

SpikeSTAG: Spatial-Temporal Forecasting via GNN-SNN Collaboration

Bang Hu, Changze Lv, Mingjie Li, Yunpeng Liu, Xiaoqing Zheng, Fengzhe Zhang, Wei cao, Fan Zhang
 School of Computer Science, Fudan University, Shanghai, China

Abstract. Spiking neural networks (SNNs), inspired by the spiking behavior of biological neurons, offer a distinctive approach for capturing the complexities of temporal data. However, their potential for spatial modeling in multivariate time-series forecasting remains largely unexplored. To bridge this gap, we introduce a brand new SNN architecture, which is among the first to seamlessly integrate graph structural learning with spike-based temporal processing for multivariate time-series forecasting. Specifically, we first embed time features and an adaptive matrix, eliminating the need for predefined graph structures. We then further learn sequence features through the Observation (OBS) Block. Building upon this, our Multi-Scale Spike Aggregation (MSSA) hierarchically aggregates neighborhood information through spiking SAGE layers, enabling multi-hop feature extraction while eliminating the need for floating-point operations. Finally, we propose Dual-Path Spike Fusion (DSF) Block to integrate spatial graph features and temporal dynamics via a spike-gated mechanism, combining LSTM-processed sequences with spiking self-attention outputs, effectively improve the model accuracy of long sequence datasets. Experiments show that our model surpasses the state-of-the-art SNN-based iSpikformer on all datasets and outperforms traditional temporal models at long horizons, thereby establishing a new paradigm for efficient spatial-temporal modeling.

1 Introduction

Spiking Neural Networks (SNNs), recognized as the third generation of neural networks, excel in processing temporal data by virtue of their precise temporal coding and intrinsically low energy consumption[1]. Consequently, they have been widely adopted for tasks including image classification [2–6], visual processing [7, 8], sequential image recognition [9, 10], and time-series classification [11, 12]. Owing to their ability to encode and transmit information through discrete spikes, SNNs exhibit exceptional spatial-temporal modeling capabilities, achieving performance comparable to or exceeding that of traditional Artificial Neural Networks (ANNs) on short-sequence tasks. For long-sequence modeling, the potential of SNNs is being progressively unlocked through novel architectures[13, 14].

Although prior studies have showcased the remarkable temporal modeling capabilities of SNNs, a truly general-purpose architecture often needs both spatial and temporal reasoning [15–17] Current SNNs, however, still underperform conventional networks in capturing rich spatial structures. Recent efforts therefore seek to equip SNNs with stronger spatial inductive biases[18, 19], thereby broadening their applicability across diverse domains.

Multivariate time-series forecasting leverages intricate inter-variable dependencies to anticipate future dynamics, thereby empowering decision-makers to optimize resource allocation, mitigate risk, and devise more effective strategies. Its applications span finance, energy, transportation, and beyond, consistently enhancing the precision and ef-

ficiency of operational decisions. Reliable forecasts demand simultaneous modeling of temporal evolution and spatial interactions; substantial research has been devoted to this dual objective. One line of work enhances predictive accuracy by coupling convolutional neural networks (CNNs) with recurrent architectures, most notably through CNN–LSTM hybrids [20–22]. Another line of research integrates recurrent networks or attention mechanisms into deep learning frameworks to further improve temporal modeling [23, 24]. Alternatively, graph neural networks provide a distinct paradigm by explicitly modeling spatial dependencies among variables [25, 26].

Despite these advances, existing GNN-based approaches remain predominantly spatial-centric; their ability to characterize fine-grained temporal dynamics is still limited, leaving significant room for improvement in capturing long-range and multi-scale temporal patterns.

By converting continuous-valued time series into meaningful spike trains, recent work has pioneered a training paradigm that, for the first time, enables SNNs to tackle multivariate forecasting, unleashing their inherent temporal modeling prowess [27]. Yet this seminal study—along with most follow-ups—largely overlooks spatial dependencies. Subsequent efforts have concentrated solely on refining the temporal inductive biases of SNNs [28]. Consequently, all existing models privilege temporal optimization and neglect spatial representation, leaving a critical void in the joint modeling of spatial and temporal patterns for multivariate time-series prediction.

In this paper, we integrate GNNs’ spatial modeling with SNNs’ sequential processing to achieve enhanced accuracy in multivariate time-series forecasting. We propose Spiking Spatial-Temporal Adaptive Graph Neural Network(SpikeSTAG), a novel architecture that seamlessly integrates adaptive graph learning with spike-driven temporal computation to address the dual spatial-temporal demands of multivariate forecasting. As illustrated in Figure 1, the framework is decomposed into three conceptual stages: (1) Data Processing: an adaptive graph is inferred from the input series, while temporal covariates are fused with the original features to yield an enriched embedding; (2) SpikeSTAG Module: the embedded series together with the learned graph are fed into SpikeSTAG to extract high-dimensional spatial-temporal representations; (3) Temporal Forecasting: the extracted representations are finally mapped to the future series. In summary, our contributions can be summarized as follows:

- SpikeSTAG Framework: To the best of our knowledge, SpikeSTAG is the first framework that synergistically combines the spatial capabilities of GNNs with the temporal dynamics of SNNs for multivariate time-series forecasting.
- We propose the Multi-Scale Spike Aggregation (MSSA) module to enable floating-point-free graph updates, ensuring energy-efficient training and inference.
- We design the Dual-Path Spike Fusion(DSF) module, which employs a learnable gating mechanism to integrate the trend representations derived from LSTM with the feature learning capabilities of SSA, thereby markedly enhancing the accuracy of the model on long-sequence datasets.
- Extensive experiments show that SpikeSTAG achieves state-of-the-art accuracy among SNN-based models on four public benchmarks, rivaling ANN counterparts, thereby establishing a new competitive baseline.

2 Related Work

GNNs have demonstrated strong potential in time-series tasks—forecasting, classification, anomaly detection, and imputation—whenever data exhibit spatial or relational structure [29]. Numerous studies cast time series as graphs of interacting segments or motifs, leveraging GNNs to capture relational patterns and enable personalized search or recommendation. These methods, however, rely on predefined topologies, constraining their capacity to learn adaptive node connections and generalize across domains [30, 31]. In forecasting tasks, models such as STGCN, MTGNN, and AGCRN explicitly construct graphs—e.g., sensor networks in traffic prediction—to encode spatial dependencies among variables [32–34]. Yet their spatial-temporal modules remain loosely coupled, limiting the interaction between spatial and temporal representations and consequently impeding predictive performance.

On the other hand, SNNs, as biologically inspired models, exhibit strong potential in temporal sequence modeling. Recent efforts—including SpikeTCN, SpikeRNN, and Spike-Transformer—have extended SNNs to

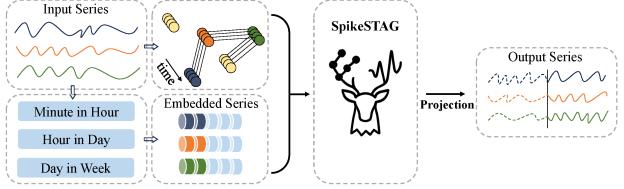


Figure 1. SpikeSTAG Framework Overview. The left four part represent the preprocessing of our model, including adaptive matrix learning and time node enhancement. The center is the SpikeSTAG main module, and the output is obtained through the prediction layer to generate the final prediction result.

time-series forecasting. These models first transform continuous signals into spike trains via dedicated encoders such as Delta or convolutional coders, and then leverage SNN neurons to capture temporal dynamics [27]. However, SNNs still face challenges in effectively integrating spatial and temporal capabilities, particularly in complex multivariate time series tasks, where both dimensions are critical for performance.

In recent years, SNNs have been increasingly investigated for graph-structured data [35–37]. Existing methods typically apply Poisson encoding to transform node features into spike trains and then perform graph message passing via SNN neurons, demonstrating viability on static-graph tasks such as node classification and link prediction. To address dynamic graphs, subsequent work has extended SNNs to temporal domains, exploiting the sparsity of spike encoding and spike-driven attention to enhance efficiency and scalability [38, 39]. Nevertheless, these approaches primarily emphasize structural exploration of the graph itself, neglecting the SNN’s inherent ability to model continuous-time dynamics.

While existing works innovatively introduce SNNs to graph domains for tasks like node classification and link prediction, they fail to leverage SNNs’ temporal dynamics for enhanced time modeling. Instead, SNNs merely replace traditional neurons in GNNs as computational units, resulting in decoupled spatial-temporal modeling. Furthermore, these approaches are restricted to structural inference tasks (e.g., node/graph classification) and overlook spatial-temporally coupled applications such as traffic flow or power load forecasting—neglecting SNNs’ inherent advantages in temporal processing.

To bridge this gap, we integrate SNNs with GNNs for multivariate time series forecasting. Targeting domains where spatial and temporal dependencies are intrinsically coupled, our approach establishes a novel event-driven baseline for complex spatial-temporal prediction tasks.

3 Preliminary

In this section, we begin by defining the multivariate time-series forecasting problem addressed in this work. We then review the LIF neuron model and the associated surrogate-gradient training algorithm.

3.1 Multi-variate Time-series Forecasting

We address the task of multivariate time-series forecasting. The input consists of N synchronous time series, each treated as graph nodes, observed over T historical time steps. At time step t , the N -dimensional observation is denoted by $\mathbf{Z}_t = [z_t^{(1)}, \dots, z_t^{(N)}]^\top$, where $z_t^{(i)}$ is the value of the i -th variable.

In addition to the raw measurements, we utilize three auxiliary temporal features: minute-of-hour, hour-of-day, and day-of-week. These temporal features are concatenated with the raw observations to form a new input vector $\mathbf{X}_t = [\mathbf{Z}_t^\top, \mathbf{m}_t^\top, \mathbf{h}_t^\top, \mathbf{d}_t^\top]^\top \in \mathbf{R}^{N+3}$, where $\mathbf{m}_t, \mathbf{h}_t, \mathbf{d}_t$ are embedded encodings of the respective temporal attributes. Our objective is, given the input sequence, to forecast the next L future values.

3.2 Spiking Neural Networks

In an SNN, one of the fundamental processing units is the Leaky Integrate-and-Fire (LIF) neuron [40], which models membrane potential as the charge stored on a capacitor. The neuronal dynamics are governed by three sequential operations: integration, firing, and reset.

$$\begin{cases} U[t] = I[t] + H[t - \Delta t], \\ S[t] = \Theta(U[t] - U_{\text{th}}), \\ H[t] = \beta U[t](1 - S[t]) + U_{\text{reset}}S[t]. \end{cases} \quad (1)$$

where $U[t]$ represents the membrane potential of the LIF neuron at time step t , and $I[t]$ denotes the input current at the same step. $H[t]$ is the internal state of the neuron, and Δt is the discrete time-step interval. The parameter U_{th} denotes the firing threshold, while U_{reset} is the reset potential applied after a spike occurs. The function $\Theta(\cdot)$ is the Heaviside step function, and β is the membrane potential decay coefficient that governs the leaky integration behavior.

To address the non-differentiability of the Heaviside step function $\Theta(\cdot)$ during back-propagation, we adopt a surrogate-gradient method that replaces the exact derivative with the derivative of an arctangent function $\sigma'(x) = \frac{\alpha}{2} \left(1 + \left(\frac{\pi}{2}\alpha x\right)^2\right)$, which exhibits a sigmoid-like shape and provides a biased but effective gradient estimator for end-to-end training [3].

4 Method

4.1 Model Architecture

SpikeSTAG is a spatial-temporal architecture that unites spiking neural computation with graph learning. Raw node features are first fused with auxiliary temporal features (minute-of-hour, hour-of-day, day-of-week) to form an enriched embedding; from this embedding, learnable node parameters induce an adaptive adjacency matrix. An Observation (OBS) Block refines each node’s representation across time, after which Multi-scale Spike Aggregation (MSSA) gathers fine-grained neighborhood information and uses spikes to aggregate. Finally, a Dual-path

Spike Fusion (DSF) layer translates the spike output into a sequence of hidden states via a lightweight LSTM, which is then discretized via spike encoding. The resulting spike trains are processed by the Spike Self-Attention (SSA) mechanism, yielding membrane potentials that are used to generate the final prediction.

The overall calculation process can be described by the following equations:

$$X_{\text{node}}, A, P = F(Z_{\text{node}}, T) \quad (2)$$

$$X_{\text{OBS}} = \text{OBS}(X_{\text{node}}, P) \quad (3)$$

$$S_{\text{MSSA}} = \text{SNN}(\text{MSSA}(X_{\text{OBS}}, A)) \quad (4)$$

$$H_{\text{DSF}} = \text{DSF}(S_{\text{MSSA}}) \quad (5)$$

$$Y = F_{\text{predict}}(H_{\text{DSF}}) \quad (6)$$

where X_{node} denotes encoded node features, A the learned adjacency matrix, and P temporal positional encodings.

4.2 Embedded Series and Adaptive Matrix

Embedded Series

In virtually all forecasting scenarios, temporal cues are indispensable for grasping the global context, and the most salient indicators revolve around “where” and “when”. Therefore, during temporal embedding, we enrich each node’s feature vector with dataset-specific side information—variables, hour-of-day, and day-of-week.

Adaptive Matrix

Learning an adjacency matrix directly from node features is non-trivial. In SpikeSTAG, we adopt the widely-used yet elegant source-target attention mechanism to build an adaptive graph [41]. The similarity between source node i and target node j is measured by an inner product and normalized into a link probability via the sigmoid function:

$$\mathbf{A} = \sigma(\mathbf{E}\mathbf{E}^\top) + \lambda \mathbf{I}, \quad (7)$$

where $\mathbf{E} \in \mathbf{R}^{N \times d}$ denote the learnable node embeddings; each row \mathbf{e}_i represents the latent feature vector of node i after fusing the raw node attributes with auxiliary temporal covariates. $\sigma(\cdot)$ is the element-wise sigmoid that converts unbounded similarities into link probabilities $\in (0, 1)$, ensuring a fully-adaptive, weighted adjacency. To further guarantee that a node’s own history remains the dominant contributor to its future state, we explicitly incorporate self-loops into the resulting matrix.

4.3 Observation Block

Figure 3 illustrates the Observation (OBS) Block, which refines node features that have already been embedded along the temporal dimension to yield time-augmented representations. However, at this stage, the node information remains insufficiently integrated; we therefore introduce a graph-attention mechanism to aggregate neighborhood information and learn the interrelations among

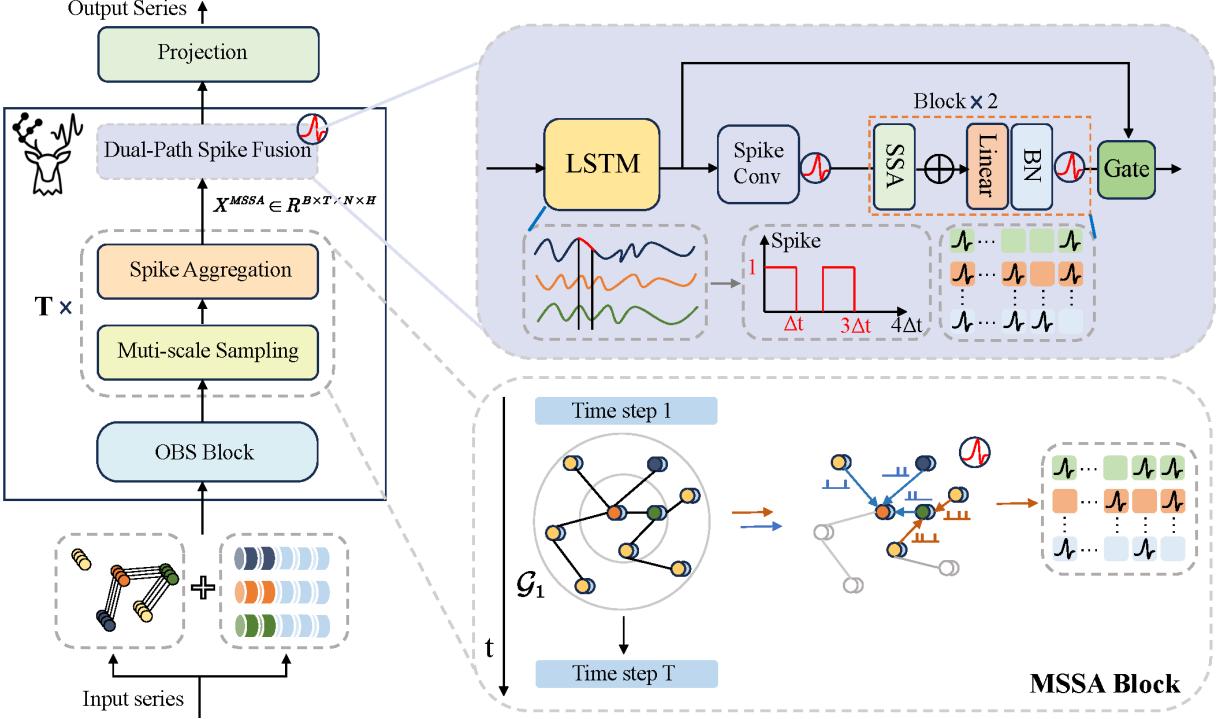


Figure 2. Detailed Architecture of SpikeSTAG. Given an input time series $\mathbf{Z} = z_1, z_2, \dots, z_T$ of length T , our goal is to predict the next L time steps $\mathbf{Y} = z_{T+1}, z_{T+2}, \dots, z_{T+L}$. The MSSA module first converts the OBS-enhanced sequence \mathbf{X}^{OBS} into a spatial-temporal spike sequence \mathbf{S} via multi-scale sampling and spike aggregation. To capture long-term dependencies, an LSTM restores \mathbf{S} to a continuous hidden-state sequence \mathbf{H} , after which a spike encoder discretizes \mathbf{H} into T_s spike time steps every Δt . The SSA then processes these spike trains to model inter-sequence relationships, and using a gating mechanism to fuse the results before the final prediction layer outputs \mathbf{Y} .

temporally embedded nodes across different timestamps. In contrast to standard attention, we restrict the computation to relationships within each node’s local neighborhood; the formula details are shown in Appendix A, where x'_i represents the enhanced model.

$$\mathbf{x}'_i = \mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{v}_j, \quad (8)$$

4.4 Multi-Scale Spiking Aggregation

Muti-scale Sampling

The Multi-Scale Spiking Aggregation (MSSA) module is proposed to refine the conventional GraphSAGE framework[42], which consists of two key components: importance-based sampling and spiking aggregation, as illustrated in Figure 2. Traditional layer-wise sampling directly feeds the raw adjacency matrix into the model, inevitably injecting noise and distracting the model from genuine relational signals. To suppress the interference of low-weight edges, we first prune the outgoing edges of node i via a learnable threshold, yielding a candidate set C_i composed of salient neighbors only.

$$C_i = \left\{ j \mid A_{i,j} > T_i \right\}, T_i = \frac{1}{A_{i,i}} \sum_{j \in \mathcal{N}(i)} A_{i,j}, \quad (9)$$

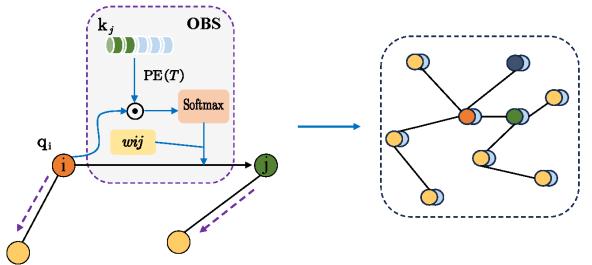


Figure 3. Details of OBS Block, which learns the enhanced relationships between nodes through a graph structure self-attention mechanism. The enhanced nodes contain both the original sequence features and time information.

where T_i is the average weight of the outgoing edges for node i , which adapts to the local neighborhood distribution without requiring any manually tuned hyperparameters.

Subsequently, we perform a perceptual-importance truncation $\text{Imp}_i = \sum_{j=1}^N A_{i,j}$, selecting the top- k highest-ranked nodes to form the neighborhood set for the current sampling layer. To further enlarge the receptive field with minimal computational overhead, we devise a two-level sampling paradigm—“local” followed by “semi-global”—which efficiently acquires a global receptive scope while maintaining sampling efficiency.

Hierarchical Spiking Aggregation

Hierarchical Spiking Aggregation enlarges the receptive field while guaranteeing both temporal causality and the binary sparsity of spike tensors. Concretely, we replace dense GEMM with Index-Mask Aggregation: whenever node features are one-bit spikes, the product \mathbf{AX} reduces to indexing active positions and summing their feature vectors. Arithmetic intensity thus falls from $O(|\mathcal{V}|^2)$ to $O(|\mathcal{E}_{\text{sample}}|)$, eliminating every dense matrix multiplication. For each node i the pre-synaptic potential of the i hop can be shown as followed:

$$\mathbf{m}_i^{(i)} = \sum_{j \in \mathcal{S}_i^{(i)}} \mathbf{x}_j \mathbf{W}^{(i)}, \quad \mathbf{W}^{(i)} \in \mathbb{R}^{T \times d_i}, \quad (10)$$

where $\mathbf{x}_j \in \{0, 1\}^T$ is the spike train of node j . This vector is then passed through an LIF layer. The resulting sparse spike trains are forwarded to downstream SNN or recurrent cells; the identical sample–index–sum–LIF routine is repeated for the two-hop neighborhood $\mathcal{S}_i^{(2)}$. Because the sampling budget $|\mathcal{S}_i^{(k)}| \ll N$, both memory footprint and runtime scale linearly with graph size, rendering the entire MSSA module entirely free of dense GEMM and conferring substantial empirical speed-ups.

$$\mathbf{s}_i^{(i)} = \Phi_{\text{LIF}}(\mathbf{m}_i^{(i)}) \in \{0, 1\}^{d_i}, \quad (11)$$

4.5 Dual-Path Spike Fusion

After processing by the MSSA block, we obtain a high-order representation S_{MSSA} that fuses the temporal dynamics of spiking neurons with the spatial expressiveness of graph neural networks.

To further leverage the efficient temporal-processing capability of SNNs, we feed the spike outputs into a lightweight LSTM module to generate hidden temporal information, forward it to the Spike Self-Attention (SSA) branch to obtain new spike sequences, and finally re-weight the outputs of the LSTM and SSA branches via a learnable, parameterized gating function, as illustrated in Figure 2.

Our designed lightweight LSTM module effectively captures long-term dependencies in time series data through its gating mechanism. Additionally, it performs continuous temporal modeling by flattening the pulse time series, ultimately producing a continuous time sequence of hidden states, which facilitates subsequent processing by the SSA module.

Before using SSA, we need to establish a precise correspondence between the time step ΔT of the time series and the time step Δt of the SNN. Specifically, we use a spike encoding technique that divides Δt uniformly into T_s segments, where each segment allows neurons with membrane potentials exceeding the threshold to generate a spike event. $\Delta T = T_s \cdot \Delta t$ this equation aligns the time steps ΔT in the time series with the time steps Δt in the SNN, ensuring that the independent variable t in the time series (denoted as $X(t)$) and the independent variables t in the SNN (denoted as $U(t), I(t), H(t), S(t)$) are semantically equivalent.

Datasets	# Samples	# Nodes	Sample Rate	Input Length	Output Length
Solar	52,560	137	10 minutes	168	1
Electricity	26,304	321	1 hour	168	1
METR-LA	34,272	207	5 minutes	12	12
PEMS-BAY	52,116	325	5 minutes	12	12

Table 1. Summary of Datasets. The solar and electricity datasets are long sequence datasets with an input length greater than 100, while METR-LA and PEMS-BAY are short sequence datasets.

Following temporal alignment operations, we execute the SSA module defined as:

$$\text{SSA}(X) = \text{SpikeSoftmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (12)$$

This mechanism amplifies micro-scale temporal features, ultimately generating event-aware representations through membrane potential decoding.

The parameterized fusion gate constitutes the core innovation of our Dual-Path Spike Fusion framework, addressing the fundamental challenge in spatial-temporal forecasting: simultaneous integration of continuous temporal evolution and discrete event-driven dynamics. The gating function operates as, where $\mathbf{G} = \sigma(\mathbf{W}[\mathbf{H}_{\text{LSTM}}; \mathbf{H}_{\text{SSA}}])$:

$$\mathbf{H}_{\text{fused}} = \mathbf{G} \odot \mathbf{H}_{\text{LSTM}} + (\mathbf{1} - \mathbf{G}) \odot \mathbf{H}_{\text{SSA}} \quad (13)$$

This mechanism enables contextual specialization by dynamically modulating contributions: emphasizing LSTM’s smooth dynamics during stable periods when $\mathbf{G} \rightarrow \mathbf{1}$, while prioritizing SSA’s event responsiveness during disruptions when $\mathbf{G} \rightarrow \mathbf{0}$.

Furthermore, it establishes a unified representation space for cross-scale interaction, where macro-scale trends, including periodic flows and daily patterns, coherently interact with micro-scale events such as accidents and sudden congestion through learned attention weights. Crucially, this framework preserves the complementary advantages of both continuous and event-driven temporal modeling paradigms.

5 Experiments

In this section, SpikeSTAG is evaluated from two complementary perspectives: predictive performance and theoretical energy consumption. First, its multivariate time-series forecasting capability is benchmarked against state-of-the-art diverse-time-series models on four reference datasets. Subsequently, the theoretical energy expenditure is quantified on the Electricity dataset.

5.1 Datasets

Table 1 summarizes the benchmark datasets employed. Long-sequence datasets consist of Electricity (hourly power consumption in kWh) and Solar (photovoltaic generation logs). Short-sequence datasets comprise METR-LA (mean traffic speed on Los Angeles County highways) and PEMS-BAY (mean traffic speed in the San Francisco

Method	Metric	Metr-la					Pems-bay					Solar					Electricity					Avg
		3	6	12	24	48	3	6	12	24	48	3	6	12	24	48	3	6	12	24	48	
SpikeTCN	$R^2 \uparrow$.845	.799	.718	<u>.602</u>	<u>.464</u>	.862	.829	.782	.681	.368	.946	.937	.893	.840	.708	.974	.970	.968	.963	.958	.807
	$RS E \downarrow$.415	.473	.560	<u>.665</u>	.772	.401	.448	.504	.582	.837	.205	.252	.409	.541	.591	.324	.333	.338	.342	.368	.468
SpikeRNN	$R^2 \uparrow$.784	.731	.661	<u>.557</u>	.440	.763	.721	.710	.693	.375	.933	.923	.903	.820	.812	.984	.978	.979	.964	.962	.785
	$RS E \downarrow$.490	.547	.614	.702	.789	.527	.571	.582	.599	.816	.246	.278	.343	.425	.435	.207	.280	.314	.317	.338	.471
iSpikformer	$R^2 \uparrow$.805	.765	.723	<u>.549</u>	.369	<u>.935</u>	<u>.884</u>	.787	.622	.348	.972	<u>.955</u>	.918	.869	.795	.982	.974	.973	.974	.972	.808
	$RS E \downarrow$.466	.512	.555	.709	.838	<u>.276</u>	<u>.369</u>	<u>.499</u>	.665	.873	.217	.218	.295	.372	.333	.214	.284	.284	.284	.338	.430
SpikeSTAG	$R^2 \uparrow$.873	<u>.822</u>	<u>.734</u>	.590	.405	.874	.835	.787	.626	.403	.973	.950	.926	.879	.836	.987	.986	.985	.984	.981	.823
	$RS E \downarrow$.375	<u>.430</u>	<u>.535</u>	.685	.834	.384	.439	.537	.661	.835	.246	.272	.315	.390	<u>.333</u>	<u>.207</u>	<u>.222</u>	<u>.224</u>	<u>.225</u>	<u>.285</u>	<u>.420</u>
GRU	$R^2 \uparrow$.803	.761	.682	.614	.324	.783	.769	.696	<u>.696</u>	.377	.962	.950	.907	.875	.781	.983	.981	.980	.972	.971	.793
	$RS E \downarrow$.448	.507	.585	.663	.833	.479	.504	.638	<u>.638</u>	<u>.818</u>	<u>.508</u>	<u>.548</u>	.569	.572	.583	.518	.522	.531	.506	.598	.578
iTTransformer	$R^2 \uparrow$	<u>.864</u>	.849	.763	.538	.379	.938	.888	.797	.629	.356	<u>.974</u>	.964	.918	<u>.879</u>	<u>.799</u>	<u>.983</u>	<u>.977</u>	<u>.977</u>	<u>.977</u>	<u>.975</u>	<u>.821</u>
	$RS E \downarrow$	<u>.344</u>	.410	.514	<u>.652</u>	<u>.783</u>	<u>.269</u>	<u>.362</u>	<u>.488</u>	.659	.868	.562	.584	.575	<u>.541</u>	<u>.438</u>	<u>.213</u>	<u>.506</u>	<u>.460</u>	<u>.305</u>	<u>.335</u>	<u>.493</u>

Table 2. Experimental results of time series prediction with different prediction lengths L on 4 benchmark datasets. The upper part is for the SNN time series prediction network, and the lower part is for the traditional time series prediction network. All optimal results are highlighted in bold, and the best results for SNN time series or traditional time series are underlined. An \uparrow (\downarrow) indicates that higher (lower) values are better. All results are the averages of 3 random seeds.

Bay Area). For datasets whose sampling granularity is one minute, the auxiliary-node features comprise three signals—minute-of-hour, hour-of-day, and day-of-week. Conversely, for datasets sampled at hourly intervals, only hour-of-day and day-of-week are provided. As evaluation metrics, we adopt the root relative squared error (RSE) and the coefficient of determination (R^2). Owing to their invariance to the absolute numerical scale of the data, these two measures are preferred over mean squared error (MSE) or mean absolute error (MAE) in time-series forecasting studies.

5.2 Main Results

Experimental results demonstrate that the proposed model achieves an average coefficient of determination (R^2) of 0.818 across the four benchmark datasets (METR-LA, PEMBAY, Solar, and Electricity), surpassing all other SNN-based architectures (e.g., 0.808 for iSpikformer) and matching the ANN-based iTTransformer (0.818). Notably, SpikeSTAG exhibits superior robustness on long-sequence forecasting tasks: under a 48-step prediction horizon, it attains R^2 values of 0.981 and 0.836 on Electricity and Solar, respectively, representing improvements of 0.009 and 0.041 over iSpikformer. This confirms that the explicit spatial-dependency modeling by the GNN effectively mitigates the performance degradation commonly observed in conventional SNNs during long-sequence modeling.

SpikeSTAG’s explicit spatial modeling yields robust generalization across domains. On spatially correlated data, the model surpasses iSpikformer notably. For the PEMBAY traffic network, SpikeSTAG records an $RS E$ of 0.661 at the 24-step horizon, 0.6 percent lower than iSpikformer’s 0.665. When the horizon extends to 48 steps, SpikeSTAG raises its R^2 to 0.403, exceeding iSpikformer’s 0.348 by 15.8 percent. These outcomes confirm SpikeSTAG as the state of the art among spiking temporal forecasting networks.

Against traditional sequence predictors, SpikeSTAG exhibits clear advantages on long-horizon tasks. On Electricity, the model delivers superior performance across all metrics. On Solar, iTTransformer suffers pronounced error accumulation, yielding an $RS E$ of 0.438 at the 48-step mark, whereas SpikeSTAG constrains the $RS E$ to 0.333 via joint spatial-temporal optimization.

5.3 Hyper-parameter Sensitivity

As an SNN-based architecture, SpikeSTAG treats the time-step T_s as a hyper-parameter that dictates the granularity at which temporal dynamics Δt are modeled. Experiments were conducted on METR-LA with a prediction horizon $L = 3$ while varying $T_s \in \{4, 8, 12, 16\}$. Figures 4(a)–4(b) reveal that a modest increase in the number of time steps yields a slight rise in the R^2 index because finer temporal resolution becomes achievable, yet the overall impact on network performance remains negligible; SpikeSTAG thereby demonstrates remarkable stability.

When $T_s = 16$, R^2 exhibits a marginal decline. This observation aligns with the “self-accumulating dynamics”[12], in which surrogate-gradient-induced error accumulation can precipitate vanishing or exploding gradients and consequently degrade model efficacy. The complementary $RS E$ metric displays an inverse trend. Consequently, in most experimental settings, a smaller T_s can be adopted to further reduce computational energy without sacrificing accuracy.

5.4 Ablation Study

To rigorously establish both the effectiveness and necessity of the proposed DSF, we conduct ablation studies on METR-LA and report the complete quantitative results. We conducted experiments under four distinct configurations : (W1) removal of the SSA module, where a

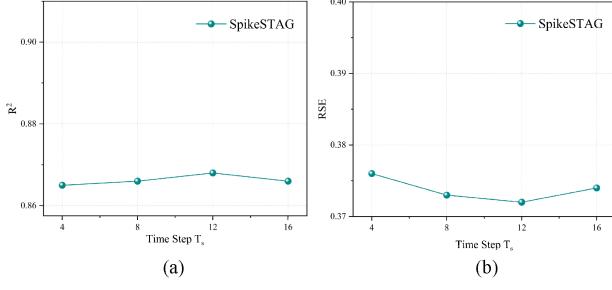


Figure 4. The impact of the time step T_s , a key hyperparameter, in SNN: (a) shows the R^2 values for different time steps, and (b) shows the RSE values for different time steps.

lightweight LSTM directly generates the temporal forecast; (w2) direct forwarding of the spiking outputs from the MSSA module to the SSA module, bypassing intermediate processing; (w3) elimination of the gating-based fusion, whereby the features are successively processed by the lightweight LSTM and the SSA module to produce the final prediction; and (W4) the intact DSF module.

Dataset	W1		W2		W3		W4	
	R^2	RSE	R^2	RSE	R^2	RSE	R^2	RSE
Metr-la	0.823	0.436	0.854	0.417	<u>0.858</u>	<u>0.412</u>	0.868	0.375
Pems-bay	0.831	0.424	<u>0.860</u>	0.417	0.854	<u>0.413</u>	0.874	0.384
Solar	0.950	0.258	0.963	0.251	0.964	<u>0.251</u>	0.973	0.246
Electricity	0.960	0.246	0.965	0.223	<u>0.966</u>	<u>0.214</u>	0.987	0.207

Table 3. Performance metrics under the four architectural configurations are reported below: W1 denotes the lightweight LSTM alone, W2 the SSA module alone, W3 the concatenation of lightweight LSTM and SSA without gating, and W4 the proposed method that fuses LSTM and SSA via a gating mechanism. Optimal results are highlighted in boldface, and second-best results are underlined.

Analysis of Table 3 demonstrates that the gating fusion mechanism confers a decisive advantage in temporal forecasting. Across the four benchmarks, the architecture that employs gating to combine LSTM with SSA (W4) uniformly surpasses all alternatives, attaining the highest R^2 of 0.868, 0.874, 0.973 and 0.987 on METR-LA, PEMSBAY, Solar and Electricity, respectively, while simultaneously delivering the lowest RSE of 0.375, 0.384, 0.246 and 0.207. The benefit is most pronounced on complex dynamic systems such as traffic data, where the gating mechanism yields an R^2 improvement of 1.0–1.4% over straightforward concatenation (W3). This gain stems from the gate’s capacity to adaptively balance LSTM’s sequential modeling strength with SSA’s spike-based feature extraction, thereby mitigating feature conflicts.

The ablations also highlight the intrinsic merit of each module. The SSA-only variant (W2) outperforms the LSTM-only one (W1) by 3.1–4.3% on three benchmarks, underscoring the superiority of spiking neural representations for raw temporal signals. The concatenated design (W3) emerges as the second-best choice on periodic

datasets such as Solar and Electricity, confirming the complementary value of the dual modules. Notably, on PEMSBAY the performance of W3 drops below that of W2, revealing the risks of unconstrained feature aggregation.

5.5 Energy Reduction

SpikeSTAG simultaneously exploits the temporal dynamics inherent to spiking neural networks and the spatial modeling capacity of graph neural networks, while inheriting the energy-efficient characteristics of SNNs. In this section, we map SpikeSTAG iSpikformer and iTransformer onto 45 nm neuromorphic hardware and derive their theoretical energy expenditure[43]. All measurements are obtained by executing inference on the Electricity dataset with a prediction horizon of $L = 3$.

Model	Param (M)	Ops (G)	Energy (mJ)	Energy Reduction	R^2
SpikeSTAG	1.566	3.75	4.39	53.64%	0.987
iSpikformer	1.634	3.55	3.19	66.30%	0.983
iTransformer	1.634	2.05	9.47	/	0.983

Table 4. Energy consumption comparison across networks. The table reports parameter counts and energy expenditure for SpikeSTAG, iSpikformer, and iTransformer. Energy reduction is calculated relative to iTransformer as the baseline.

Our SpikeSTAG achieves 53.6% energy reduction compared to iTransformer (4.39mJ versus 9.47mJ), demonstrating the efficacy of spiking graph architectures. Although its 3.75G OPs are higher than iTransformer’s 2.05G, the spike-driven event-based computation enables lower energy consumption.

6 Conclusion

In this paper, we present a novel spiking neural network architecture that unifies graph structural learning with spike-based temporal processing for multivariate time-series forecasting. By introducing Observation (OBS) Block and using Adaptive connection matrix, our model dynamically constructs graph topologies through learnable node embeddings, enabling spike-aware message passing without relying on predefined graphs. Additionally, the proposed Multi-Scale Spike Aggregation (MSSA) module hierarchically captures multi-hop spatial dependencies using spiking SAGE layers in a fully event-driven manner. To further enhance spatial-temporal integration, we design a spike-gated fusion mechanism that combines LSTM-based sequence modeling with spiking self-attention, named Dual-Path Spike Fusion(DSF). Extensive experiments demonstrate that our method consistently outperforms existing SNN and traditional models across multiple benchmarks, offering a biologically inspired and energy-efficient solution for complex spatial-temporal forecasting tasks

References

- [1] W. Maass, Neural Networks **10**, 1659 (1997)

- [2] Y. Hu, H. Tang, G. Pan, IEEE Transactions on Neural Networks and Learning Systems **34**, 5200 (2021)
- [3] W. Fang, Z. Yu, Y. Chen et al., Advances in Neural Information Processing Systems **34**, 21056 (2021)
- [4] J. Ding, Z. Yu, Y. Tian et al., arXiv preprint arXiv:2105.11654 (2021)
- [5] Z. Zhou, Y. Zhu, C. He et al., arXiv preprint arXiv:2209.15425 (2022)
- [6] M. Yao, J. Hu, Z. Zhou et al., Advances in Neural Information Processing Systems **36**, 64043 (2023)
- [7] B. Meftah, O. Lezoray, A. Benyettou, Neural Processing Letters **32**, 131 (2010)
- [8] M.J. Escobar, G.S. Masson, T. Vieville et al., International Journal of Computer Vision **82**, 284 (2009)
- [9] A. Jeffares, Q. Guo, P. Stenetorp et al., arXiv preprint arXiv:2110.02865 (2021)
- [10] X. Chen, J. Wu, H. Tang et al., arXiv preprint arXiv:2308.15150 (2023)
- [11] J.P. Dominguez-Morales, Q. Liu, R. James et al., *Deep Spiking Neural Network Model for Time-Variant Signals Classification: A Real-Time Speech Recognition Approach*, in *2018 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2018), pp. 1–8
- [12] H. Fang, A. Shrestha, Q. Qiu, *Multivariate Time Series Classification Using Spiking Neural Networks*, in *2020 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2020), pp. 1–7
- [13] S. Shen, C. Wang, R. Huang et al., *SpikingSSMs: Learning Long Sequences with Sparse and Parallel Spiking State Space Models*, in *Proceedings of the AAAI Conference on Artificial Intelligence* (2025), Vol. 39, pp. 20380–20388
- [14] Y. Zhong, R. Zhao, C. Wang et al., arXiv preprint arXiv:2410.17268 (2024)
- [15] A. Ain, A.R. Zamir, S. Savarese et al., *Structural-RNN: Deep Learning on Spatio-Temporal Graphs*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 5308–5317
- [16] S. Yan, Y. Xiong, D. Lin, *Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition*, in *Proceedings of the AAAI Conference on Artificial Intelligence* (2018), Vol. 32
- [17] A.A. Musa, A. Hussaini, W. Liao et al., Future Internet **15**, 199 (2023)
- [18] T. Zhang, K. Yu, X. Zhong et al., *STAA-SNN: Spatial-Temporal Attention Aggregator for Spiking Neural Networks*, in *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 13959–13969
- [19] C. Yu, Z. Gu, D. Li et al., Frontiers in Neuroscience **16**, 1079357 (2022)
- [20] X. Jin, X. Yu, X. Wang et al., *Prediction for Time Series with CNN and LSTM*, in *Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019)* (Springer Singapore, 2019), pp. 631–641
- [21] I.E. Livieris, E. Pintelas, P. Pintelas, Neural Computing and Applications **32**, 17351 (2020)
- [22] H. Widiputra, A. Mailangkay, E. Gautama, Complexity **2021**, 9903518 (2021)
- [23] J. Zhang, K.F. Man, *Time Series Prediction Using RNN in Multi-Dimension Embedding Phase Space*, in *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics (SMC'98)* (IEEE, 1998), Vol. 2, pp. 1868–1873
- [24] Y. Liu, T. Hu, H. Zhang et al., arXiv preprint arXiv:2310.06625 (2023)
- [25] K. Yi, Q. Zhang, W. Fan et al., Advances in Neural Information Processing Systems **36**, 69638 (2023)
- [26] D. Cao, Y. Wang, J. Duan et al., Advances in Neural Information Processing Systems **33**, 17766 (2020)
- [27] C. Lv, Y. Wang, D. Han et al., arXiv preprint arXiv:2402.01533 (2024)
- [28] F. Shibo, W. Feng, X. Gao et al., *TS-LIF: A Temporal Segment Spiking Neuron Network for Time Series Forecasting*, in *Proceedings of the Thirteenth International Conference on Learning Representations* (2025)
- [29] M. Jin, H.Y. Koh, Q. Wen et al., IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
- [30] C. Yang, W. Xue, S. Zhou et al., Proceedings of the VLDB Endowment **12**, 2094 (2019)
- [31] A. Pal, C. Eksombatchai, Y. Zhou et al., *Pinner-Sage: Multi-Modal User Embedding Framework for Recommendations at Pinterest*, in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2020), pp. 2311–2320
- [32] H. Han, M. Zhang, M. Hou et al., *STGCN: A Spatial-Temporal Aware Graph Learning Method for POI Recommendation*, in *2020 IEEE International Conference on Data Mining (ICDM)* (IEEE, 2020), pp. 1052–1057
- [33] Z. Wu, S. Pan, G. Long et al., *Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks*, in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 753–763
- [34] L. Bai, L. Yao, C. Li et al., Advances in Neural Information Processing Systems **33**, 17804 (2020)
- [35] D. Dold, J.S. Garrido, *Spike: Spike-Based Embeddings for Multi-Relational Graph Data*, in *2021 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2021), pp. 1–8
- [36] V.C. Chian, M. Hildebrandt, T. Runkler et al., *Learning Through Structure: Towards Deep Neuromorphic Knowledge Graph Embeddings*, in *2021 International Conference on Neuromorphic Computing (ICNC)* (IEEE, 2021), pp. 61–70
- [37] M. Xu, Y. Wu, L. Deng et al., arXiv preprint arXiv:2107.06865 (2021)
- [38] J. Li, Z. Yu, Z. Zhu et al., *Scaling Up Dynamic Graph Representation Learning via Spiking Neural Networks*, in *Proceedings of the AAAI Conference on*

- Artificial Intelligence* (2023), Vol. 37, pp. 8588–8596
- [39] Y. Sun, D. Zhu, Y. Wang et al., arXiv preprint arXiv:2403.15480 (2024)
 - [40] W. Gerstner, W.M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity* (Cambridge University Press, 2002)
 - [41] Z. Wu, S. Pan, G. Long et al., arXiv preprint arXiv:1906.00121 (2019)
 - [42] W. Hamilton, Z. Ying, J. Leskovec, Advances in Neural Information Processing Systems **30** (2017)
 - [43] M. Horowitz, *1.1 Computing’s Energy Problem (and What We Can Do About It)*, in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (IEEE, 2014), pp. 10–14