVATION HORIZON

Dataset	Method	100%	75%	50%	25%
METR-LA	GWN-EE	100%	75.4±0.6%	49.8±0.8%	25.4±0.9%
	EARLIEST	100%	74.6±0.9%	50.7±1.1%	25.6±1.2%
	RAIM	100%	75.7±1.1%	49.4±1.3%	25.2±1.4%
	STEMO	100%	74.9±0.6%	50.3±0.7%	24.8±0.8%
	ASTER	100%	75.2±0.5%	49.7±0.6%	25.1±0.7%
	ESTMGCN	100%	74.8±0.4%	50.2±0.5%	25.3±0.6%
EMS	GWN-EE	100%	75.6±0.5%	50.4±0.6%	25.2±0.7%
	EARLIEST	100%	74.7±0.8%	50.8±0.9%	25.4±1.0%
	RAIM	100%	75.3±0.9%	49.6±1.0%	24.7±1.1%
	STEMO	100%	75.1±0.5%	50.5±0.6%	24.9±0.7%
	ASTER	100%	74.8±0.4%	49.9±0.5%	25.2±0.6%
	ESTMGCN	100%	75.4±0.3%	50.1±0.4%	25.5±0.5%
NYPD	GWN-EE	100%	75.5±0.6%	49.7±0.8%	25.4±0.9%
	EARLIEST	100%	74.5±0.9%	50.6±1.0%	24.8±1.1%
	RAIM	100%	75.8±1.0%	49.5±1.2%	25.7±1.3%
	STEMO	100%	75.2±0.6%	50.2±0.7%	24.7±0.8%
	ASTER	100%	74.9±0.5%	49.8±0.6%	25.1±0.7%
	ESTMGCN	100%	75.1±0.4%	50.4±0.5%	25.2±0.6%
Air Quality	GWN-EE	100%	75.2±0.5%	50.3±0.6%	24.9±0.7%
	EARLIEST	100%	74.7±0.8%	50.5±0.9%	25.6±1.0%
	RAIM	100%	75.6±0.9%	49.7±1.0%	25.4±1.1%
	STEMO	100%	75.0±0.5%	50.1±0.6%	24.8±0.7%
	ASTER	100%	74.9±0.4%	49.9±0.5%	25.2±0.6%
	ESTMGCN	100%	75.3±0.4%	50.2±0.5%	25.1±0.6%

TABLE VI

ABLATION STUDY ON THE NYPD DATASET UNDER FOUR TARGET OBSERVATION BUDGETS. KEY COMPONENTS, INCLUDING THE DYNAMIC ADJACENCY MATRIX, TRANSFORM-ATTENTION MODULE, SPATIAL EMBEDDING, TEMPORAL EMBEDDING, AND CONTROLLER, ARE REMOVED FROM ESTMGCN TO EXAMINE THEIR INDIVIDUAL CONTRIBUTIONS. THE RESULTING VARIANTS ARE EVALUATED IN TERMS OF MAE AND RMSE

Variant	100%		75%		50%		25%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/o DA	1.45	2.02	1.53	2.24	1.67	2.98	1.73	3.03
w/o TA	1.38	1.84	1.40	1.92	1.52	1.94	1.61	2.03
w/o SE	1.38	1.83	1.39	1.92	1.52	1.93	1.60	2.09
w/o TE	1.37	1.79	1.37	1.86	1.39	1.93	**1.41**	1.99
w/o C	**1.33**	**1.78**	1.41	1.95	1.57	2.02	1.68	2.14
w/o (DA + TA)	1.52	2.13	1.64	2.41	1.79	3.17	1.87	3.29
w/o (DA + C)	1.51	2.08	1.66	2.45	1.84	3.31	1.93	3.44
w/o (TA + C)	1.44	1.93	1.50	2.07	1.65	2.14	1.78	2.31
w/o (DA + TA + C)	1.63	2.31	1.76	2.57	1.97	3.41	2.08	3.58
ESTMGCN	**1.33**	**1.78**	**1.35**	**1.82**	**1.37**	**1.85**	**1.41**	**1.91**

the largest MAE, while faster exploration decay causes one premature convergence failure. Halving the earliness weight slightly lowers MAE but increases actual used time to 56.7%, illustrating the intended accuracy–timeliness tradeoff.

6) Ablation Study: To evaluate the contribution of each component in ESTMGCN, we conduct an ablation study on the NYPD dataset, as shown in Table VI. Specifically, we remove the dynamic adjacency matrix (w/o DA), transform-attention module (w/o TA), spatial embedding (w/o SE), temporal embedding (w/o TE), and controller (w/o C), as well as several combinations of these components.

The results show that removing any component degrades the forecasting performance to different degrees, which verifies the effectiveness of the overall design. Among the single-component variants, **w/o DA** causes the most significant performance drop under all target observation budgets. For example, its MAE increases from 1.33 to 1.45, 1.35 to 1.53, 1.37 to 1.67, and 1.41 to 1.73 under the 100%, 75%, 50%, and

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

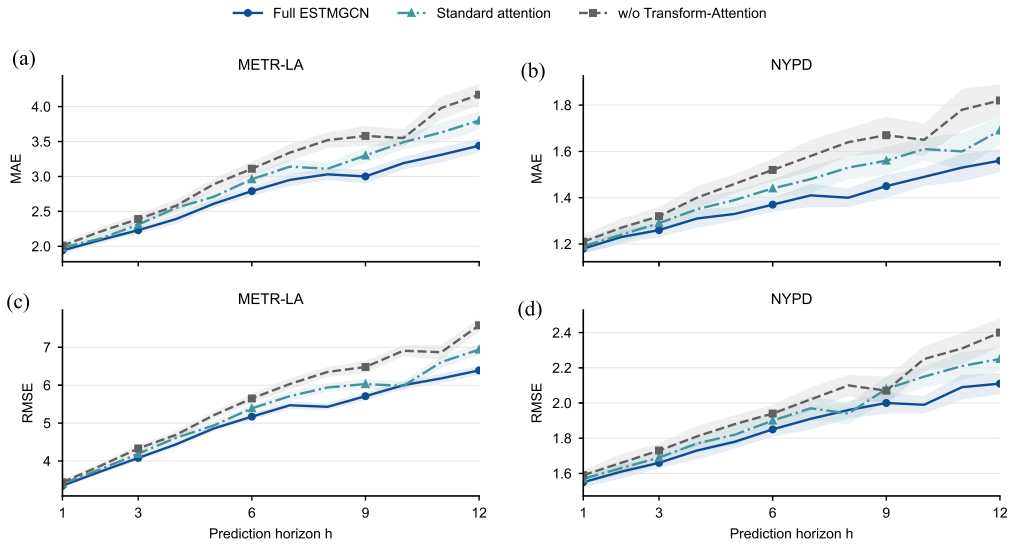


Fig. 5. Prediction error across the 12-step autoregressive rollout when only X_0 is observed at $t = 0$. The decoder recursively predicts $X_1, \ldots, X_{12}$, and h denotes the rollout horizon. (a) and (b) MAE on METR-LA and NYPD. (c) and (d) RMSE on METR-LA and NYPD. Lines and shaded bands denote mean and standard deviation over nine runs. The standard-attention variant replaces only the transform-attention bridge; all remaining modules and training settings are unchanged.

TABLE VII

STATISTICAL SIGNIFICANCE OF ESTMGCN AGAINST FOUR BASELINES USING MAE. TESTS ARE PAIRED ONE-SIDED T-TESTS OVER THE SAME NINE RANDOM SEEDS, WITH THE ALTERNATIVE THAT ESTMGCN HAS LOWER MAE. HOLM CORRECTION IS APPLIED TO THE FOUR BASELINE COMPARISONS WITHIN EACH DATASET AND TARGET OBSERVATION BUDGET. "*," "**," AND "***" DENOTE ADJUSTED $p < 0.05$, $p < 0.01$, AND $p < 0.001$, RESPECTIVELY; "–" INDICATES NO SIGNIFICANT DIFFERENCE

Dataset	Budget	GWN-EE	Graph WaveNet	STEMO	ASTER
METR-LA	100%	***	***	*	–
	75%	***	***	*	*
	50%	***	***	***	***
	25%	***	***	***	**
EMS	100%	***	***	**	–
	75%	***	***	**	*
	50%	***	***	***	**
	25%	***	***	***	**
NYPD	100%	***	***	–	*
	75%	***	***	*	**
	50%	**	**	–	–
	25%	***	***	*	–
Air Quality	100%	***	***	***	**
	75%	***	***	***	**
	50%	***	***	***	*
	25%	***	***	**	–

25% target budgets, respectively, indicating the importance of modeling dynamic spatial dependencies.

In addition, removing the transform-attention module or the spatial/temporal embeddings also consistently degrades the results, showing that these components provide useful spatiotemporal context for prediction. The effect of the controller is relatively limited under the 100% target observation budget, where sufficient observations are already available, but becomes much more evident as the target observation budget decreases. For instance, under the 25% target observation budget, removing the controller increases the RMSE from 1.91 to 2.14, demonstrating its importance for early forecasting.

Moreover, multicomponent removal leads to more severe degradation than single-component removal. In particular, **w/o (DA + TA + C)** achieves the worst performance under all settings, which further confirms that these components are complementary. Overall, the ablation results validate that each module contributes to ESTMGCN, while the dynamic adjacency matrix and controller are particularly important for robust early spatiotemporal forecasting.

7) Hyperparameter Study: To investigate the influence of hyperparameters on the performance of ESTMGCN, we conducted a study focusing on three core hyperparameters: the dimension of the hidden state, the weight parameter ω_1, and the weight parameter ω_2. The hidden-state dimension was varied between 16 and 64, ω_1 was varied over $\{0.1, 0.5, 1.0, 2.0, 5.0\}$ with ω_2 fixed at 0.01, and ω_2 ranged from 0 to 0.1 with ω_1 fixed at 1.

Fig. 6 presents the results of the hyperparameter study on the NYPD dataset. Increasing the dimension of the hidden state enhances the representation ability of ESTMGCN and generally reduces MAE, although excessively large dimensions may negatively affect performance. Regarding ω_1, the MAE exhibits a U-shaped curve with a minimum around $\omega_1 = 1$, while the actual used time increases monotonically. A too-small ω_1 weakens controller optimization and leads to premature decisions, whereas a too-large ω_1 destabilizes training due to high-variance gradients. These results justify fixing $\omega_1 = 1$ as the default.

Regarding ω_2, increasing its value reduces the amount of observed data and improves timeliness, but an excessively large ω_2 may hurt prediction accuracy because the model tends to favor overly early decisions. These observations are

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

TABLE VIII

CONTROLLER STABILITY ON NYPD AT THE 50% TARGET OBSERVATION BUDGET OVER NINE RANDOM SEEDS. MEAN ± STANDARD DEVIATION VALUES SUMMARIZE SUCCESSFUL RUNS. REWARD SCALE MULTIPLIES THE REWARD BY 0.5; FASTER EXPLORATION DECAY DOUBLES THE ϵ DECAY RATE WHILE KEEPING ITS ENDPOINTS UNCHANGED; HALF EARLINESS WEIGHT USES 0.5 TIMES THE DEFAULT EARLINESS WEIGHT

Setting	Success/failure	Convergence epoch	MAE	Actual used time
Default	9/0	72.4±8.7	1.37±0.03	50.4±0.5%
Reward scale ×0.5	9/0	80.7±11.3	1.40±0.04	48.9±0.9%
No learned baseline	7/2	88.6±13.9	1.44±0.07	47.6±2.3%
Faster exploration decay	8/1	65.8±10.6	1.41±0.05	48.4±1.5%
Half earliness weight	9/0	70.1±9.4	1.35±0.03	56.7±1.2%

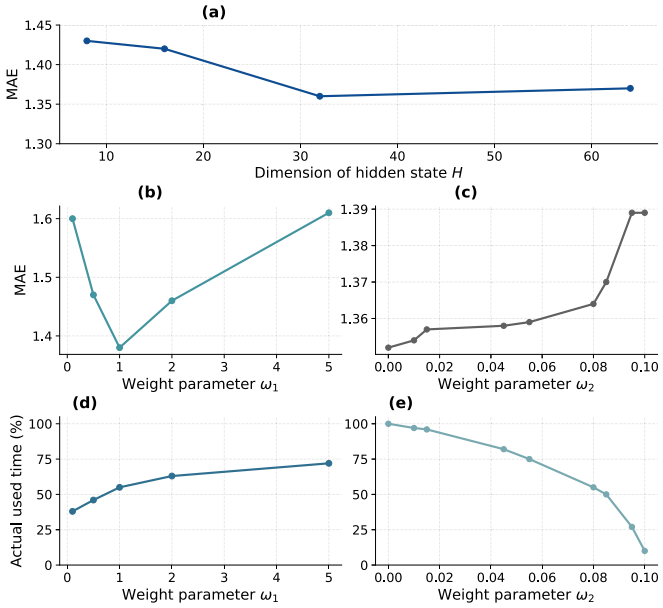


Fig. 6. Experimental results of the hyperparameter study on the NYPD dataset. The figure illustrates the effects of the hidden state dimension and the weight parameters ω_1 and ω_2 on prediction accuracy and timeliness. (a) Hidden state dimension versus MAE. (b) ω_1 versus MAE. (c) ω_2 versus MAE. (d) ω_1 versus actual used time %. (e) ω_2 versus actual used time %.

consistent with the roles of ω_1 and ω_2 in (24), where ω_1 governs controller optimization and ω_2 governs the earliness preference.

E. Computational Complexity and Runtime Analysis

We briefly analyze the efficiency of ESTMGCN from both theoretical and empirical perspectives. Let B, N, T, H, and D denote the batch size, number of nodes, observation horizon, hidden dimension, and embedding dimension, respectively. Decoder-side rollout, transform-attention, and exact online DTW require $\mathcal{O}(BT^2N^2H)$, $\mathcal{O}(BNDT^3)$, and $\mathcal{O}(BN^2T^2)$ time, respectively. The overall time complexity is $\mathcal{O}(BT^2N^2H + BNDT^3)$, and the space complexity is $\mathcal{O}(BTNH + BN^2T^2)$ under dense graph storage.

Batch-one inference latency and memory were benchmarked on an NVIDIA GeForce RTX 5090 in FP32. Table IX reports both the METR-LA module breakdown and ESTMGCN scaling with the number of nodes and observation steps.

On METR-LA, ESTMGCN requires 5.89 ± 0.21 ms per sample with a p95 latency of 6.43 ms, including 2.24 ± 0.09 ms for graph/DTW construction. Scaling N from 100 to 1000 increases total latency from 2.43 to 81.37 ms and peak memory from 61.8 to 897.4 MB; increasing T from 6 to 24 at $N = 207$ increases latency from 2.27 to 19.42 ms. The METR-LA p95 latency is small relative to its 5-min sampling interval, while the dense quadratic node scaling remains a limitation for much larger networks. The reported implementation recomputes exact prefix-DTW online using one dynamic-programming table per node pair and stores dense graphs. Top-k graph sparsification and cached or approximate DTW provide direct routes to lower large-network cost.

F. Interpretability of the Prediction-Time Controller

To improve interpretability, we analyze the behavior of the prediction-time controller by visualizing the nodewise decision probability over time. At each observation step t, the controller outputs a decision probability p_t^i for node i, which reflects the confidence that the currently observed information is sufficient for selecting the prediction generated at time t.

Fig. 7(a) and (b) shows two representative nodes. For the early-decision case, the controller quickly increases its confidence and assigns a high decision probability at observation step 3, with an actual used time of 25% and a low prediction error (MAE = 1.868). For the late-decision case, the controller keeps a low confidence in the early stages and assigns higher decision probability at observation step 9, with an actual used time of 75% (MAE = 4.645). This indicates that the controller adaptively adjusts prediction-time selection according to the difficulty of node dynamics.

To further examine the overall trend, we divide nodes into low-error and high-error groups according to the median of nodewise MAE on the test set. As shown in Fig. 7(c), the low-error group reaches high decision confidence earlier, while the high-error group shows slower confidence accumulation and larger uncertainty in the middle stages.

The selected-count distributions and controller correlations are summarized in Table X.

Across the four datasets, the middle interval contains the largest share of selected counts, while each dataset retains substantial early and late decisions. The selected counts are positively correlated with absolute error (ρ = 0.27–0.36) and sequence volatility (ρ = 0.21–0.34), indicating that more

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

TABLE IX

BATCH-ONE INFERENCE LATENCY AND SCALING ON AN NVIDIA GEFORCE RTX 5090 IN FP32. MEASUREMENTS USE 20 WARM-UP RUNS AND 100 TIMED RUNS. GRAPH/DTW INCLUDES GRAPH CONSTRUCTION AND EXACT ONLINE DTW WHERE APPLICABLE. VALUES WITH ± ARE MEAN ± STANDARD DEVIATION IN MILLISECONDS; P95 IS IN MILLISECONDS AND PEAK MEMORY IS IN MB. ALL SETTINGS USE $H = 32$ WITHOUT DTW CACHING OR APPROXIMATION

(a)

Method	Graph/DTW	Backbone	Stop head	Total	p95	Peak memory
ESTMGCN	2.24 ± 0.09	3.47 ± 0.13	0.18 ± 0.02	5.89 ± 0.21	6.43	83.9
GWN-EE	0.42 ± 0.03	2.07 ± 0.09	0.14 ± 0.01	2.69 ± 0.12	2.96	66.9
Graph WaveNet	0.41 ± 0.03	2.05 ± 0.08	–	2.50 ± 0.11	2.76	65.4
DCRNN	0.36 ± 0.02	5.72 ± 0.18	–	6.18 ± 0.20	6.69	59.8
MTGNN	0.47 ± 0.03	2.84 ± 0.11	–	3.38 ± 0.14	3.71	70.6

(b)

Setting	N	T	Graph/DTW	Prediction	Total	p95	Peak memory
$N = 100$	100	12	0.74 ± 0.04	1.65 ± 0.07	2.43 ± 0.09	2.68	61.8
METR-LA	207	12	2.24 ± 0.09	3.47 ± 0.13	5.89 ± 0.21	6.43	83.9
$N = 500$	500	12	10.96 ± 0.38	11.72 ± 0.41	23.14 ± 0.72	25.01	257.6
$N = 1000$	1000	12	43.85 ± 1.41	36.62 ± 1.18	81.37 ± 2.35	87.96	897.4
$T = 6$	207	6	0.81 ± 0.04	1.38 ± 0.07	2.27 ± 0.09	2.51	71.2
$T = 24$	207	24	7.91 ± 0.24	11.08 ± 0.36	19.42 ± 0.58	21.17	164.7

TABLE X

SELECTED OBSERVATION-COUNT DISTRIBUTION AND CONTROLLER CORRELATIONS AT THE 50% TARGET OBSERVATION BUDGET. THE SELECTED COUNT IS $t_i^* + 1$; EARLY, MIDDLE, AND LATE CORRESPOND TO COUNTS 1–4, 5–8, AND 9–12, RESPECTIVELY. VALUES ARE MEAN ± STANDARD DEVIATION OVER NINE RUNS. SPEARMAN CORRELATIONS COMPARE THE SELECTED COUNT WITH ABSOLUTE PREDICTION ERROR AND SEQUENCE VOLATILITY ACROSS TEST NODE INSTANCES; ALL REPORTED CORRELATIONS HAVE $p < 5 \times 10^{-5}$

Dataset	Mean count	Early (%)	Middle (%)	Late (%)	ρ(error)	ρ(volatility)
METR-LA	6.02 ± 0.06	30.1 ± 1.3	49.7 ± 1.6	20.2 ± 1.1	0.36 ± 0.05	0.31 ± 0.06
EMS	6.01 ± 0.05	34.0 ± 1.5	42.9 ± 1.8	23.1 ± 1.3	0.28 ± 0.06	0.21 ± 0.08
NYPD	6.05 ± 0.06	35.1 ± 1.8	39.5 ± 2.0	25.4 ± 1.6	0.33 ± 0.07	0.26 ± 0.08
Air Quality	6.02 ± 0.06	27.6 ± 1.2	53.6 ± 1.5	18.8 ± 1.0	0.27 ± 0.06	0.34 ± 0.05

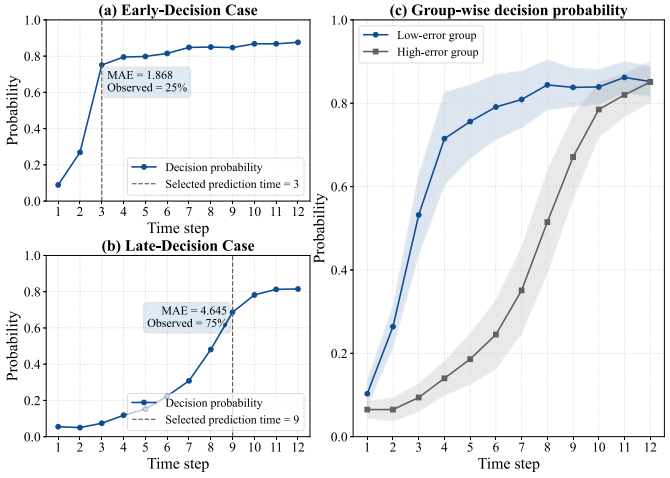


Fig. 7. Case study of the prediction-time controller. (a) Decision probability of a representative early-decision node, whose prediction is selected at observation step 3 with an actual used time of 25% (MAE = 1.868). (b) Decision probability of a representative late-decision node, whose prediction is selected at observation step 9 with an actual used time of 75% (MAE = 4.645). (c) Average decision probability of low-error and high-error node groups with standard-deviation bands. The low-error group reaches high confidence earlier, while the high-error group shows slower confidence accumulation and larger midstage uncertainty.

difficult or volatile node instances generally require later predictions.

VI. DISCUSSION

The results show that ESTMGCN improves early spatiotemporal forecasting by better balancing prediction accuracy and prediction timing. Its advantage mainly comes from nodewise adaptive prediction timing, which allows different nodes to make predictions at different times according to their uncertainty evolution, making it well suited to heterogeneous IoT spatiotemporal systems. The results also highlight the importance of dependency modeling under partial observations. By integrating static spatial structure, dynamic internode correlations, and temporal similarity, ESTMGCN preserves informative spatiotemporal dependencies even when observations are incomplete, while the ablation study further confirms the critical role of the dynamic graph component. Moreover, ESTMGCN degrades more gracefully as the target observation budget decreases, indicating stronger robustness in low-observation settings common in practical IoT applications.

VII. CONCLUSION

This article proposed ESTMGCN for early spatiotemporal forecasting in IoT environments. By combining a dynamic multigraph encoder, a transform-attention decoder bridge, and an adaptive prediction-time controller, the proposed method enables nodewise forecasting while preserving forecasting accuracy under partial observations. Experiments on four

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.

real-world datasets demonstrate that ESTMGCN consistently outperforms strong baselines across different target observation budgets. These results indicate that adaptive early forecasting is a promising direction for IoT applications requiring timely and reliable decision-making.

ACKNOWLEDGMENT

The experiments were performed at the School of Telecommunications Engineering, Xidian University.

REFERENCES

[1] F. Li et al., "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 1, pp. 1–21, Feb. 2023.

[2] W. Shao et al., "Long-term spatio-temporal forecasting via dynamic multiple-graph attention," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 2225–2232.

[3] W. Shao et al., "Transferrable contextual feature clusters for parking occupancy prediction," *Pervas. Mobile Comput.*, vol. 97, Jan. 2024, Art. no. 101831. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574119223000895

[4] M. B. Joseph et al., "Spatiotemporal prediction of wildfire size extremes with Bayesian finite sample maxima," *Ecological Appl.*, vol. 29, no. 6, p. 01898, Sep. 2019.

[5] L. Xia et al., "Spatial-temporal sequential hypergraph network for crime prediction with dynamic multiplex relation learning," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1631–1637.

[6] Y. Kang, M. S. Rahaman, Y. Ren, M. Sanderson, R. W. White, and F. D. Salim, "App usage on-the-move: Context- and commute-aware next app prediction," *Pervas. Mobile Comput.*, vol. 87, Dec. 2022, Art. no. 101704.

[7] D. Li, J. Chen, R. Huang, Z. Chen, and W. Li, "Sensor-aware CapsNet: Towards trustworthy multisensory fusion for remaining useful life prediction," *J. Manuf. Syst.*, vol. 72, pp. 26–37, Feb. 2024.

[8] R. Huang, Y. Chen, and C. Liu, "Revealing state-feature dependencies in dynamic degradation: An exofeature-aware transformer for battery state-of-health prediction," *IEEE Trans. Consum. Electron.*, vol. 72, no. 1, pp. 719–730, Feb. 2026.

[9] T. Hartvigsen, C. Sen, X. Kong, and E. Rundensteiner, "Recurrent halting chain for early multi-label classification," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1382–1392.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[11] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[12] T. Hartvigsen, C. Sen, X. Kong, and E. Rundensteiner, "Adaptive-halting policy network for early classification," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 101–110.

[13] W. Shao et al., "STEMO: Early spatio-temporal forecasting with multi-objective reinforcement learning," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2024, pp. 2618–2627.

[14] R. Castro, Y. M. Souto, E. Ogasawara, F. Porto, and E. Bezerra, "STConvS2S: Spatiotemporal convolutional sequence to sequence network for weather forecasting," *Neurocomputing*, vol. 426, pp. 285–298, Feb. 2021.

[15] A. Ermagun and D. Levinson, "Spatiotemporal traffic forecasting: Review and proposed directions," *Transp. Rev.*, vol. 38, no. 6, pp. 1–29, Nov. 2018.

[16] D. Zhu, Y. Yang, F. Ren, S. Murai, C. Cheng, and M. Huang, "Novel intelligent spatiotemporal grid earthquake early-warning model," *Remote Sens.*, vol. 13, no. 17, p. 3426, Aug. 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/17/3426

[17] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1907–1913.

[18] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.

[19] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1234–1241.

[20] L. Bai, L. Yao, C. Li, and X. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17804–17815.

[21] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 753–763.

[22] P. Schäfer and U. Leser, "TEASER: Early and accurate time series classification," *Data Mining Knowl. Discovery*, vol. 34, no. 5, pp. 1336–1362, Sep. 2020.

[23] Y. Xu, S. Biswal, S. R. Deshpande, K. O. Maher, and J. Sun, "RAIM: Recurrent attentive and intensive model of multimodal patient monitoring data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2565–2573.

[24] S. Wang, Y. Li, J. Zhang, Q. Meng, L. Meng, and F. Gao, "PM2.5-GNN: A domain knowledge enhanced graph neural network for PM2.5 forecasting," in *Proc. 28th Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2020, pp. 163–166, doi: 10.1145/3397536.3422208.

[25] J. Liu and W. Guan, "A summary of traffic flow forecasting methods," *J. Highway Transp. Res. Develop.*, vol. 3, pp. 82–85, Apr. 2004.

[26] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box–Jenkins techniques," Transportation Research Board, National Research Council, Washington, DC, USA, Transportation Research Record 722, pp. 1–9, 1979.

[27] S. Chen, W. Shao, F. D. Salim, and H. Xue, "ASTER: Adaptive spatio-temporal early decision model for dynamic resource allocation," 2025, *arXiv:2506.17929*.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.



Xiao Xiao (Member, IEEE) received the bachelor's degree in control technology and instrumentation from Xidian University, Xi'an, China, in 2004, and the Ph.D. degree in measuring and testing technologies and instruments from Zhejiang University, Hangzhou, China, in 2009.

In 2019, he joined the Min H. Kao Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, Knoxville, USA, as a Research Fellow. He is currently an Associate Professor with the School of Telecommunications Engineering, Xidian University. His research interests include wireless network architecture, intelligent transportation systems, computer vision, and deep learning.



Liwen Wang received the B.S. degree in artificial intelligence from Chongqing Jiaotong University, Chongqing, China, in 2024. She is currently pursuing the M.S. degree in the next-generation electronic information technology with Xidian University, Xi'an, China.

Her research interests include graph neural networks, deep learning, spatiotemporal forecasting, and intelligent transportation systems.



Ziyan Peng received the master's degree in communication engineering from Xidian University, Xi'an, China, in 2024.

Her research interests include graph neural networks, deep learning, spatiotemporal prediction, and intelligent transportation systems.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.



Zhisheng Yin (Member, IEEE) received the B.E. degree from Wuhan Institute of Technology, Wuhan, China, in 2012, the B.B.A. degree from Zhongnan University of Economics and Law, Wuhan, China, in 2012, the M.Sc. degree from the Civil Aviation University of China, Tianjin, China, in 2016, and the Ph.D. degree from the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China, in 2020.

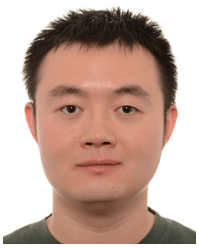
From September 2018 to September 2019, he visited the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor with the School of Telecommunications Engineering, Xidian University, Xi'an, China. His research interests include space-air-ground integrated networks, wireless communications, cyber twin, and physical-layer security.

Dr. Yin is an Associate Editor of IEEE INTERNET OF THINGS JOURNAL.



Yuan Zhang received the M.Eng. degree from the University of Shanghai for Science and Technology, Shanghai, Anhui, China, in September 1993.

Her research interests include the overall design of weapon systems and model-based systems engineering (MBSE).



Xuzhi He received the M.S. degree in electrical and computer engineering and the Ph.D. degree in biomedical engineering from the University of California, Davis, Davis, CA, USA, in 2020 and 2026, respectively.

His doctoral research focused on the development of advanced detector technologies for ultrafast positron emission tomography (PET), with an emphasis on Monte Carlo simulation of radiation and optical photon transport, depth-of-interaction-based timing correction, and photonic-crystal-enhanced scintillation detectors. His research interests include medical imaging, radiation detector physics, Monte Carlo simulation, photonic crystals, computational optics, and optical system design.



Zhiqiang Li received the B.S. degree from the College of Mathematics and Information Science, Hebei Normal University, Shijiazhuang, China, in 2004, the M.S. degree from the Department of Mathematics, Beijing Institute of Technology (BIT), Beijing, China, in 2006, and the Doctor of Engineering degree from BIT, in 2014.

He joined the Beijing Electro-Mechanical Engineering Institute, BIT. His current research interests include signal processing and pattern recognition.



Guoqiang Mao (Fellow, IEEE) has authored or co-authored more than 200 papers in international conferences and journals, which have been cited more than 9000 times. His research interests include intelligent transport systems, applied graph theory and its applications in telecommunications, the Internet of Things, wireless sensor networks, wireless localization techniques, and network modeling and performance analysis.

Dr. Mao received the Top Editor Award for outstanding contributions to IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY in 2011, 2014, and 2015. He was the Co-Chair of the IEEE Intelligent Transport Systems Society Technical Committee on Communication Networks. He has served as an Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS since 2018, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2014 to 2019, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY since 2010.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:56 UTC from IEEE Xplore. Restrictions apply.