NOT ALL EVENTS ARE EQUAL: CAPTURING HETEROGENEOUS IMPACT OF APERIODIC EVENTS IN LONG-TERM SPATIO-TEMPORAL TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Aperiodic events, such as traffic incidents and extreme weather, induce irregular deviations from recurring spatio-temporal patterns. Accurately forecasting these deviations is particularly challenging over long horizons. Existing methods typically approach this challenge by decomposing signals into regular and aperiodic components, but usually process all aperiodic components with an event-invariant parameterization. This overlooks the fact that the impact of an aperiodic event is heterogeneous, depending on when and where the event occurs and what type it is. A event-invariant model, agnostic to this heterogeneity, cannot adequately capture it. To address this challenge, we propose **HEAM** (**H**eterogeneous **E**vent-**A**daptive **M**odulation), a plug-and-play module that can be integrated into existing forecasting backbones to achieve event-adaptive modeling of the aperiodic component instead of event-invariant modeling. Specifically, we propose a *Decompose-Situate-Identify* pipeline, that generates modulation vectors from a library of modulator pools, conditioned on the spatio-temporal context and the latent event type of the aperiodic component. These vectors then modulate the aperiodic representation, so that the heterogeneous impact is explicitly captured. Experiments on datasets spanning multiple domains show that HEAM improves strong backbones of different architectural families, on average across all datasets and horizons. The gains are largest at the longest forecast horizon and in event-dense periods, particularly where the modeling of the heterogeneous impact of aperiodic events matters most.

1 INTRODUCTION

Long-term spatio-temporal time series forecasting supports a wide range of real-world applications, including traffic management and renewable energy scheduling. Its practical importance is particularly pronounced during aperiodic events, such as traffic incidents and extreme weather, which can abruptly disrupt the normal evolution of spatio-temporal systems. Thus, accurately forecast the deviation induced by such events in long-term forecasting is therefore a critical problem.

However, predicting these deviations, which we refer to as aperiodic component, is inherently challenging. The difficulty is that an event's impact varies with when and where it occurs and what type it is, which means the distribution of aperiodic components varies across different time and location. We term this phenomenon the **heterogeneous impact** of aperiodic events. To demonstrate this, we conduct an empirical analysis on PEMS08, a real world traffic flow dataset (Figure 1). As shown in Figure 1a, 1b, at sensor 67, two comparable surges of traffic flow take place in different day of week, but followed by totally different future variations. This reveals a **temporal heterogeneity of event impacts**. As shown in Figure 1a,b, at sensor 67, two comparable drops in traffic flow occurred at different times, followed by totally different future variations. This reveals a **temporal heterogeneity of event impacts**. A similar contrast emerges across space. We visualize the conditional distribution of future deviation amplitudes given past deviations at two sensors of PEMS08 dataset (Figure 1 c, d). While large deviations may persist or amplify at one sensor, they tend to revert towards regular level at another sensor. This reveals a **spatial heterogeneity of event impacts**. A closer look at the Figure 1d reveals a third dimension—a **type of heterogeneity of event impacts**: deviation with sim-

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.

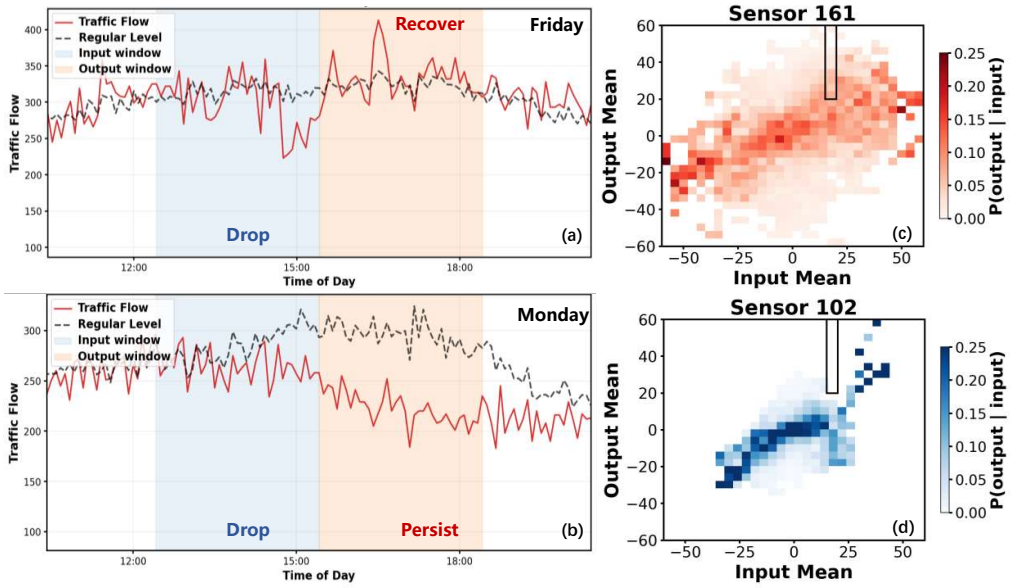


Figure 1: Heterogeneous impact of aperiodic traffic events on PEMS08. (a, b) Similar surge of traffic volume in sensor 1 with different future pattern in different time revealing temporal heterogeneity of event's impact (c, d) Comparable input deviations may amplify at the sensor 161 but almost never at the sensor 102 revealing spatial heterogeneity of event's impact. Aperiodic variation of same amplitude will results in different future pattern demonstrating the type heterogeneity of event's impact.

ilar amplitude in the past may evolve differently in the future, implying that an event's impact also depends on the pattern of the aperiodic components. These patterns serve as observable signatures of latent event types, even when explicit event annotations are unavailable. Together, these observations suggest that event-induced deviations are jointly shaped by their temporal context, spatial context, and latent dynamic patterns.

Capturing such heterogeneous impacts requires the modeling itself to respond to each event individually. Existing methods (as summarized in Figure 2), however, fall short of this requirement. A large body of work focuses on modeling dependencies among nodes, typically through graph neural networks or self-attention mechanisms Zhang et al. (2026); Li et al. (2018); Bai et al. (2020); Wu et al. (2019); Han et al. (2021); Kong et al. (2024); Zheng et al. (2020), enriching forecasting with spaitio-temporal contexts Dong et al. (2024); Shao et al. (2022); Liu et al. (2023); Guo et al. (2023) or introducing dynamic parameterization to adapt the model to varying data distributions Dong et al. (2024); Zhong et al. (2026). However, without explicit decomposition, event-induced deviations remain entangled with periodic variation, and they tend to be overlooked when only the overall forecasting loss is optimized, as periodic patterns dominate the training signal Su et al. (2025). A few recent methods explicitly disentangle the aperiodic component from the series Cao et al. (2025); Su et al. (2025); Fang et al. (2023); Shao et al. (2026); Lin et al. (2024); however, their efforts are devoted to better decomposition, while the disentangled component is still modeled with an event-invariant parameterization, assuming the aperiodic deviation

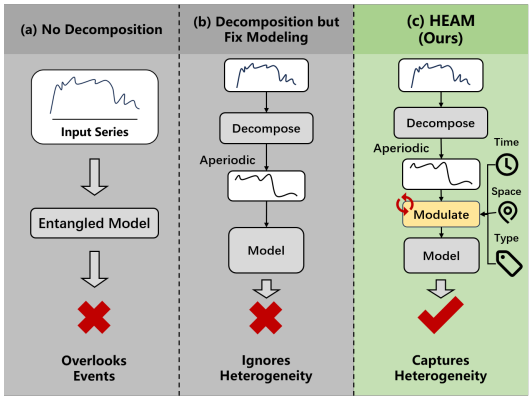


Figure 2: Illustration of three paradigms of handling aperiodic components.

pattern does not vary with spatio-temporal context and event type. And such fixed modeling cannot adapt to events that differ in when and where they occur and what type they are.

To address this challenge, we propose **HEAM** (**H**eterogeneous **E**vent-**A**daptive **M**odulation), a plug-and-play module that can be integrated into existing forecasting backbones to endow the forecastor with the ability of event-adaptive modeling. Specifically, following a *Decompose-Situate-Identify* pipeline, HEAM generates a modulation vector for each event from a library of modulator pools, each holding a set of candidate modulators. It first decomposes the overall representation of spatio-temporal series into a regular context component and an aperiodic component which captures the spatio-temporal context and the pattern of the event. It then situates each event against the library through its spatio-temporal context, selecting the pools according to when and where the event occurs, and then identifies candidate modulators within each pool according to the latent type of each event. HEAM finally responds to the event by assembling the selected modulators into a single modulation vector adapted to both its spatio-temporal context and its latent type, such that the heterogeneous impact of each event is explicitly captured through the modulation. Our contributions are summarized as follows:

- We identify the limitation of current work in capturing the heterogeneous impact of aperiodic events. To fill this gap, we propose HEAM, a plug-and-play module that modulates the representation of the aperiodic component in an event-adaptive manner.
- We design a *Decompose-Situate-Identify* pipeline to effectively generate modulation vectors tailored to each event's context and latent type, such that the heterogeneous impact of events can be captured.
- Extensive experiments on traffic and solar benchmarks show that HEAM can benefit strong backbones with various architectures. The improvements concentrate on long forecast horizons and in event-dense periods, particularly where the modeling of the heterogeneous impact of aperiodic events matters most.

2 Related Work

Spatio-temporal forecasting has received broad attention in recent years. Early studies mainly focused on modeling relations among nodes, as exemplified by DCRNN Li et al. (2018), Graph WaveNet Wu et al. (2019), AGCRN Bai et al. (2020) and GMAN Zheng et al. (2020) which exploit predefined graph structures, learned adjacency matrices or attention mechanisms. More recently, another line of work recognizes that the underlying data distribution itself also varies across space and time. This idea is reflected in HimNet Dong et al. (2024), STAEFormer Liu et al. (2023), STID Shao et al. (2022) and DPRNet Zhong et al. (2026), which assign different embeddings or parameters to different spatio-temporal states. These methods, however, do not explicitly decompose the series, so the impact of aperiodic events remains entangled with periodic variation that dominates the training objective. Consequently, even with state-dependent parameters, the aperiodic component induced by events is still easily overlooked Su et al. (2025).

A recent line of work addresses this issue by explicitly decomposing the series before forecasting. DualCast Su et al. (2025), HyperD Shao et al. (2026), CycleNet Lin et al. (2024), and STDN Cao et al. (2025) represent this direction through decomposing spatio-temporal signal into regular and aperiodic components and model them respectively. However, they model all aperiodic deviations with input-invariant parameterization, leaving the heterogeneous impacts of events unaddressed.

Our HEAM framework differs from these works in how the aperiodic component is modeled after being exposed. Rather than assigning one shared response to all deviations, HEAM modulates the representation of the aperiodic components in an event-adaptive way, according to its spatio-temporal context in which an event occurs and the latent type of deviation it exhibits. This design models the heterogeneous impact of event explicitly, and forecasting quality can be improved.

3 Task Formulation

We study multi-step spatio-temporal forecasting over N spatially distributed sensors. Let $X_{t-T+1:t} \in \mathbb{R}^{T\times N\times C}$ denote the historical observations within a look-back window of T steps,

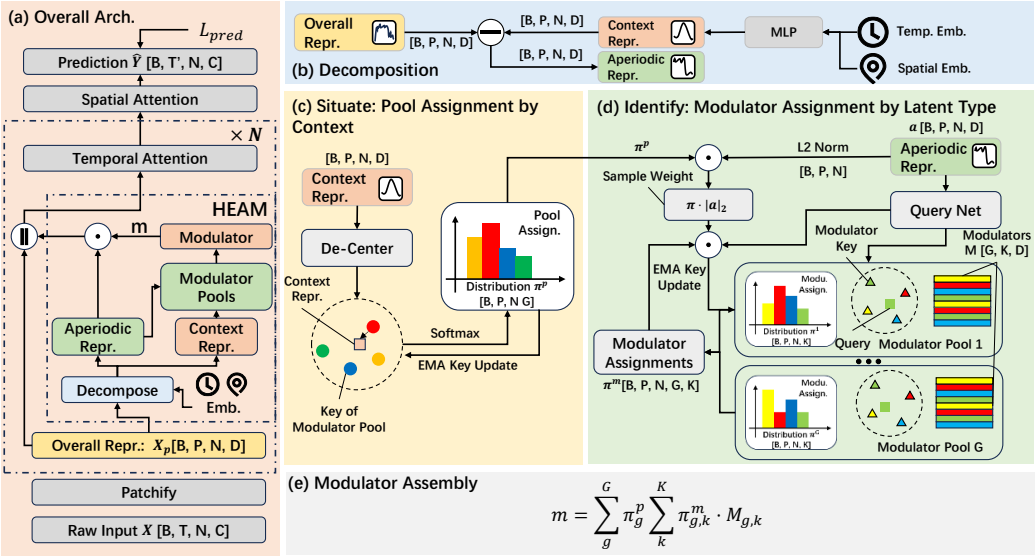


Figure 3: Overview of HEAM, which generates an event-specific modulation vector from a library of modulator pools and applies it to the aperiodic representation. (a) The overall architecture: the aperiodic representation is modulated by the generated vector m and combined with the original representation to drive the final prediction. (b) Decomposition: obtain context and aperiodic representation of the spatio-temporal series. (c) Situate: pool assignment π^p from the de-centered context representation. (d) Identify: in-pool modulator assignment π^m from the aperiodic representation. (e) Modulator Assembly: assembly of the modulation vector m from π^p and π^m.

where C counts the recorded channels, including the target variable (e.g., traffic flow or photovoltaic power) and time covariates (e.g., time-of-day and day-of-year). Given a graph $G \in \mathbb{R}^{N \times N}$ that encodes pairwise spatial relations among sensors, the goal is to learn a mapping F_θ that forecasts the evolution of the target variable over the next T' steps:

$$\hat{Y}_{t+1:t+T'} = F_\theta(G, X_{t-T+1:t}), \tag{1}$$

where $\hat{Y}_{t+1:t+T'} \in \mathbb{R}^{T' \times N \times C'}$ and C' denotes the number of target channels.

4 Methodology

4.1 Overview

HEAM is a plug-and-play module that can be inserted into existing spatio-temporal forecasting backbones without architectural changes. It consumes the backbone's overall representation of the spatio-temporal series and modulates the aperiodic representation derived from it (Figure 3b-e). The modulated representation, combined with the original representation, drives the final prediction, improving forecasting quality (Figure 3a). The remainder of the paper is organized as follow: we detail the *Decompose-Situate-Identify* pipeline of HEAM in Sections 4.2–4.4. Section 4.5 then describes how the selected modulators are assembled into a modulation vector and applied to the aperiodic representation. We analyze the complexity of HEAM module in Section 4.6.

4.2 Decomposition

As shown in Figure 3b, in the decomposition step of HEAM module, we aim to decompose the overall representation of spatio-temporal series $Z^{(l)} \in \mathbb{R}^{P \times N \times D}$ at layer l into a regular context representation $Z_c^{(l)}$, capturing the spatio-temporal context in which events unfold, and an aperiodic representation $Z_a^{(l)}$, capturing the event-induced deviations themselves (l is omitted in the following

section). Here P, N, and D denote the numbers of temporal tokens (e.g., patches or convolution outputs), nodes, and hidden dimensions, respectively. Specifically, the context representation is produced by a lightweight MLP-based extractor that consumes only temporal and spatial embeddings:

$$Z_c = \mathrm{MLP}\big([\, E_{tod}(c_{tod});\, E_{dow}(c_{dow});\, E_{node}(c_n)\,]\big), \tag{2}$$

where $c_{tod}, c_{dow} \in \mathbb{R}^P$ and $c_n \in \mathbb{R}^N$ are taken directly from the raw input, denoting the time-of-day and day-of-week of each temporal token and the node index, respectively; they are broadcast to $P \times N$ fields. $E_{tod} \in \mathbb{R}^{S \times d_t}$, $E_{dow} \in \mathbb{R}^{7 \times d_w}$, and $E_{node} \in \mathbb{R}^{N \times d_n}$ are learnable embedding tables (S is the number of time steps per day), and $E(\cdot)$ denotes the row selected by the corresponding index. We read c_{tod} and c_{dow} at the temporal position represented by each token. This position is the patch center for patch-based backbones, and the last position of the receptive field for convolutional ones.

$$Z_a = Z - Z_c \tag{3}$$

To ensure that Z_c absorbs the regular variation of the series, the extractor is trained to reconstruct the backbone representation from temporal and spatial embeddings alone, with gradients flowing only to the extractor:

$$\mathcal{L}_{\mathrm{dec}} = \big\|\mathrm{sg}(Z) - Z_c\big\|_2^2, \tag{4}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation, which detaches the backbone features from the computational graph of the decomposition loss preventing the overall representation being pulled toward the regular component. This structural design ensures that $Z_c^{(l)}$ is confined to the regular variation of the spatio-temporal series and Z_a retains the aperiodic deviations left by events, which cannot be explained by the regular variation of the series.

4.3 Pool Assignment by Context Representation

HEAM module aims to model the heterogeneous impact of aperiodic events by modulating the aperiodic representation according to its spatio-temporal context and latent event type. The Situate stage realizes the context half of this goal: it routes the context representation of each temporal token to the modulator pools in a library in which each pool holds a set of candidate modulators, yielding the pool assignment π^p. The assignment is conditioned on the de-centered context representation: $\bar{Z}_c = Z_c - \mu \in \mathbb{R}^{P \times N \times D}$ where $\mu \in \mathbb{R}^D$ is the global mean of the context representation Z_c, averaged over the batch, temporal-token, and node dimensions, and maintained as an exponential moving average (EMA) across training steps. Subtracting μ removes the direction shared by all context representations, so that their cosine similarity reflects differences of spatio-temporal context rather than this common component. The pool assignment of each temporal token is then computed by cosine similarity between $\bar{Z}_c$ and a set of G pool prototypes $P \in \mathbb{R}^{G \times D}$:

$$\pi^p = \mathrm{softmax}_g\big(\tau_p \cdot \langle \bar{Z}_c, P \rangle\big) \in \mathbb{R}^{P \times N \times G}, \tag{5}$$

where τ_p is a learnable scale parameter and $\langle \cdot, \cdot \rangle$ denotes cosine similarity, so that the assignment logit can be stabilized during training period. After each training step, every context representation is hard-assigned to its most probable pool prototype, and each prototype is updated by exponential moving average toward the mean of the normalized de-centered context representations assigned to it. In this way, the prototypes automatically adapt to the update process of the context representation during training, which stabilizes the training process. The full update procedure of the pool prototypes and global mean of context representation μ is detailed in Appendix A.

4.4 Modulator Assignment by Latent Event Type

HEAM modulates the aperiodic representation according to each event's latent event type to capture the type heterogeneity of event's impact. The Identify stage achieve this goal by computing an in-pool assignment $\pi^m \in \mathbb{R}^{P \times N \times G \times K}$ from aperiodic representation Z_a to identify the modulator that match the event's latent type. Specifically, the aperiodic representation is first mapped by per-pool linear heads into a low-dimensional query space $q \in \mathbb{R}^{P \times N \times G \times d_q}$ with $d_q \ll D$. The query is then matched against K latent event prototypes per pool, each $e \in \mathbb{R}^{d_q}$ representing one latent event type under spatio-temporal context represented by pool g, via cosine similarity:

$$\pi^m = \mathrm{softmax}_k\big(\tau_g \cdot \langle q, e \rangle\big) \in \mathbb{R}^{P \times N \times G \times K}, \tag{6}$$

where τ_g is a per-pool learnable temperature and $\langle \cdot, \cdot \rangle$ denotes cosine similarity, which stabilize the logit of assignments. As in the Situate stage, the event prototypes are not trained by gradient descent but are updated by an exponential moving average toward the mean of the queries hard-assigned to them, each weighted by:

$$w = \min\left(1, \frac{\|Z_a\|}{\mathrm{med}(\|Z_a\|)}\right) \cdot \pi^p \in \mathbb{R}^{P \times N \times G}, \tag{7}$$

where the aperiodic magnitude $\|Z_a\|$ represents the energy of the event and $\mathrm{med}(\cdot)$ denotes the median, estimated over training batches by exponential moving average. The energy factor prevents the prototypes from being contaminated by event-free samples, which dominate the data volume and would otherwise pull the update toward the event-free state. The pool assignment factor, in contrast, ensures that each pool's prototypes are updated mainly by the samples of its own context. This ensures that each modulator learns how to modulate the aperiodic representation under this specific spatio-temporal context and the latent event type. The resulting π^m serves as the in-pool weights for assembling the modulation vector, as described in the next section. The procedure for the EMA update of latent event prototype e is detailed in Appendix A.

4.5 Modulation

The Modulation stage assembles the modulation vector from the two routing levels and applies it to the aperiodic representation, realizing the event-adaptive modulation. Each pool holds K modulators $M_{g,k} \in \mathbb{R}^D$, and the vector is assembled by combining the modulators within each pool by π^m and across pools by π^p:

$$m = \sum_{g=1}^{G} \pi_g^p \sum_{k=1}^{K} \pi_{g,k}^m M_{g,k} \in \mathbb{R}^{P \times N \times D} \tag{8}$$

The aperiodic representation is then modulated by m through a scalar gate γ: $\tilde{Z}_a = (1 + \gamma\, m) \odot Z_a$ where γ is initialized to zero, so that no modulation signal is injected at the beginning of training and the backbone's behavior is gradually refined as γ grows. The modulated representation is concatenated with the original overall representation and drives the final prediction (Figure 3a). To keep the K modulators within each pool from degenerating into redundant directions, an orthogonality regularization is applied to each pool's modulators Zhong et al. (2026): $\mathcal{L}_{\mathrm{ortho}} = \frac{1}{G}\sum_{g=1}^{G} \frac{1}{K}\left\|\hat{M}_g \hat{M}_g^\top - I_K\right\|_F^2$ where $\hat{M}_g \in \mathbb{R}^{K \times D}$ denotes the row-normalized modulator matrix of pool g. This keeps the modulators diverse, so that the K modulators of each pool encode distinct responses to latent event types under different context. Eventually, the model parameter is optimized by the weighted sum of L_{ortho}, L_{dec}, and the forecasting loss based on huber loss:

$$L = L_{forecast} + \lambda_{dec} L_{dec} + \lambda_{ortho} L_{ortho} \tag{9}$$

By construction, the *Decompose-Situate-Identify* pipeline of HEAM explicitly modulates the aperiodic representation in a context- and type-specific manner that a shared, flat routing cannot provide; thereby, the heterogeneous impact of events are well captured. The detailed process of plugging HEAM into backbones of various architecture is detailed in Appendix B.

4.6 Complexity Analysis of HEAM Module

In terms of computational complexity, HEAM operates independently on each temporal token and each node, without any cross-token or cross-node aggregation. Thus, the additional computational complexity of HEAM is $O(NT)$. The added parameters scale likewise: the only data-dependent terms are the node embeddings ($O(N \cdot d_n)$), which is linear in N. While many spatio-temporal forecasting backbones involve operations with quadratic terms $O(N^2)$ or $O(T^2)$ in computational complexity, HEAM does not change the asymptotic complexity of the backbone.

5 Experiments

We evaluate HEAM on long-term spatio-temporal forecasting across three datasets spanning traffic flow, traffic speed, and solar power generation. We first detail our experiment settings and we organize the remainder of experiments section around the following research questions:

- **RQ1:** (Overall Improvement) How much does HEAM improve the performance of its host backbones, and how do the resulting models compare with state-of-the-art methods?
- **RQ2:** (Improvement Analysis) Do HEAM's gains concentrate in event-dense periods?
- **RQ3:** (Ablation Study) How does each component of HEAM contribute to the overall improvement?
- **RQ4:** (Structural Analysis) Does the *Decompose-Situate-Identify* pipeline learn meaningful context and latent event-type structures to facilitate event-adaptive modulations?

Datasets. We conduct experiments on three public benchmarks spanning different domains. PEMS08 and METR-LA are traffic datasets recording traffic flow and vehicle speed, respectively, at 5-minute intervals, while Solar-AL records photovoltaic power generation at 10-minute intervals. Full dataset statistics and descriptions are provided in Appendix C.

Baselines and Evaluation Setup. We compare HEAM against three families of state-of-the-art methods, including graph-based, heterogeneity-aware, and decomposition-based methods. We use GraphWaveNet Wu et al. (2019) and VisiFold Zhang et al. (2026) as two strong backbones of different architectural families, and report the improvement of HEAM over each backbone and over the strongest baselines. Detailed descriptions of the baselines and backbone models are provided in Appendix F. We follow the standard chronological train/validation/test splits of each dataset and evaluate all models at forecast horizons of 24, 36, and 48 steps. Following VisiFold Zhang et al. (2026), we use an input window of the same length as the forecast horizon, i.e., 24, 36, and 48 steps respectively. We adopt MAE, RMSE, and MAPE as evaluation metrics; MAPE is not reported on Solar-AL, where zero values at night make the metric unstable.

Implementation Details. All baselines are implemented with configurations adapted from the BasicTS Shao et al. (2024) framework. HEAM uses the same training configuration as its host backbone to ensure a fair comparison. HEAM uses the same optimizer, learning rate schedule, batch size, weight decay, and epoch budget as its host backbone to ensure a fair comparison. Appendix B lists the complete hyperparameter settings, including the number of pools G, the modulators per pool K, the routing scale parameters τ_p and τ_g, and the loss weights λ_{dec} and λ_{ortho}, and describes how the module is inserted into each backbone. Hyperparameter study results suggest that HEAM is relatively insensitive to the setting of G, K (see Appendix E for hyperparameter study results.

5.1 RQ1: Overall Performance of HEAM Module

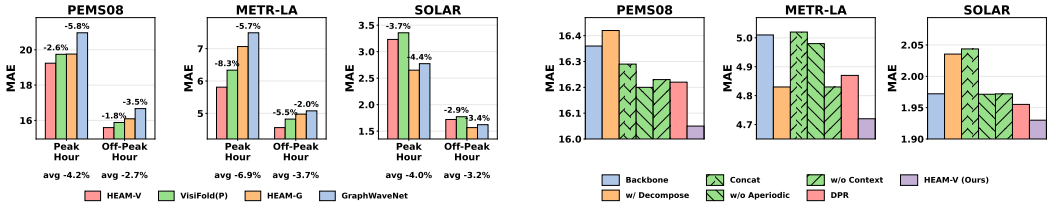
We report the performance of baseline models, HEAM's backbone models, and HEAM equipped models in Table 1. Each HEAM equipped model and backbone model is run under five different seeds, and we report their mean performance in TABLE 1. Their standard deviations are reported in Table 4. As shown in Table 1, HEAM improves its host backbones on average consistently across datasets and horizons and achieves largest average improvement over the backbone model under the longest 48-step ahead forecast scenario. Specifically, HEAM improves the performance of strong baseline by 2.88%, 4.27% and 3.35% on 48 step ahead forecasting on PEMS08, METR-LA and SOLAR datasets respectively, demonstrating its effectiveness in long-term spatio-temporal forecasting. Beyond improving its backbones, the HEAM-equipped models also compare favorably with existing methods: at 48-step ahead forecast scenario, they attain the lowest MAE among all models, reducing MAE by 1.8% over the best baseline on PEMS08 (16.05 vs. 16.35), 6.2% on METR-LA (4.72 vs. 5.03) and by 2.3% on SOLAR (1.70 vs 1.74). Both comparisons indicate the effectiveness of the HEAM module in improving long-term forecast performance.

5.2 RQ2: Analysis of HEAM's Improvement

To understand where HEAM's improvement originates, we compare the relative reduction of MAE of the HEAM-equipped model and its host backbone on PEMS08 at horizon 48, separately over peak hours, when deviations from the regular pattern are most consequential, and over off-peak hours. As shown in Figure 4a, HEAM reduces MAE by 5.03% over peak hours against 3.20% over off-peak hours. This indicates that the benefit of HEAM is concentrated in the periods where the forecast of aperiodic deviations matter most, consistent with the event-adaptive design of the module. Results for the other horizons and the definitions of the peak hours are provided in Appendix G.

	Method	24 Time Steps			36 Time Steps			48 Time Steps		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
PEMS08	DCRNN	33.34	22.60	15.46	39.37	25.82	18.53	45.64	30.47	25.10
	AGCRN	28.05	17.45	11.25	30.96	19.39	12.73	31.11	19.46	12.88
	STID	25.67	15.50	10.41	26.97	16.29	10.91	28.06	17.00	11.67
	STAEformer	25.71	**14.60**	**9.84**	27.50	15.72	10.45	28.97	16.54	10.99
	VisiFold	25.64	15.26	10.26	**26.71**	16.07	10.78	**27.76**	16.35	10.95
	HimNet	26.19	15.58	10.14	27.47	15.92	10.61	29.36	17.30	11.31
	STDN	27.04	16.34	10.85	29.14	17.07	11.35	30.18	17.67	12.12
	HyperD	25.68	15.19	10.17	27.71	16.46	11.28	28.81	16.93	11.75
	CycleNet	34.93	21.99	14.21	41.53	26.15	17.51	45.30	28.49	20.02
	DualCast	28.59	18.31	12.10	30.29	19.66	13.13	36.97	26.09	26.69
	GraphWaveNet	25.97	15.98	10.64	27.14	16.45	10.89	28.23	17.20	11.47
	HEAM-G (Ours)	**25.56**	15.17	9.93	27.01	15.95	10.48	27.89	16.55	10.96
	VisiFold (P)	25.61	14.97	10.02	27.03	15.78	10.63	28.19	16.36	10.89
	HEAM-V (Ours)	25.66	14.91	9.88	26.93	**15.63**	**10.41**	27.79	**16.05**	**10.66**
	ΔBackbone	**+0.70%**	**+2.75%**	**+4.03%**	**+0.43%**	**+2.00%**	**+2.93%**	**+1.32%**	**+2.88%**	**+3.28%**
METR-LA	DCRNN	24.85	19.34	52.18	15.10	8.16	23.83	17.40	9.19	23.51
	AGCRN	**10.45**	4.50	11.60	11.46	5.00	13.12	11.83	5.18	13.51
	STID	10.95	4.79	12.39	10.94	4.80	12.95	11.57	5.03	13.83
	STAEformer	10.53	4.47	**11.46**	12.27	5.19	13.52	11.84	5.41	14.65
	VisiFold	10.57	4.46	11.99	11.79	5.08	13.41	12.22	5.53	15.19
	HimNet	11.31	4.52	11.59	12.72	5.14	13.05	13.52	5.51	13.81
	STDN	10.47	4.47	12.31	11.09	4.85	13.23	11.83	5.08	13.84
	HyperD	10.70	4.84	13.10	**10.89**	4.96	13.51	11.42	5.28	14.60
	CycleNet	11.81	5.45	14.43	12.80	6.18	17.03	13.34	6.55	18.38
	DualCast	12.48	5.22	12.55	12.55	5.38	14.32	12.89	5.55	13.82
	GraphWaveNet	10.82	4.63	11.81	11.62	5.04	13.37	12.46	5.38	14.25
	HEAM-G (Ours)	10.46	**4.37**	11.53	11.44	4.80	12.55	12.34	5.24	13.64
	VisiFold (P)	10.54	4.42	11.69	11.31	4.82	12.95	11.73	5.01	13.46
	HEAM-V (Ours)	10.51	4.43	11.79	11.06	**4.67**	**12.50**	**11.11**	**4.72**	**12.88**
	ΔBackbone	**+1.86%**	**+2.78%**	**+0.73%**	**+1.88%**	**+4.00%**	**+4.80%**	**+3.15%**	**+4.27%**	**+4.29%**
Solar-AL	DCRNN	3.69	1.65	-	6.89	2.95	-	5.09	2.40	-
	AGCRN	3.11	1.32	-	3.85	1.65	-	4.34	1.80	-
	STID	3.74	1.73	-	4.30	1.96	-	4.62	2.07	-
	STAEformer	4.94	3.07	-	6.24	4.01	-	7.03	4.48	-
	VisiFold	3.28	1.38	-	4.08	1.77	-	4.53	1.97	-
	HimNet	3.09	1.23	-	3.95	1.62	-	4.30	1.74	-
	STDN	3.28	1.49	-	4.44	2.11	-	4.45	1.97	-
	HyperD	3.76	1.85	-	4.11	1.94	-	4.39	1.98	-
	CycleNet	5.45	2.41	-	6.74	3.16	-	7.49	3.56	-
	DualCast	3.22	1.37	-	4.01	1.71	-	4.20	1.86	-
	GraphWaveNet	3.04	1.25	-	3.77	1.60	-	4.23	1.77	-
	HEAM-G (Ours)	**2.94**	**1.22**	-	**3.73**	**1.56**	-	**4.13**	**1.70**	-
	VisiFold (P)	3.12	1.37	-	3.97	1.79	-	4.47	1.97	-
	HEAM-V (Ours)	3.08	1.35	-	3.95	1.76	-	4.38	1.91	-
	ΔBackbone	**+2.19%**	**+2.06%**	-	**+0.77%**	**+1.78%**	-	**+2.19%**	**+3.35%**	-

Table 1: Performance of HEAM in long-term spatio-temporal forecasting scenarios. The values of the three metrics are the smaller, the better. The best results are indicated in **bold**, and the second-best results are underlined. Δ **Backbone** line denote the average improvement of backbone models after equipped with HEAM module.



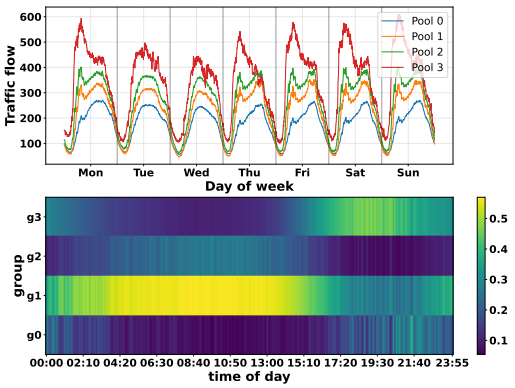
(a) Reduction of MAE brought by HEAM over peak and off-peak periods.

(b) MAE of ablation study results across the three datasets of HEAM-V model.

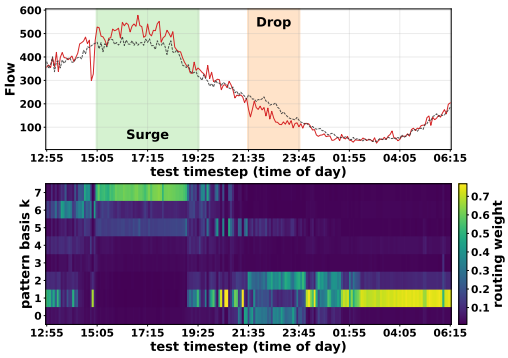
Figure 4: Further analysis of the HEAM module.

5.3 RQ3: Ablation Study

Setup. We ablate HEAM-V on all three datasets at horizon 48 (Figure 4b), comparing the full module against **Backbone** (VisiFold(P) without HEAM), *w/ Decompose* (Decompose only, no modulation), *w/o Aperiodic* and *w/o Context* (Identify and Situate removed, respectively), *Concat* (flat



(a) Traffic flow pattern of node selecting each pool (top) Selection rate of each modulator pool across time of day (bottom).



(b) In-pool modulator selection: flow (red) vs regular (gray), with a surge (green) and a drop (orange) window (top) and the selection of modulators (bottom).

Figure 5: Visualization of the pool-level and in-pool selection patterns of HEAM-V on PEMS08 at forecast horizon 48.

routing over the concatenation), and *DPR* Zhong et al. (2026) (recalibration of the entire hidden representation).

Results. *w/ Decompose* shows that isolating the aperiodic component is not sufficient without modulating it. The results of *w/o Aperiodic* and *w/o Context* variants indicate that both the spatiotemporal context and the latent event type are required for event-adaptive modeling. *Concat* variant suggests that the two-level structure is necessary for HEAM's event-adaptive modeling. *DPR*, the closest concurrent work, shows that modulating the disentangled component is preferable to recalibrating the whole representation.

5.4 RQ4: Analysis of HEAM's Selection Pattern of Modulators

To investigate how HEAM selects modulators under the heterogeneous impact of events, we analyze the selection behavior of HEAM-V on PEMS08 at both the pool and the in-pool level (Figure 5). Figure 5a a shows the variaton pattern of traffic flow on nodes selecting different pools (top) and pool selection rate across time of day (bottom). As shown in Figure 5a, Nodes attributed to different pool demonstrate different variation patterns, and pools were selected on different time of day. This indicates that the pool assignment is conditioned on where and when an event occurs. Figure 5b shows the in-pool modulator selection rate of node #93, together with the traffic flow and its regular component. The selected modulator differs between a surge of flow above the regular component (green) and a drop below it (orange). Indicating that modulators represent distinct latent event type. This suggests that HEAM can modulate the aperiodic representation of spatio-temporal series according to its spatio-temporal context and latent-event-type.

6 Conclusion

In this paper, we study long-term spatio-temporal forecasting in the presence of aperiodic events, and argue that their impact is heterogeneous: the same deviation unfolds differently depending on when and where it occurs and what type it is. To capture this heterogeneity, we proposed HEAM, a plug-and-play module that isolates the aperiodic component from the backbone's representation and modulates it through a two-level routing over a library of modulator pools. Experiments across three datasets and two architecturally different backbones show that HEAM consistently improves forecasting, with gains that grow with the forecast horizon and concentrate in event-rich periods. Our analysis further indicates that the routing learns interpretable structure: pools align with spatiotemporal regimes, and modulators separate distinct event patterns.

AI USE STATEMENT

We used generative AI tools solely to edit the manuscript for grammar, wording, and readability. We did not use generative AI tools for any other purpose, and all other tasks listed in the ICLR 2027 AI Policy for Authors are not applicable to this work. We have reviewed all AI-assisted text and take responsibility for the final content of this work.

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.

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.

Yuchen Fang, Yanjun Qin, Haiyong Luo, Fang Zhao, Bingbing Xu, Liang Zeng, and Chenxing Wang. When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks. In *2023 IEEE 39th international conference on data engineering (ICDE)*, pp. 517–529. IEEE, 2023.

Shengnan Guo, Youfang Lin, Letian Gong, Chenyu Wang, Zeyu Zhou, Zekai Shen, Yiheng Huang, and Huaiyu Wan. Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1585–1596. IEEE, 2023.

Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 547–555, 2021.

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.

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.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37:106315–106345, 2024.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 4125–4129, 2023.

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 AAAI Conference on Artificial Intelligence*, volume 40, pp. 15689–15697, 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.

Xinyu Su, Feng Liu, Yanchuan Chang, Egemen Tanin, Majid Sarvi, and Jianzhong Qi. Dualcast: a model to disentangle aperiodic events from traffic series. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3290–3298, 2025.

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

Zhiwei Zhang, Xinyi Du, Weihao Wang, Xuanchi Guo, and Wenjuan Han. Visifold: Long-term traffic forecasting via temporal folding graph and node visibility. *arXiv preprint arXiv:2603.11816*, 2026.

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

Siru Zhong, Zhao Meng, Haohuan Fu, Haoyang Li, Qingsong Wen, and Yuxuan Liang. Perceive, route and modulate: Dynamic pattern recalibration for time series forecasting. *arXiv preprint arXiv:2605.06310*, 2026.

A Algorithm Details

In this section, we detail the EMA update process of the context prototypes and the latent event type prototypes. We write the batch dimension explicitly here, whereas the main text omits it and describes a single sample. For a mini-batch $\mathcal{B}$ of size B, let $\langle\cdot\rangle_{\mathcal{B}}$ denote the mean over its $B \times P \times N$ tokens and let $\mathrm{norm}(x) = x/\|x\|_2$.

EMA Update Procedure of Context Prototypes. The context prototypes are updated by a batch-wise exponential moving average of the context representations assigned to them, so that each prototype settles on a distinct spatio-temporal context. In this way, the update of the prototypes is detached from the compute graph, which stabilizes the update of the model parameters. Let α denote the EMA decay. After each training step, every temporal token is hard-assigned to its most probable prototype:

$$g^{\star}_{b,p,n} = \arg\max_{g\in\{1,\dots,G\}} \pi^{p}_{b,p,n,g} \;\in\; \{1,\dots,G\}, \tag{10}$$

the global mean μ, and the assignment sum S_g are updated with the same decay α:

$$\mu \leftarrow \alpha\,\mu + (1-\alpha)\,\langle Z_c\rangle_{\mathcal{B}}, \tag{11}$$

$$S_g \leftarrow \alpha\,S_g + (1-\alpha)\sum_{b,p,n} \mathbb{1}\big[g^{\star}_{b,p,n} = g\big]\;\mathrm{norm}\big(\bar{Z}_c[b,p,n]\big)\,, \tag{12}$$

where the sums run over all tokens of the batch, with b indexing the batch, p the P temporal tokens, and n the N nodes.$\mathbb{1}[\cdot]$ denotes the indicator function, equal to one when its argument holds and zero otherwise. Each prototype is then computed as the norm of the assignment sum $P_g \leftarrow \mathrm{norm}(S_g)$. We set $\alpha = 0.9$ for the first 10^3 steps and $\alpha = 0.99$ thereafter, so that the prototypes adapt quickly at the beginning of training and stabilize afterwards.

EMA Update Procedure of Latent Event Prototypes. The latent event prototypes follow the same procedure as the context prototypes, differing only in the weight of each token. After each training step, every token is hard-assigned to its most probable prototype within each pool,

$$k^{\star}_{b,p,n,g} = \arg\max_{k\in\{1,\dots,K\}} \pi^{m}_{b,p,n,g,k} \;\in\; \{1,\dots,K\}, \tag{13}$$

and the weighted assignment sum $C_{g,k}$ is updated with the same decay α,

$$C_{g,k} \leftarrow \alpha\, C_{g,k} + (1-\alpha) \sum_{b,p,n} \mathbb{1}\big[k^{\star}_{b,p,n,g} = k\big]\; w_{b,p,n,g}\, \mathrm{norm}(q_{b,p,n,g})\,, \tag{14}$$

where $q_{b,p,n,g}$ is the per-pool low-dimension projection of aperiodic representations and

$$w_{b,p,n,g} = \min\!\left(1, \frac{\|Z_a[b,p,n]\|}{\mathrm{med}(\|Z_a\|)}\right) \pi^{p}_{b,p,n,g} \tag{15}$$

where $\pi^{p}_{b,p,n,g}$ is the pool assignment probability of token (b,p,n) to pool g, obtained in the Situate stage. And the running median of aperiodic magnitude $\mathrm{med}(\|Z_a\|)$ is maintained by an exponential moving average over per-batch medians, $\mathrm{med}(\|Z_a\|) \leftarrow \alpha\,\mathrm{med}(\|Z_a\|) + (1-\beta)\,\mathrm{median}_{b,p,n}\|Z_a[b,p,n]\|$. Each prototype is then computed as $\mathrm{norm}(C_{g,k})$, with unassigned prototypes and the decay schedule treated as above.

B Implementation Details

In this section, we introduce how HEAM module is integrated into existing backbones and implementation details regarding the training of HEAM and its backbones.

Details of HEAM's Integration into Backbones. HEAM is inserted once per layer rather than once per model, and each insertion site owns an independent set of pools, prototypes, and modulators. In VisiFold(P) the module is placed in front of every patch temporal transformer layer; in GraphWaveNet it is placed in front of each gated convolution block. At each site the module consumes the layer's input representation $Z^{(l)}$, decomposes it into a context $Z_c^{(l)}$ and an aperiodic component $Z_a^{(l)}$, and produces the modulated aperiodic representation $\tilde{Z}_a^{(l)}$, which is then concatenated with $Z^{(l)}$ to form the input of the following layer. Because concatenation doubles the input width, the consuming layer is widened accordingly: the input dimension of the query, key, and value projection is doubled in VisiFold(P), and so is the input channel count of the dilated convolutions in GraphWaveNet.

Model Training and Hyperparameter Details. All models are optimized with Adam for a maximum of 100 epochs, and the checkpoint with the lowest validation MAE is used for testing. The backbone alone and the backbone equipped with HEAM receive the same optimizer, learning rate, schedule, batch size, weight decay, and epoch budget, so that the comparison isolates the effect of the module rather than that of the training configuration. HEAM uses $G = 4$ modulator pools and $K = 8$ modulators per pool on all datasets except for HEAM-G in SOLAR-AL, where we use $K = 32$. The prototypes are updated with $\alpha = 0.99$ after a warm-up of $\alpha = 0.9$ over the first 1000 steps. For VisiFold(P) we use 3 temporal layers with patch size 4 and stride 2, and otherwise follow the PEMS08 configuration of the original VisiFold Zhang et al. (2026), including the masking ratio of 0.2 and the subgraph size of 30 used during training. For GraphWaveNet we use kernel sizes of 5, 4, and 3 for horizons 48, 36, and 24, and otherwise follow the PEMS08 configuration of GraphWaveNet in BasicTS Shao et al. (2024). All remaining settings, including the learning rate and its schedule, the batch size, the loss weights λ_{dec} and λ_{ortho}, and the initial value of routing scale parameters τ_p and τ_g, are listed in Table 2.

C Dataset Statistics

We evaluate on three public benchmarks drawn from different domains and sampling rates. PEMS08 reports traffic flow from 170 sensors on eight freeways in the San Bernardino and Riverside areas from July to August 2016. METR-LA reports traffic speed from 207 loop detectors on the highways of Los Angeles County from March to June 2012. SOLAR-AL reports photovoltaic power generation from 137 plants in Alabama at 10-minute intervals over 2006. All three datasets are split chronologically into training, validation, and test sets. Table 3 summarizes the statistics.

Table 2: Hyperparameter settings. Values shared across models are stated once.

Setting	HEAM-V	HEAM-G
Optimizer	Adam	
Learning rate	0.001	
Learning rate milestones	20,25	50,80
Batch size	16	32
Max Epochs	100	
G (modulator pools)	4	
K (modulators per pool)	8(32 for SOLAR-AL dataset)	
d_q	$\max(D/4, 16)$	
α (prototype EMA)	0.99, 0.9 for first 10^3 steps	
λ_{dec}, λ_{ortho}	0.1, 0.05	
τ_p, τ_g	1, 1	

Table 3: Statistics of the three benchmark datasets.

Dataset	#Nodes	Timesteps	Interval	Split (train/val/test)
PEMS08	170	17,856	5 min	6:2:2
METR-LA	207	34,272	5 min	7:1:2
SOLAR-AL	137	52,560	10 min	6:2:2

D Robustness of HEAM Module

We run the HEAM-equipped model and its backbone with five random seeds and report the mean and standard deviation of their performance in Table 4; the subscript denotes the standard deviation in units of 10^{-2}.

	Method	24 Time Steps			36 Time Steps			48 Time Steps		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
PEMS08	GraphWaveNet	25.97_{41}	15.98_{46}	10.64_{46}	27.14_{25}	16.45_{29}	10.89_{28}	28.23_{25}	17.20_{47}	11.47_{43}
	HEAM-G (Ours)	25.56_{08}	15.17_{04}	9.93_{05}	27.01_{10}	15.95_{03}	10.48_{06}	27.89_{11}	16.55_{04}	10.96_{07}
	VisiFold (P)	25.61_{03}	14.97_{02}	10.02_{04}	27.03_{08}	15.78_{04}	10.63_{12}	28.19_{13}	16.36_{09}	10.89_{11}
	HEAM-V (Ours)	25.66_{08}	14.91_{05}	9.88_{04}	26.93_{09}	15.63_{10}	10.41_{10}	27.79_{07}	16.05_{04}	10.66_{03}
	ΔBackbone	**+0.70%**	**+2.75%**	**+4.03%**	**+0.43%**	**+2.00%**	**+2.93%**	**+1.32%**	**+2.88%**	**+3.28%**
METR-LA	GraphWaveNet	10.82_{05}	4.63_{03}	11.81_{30}	11.62_{32}	5.04_{10}	13.37_{53}	12.46_{33}	5.38_{12}	14.25_{37}
	HEAM-G (Ours)	10.46_{30}	4.37_{14}	11.53_{38}	11.44_{32}	4.80_{10}	12.55_{22}	12.34_{27}	5.24_{09}	13.64_{22}
	VisiFold (P)	10.54_{16}	4.42_{06}	11.69_{10}	11.31_{36}	4.82_{12}	12.95_{13}	11.73_{53}	5.01_{24}	13.46_{45}
	HEAM-V (Ours)	10.51_{15}	4.43_{06}	11.79_{26}	11.06_{29}	4.67_{10}	12.50_{18}	11.11_{27}	4.72_{10}	12.88_{20}
	ΔBackbone	**+1.86%**	**+2.78%**	**+0.73%**	**+1.88%**	**+4.00%**	**+4.80%**	**+3.15%**	**+4.27%**	**+4.29%**
Solar-AL	GraphWaveNet	3.04_{08}	1.25_{03}	-	3.77_{09}	1.60_{05}	-	4.23_{12}	1.77_{07}	-
	HEAM-G (Ours)	2.94_{10}	1.22_{03}	-	3.73_{07}	1.56_{03}	-	4.13_{06}	1.70_{04}	-
	VisiFold (P)	3.12_{05}	1.37_{02}	-	3.97_{05}	1.79_{04}	-	4.47_{09}	1.97_{02}	-
	HEAM-V (Ours)	3.08_{02}	1.35_{02}	-	3.95_{06}	1.76_{03}	-	4.38_{06}	1.91_{02}	-
	ΔBackbone	**+2.19%**	**+2.06%**	-	**+0.77%**	**+1.78%**	-	**+2.19%**	**+3.35%**	-

Table 4: Mean and standard deviation over five random seeds. The subscript denotes the standard deviation in units of 10^{-2}. **Δ Backbone** is the average relative improvement of the two backbones after equipping them with HEAM.

E Hyperparameter Study of HEAM Module

We study the sensitivity of HEAM to the number of modulator pools G and the number of modulators per pool K by comparing the performance of HEAM-V on PEMS08 at horizon 48 for $G \in \{1, 2, 4, 8, 16\}$ and $K \in \{1, 2, 4, 8, 16\}$. We report RMSE, MAE, and MAPE in Figure 6. Two conclusions follow. First, the degenerate settings $G = 1$ and $K = 1$, which disable the context-conditioned and the type-conditioned routing respectively, degrade performance markedly, consistent with the ablation study results. This indicates that both the spatio-temporal context and

the latent event type are necessary for event-adaptive modeling. Second, performance remains stable for $G, K \geq 2$. This suggests that the module is robust to both choices of G and K.

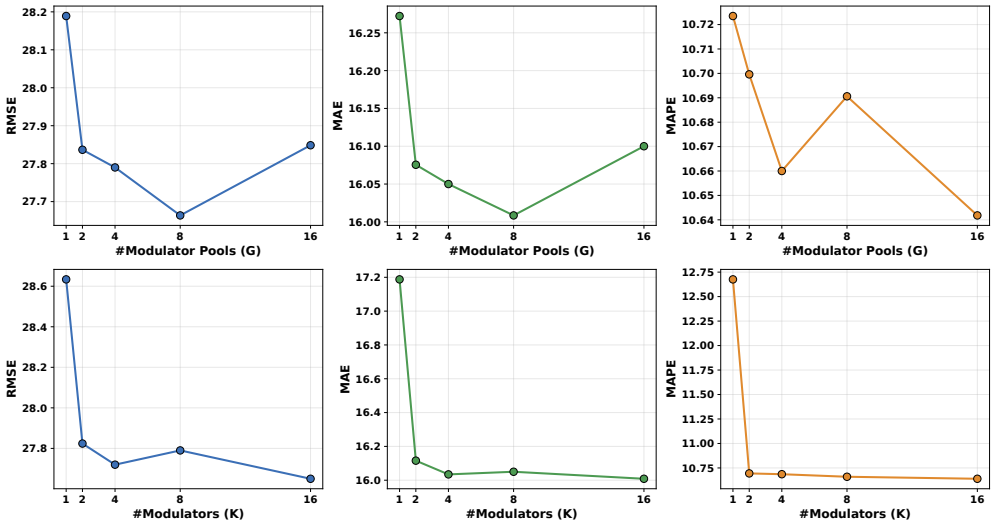


Figure 6: Hyperparameter study results. We report RMSE, MAE and MAPE of HEAM-V model on PEMS08 dataset with horizon of 48 under different number of modulator pools (G) and number of modulators per pool (K).

F Baseline and Backbone Details

We select the following baselines for comparison: (i) Graph-based methods model spatial dependencies through predefined or learned graphs, including DCRNN Li et al. (2018), AGCRN Bai et al. (2020), and GraphWaveNet Wu et al. (2019). (ii) Heterogeneity-aware methods assign different embeddings or parameters to different spatiotemporal states, including STID Shao et al. (2022), STAEformer Liu et al. (2023), HimNet Dong et al. (2024) and VisiFold Zhang et al. (2026). (iii) Decomposition-based methods explicitly separate the regular and aperiodic components, including STDN Cao et al. (2025), HyperD Shao et al. (2026), DualCast Su et al. (2025) and CycleNet Lin et al. (2024). We select strong attention-based model VisiFold and graph-based model GraphWaveNet as our backbone. However, VisiFold applies a temporal MLP over the time dimension and therefore does not produce per-timestep or per-patch representations, which HEAM requires for decomposition. We therefore replace the temporal transform with a PatchTST-style patch transformer and use the resulting variant, denoted VisiFold (P), as the host backbone. The original VisiFold is retained as a separate baseline in Table 1. For each host backbone, we report both the backbone itself and HEAM plugged into it, so that the improvement attributable to the module is directly readable. We adopt RMSE, MAE, and MAPE as evaluation metrics.

G Full Results on Improvement Distribution

Criteria of Peak Hours For METR-LA dataset, we select forecasting windows whose center falls into 7:30-9:00 and 16:00-17:30 as peak hour and other samples were called off-peak Hour samples. Because the average speed of all sensor reduces rapidly in this period, representing morning and evening rush hour. For PEMS08 dataset, the peak hours were selected as 6:30 8:00 and 16:00-17:30. For SOLAR-AL dataset, forecasting windows whose center fall into 12:00 15:00 were selected at peak hour, because aperiodic events are most consequential at this time.

Additional Results on Improvement Distribution We also report the performance reduction brought by HEAM under forecasting horizon of 36 and 24 in Figure 7, 8. As shown in the figure, under five of six scenarios of different forecasting horizon and datasets, the reduction brought

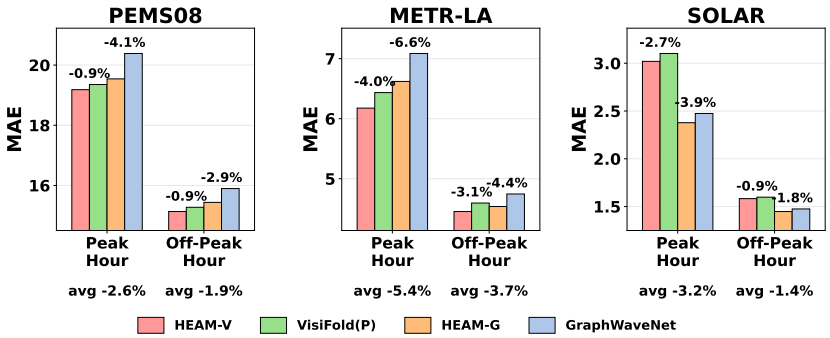


Figure 7: MAE reduction of HEAM equipped model over three datasets with a forecasting horizon of 36

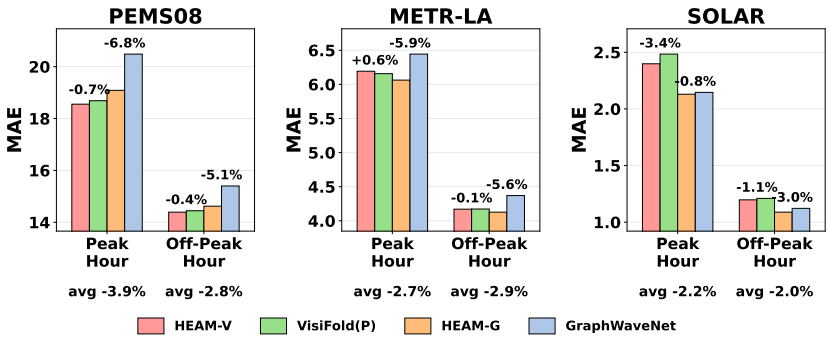


Figure 8: MAE reduction of HEAM equipped model over three datasets with a forecasting horizon of 24

by HEAM concentrates in peak hours where the impact of aperiodic events are more consequential. This once again demonstrates the importance of event-adaptive modeling mechanism of HEAM.