SPATIO-TEMPORAL SPECTRAL BACKPROPAGATION FOR TRAINING SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking Neural Networks (SNNs) offer significant advantages in time-series modeling due to their event-driven nature and biological plausibility, yet their training remains challenging. In existing backpropagation algorithms, parameters are deeply coupled within the temporal dimension, hindering the quantification of individual influences and limiting the analytical depth of gradient dynamics and theoretical interpretability. This paper proposes the Spatio-Temporal Spectral Backpropagation (STSBP) framework, which decomposes SNN error propagation in the spatio-temporal frequency domain into a product of four terms: (1) a spatial frequency kernel dependent on network structure and surrogate gradients; (2) a temporal frequency kernel governed by the leakage factor β; (3) an error spatio-temporal spectrum derived from task errors; and (4) an input signal spectrum. The framework reveals that SNNs prioritize low-frequency components in the spatial domain. In the temporal domain, the learning order is jointly determined by the temporal frequency kernel, the spatial-frequency integral of the error spectrum, and the input signal spectrum. Based on this theory, this paper presents the Frequency-LIF (Frequency Leaky Integrate-and-Fire) model, which comprises two core modules: the Spatial Frequency Dynamic Adjustment Mechanism (SFDAM) and the Temporal Frequency Dynamic Adjustment Mechanism (TFDAM). In the spatial domain, SFDAM detects low-frequency dominance, then attenuates low-frequency weights and enhances high-frequency learning, guiding the model from global trends to local details. In the temporal domain, TFDAM exponentially decays the leakage factor β over timesteps, steering the model from initial low-frequency focus toward high-frequency learning. Experimental results on eight time-series prediction benchmark datasets demonstrate that Frequency-LIF outperforms current state-of-the-art methods. The relevant code is available at: `https://anonymous.4open.science/r/Frequency-LIF-86F4`.

1 INTRODUCTION

Spiking Neural Networks (SNNs) have attracted broad interest in time-series prediction and low-power AI Feng et al. (2025); Wunderlich & Pehle (2021); Li et al. (2023); Hu et al. (2024a) due to their event-driven computation and inherent temporal encoding. Unlike Artificial Neural Networks (ANNs), SNNs transmit information via sparse spikes over time Hu et al. (2024b); He et al. (2025), enabling more natural capture of temporal dependencies and offering unique advantages in time-series forecasting Gast et al. (2024); Zhu et al. (2024). However, training SNNs is challenging because the spike firing function is non-differentiable, precluding standard gradient-based optimization Lee et al. (2016); Tavanaei & Maida (2016). To overcome this, surrogate gradient methods approximate the derivative of the step function with a smooth function during backpropagation while retaining it in the forward pass Neftci et al. (2019); Gygax & Zenke (2025), enabling efficient SNN training via backpropagation. Among neuron models, the Leaky Integrate-and-Fire (LIF) model is widely adopted for its computational efficiency and biological plausibility Datta et al. (2024); Wan et al..

Existing surrogate-gradient-based training methods, such as SuperSpike Zenke & Ganguli (2018), STBP Wu et al. (2018), and other proxy gradient algorithms Liu et al. (2025), predominantly utilize LIF neurons and rely on Backpropagation Through Time (BPTT) for parameter updates. However, these methods encounter a critical bottleneck: parameters are deeply coupled within the temporal

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.

domain Jiang et al. (2026); Kook et al. (2026). Specifically, the leakage factor β and surrogate gradient shape parameter α are deeply entangled in BPTT recursion. This coupling hinders analytical interpretability of gradient dynamics and quantification of individual parameter influences, rendering hyperparameter tuning a black box. For instance, it remains difficult to assess how α or β variations directly affect learning and network responses, limiting theoretical insights into SNN mechanisms and task-specific optimization. Therefore, this research investigates methodologies for mitigating deep parameter coupling in BPTT and quantifies the independent contribution of each parameter to the learning behavior of the network.

This paper introduces the Spatio-Temporal Spectral Backpropagation (STSBP) theoretical framework, which decouples error propagation in SNN training into the spatio-temporal frequency domain. By integrating spatial Fourier transforms and Discrete-Time Fourier Transform (DTFT), this framework decomposes the error backpropagation gradient into a product of four independent terms: the spatial frequency kernel, the temporal frequency kernel, the error spatio-temporal spectrum and the input signal spectrum. The spatial frequency kernel depends on the network topology and the surrogate gradient parameter α; the temporal frequency kernel is primarily controlled by the neuron leakage factor β; the error spatio-temporal spectrum is determined by task data and network output; and the input signal spectrum depends on the input data. This theory reveals that SNNs act as spatial low-pass filters in the spatial frequency domain, while their frequency preference in the temporal frequency domain is jointly driven by the temporal frequency kernel, the integral of the error spatio-temporal spectrum over spatial frequency, and the input signal spectrum.

Based on this theoretical framework, this paper introduces the Frequency-LIF (Frequency Leaky Integrate-and-Fire) model, which dynamically modulates the preference of the network for spatio-temporal frequencies throughout the training process. In the spatial frequency domain, SFDAM continuously monitors the learning status of the network across various spatial frequency components. The network exhibits a pronounced preference for low-frequency components during the early training phase. Upon detecting that low-frequency components dominate the learning process, SFDAM promptly attenuates their corresponding weights while intensifying high-frequency learning, thereby steering the model from global trend extraction toward local detail refinement. Regarding the temporal frequency dimension, the Temporal Frequency Dynamic Adjustment Mechanism (TFDAM) introduces an exponential decay function to dynamically adjust the neuron leakage factor β. In the initial stage, a larger β guides the network to focus on low-frequency features of the time series; as the time step increases, β decays exponentially, prompting the network to transition toward the extraction and learning of high-frequency information.

The main contributions of this paper are summarized as follows:

- Through a theoretical derivation, this paper proposes the STSBP framework, which decouples the error backpropagation of SNNs into the spatio-temporal frequency domain by factorizing gradients into spatial and temporal kernels, error spectra and input signal spectrum. This approach mitigates the coupling of parameters and enables the quantification of the independent influence of parameters within BPTT, providing theoretical insights for the learning mechanisms of SNNs and the design of models.
- Based in the theoretical analysis of STSBP, this paper proposes the Frequency-LIF model. In the spatial domain, a critical-point switching mechanism dictates the order of learning from low to high frequencies. Simultaneously, in the temporal domain, the exponential decay of β steers the evolution of learning from long-term dependencies toward transient events.
- Experiments on eight time-series prediction benchmark datasets demonstrate that the performance of Frequency-LIF outperforms current state-of-the-art methods, which validates the effectiveness of the theoretical framework proposed in this paper.

2 Theoretical Framework

2.1 Preliminaries and Motivation

LIF neurons have become the most widely adopted model in spiking neural networks due to their biological plausibility and low computational complexity. In discrete-time implementations, the

dynamics of LIF neurons are typically described as follows:

$$\begin{cases} v_j[t] = \beta v_j[t-1] + c[t] \\ o_j[t] = H(v_j[t] - V_{th}) \end{cases} \tag{1}$$

where $v_j[t]$ represents the membrane potential of neuron j at time step t, $c[t]$ is the input current, o_j denotes the output spike, and β is the decay factor. The output spike is determined by the step function $H(\cdot)$: when $x \geq 0$, $H(x) = 1$, and otherwise 0. This function determines whether the membrane potential reaches the threshold V_{th} to trigger a spike.

Training SNNs typically adopts a scheme of BPTT combined with surrogate gradients to overcome the challenge of the non-differentiability of the spike function. Under this framework, the error signal $d_j[t] = \partial \mathcal{L}/\partial v_j[t]$ of neuron j at time step t backpropagates along both spatial and temporal dimensions. Its recursive dynamics can be expressed as:

$$d_j[t] = \frac{\partial \mathcal{L}[t]}{\partial o_j[t]} \cdot g(v_j[t] - V_{th}) + d_j[t+1] \cdot \beta, \tag{2}$$

where $g(\cdot)$ is a surrogate gradient function parameterized by α, used to approximate the derivative of the step function.

From the above recursive formulation, it can be observed that the propagation of the error signal $d_j[t]$ is governed by multiple factors. First, the error signal at the current timestep originates from the superposition of two pathways: one is the combination of the current output's contribution to the loss $\partial \mathcal{L}[t]/\partial o_j[t]$ with the surrogate gradient $g(\cdot)$, where α regulates the shape and magnitude of the gradient; the other is the attenuated feedback of the future error $d_j[t+1]$ through the decay factor β. Second, the contributions of these two pathways are coupled additively within the same equation, making it difficult to isolate the independent effects of α and β on the error. Furthermore, the term $\partial \mathcal{L}[t]/\partial o_j[t]$ itself depends on the input data $c[t]$, network weights, and the specific form of the task loss function. These factors, through the recursive expansion of BPTT, further intertwine with α and β, forming intricate dependency relationships.

This multi-level coupling complicates the quantification of individual parameter contributions and hinders in-depth theoretical analysis of the SNN learning process. Consequently, hyperparameter tuning remains heavily reliant on empirical trial-and-error, lacking clear theoretical guidance.

2.2 Spatio-Temporal Spectral Backpropagation

This section establishes a frequency-domain theoretical framework to provide a deep understanding of the training process in SNNs. First, consider a Deep Neural Network (DNN) featuring a single hidden layer, a one-dimensional input x, and a one-dimensional output:

$$\Gamma(x) = \sum_{j=1}^{N} a_j \sigma(w_j x + b_j), \quad w_j, b_j, a_j \in \mathbb{R} \tag{3}$$

where $\Gamma(x)$ denotes the network output, w_j and a_j represent the model weight and output weight, respectively, b_j is the bias term, and $\sigma(\cdot)$ is the activation function. The Fourier transform of $\sigma(w_j x + b_j)$ is given by:

$$\mathcal{F}[\sigma(w_j x + b_j)](k) = \frac{1}{|w_j|} e^{\frac{ikb_j}{w_j}} \mathcal{F}[\sigma]\left(\frac{k}{w_j}\right) \tag{4}$$

where $\mathcal{F}$ denotes the Fourier transform and k represents the frequency in the spatial frequency domain. Summing the contributions of all neurons yields:

$$\mathcal{F}[\Gamma](k) = \sum_{j=1}^{N} a_j \left[\frac{1}{|w_j|} e^{\frac{ikb_j}{w_j}} \mathcal{F}[\sigma]\left(\frac{k}{w_j}\right)\right] \tag{5}$$

Defining the Fourier transform of the target function $f(x)$ as $\mathcal{F}[f](k)$, the error spectrum follows:

$$E(k) = \mathcal{F}[\Gamma](k) - \mathcal{F}[f](k) = A(k) e^{i\theta(k)} \tag{6}$$

where $A(k) = |E(k)|$ is the magnitude spectrum and $\theta(k) \in [-\pi, \pi]$ is the phase spectrum. According to the Theorem of Parseval, the mean squared error in the temporal domain is equivalent to the energy integral in the frequency domain:

$$\mathcal{L} = \frac{1}{2}\int_{-\infty}^{+\infty} |\Gamma(x) - f(x)|^2 dx = \frac{1}{4\pi}\int_{-\infty}^{+\infty} A(k)^2 dk \tag{7}$$

Consequently, the loss function is converted from the time domain into the frequency domain.

Regarding SNNs, this study employs LIF neurons for subsequent derivations, with their dynamic equations defined as follows:

$$\begin{cases} v_j[t] = \beta v_j[t-1] + (w_j x[t] + b_j) \\ o_j[t] = H(v_j[t] - V_{th}) \end{cases} \tag{8}$$

where $w_j x[t] + b_j$ represents the input current at time t. The step function $H(x)$ serves as the activation function in the forward pass. As the step function is non-differentiable, this work adopts the Cauchy distribution as the surrogate gradient:

$$g(x) = \frac{\alpha}{\pi(\alpha^2 + x^2)}, \quad x \in \mathbb{R} \tag{9}$$

where $\alpha > 0$ is a hyperparameter that approaches the derivative of the step function as $\alpha \to 0$. Applying translation and scaling transformations to the surrogate gradient yields:

$$\mathcal{F}[g(w_j x + b_j)](k) = \frac{1}{|w_j|} e^{\frac{ikb_j}{w_j}} \mathcal{F}[g]\left(\frac{k}{w_j}\right) = \frac{1}{|w_j|} e^{\frac{ikb_j}{w_j}} e^{-\frac{\alpha|k|}{|w_j|}} \tag{10}$$

The gradient expressions for each parameter are derived as follows:

$$\frac{\partial \mathcal{L}}{\partial w_j} = \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\infty}^{\infty} S_j(k) T(\omega) X^*(\omega) E(k,\omega) dk d\omega \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial b_j} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\int_{-\infty}^{\infty} S_j(k) T(\omega) E(k,\omega) dk d\omega \tag{12}$$

$$\frac{\partial \mathcal{L}}{\partial a_j} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\int_{-\infty}^{\infty} O_j(\omega)^* E(k,\omega) dk d\omega \tag{13}$$

where:

$$S_j(k) = \frac{a_j}{2\pi w_j |w_j|} e^{-\frac{\alpha|k|}{|w_j|}} (ib_j - \alpha \cdot \text{sgn}(k) \cdot \text{sgn}(w_j)) e^{-\frac{ikb_j}{w_j}}, T(\omega) = \frac{1}{1 - \beta e^{i\omega}} \tag{14}$$

$$E(k,\omega) = \sum_{n=-\infty}^{\infty} \mathcal{F}[\Gamma[n] - f[n]](k) e^{-i\omega n} \tag{15}$$

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-i\omega n}, \quad O_j(\omega) = \sum_{n=-\infty}^{\infty} o_j[n] e^{-i\omega n} \tag{16}$$

In this formulation, $S_j(k)$, $T(\omega)$, $E(k,\omega)$, $X(\omega)$, and $O_j(\omega)$ denote the spatial frequency kernel, the temporal frequency kernel, the spatio-temporal spectrum of the error, the Fourier transform of the input signal in the temporal frequency dimension, and the Fourier transform of the LIF model output in the temporal frequency dimension, respectively. $X^*(\omega)$ and $O_j(\omega)^*$ are the complex conjugates of $X(\omega)$ and $O_j(\omega)$, respectively. k and ω represent the frequencies in the spatial and temporal frequency domains, respectively.

The detailed derivation is provided in the Appendix A.3. Through the aforementioned analytical framework, this paper derives the following theorems:

Theorem 1 *Consider a SNN employing LIF neurons and a Cauchy surrogate gradient* $g(z) = \frac{\alpha}{\pi(\alpha^2+x^2)}$. *For any two spatial frequencies* $k_2 > k_1 > 0$*, assume there exist positive constants*

c_1, c_2 such that $A(k_1) > c_1 > 0$ and $A(k_2) < c_2 < \infty$. Furthermore, assume a constant $\rho > 0$ exists such that the correlation condition $\frac{|\langle TX, E(k,\cdot)\rangle|}{\|TX\|\cdot\|E(k,\cdot)\|} \geq \rho > 0$ holds. It follows that:

$$\lim_{\epsilon \to 0} \frac{\mu(\{w_j \in B_\epsilon : |\frac{\partial \mathcal{L}(k_1)}{\partial w_j}| > |\frac{\partial \mathcal{L}(k_2)}{\partial w_j}|\})}{\mu(B_\epsilon)} = 1 \tag{17}$$

where B_ϵ is a ball centered at the origin with radius ϵ, $\mu(\cdot)$ is the Lebesgue measure. Similar conclusions hold for parameters b_j and a_j.

Theorem 1 reveals that in the spatial frequency domain, for any two unconverged spatial frequencies under sufficiently small weight initialization, the gradient-based training process assigns higher learning priority to low-frequency components.

Theorem 2 *Consider a SNN that utilizes LIF neurons and a Cauchy distribution $g(x) = \frac{\alpha}{\pi(\alpha^2+x^2)}$ as the surrogate gradient. Let $k_2 > k_1 > 0$ and $|\mathcal{F}[f](k_1)| > 0$. Assume there exists a positive constant $\rho > 0$ such that for any frequency k, the condition $\frac{|\langle TX, E(k,\cdot)\rangle|}{\|TX\|\cdot\|E(k,\cdot)\|} \geq \rho > 0$ holds. If there exists a constant $C_a > 0$ such that $A(k_2) < C_a$, then for any $\epsilon > 0$, there exists $M > 0$ such that for any $w_j \in [-M, M] \setminus \{0\}$, the condition:*

$$\left|\frac{\partial \mathcal{L}(k_1)}{\partial w_j}\right| = \left|\frac{\partial \mathcal{L}(k_2)}{\partial w_j}\right| \neq 0 \tag{18}$$

holds, then the relative error satisfies $\Delta_F(k_1) = \frac{A(k_1)}{|\mathcal{F}[f](k_1)|} < \epsilon$.

Theorem 2 indicates that when the magnitude spectrum $A(k_2)$ of spatial high-frequency components is bounded, it is always possible to find sufficiently small weights such that when high-frequency components obtain a priority similar to low-frequency components, the relative error of the low-frequency components remains extremely small.

Theorem 3 *Consider a SNN that employs LIF neurons and the Cauchy distribution $g(x) = \frac{\alpha}{\pi(\alpha^2+x^2)}$ as the surrogate gradient. Let $k_2 > k_1 > 0$ and $|\mathcal{F}[f](k_1)| > 0$. Assume that there exists a positive constant $\rho > 0$ such that the correlation condition $\frac{|\langle TX, E(k,\cdot)\rangle|}{\|TX\|\cdot\|E(k,\cdot)\|} \geq \rho > 0$ holds. For any $\Lambda > 0$, there exists $M > 0$ such that if $A(k_2) > M$ and the condition:*

$$\left|\frac{\partial \mathcal{L}(k_1)}{\partial w_j}\right| = \left|\frac{\partial \mathcal{L}(k_2)}{\partial w_j}\right| \neq 0 \tag{19}$$

holds, then the relative error $\Delta_F(k_1)$, defined as $\frac{A(k_1)}{|\mathcal{F}[f](k_1)|}$, satisfies $\Delta_F(k_1) > \Lambda$.

Theorem 3 indicates that when the target function is dominated by spatial high-frequency components (i.e., $A(k_2)$ is large), the spatial low-frequency component may already be significantly interfered with by high-frequency components before it converges.

Theorem 4 *Consider a SNN that employs LIF neurons and the Cauchy distribution $g(x) = \frac{\alpha}{\pi(\alpha^2+x^2)}$ as the surrogate gradient. The gradient density at temporal frequency ω decomposes into a product of three factors: $\left|\frac{\partial \mathcal{L}}{\partial w_j}(\omega)\right| = |T(\omega)| \cdot |X^*(\omega)| \cdot |SE_j(\omega)|$. The learning preference of the network for two distinct temporal frequencies $\omega_1 < \omega_2$ is determined by their ratio,* $\mathrm{Ratio}(\omega_1, \omega_2)$*:*

$$\mathrm{Ratio}(\omega_1, \omega_2) = \frac{|T(\omega_1)|}{|T(\omega_2)|} \cdot \frac{|X^*(\omega_1)|}{|X^*(\omega_2)|} \cdot \frac{|SE_j(\omega_1)|}{|SE_j(\omega_2)|} \tag{20}$$

The network prioritizes ω_1 when $\mathrm{Ratio} > 1$ *and ω_2 when* $\mathrm{Ratio} < 1$*, while* $\mathrm{Ratio} = 1$ *indicates equal priority. Thus, the temporal learning trajectory is dynamically governed by three factors—the temporal kernel $T(\omega)$, the input spectrum $X(\omega)$, and the integrated error magnitude $SE_j(\omega)$—precluding any inherent low-frequency bias.*

Theorem 4 indicates that when $\beta \to 1$, the temporal frequency kernel dominates, and the network prioritizes temporal low-frequency information. When the input signal is concentrated at high frequencies, the input spectrum dominates, leading to a preference for high frequencies. When the temporal high-frequency error is significantly larger than the low-frequency error, the integrated magnitude term dominates, and the network likewise prioritizes learning temporal high-frequency information.

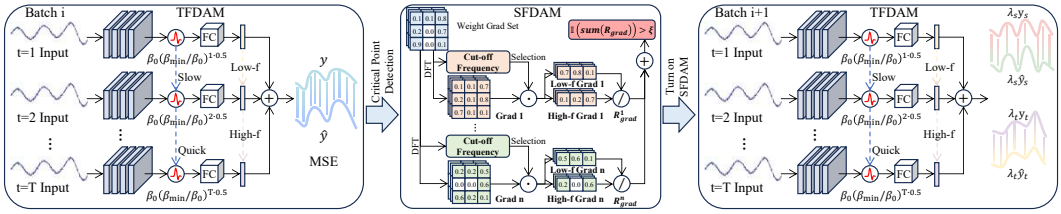

Figure 1: The framework of Frequency-LIF comprises two core modules: SFDAM and TFDAM. These modules dynamically regulate learning preference for frequency components across the spatial and temporal frequency domains, respectively.

3 METHODOLOGY

Based on the STSBP theoretical analysis, this section introduces the Frequency-LIF model. Figure 1 and Algorithm 1 illustrate its architectural framework and pseudocode implementation, respectively. The model dynamically regulates the learning preference for frequency components in both spatial and temporal frequency domains.

3.1 TIME SERIES PROBLEM SETTING

Time series forecasting employs historical time series $x \in \mathbb{R}^{C\times L}$ to predict future values $y \in \mathbb{R}^{C\times H}$. Where, C denotes the number of variables or channels, L is the length of the lookback window, and H represents the forecast horizon. When $C > 1$, this corresponds to a multivariate time series; otherwise, it is referred to as a univariate time series. This method performs point estimation via a function f: $\hat{y} = f(x)$.

3.2 SPATIAL FREQUENCY DYNAMIC ADJUSTMENT MECHANISM

Based on Theorem 1, SNNs act as spatial low-pass filters under small initialization. Theorems 2 and 3 indicate that bounded high-frequency bias implies adequate low-frequency fitting, while unbounded high-frequency bias may interfere with it. Motivated by this, SFDAM monitors gradient energy distribution in the spatial frequency domain and triggers a shift from global trends to local details when low-frequency gradients dominate, realizing theory-aligned dynamic regulation to prevent high-frequency interference with low-frequency learning.

To quantify this phenomenon, this paper introduces the Gradient Spectral Ratio:

$$R_{grad} = \frac{\mathbb{E}[|\mathcal{F}\{\nabla_\vartheta \mathcal{L}\}|_{k>k_l}]}{\mathbb{E}[|\mathcal{F}\{\nabla_\vartheta \mathcal{L}\}|_{k<k_h}]} \tag{21}$$

where k_l and k_h represent the low-frequency and high-frequency cutoff frequencies, respectively, and $\mathcal{F}$ denotes the Fourier transform. When $R_{grad} > 1$, the gradient is dominated by low-frequency components.

To transition to high-frequency detail learning once the low-frequency components are adequately learned, SFDAM employs a critical point detection mechanism. By recording the gradient spectral ratios over the N_{batch} most recent batches, the switching indicator function is defined as follows:

$$\text{Switch} = \mathbb{I}\left(\sum_{j=i-\mathcal{N}_{batch}+1}^{i} \mathbb{I}(R_{grad}^{(j)} > 1) > \xi\right) \tag{22}$$

where ξ is the tolerance threshold and $\mathbb{I}(\cdot)$ is the indicator function. When this condition consistently holds, the high-frequency learning mode is activated, indicating that the network's learning focus has transitioned from the low-frequency-dominated stage to the high-frequency detail fitting stage.

Following the activation of the high-frequency learning mode, the loss function is adjusted accordingly. SFDAM decomposes the predicted output into trend and seasonal components Wu et al. (2021). The trend component captures the long-term progression of the sequence, corresponding

to low-frequency information, while the seasonal component captures periodic fluctuations, corresponding to high-frequency information. Let $\hat{y}$ denote the original prediction and y denote the ground truth. Sequence decomposition yields the seasonal terms $(\hat{y}_s, y_s)$ and trend terms $(\hat{y}_t, y_t)$, leading to the definition of the trend-seasonal decomposition loss:

$$\mathcal{L}_{TS} = \lambda_s \cdot \text{MSE}(\hat{y}_s, y_s) + \lambda_t \cdot \text{MSE}(\hat{y}_t, y_t) \tag{23}$$

where λ_s and λ_t are the weights for the seasonal and trend components, respectively. In the absence of SFDAM, the standard mean squared error loss $\mathcal{L} = \text{MSE}(\hat{y}, y)$ is utilized. Upon switching to high-frequency learning, the weight coefficients are dynamically adjusted as follows:

$$\lambda_s = 1 + \eta, \quad \lambda_t = 1 - \eta \tag{24}$$

where $\eta \in (0, 1)$ is the Frequency Factor. Increasing the weight of the seasonal component guides the network toward high-frequency details, while decreasing the weight of the trend component reduces the bias toward the already learned low-frequency principal components.

3.3 Temporal Frequency Dynamic Adjustment Mechanism

Theorem 4 reveals that in the temporal frequency domain, the learning order of frequency components is jointly driven by $|T(\omega)| \cdot |X^*(\omega)| \cdot |E_j(\omega)|$. In practice, $T(\omega)$ is regulated by the leakage factor β and offers high adjustability. Therefore, this paper primarily optimizes the learning of temporal frequencies through the dynamic adjustment of β. This parameter determines the memory length of the neuron: a higher β yields a longer effective memory window, which assists the network in capturing low-frequency temporal components; a lower β shortens the memory duration, thereby enhancing the response to high-frequency transient events.

Based on this property, TFDAM introduces an exponential decay strategy within the temporal dimension. Let the current timestep be $t \in \{0, 1, \ldots, T-1\}$, where T denotes the total number of timesteps in the spiking neural network. The dynamic leakage factor is defined as follows:

$$\beta[t] = \max\left(\beta_0 \cdot \left(\frac{\beta_{\min}}{\beta_0}\right)^{t \cdot t_{decay}}, \beta_{\min}\right) \tag{25}$$

where β_0 is the initial leakage factor, $\beta_{\min}$ is the minimum leakage factor, and t_{decay} represents the exponential decay factor. This function exhibits exponential decay across the temporal dimension: in the initial stage, a larger β provides the network with a longer memory duration, allowing it to prioritize the capture of low-frequency trend information in time series; as the timestep increases, β decays exponentially toward $\beta_{\min}$, which shortens the network memory and shifts the focus toward learning high-frequency transient events.

With the integration of TFDAM, the membrane potential update rule for the neuron is expressed as:

$$\begin{cases} v_j[t] = \beta[t] v_j[t-1] + c[t] \\ o_j[t] = H(v_j[t] - V_{th}) \end{cases} \tag{26}$$

where $c[t]$ is the input current at time t, V_{th} is the firing threshold, and H is the step function. This rule provides individual neurons with dynamically changing time constants across different timesteps, thereby achieving adaptive learning of various frequency components within the time series.

4 Experiments

Dataset To comprehensively evaluate model performance, this study compares the proposed Frequency-LIF model with various state-of-the-art methods across eight commonly used real-world benchmark datasets for the Multivariate Long-term Time-series Forecasting (MTSF) task. Detailed dataset descriptions are provided in Appendix A.4.1

Baseline Methods This paper selects the following ten representative state-of-the-art (SOTA) methods as comparison baselines: 1) AMP2 Song et al. (2026); 2) DLIF Ma et al.; 3) TS-LIF Feng et al. (2025); 4) AGMM Liang et al. (2025); 5) AT-LIF Bu et al.; 6) TC-LIF Zhang et al. (2024); 7) LM-H Hao et al. (2023); 8) LM_HT Hao et al. (2024); 9) LIF; and 10) GELU. Among these, GELU is a widely applied continuous activation function, and LIF refers to the standard vanilla Leaky Integrate-and-Fire model.

Table 1: Frequency-LIF compares the performance of advanced methods in the SNNs field in the past three years. GELU (Original) denotes the original iTransformer Liu et al. (2023), while others use SNNs as activation functions. The look-back window length L is set to 96, and the results are averaged across four forecasting horizons $H \in \{96, 192, 336, 720\}$. The best and second-best results are red and blue, respectively. Full results are listed in Appendix 6.

Methods	**Frequency-LIF** (Ours)		**AMP2** (AAAI2026)		**DLIF** (ICLR2026)		**TS-LIF** (ICLR2025)		**AGMM** (AAAI2025)		**AT-LIF** (NeurIPS2025)		**TC-LIF** (AAAI2024)		**LM-H** (ICLR2024)		**LM_HT** (NeurIPS2024)		**LIF** (Vanilla)		**GELU** (Original)	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTm1	0.407	0.412	0.415	0.417	0.424	0.420	0.450	0.434	0.428	0.423	0.426	0.423	0.424	0.421	0.419	0.420	0.426	0.424	0.416	0.418	0.426	0.421
ETTm2	0.288	0.331	0.290	0.332	0.296	0.337	0.302	0.342	0.305	0.344	0.301	0.340	0.298	0.338	0.302	0.340	0.302	0.340	0.304	0.341	0.297	0.338
ETTh1	0.452	0.446	0.459	0.450	0.458	0.448	0.454	0.448	0.459	0.450	0.458	0.450	0.454	0.448	0.461	0.451	0.462	0.450	0.461	0.450	0.454	0.448
ETTh2	0.378	0.402	0.392	0.410	0.400	0.415	0.396	0.412	0.388	0.408	0.390	0.408	0.389	0.408	0.390	0.408	0.395	0.412	0.393	0.409	0.394	0.412
Exchange	0.365	0.408	0.372	0.409	0.373	0.413	0.477	0.447	0.372	0.413	0.382	0.414	0.374	0.416	0.407	0.428	0.372	0.414	0.387	0.417	0.392	0.420
Weather	0.258	0.279	0.256	0.277	0.260	0.280	0.256	0.283	0.260	0.281	0.258	0.279	0.258	0.280	0.262	0.283	0.266	0.287	0.258	0.279	0.259	0.280
Traffic	0.444	0.296	0.455	0.307	0.456	0.306	0.461	0.310	0.458	0.306	0.456	0.306	0.460	0.307	0.472	0.318	0.480	0.327	0.454	0.302	0.466	0.313
Solar-energy	0.239	0.265	0.239	0.271	0.244	0.276	0.256	0.286	0.241	0.274	0.242	0.274	0.241	0.272	0.251	0.283	0.268	0.308	0.240	0.270	0.242	0.273
1st Count	7	7	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.1 Comparison with State-of-the-Art Methods

In this section, we evaluate the prediction performance of Frequency-LIF using iTransformer Liu et al. (2023) as the backbone architecture. Specifically, we replace the original GELU activation function (Original) with representative SNN models, while keeping all other conditions unchanged to ensure a fair comparison (All SNN models adopt iTransformer as the backbone architecture). Table 1 summarizes the prediction results, where the best results are highlighted in red and the second-best in blue; lower MSE/MAE values indicate more accurate prediction results. Overall, Frequency-LIF achieves the best average performance across various prediction tasks. Among the comparison baselines, AMP2 and TS-LIF also exhibit strong competitiveness.

AMP2 replaces timestep accumulation with spatial adjacent accumulation, which aims to save computational resources within a single timestep and introduces membrane potential residual connections to alleviate the spike degradation problem. TS-LIF adopts a dual-compartment structure, using a dendritic compartment to capture long-term features and a somatic compartment to capture short-term changes. In contrast, Frequency-LIF achieves progressive learning from long-term trends to transient changes in the temporal frequency dimension and realizes an adaptive evolution from low to high frequencies in the spatial frequency dimension.

Specifically, although AMP2 transmits spikes between spatially adjacent neurons, its strategy ignores information accumulation in the temporal dimension, which makes it difficult to model fine temporal dynamics. Frequency-LIF adopts an exponential β decay strategy in the temporal dimension, which guides the model to capture long-term trends in the initial stage and gradually focus on transient changes as timesteps increase. Although TS-LIF utilizes a dual-compartment structure to process long- and short-term features, the parallel processing mode of it may cause the dendritic compartment to be vulnerable to noise in the early stages of training, thereby affecting the modeling of the somatic compartment regarding short-term changes. In contrast, Frequency-LIF monitors the learning status of spatial frequency components in real-time and shifts to high-frequency learning when low-frequency information dominates the model.

Table 2: The results are averaged across four forecasting horizons. Original denotes model performance with the original activation function. Frequency-LIF represents the performance when applied to the aforementioned baseline models without altering the original architecture or execution environment. Full results are listed in Appendix 7.

Dataset	**Transformer**				**TimeXer**				**MoFo**				**DeepBooTS**				**ChannelTokenFormer**			
	Original		Frequency-LIF		Original		Frequency-LIF		Original		Frequency-LIF		Original		Frequency-LIF		Original		Frequency-LIF	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh1	1.126	0.860	0.944	0.781	0.447	0.445	0.442	0.438	0.442	0.440	0.434	0.437	0.461	0.455	0.451	0.449	0.447	0.448	0.429	0.431
ETTh2	3.728	1.589	3.594	1.563	0.381	0.404	0.371	0.400	0.395	0.415	0.376	0.405	0.408	0.415	0.400	0.413	0.388	0.406	0.375	0.398
ETTm1	0.860	0.694	0.769	0.654	0.394	0.406	0.392	0.405	0.369	0.395	0.366	0.394	0.422	0.418	0.406	0.411	0.386	0.400	0.380	0.397
ETTm2	1.566	0.894	1.408	0.838	0.287	0.330	0.283	0.328	0.281	0.327	0.280	0.327	0.285	0.328	0.282	0.327	0.280	0.325	0.277	0.324

4.2 Generality Analysis of Frequency-LIF

To verify the generality of Frequency-LIF, we integrate it into five MTSF benchmark models, namely Transformer Vaswani et al. (2017), TimeXer Wang et al. (2024), MoFo Ma et al. (2025), DeepBooTS Liang et al. (2026), and ChannelTokenFormer Jang et al. (2025). Specifically, we replace the original activation function (Original) of the benchmark models with Frequency-LIF, while keeping all other conditions unchanged to ensure a fair comparison. Experiments are conducted on four benchmark datasets. The experimental results are shown in Table 2.

Experimental results demonstrate that the introduction of Frequency-LIF significantly improves the average performance of baseline models across four prediction horizons in all test scenarios. Taking the ETTh1 dataset as an example, Frequency-LIF reduces the MSE and MAE of the Transformer by 0.182 and 0.079, respectively. The above results demonstrate the excellent generality of Frequency-LIF. Regardless of the backbone architecture, Frequency-LIF guides the model to adaptively capture low- and high-frequency information at different timesteps and training stages by dynamically adjusting the learning preference of frequency components across spatio-temporal dimensions, which significantly enhances the adaptive learning capability of the model for different frequency components.

Table 3: The average results of the ablation study on removing TFDAM and SFDAM. ✓ indicates the use of the component, while × means the component is removed.

	TFDAM	**SFDAM**	**ETTm1**		**ETTm2**		**ETTh1**		**ETTh2**	
Case	**w/**	**w/**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
①	×	×	0.419	0.419	0.295	0.336	0.459	0.450	0.394	0.412
②	✓	×	0.417	0.418	0.292	0.333	0.458	0.450	0.391	0.410
③	×	✓	0.412	0.415	0.290	0.333	0.455	0.449	0.385	0.406
④	✓	✓	0.407	0.412	0.288	0.331	0.452	0.446	0.378	0.402

4.3 Ablation Study

This section conducts a comprehensive ablation study to quantitatively evaluate the effectiveness of the core components of Frequency-LIF; the results are summarized in Table 3.

Overall, experimental results demonstrate that TFDAM and SFDAM are the key modules driving model performance enhancement. For example, on the ETTm1 dataset, removing TFDAM and SFDAM increases MSE by 0.012 and 0.009 on average, respectively. The specific analysis is as follows: 1) w/o TFDAM: The absence of the progressive low-to-high frequency learning strategy in the temporal domain hinders effective frequency capture and impairs the balance between long-term dependencies and transient events. 2) w/o SFDAM: Without the active detection mechanism, the network cannot ascertain whether low-frequency components have dominated learning, leading to overfitting of low-frequency principal components and compromised generalization. 3) w/o TFDAM_SFDAM: Removing both modules yields the most severe performance drop.

CONCLUSION

This paper proposes the STSBP theoretical framework, which decouples the error backpropagation process into the spatio-temporal frequency domain, thereby addressing the challenges of deep parameter coupling and quantification of individual parameter influences in BPTT for SNNs. Based on this framework, the proposed Frequency-LIF model is evaluated on eight time-series benchmark datasets, and the experimental results demonstrate that Frequency-LIF significantly outperforms current state-of-the-art methods.

Although this study achieves significant progress, there remains room for further optimization. First, Frequency-LIF can adopt adaptive parameter strategies to control frequency learning across spatial and temporal dimensions, thereby enhancing model flexibility. Second, the STSBP theoretical framework is not limited to time-series forecasting, and future work can explore its generalization to other tasks.

REFERENCES

Ernesto Araya, Massimiliano Datres, and Gitta Kutyniok. Random spiking neural networks are stable and spectrally simple. *arXiv preprint arXiv:2511.00904*, 2025.

Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):3625, 2020.

Tong Bu, Xinyu Shi, and Zhaofei Yu. Activity pruning for efficient spiking neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Long Chen, Wei Miao, Xin Gao, Yunzhi Zhuge, Hongming Xu, Yaxin Li, and Qi Xu. Spatial-frequency spiking neural network for underwater object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20217–20225, 2026.

Xinyi Chen, Qu Yang, Jibin Wu, Haizhou Li, and Kay Chen Tan. A hybrid neural coding approach for pattern recognition with spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 46(5):3064–3078, 2023.

Gourav Datta, Zeyu Liu, and Peter Anthony Beerel. Can we get the best of both binary neural networks and spiking neural networks for efficient computer vision? In *The Twelfth International Conference on Learning Representations*, 2024.

Yuetong Fang, Deming Zhou, Ziqing Wang, Hongwei Ren, Zecui Zeng, Lusong Li, Shibo Zhou, and Renjing Xu. Spiking neural networks need high frequency information. *arXiv preprint arXiv:2505.18608*, 2025.

Jingru Fei, Kun Yi, Wei Fan, Qi Zhang, and Zhendong Niu. Amplifier: Bringing attention to neglected low-energy components in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 11645–11653, 2025.

Shibo Feng, Wanjin Feng, Xingyu Gao, Peilin Zhao, and Zhiqi Shen. Ts-lif: A temporal segment spiking neuron network for time series forecasting. *arXiv preprint arXiv:2503.05108*, 2025.

Richard Gast, Sara A Solla, and Ann Kennedy. Neural heterogeneity controls computations in spiking neural networks. *Proceedings of the National Academy of Sciences*, 121(3):e2311885121, 2024.

Julia Gygax and Friedemann Zenke. Elucidating the theoretical underpinnings of surrogate gradient learning in spiking neural networks. *Neural Computation*, 37(5):886–925, 2025.

Zecheng Hao, Xinyu Shi, Zihan Huang, Tong Bu, Zhaofei Yu, and Tiejun Huang. A progressive training framework for spiking neural networks with learnable multi-hierarchical model. In *The twelfth international conference on learning representations*, 2023.

Zecheng Hao, Xinyu Shi, Yujia Liu, Zhaofei Yu, and Tiejun Huang. Lm-ht snn: Enhancing the performance of snn to ann counterpart through learnable multi-hierarchical threshold model. *Advances in Neural Information Processing Systems*, 37:101905–101927, 2024.

Ping He, Rong Xiao, Chenwei Tang, Shudong Huang, Jiancheng Lv, and Huajin Tang. Stsf: Spiking time sparse feedback learning for spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):11479–11492, 2025.

JiaKui Hu, Man Yao, Xuerui Qiu, Yuhong Chou, Yuxuan Cai, Ning Qiao, Yonghong Tian, Bo Xu, and Guoqi Li. High-performance temporal reversible spiking neural networks with $o(l)$ training memory and $o(1)$ inference cost. *arXiv preprint arXiv:2405.16466*, 2024a.

Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing spiking neural networks toward deep residual learning. *IEEE transactions on neural networks and learning systems*, 36(2):2353–2367, 2024b.

Jinkwan Jang, Hyungjin Park, Jinmyeong Choi, and Taesup Kim. Towards robust real-world multivariate time series forecasting: A unified framework for dependency, asynchrony, and missingness. *arXiv preprint arXiv:2506.08660*, 2025.

Runhao Jiang, Chengzhi Jiang, Rui Yan, and Huajin Tang. Mpd-sgr: Robust spiking neural networks with membrane potential distribution-driven surrogate gradient regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1864–1872, 2026.

Yizhou Jiang, Kunlin Hu, Tianren Zhang, Haichuan Gao, Yuqian Liu, Ying Fang, and Feng Chen. Spatio-temporal approximation: A training-free snn conversion for transformers. In *The twelfth international conference on learning representations*, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Hyunho Kook, Byeongho Yu, Jeong Min Oh, and Eunhyeok Park. Stabilizing direct training of spiking neural networks: Membrane potential initialization and threshold-robust surrogate gradient. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 8500–8510, 2026.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in neuroscience*, 10:508, 2016.

Yuhang Li, Tamar Geller, Youngeun Kim, and Priyadarshini Panda. Seenn: Towards temporal spiking early exit neural networks. *Advances in Neural Information Processing Systems*, 36: 63327–63342, 2023.

Daojun Liang, Jing Chen, Xiao Wang, Yinglong Wang, and Shuo Li. Deepboots: Dual-stream residual boosting for drift-resilient time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23391–23400, 2026.

Yu Liang, Wenjie Wei, Ammar Belatreche, Honglin Cao, Zijian Zhou, Shuai Wang, Malu Zhang, and Yang Yang. Towards accurate binary spiking neural networks: Learning with adaptive gradient modulation mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1402–1410, 2025.

Wei Liu, Li Yang, Mingxuan Zhao, Shuxun Wang, Jin Gao, Wenjuan Li, Bing Li, and Weiming Hu. Deeptage: Deep temporal-aligned gradient enhancement for optimizing spiking neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Yijie Lu, Zhiyi Pan, Renrui Zhang, Yanhao Jia, Ronggang Wang, and Zhaokun Zhou. Spiking discrepancy transformer for point cloud analysis. In *The Fourteenth International Conference on Learning Representations*.

Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. Mofo: Empowering long-term time series forecasting with periodic pattern modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jingyang Ma, Chongming Liu, Songting Li, and Douglas Zhou. Beyond linear processing: Dendritic bilinear integration in spiking neural networks. In *The Fourteenth International Conference on Learning Representations*.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Yang Ouyang, Zihan Cheng, Xiaotong Luo, Guoqi Li, and Yanyun Qu. Spikingir: A novel converted spiking neural network for efficient image restoration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8215–8223, 2026.

Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in neuroscience*, 12:409662, 2018.

Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

Jian Song, Xiangfei Yang, Shangke Lyu, and Donglin Wang. Activation-wise propagation: A one-timestep strategy for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2056–2064, 2026.

Qi Sun, Zhen Cao, Kaige Geng, Ziyi Zhang, and Biao Hou. Optimization method for surrogate function in spiking neural networks based on membrane potential distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25718–25726, 2026.

Amirhossein Tavanaei and Anthony S Maida. Bio-inspired spiking convolutional neural network using layer-wise sparse coding and stdp learning. *arXiv preprint arXiv:1611.03000*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Gang Wan, Qinlong Lan, Huimin Wang, Wu Yitian, Wanhua Li, Yufei Guo, et al. Spik-nerf: Spiking neural networks for neural radiance fields. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Hongzhi Wang, Xiubo Liang, Jinxing Han, and Weidong Geng. Spike-retinexformer: Rethinking low-light image enhancement with spiking neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Shuai Wang, Haorui Zheng, Yukun Chen, Ammar Belatreche, Guoqing Wang, Yeying Jin, Jibin Wu, Malu Zhang, Yang Yang, and Haizhou Li. Snn-ft: Temporal-coded spiking neural networks for fourier transform. *IEEE Transactions on Neural Networks and Learning Systems*, 2025b.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

Timo C Wunderlich and Christian Pehle. Event-based backpropagation can compute exact gradients for spiking neural networks. *Scientific Reports*, 11(1):12829, 2021.

Mengting Xu, De Ma, HuaJin Tang, Qian Zheng, and Gang Pan. Feel-snn: Robust spiking neural networks with frequency encoding and evolutionary leak factor. *Advances in Neural Information Processing Systems*, 37:91930–91950, 2024.

Chengting Yu, Lei Liu, Gaoang Wang, Erping Li, and Aili Wang. Advancing training efficiency of deep spiking neural networks through rate-based backpropagation. *Advances in Neural Information Processing Systems*, 37:115786–115815, 2024.

Kairong Yu, Tianqing Zhang, Hongwei Wang, and Qi Xu. Fsta-snn: Frequency-based spatial-temporal attention module for spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 22227–22235, 2025.

Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

Dehao Zhang, Fukai Guo, Shuai Wang, Jingya Wang, Jieyuan Zhang, Yimeng Shan, Malu Zhang, Yang Yang, and Haizhou Li. Neural dynamics self-attention for spiking transformers. *arXiv preprint arXiv:2603.19290*, 2026.

Shimin Zhang, Qu Yang, Chenxiang Ma, Jibin Wu, Haizhou Li, and Kay Chen Tan. Tc-lif: A two-compartment spiking neuron model for long-term sequential modelling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16838–16847, 2024.

Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Han Zhang, Jiaqi Wang, Huihui Zhou, Zhengyu Ma, and Yonghong Tian. Spikingformer: A key foundation model for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2236–2244, 2026.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

Rui-Jie Zhu, Malu Zhang, Qihang Zhao, Haoyu Deng, Yule Duan, and Liang-Jian Deng. Tcja-snn: Temporal-channel joint attention for spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):5112–5125, 2024.

A Appendix

A.1 LLM Usage

We used LLMs exclusively for language polishing and readability improvements. They played no role in the research design, experimental execution, data analysis, or final conclusions. All scientific contributions and judgments are entirely our own.

A.2 Related Work

A.2.1 Spiking Neural Networks and Their Training Algorithms

Spiking Neural Networks (SNNs) attract significant attention due to their event-driven computational nature and biological plausibility, offering promising pathways for the development of neuromorphic computing and low-power artificial intelligence Pfeiffer & Pfeil (2018); Chen et al. (2023). For example, STA pre-trains a universal group operator using synthetic data and converts it into an SNN, while utilizing a temporally-calibrated self-attention layer to stabilize the output spike sequence Jiang et al. (2024). LRF-Dyn employs local receptive fields to enhance the local modeling capability of spike self-attention and utilizes neuron dynamics to approximate attention computation, thereby significantly reducing inference-time memory overhead Zhang et al. (2026). Most existing studies focus on attention mechanisms built upon spike features Wang et al. (2025a); Lu et al.; Zhou et al. (2026). However, the non-differentiability of the spike generation function prevents the direct application of gradient-based optimization methods in SNNs.

To address this issue, Surrogate Gradient methods emerge as the mainstream approach, wherein a smooth function approximates the derivative of the spike function during backpropagation Wunderlich & Pehle (2021); Sun et al. (2026); Ouyang et al. (2026). For instance, SLAYER overcomes the non-differentiability of spike firing through surrogate gradients and utilizes a spike response kernel to achieve non-local credit assignment across multiple timesteps Shrestha & Orchard (2018). e-prop is an online learning algorithm for recurrent spiking neural networks that propagates gradient information via eligibility traces Bellec et al. (2020). Rate-based Backpropagation employs time-averaged statistics during backpropagation to substitute for the full time-series expansion, thereby simplifying training into a single spatial backpropagation pass Yu et al. (2024). Although surrogate gradient methods are effective, these BPTT-based approaches generally suffer from deep parameter coupling, where parameters controlling the surrogate gradient shape and neuron dynamics are

highly entangled in the temporal domain, limiting theoretical interpretability and in-depth model understanding.

To address these limitations, this paper decouples SNN error backpropagation into the spatio-temporal frequency domain, factorizing it into four independent terms: (1) a spatial frequency kernel depending on α and network weights; (2) a temporal frequency kernel depending on β; (3) an error spatio-temporal spectrum determined by task data; (4) an input signal spectrum determined by input data.

A.2.2 Frequency Analysis of Spiking Neural Networks

In ANNs, research reveals the Spectral Bias phenomenon: networks tend to prioritize learning low-frequency components of the target function before capturing high-frequency details Zhou et al. (2022); Fei et al. (2025). In the field of SNNs, existing frequency-domain research is primarily divided into two categories: frequency-domain-based SNN architecture design Yu et al. (2025); Chen et al. (2026); Wang et al. (2025b) and frequency-domain-based theoretical analysis of SNNs Xu et al. (2024). The former mostly adapts frequency-domain theories from ANNs to guide SNN design; however, due to the unique spike firing mechanism of SNNs, ANN frequency theories are not fully applicable.

To address this issue, Fang et al.Fang et al. (2025) show via Fourier spectral analysis that spike generation rapidly attenuates high-frequency components during forward propagation, rendering spiking neurons as low-pass filters. Araya et al. Araya et al. (2025) demonstrate through Boolean function analysis that randomly initialized SNNs possess quantifiable noise robustness, with Fourier spectra naturally concentrated in low frequencies. However, existing research is largely limited to the frequency analysis of static characteristics and fails to quantify how various internal parameters affect network learning behavior in an explicit mathematical form during the training process.

Addressing these limitations, we derive a closed-form gradient expression that precisely quantifies how internal SNN parameters influence learning dynamics. The theory reveals an inherent low-frequency preference of SNNs in the spatial domain. In the temporal domain, however, the preference is not a simple low-frequency priority but is jointly shaped by the temporal kernel, the integral of the error spectrum over spatial frequencies, and the input signal spectrum.

A.3 Detailed Derivation of Spatio-Temporal Spectral Backpropagation

The theoretical derivation is organized as follows. Section A.3.1 establishes the foundational frequency-domain derivation for a standard ANN with a single hidden layer, using the step function as the activation and the Cauchy distribution as the surrogate gradient, which serves as the baseline for subsequent extensions. Section A.3.2 extends this framework to SNNs by incorporating leaky integrate-and-fire (LIF) neurons, where temporal dynamics are introduced via backpropagation through time (BPTT). Building upon the conclusions of Section A.3.2, Sections A.3.3–A.3.6 provide rigorous proofs of Theorems 1–4, respectively. Section A.3.7 demonstrates that the theoretical framework—from Section A.3.2 through Section A.3.6—is not restricted to the Cauchy surrogate gradient, but applies broadly to any surrogate gradient whose Fourier transform exhibits exponential decay with a polynomially bounded prefactor. Finally, Section A.3.8 discusses the generalization of the STSBP theoretical framework to more complex SNN architectures.

A.3.1 ANN Backpropagation under Surrogate Gradients

Consider a neural network with a single hidden layer that processes a one-dimensional input x and produces a one-dimensional output, where the target function is $f(x)$. Let the hidden layer contain N neurons; the network output is expressed as:

$$\Gamma(x) = \sum_{j=1}^{N} a_j \sigma(w_j x + b_j), \quad w_j, b_j, a_j \in \mathbb{R} \tag{27}$$

where $\Gamma(x)$ denotes the network output, σ is the activation function, a_j is the output weight, and w_j and b_j are the model parameters. The mean squared error loss function is defined as:

$$\mathcal{L} = \frac{1}{2} \int_{-\infty}^{+\infty} |\Gamma(x) - f(x)|^2 dx \tag{28}$$

When the step function $H(x)$ serves as the activation function:

$$H(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{29}$$

The Fourier transform of the step function $H(x)$ is given by:

$$\mathcal{F}[H(x)](k) = \pi\delta(k) + \frac{1}{ik}, \quad k \in \mathbb{R} \tag{30}$$

where $\mathcal{F}$ denotes the Fourier transform, $\delta(k)$ is the Dirac delta function, i is the imaginary unit, and k represents the frequency. For $k \neq 0$, the term $1/(ik)$ takes its usual value. At $k = 0$, the singularity of $1/(ik)$ is understood to be absorbed by the $\pi\delta(k)$ term in the distributional sense; hence it is not evaluated separately. Since all subsequent operations are carried out under integrals, we never need to evaluate the function at the isolated point $k = 0$.

By utilizing the shift and scaling properties of the Fourier transform, $\mathcal{F}[H(w_j x + b_j)](k)$ is transformed as:

$$\mathcal{F}[H(w_j x + b_j)](k) = e^{\frac{ikb_j}{w_j}} \left[\pi\delta(k) + \frac{1}{ik}\right] \tag{31}$$

Summing the contributions of all neurons yields the spectrum of the network output:

$$\mathcal{F}[\Gamma](k) = \sum_{j=1}^{N} a_j \left[\pi\delta(k) e^{\frac{ikb_j}{w_j}} + \frac{1}{ik} e^{\frac{ikb_j}{w_j}}\right] \tag{32}$$

where $w_j \neq 0$. For the degenerate case $w_j = 0$, the neuron output reduces to a constant, whose Fourier transform degenerates to $\pi\delta(k)$; this isolated point does not affect the subsequent derivations, as all operations are carried out under integrals. Let $\mathcal{F}[f](k)$ be the Fourier transform of the target function $f(x)$; the error spectrum is defined as:

$$E(k) = \mathcal{F}[\Gamma](k) - \mathcal{F}[f](k) = A(k)e^{i\theta(k)} \tag{33}$$

where $A(k) = |E(k)|$ is the magnitude spectrum and $\theta(k) \in [-\pi, \pi]$ is the phase spectrum. According to the Theorem of Parseval, the mean squared error in the time domain is equivalent to the energy integral in the frequency domain:

$$\mathcal{L} = \frac{1}{2}\int_{-\infty}^{+\infty} |\Gamma(x) - f(x)|^2 dx = \frac{1}{4\pi}\int_{-\infty}^{+\infty} A(k)^2 dk \tag{34}$$

Thus, the time-domain loss function is converted into the frequency domain.

Since the step function is mathematically non-differentiable, it presents challenges for backpropagation. We adopt the Cauchy distribution as a surrogate gradient to approximate the derivative of the step function:

$$g(x) = \frac{\alpha}{\pi(\alpha^2 + x^2)}, \quad x \in \mathbb{R} \tag{35}$$

$$\hat{g}(k) = \mathcal{F}[g](k) = e^{-\alpha|k|}, \quad k \in \mathbb{R} \tag{36}$$

where g represents the surrogate gradient function and $\alpha > 0$ is the smoothing parameter (as $\alpha \to 0$, g approaches the derivative of the step function). $\hat{g}$ represents the Fourier transform of g. Applying shift and scaling transformations to the surrogate gradient yields:

$$\mathcal{F}[g(w_j x + b_j)](k) = \frac{1}{|w_j|} e^{\frac{ikb_j}{w_j}} \hat{g}\left(\frac{k}{w_j}\right) = \frac{1}{|w_j|} e^{\frac{ikb_j}{w_j}} e^{-\frac{\alpha|k|}{|w_j|}} \tag{37}$$

The gradient of the loss function with respect to any parameter Θ is expressed in the frequency domain as:

$$\frac{\partial \mathcal{L}}{\partial \Theta} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \mathfrak{R}\left[A(k)e^{-i\theta(k)} \frac{\partial E(k)}{\partial \Theta}\right] dk \tag{38}$$

where $E(k)^*$ represents the complex conjugate of $E(k)$, and $\mathfrak{R}[\cdot]$ denotes the real part.

The derivative with respect to w_j is expressed as:

$$\frac{\partial E(k)}{\partial w_j} = a_j \frac{\partial}{\partial w_j} \mathcal{F}[H(w_j x + b_j)](k) = a_j \mathcal{F}[g(w_j x + b_j) \cdot x](k) \tag{39}$$

Utilizing the differential property of the Fourier transform $\mathcal{F}[x \cdot h(x)](k) = i\frac{d}{dk}\mathcal{F}[h](k)$, we obtain:

$$\mathcal{F}[g(w_j x + b_j) \cdot x](k) = i\frac{d}{dk}\mathcal{F}[g(w_j x + b_j)](k) \tag{40}$$

The calculation of the derivative yields:

$$\mathcal{F}[g(w_j x + b_j) \cdot x](k) = \frac{1}{|w_j|} e^{\frac{ikb_j}{w_j}} e^{-\frac{\alpha|k|}{|w_j|}} \left[-\frac{b_j}{w_j} - \frac{i\alpha}{|w_j|} \cdot \text{sgn}(k)\right] \tag{41}$$

By rearranging terms, we obtain:

$$\frac{\partial E(k)}{\partial w_j} = \frac{a_j}{w_j|w_j|} e^{\frac{ikb_j}{w_j}} e^{-\frac{\alpha|k|}{|w_j|}} \left[-b_j - i\alpha \cdot \text{sgn}(k) \cdot \text{sgn}(w_j)\right] \tag{42}$$

Substituting this into the gradient formula and simplifying yields:

$$\frac{\partial \mathcal{L}}{\partial w_j} = \frac{a_j}{2\pi w_j|w_j|} \int_{-\infty}^{+\infty} A(k) e^{-\frac{\alpha|k|}{|w_j|}} \Re\left[(ib_j - \alpha \cdot \text{sgn}(k) \cdot \text{sgn}(w_j)) e^{i(\theta(k) - \frac{kb_j}{w_j})}\right] dk \tag{43}$$

Similarly, the partial derivatives with respect to b_j and a_j are:

$$\frac{\partial \mathcal{L}}{\partial b_j} = \frac{a_j}{2\pi|w_j|} \int_{-\infty}^{+\infty} A(k) e^{-\frac{\alpha|k|}{|w_j|}} \Re\left[e^{i(\theta(k) - \frac{kb_j}{w_j})}\right] dk \tag{44}$$

$$\frac{\partial \mathcal{L}}{\partial a_j} = \frac{1}{2\pi|w_j|} \int_{-\infty}^{+\infty} A(k) e^{-\frac{\alpha|k|}{|w_j|}} \Re\left[e^{i(\theta(k) - \frac{kb_j}{w_j})}\right] dk \tag{45}$$

A.3.2 Spatio-Temporal Spectral Backpropagation

LIF neuron update equations:

$$v_j[t] = \beta v_j[t-1] + (w_j x[t] + b_j) \tag{46}$$

$$o_j[t] = H(v_j[t] - V_{\text{th}}) \tag{47}$$

where $v_j[t]$ denotes the membrane potential of the j-th neuron at time t, and o_j denotes the output; β is the leakage factor that controls the potential decay; $x[t]$ is the value of the input signal at time t; and V_{th} is the firing threshold. Similarly, let $\Gamma[t]$ be the network output at time t, and denote the loss at the corresponding time as $\mathcal{L}[t]$, with the Cauchy distribution as the surrogate gradient. The total loss is the accumulation over all time steps:

$$\mathcal{L} = \sum_{t=1}^{T} \mathcal{L}[t] \tag{48}$$

where $\mathcal{L}$ represents the loss. The gradient of backpropagation through time is expressed as:

$$\frac{\partial \mathcal{L}}{\partial v_j[t]} = \frac{\partial \mathcal{L}[t]}{\partial v_j[t]} + \frac{\partial \mathcal{L}}{\partial v_j[t+1]} \cdot \frac{\partial v_j[t+1]}{\partial v_j[t]} = d_j[t] + \beta \cdot \frac{\partial \mathcal{L}}{\partial v_j[t+1]} \tag{49}$$

$$d_j[t] = \frac{\partial \mathcal{L}}{\partial o_j[t]} \cdot \frac{\partial o_j[t]}{\partial v_j[t]} = (\Gamma[t] - f[t]) \cdot a_j \cdot g(v_j[t] - V_{\text{th}}) \tag{50}$$

where $\frac{\partial \mathcal{L}[t]}{\partial v_j[t]} = d_j[t]$ and $\frac{\partial v_j[t+1]}{\partial v_j[t]} = \beta$. Backpropagation starts from the last time step T:

$$\frac{\partial \mathcal{L}}{\partial v_j[T]} = d_j[T] \tag{51}$$

$$\frac{\partial \mathcal{L}}{\partial v_j[T-1]} = d_j[T-1] + \beta d_j[T] \tag{52}$$

$$\frac{\partial \mathcal{L}}{\partial v_j[T-2]} = d_j[T-2] + \beta d_j[T-1] + \beta^2 d_j[T] \tag{53}$$

This can be generalized to the following form:

$$\frac{\partial \mathcal{L}}{\partial v_j[t]} = \sum_{\tau=0}^{T-t} \beta^\tau d_j[t+\tau] \tag{54}$$

Considering an infinite number of time steps:

$$\frac{\partial \mathcal{L}}{\partial v_j[t]} = \sum_{\tau=0}^{\infty} \beta^\tau d_j[t+\tau] \tag{55}$$

$\frac{\partial \mathcal{L}}{\partial v_j[t]}$ indicates that the gradient of the membrane potential at the current time is the attenuated sum of the instantaneous gradients at all future times. Therefore, the derivative with respect to w_j is expressed as:

$$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial v_j[t]} \frac{\partial v_j[t]}{\partial w_j} = \sum_{t=1}^{T} \left(\sum_{\tau=0}^{\infty} \beta^\tau d_j[t+\tau] \right) x[t] \tag{56}$$

Applying the DTFT to $\frac{\partial \mathcal{L}}{\partial v_j[t]} = \sum_{\tau=0}^{\infty} \beta^\tau d_j[t+\tau]$:

$$V_j(\omega) = \sum_{n=-\infty}^{\infty} \left(\sum_{\tau=0}^{\infty} \beta^\tau d_j[n+\tau] \right) e^{-i\omega n} \tag{57}$$

Let $n' = n + \tau$, then $n = n' - \tau$. Rearranging gives:

$$V_j(\omega) = \sum_{\tau=0}^{\infty} \beta^\tau e^{i\omega\tau} D_j(\omega) = \frac{1}{1 - \beta e^{i\omega}} D_j(\omega) \tag{58}$$

Applying the DTFT to $\frac{\partial \mathcal{L}}{\partial w_j} = \sum_t \frac{\partial \mathcal{L}}{\partial v_j[t]} x[t]$:

$$\frac{\partial \mathcal{L}}{\partial w_j}(\omega) = V_j(\omega) \cdot X^*(\omega) = \frac{1}{1 - \beta e^{i\omega}} \cdot D_j(\omega) \cdot X^*(\omega) \tag{59}$$

where

$$D_j(\omega) = \sum_{n=-\infty}^{\infty} d_j[n] e^{-i\omega n}, \quad X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-i\omega n} \tag{60}$$

The frequency-domain expression of the weight gradient contains $D_j(\omega)$, which is the DTFT of $d_j[t]$. To relate $D_j(\omega)$ to the spatial frequency, we introduce the Fourier transform in the spatial direction and express $d_j[t]$ in terms of the spatial spectrum of the error. In a static network, for the step activation function and the surrogate gradient, the frequency-domain expression of the weight gradient leads to the abstraction of a spatial frequency-domain kernel $S_j(k)$ such that:

$$d_j[t] = \int_{-\infty}^{\infty} S_j(k) \cdot E_t(k)\, dk \tag{61}$$

The form of $S_j(k)$ is:

$$S_j(k) = \frac{a_j}{2\pi w_j |w_j|} e^{-\alpha|k|/|w_j|} \left(i b_j - \alpha \cdot \mathrm{sgn}(k) \cdot \mathrm{sgn}(w_j) \right) e^{-i(k b_j)/w_j} \tag{62}$$

$S_j(k)$ describes the sensitivity of neuron j to the spatial frequency k and incorporates the effect of the surrogate gradient. The spatial spectrum of the error at each time step is:

$$E_t(k) = \mathcal{F}[\Gamma[t] - f[t]](k) \tag{63}$$

where $\mathcal{F}[\cdot]$ denotes the Fourier transform with respect to the spatial variable x. Applying the DTFT in the time direction to $E_t(k)$ yields the spatiotemporal spectrum of the error:

$$E(k, \omega) = \sum_{n=-\infty}^{\infty} E_t(k) e^{-i\omega n} = \sum_{n=-\infty}^{\infty} \mathcal{F}[\Gamma[n] - f[n]](k) e^{-i\omega n} \tag{64}$$

Applying the DTFT to both sides of $d_j[t] = \int S_j(k)E_t(k)dk$ gives:

$$D_j(\omega) = \int_{-\infty}^{\infty} S_j(k) \cdot E(k,\omega)\, dk \tag{65}$$

$D_j(\omega)$ establishes a connection between the temporal frequency ω and the spatial frequency k. Substituting $D_j(\omega)$ into the frequency-domain expression of the weight gradient yields:

$$\frac{\partial \mathcal{L}}{\partial w_j}(\omega) = \frac{1}{1-\beta e^{i\omega}} \cdot \left(\int_{-\infty}^{\infty} S_j(k) \cdot E(k,\omega)\, dk \right) \cdot X^*(\omega) \tag{66}$$

For a fixed spatial frequency k and temporal frequency ω, the gradient can be written as:

$$\frac{\partial \mathcal{L}}{\partial w_j}(k,\omega) = S_j(k) \cdot T(\omega) \cdot X^*(\omega) \cdot E(k,\omega) \tag{67}$$

$$T(\omega) = \frac{1}{1-\beta e^{i\omega}} \tag{68}$$

where $X^*(\omega)$ is the complex conjugate of $X(\omega)$, and $T(\omega)$ is the temporal frequency-domain kernel. To obtain the actual gradient value of the weight, we integrate over all spatial frequencies k and temporal frequencies ω:

$$\frac{\partial \mathcal{L}}{\partial w_j} = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\pi}^{\pi} S_j(k) \cdot T(\omega) \cdot X^*(\omega) \cdot E(k,\omega)\, d\omega\, dk \tag{69}$$

where $\frac{1}{4\pi^2}$ is the normalization factor of the two-dimensional inverse Fourier transform. Similarly, the derivatives with respect to b_j and a_j are computed as:

$$\frac{\partial \mathcal{L}}{\partial b_j} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \int_{-\infty}^{\infty} T(\omega) S_j(k) E(k,\omega)\, dk\, d\omega \tag{70}$$

$$\frac{\partial \mathcal{L}}{\partial a_j} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \int_{-\infty}^{\infty} O_j^*(\omega) E(k,\omega)\, dk\, d\omega \tag{71}$$

where $O_j(\omega)$ is the DTFT of the spike train $o_j[t]$, and $O_j^*(\omega)$ is the complex conjugate of $O_j(\omega)$.

A.3.3 Proof of Theorem 1

Theorem 1 *Consider a SNN employing LIF neurons and a Cauchy surrogate gradient $g(x) = \frac{\alpha}{\pi(\alpha^2+x^2)}$. For any two spatial frequencies $k_2 > k_1 > 0$, assume there exist positive constants c_1, c_2 such that $A(k_1) > c_1 > 0$ and $A(k_2) < c_2 < \infty$. Furthermore, assume a constant $\rho > 0$ exists such that the correlation condition $\frac{|\langle TX, E(k,\cdot)\rangle|}{\|TX\| \cdot \|E(k,\cdot)\|} \geq \rho > 0$ holds. It follows that:*

$$\lim_{\epsilon \to 0} \frac{\mu(\{w_j \in B_\epsilon : |\frac{\partial \mathcal{L}(k_1)}{\partial w_j}| > |\frac{\partial \mathcal{L}(k_2)}{\partial w_j}|\})}{\mu(B_\epsilon)} = 1 \tag{72}$$

where B_ϵ is a ball centered at the origin with radius ϵ, $\mu(\cdot)$ is the Lebesgue measure, and $|\frac{\partial \mathcal{L}(k)}{\partial w_j}|$ denotes the magnitude of the gradient contribution from spatial frequency k, where $TX = T(\omega)X^*(\omega)$ and $E(k,\cdot)$ represents the spatial frequency domain error.

The magnitude of the gradient contribution from spatial frequency k is defined as:

$$|\frac{\partial \mathcal{L}(k)}{\partial w_j}| = \frac{1}{4\pi^2} \left| \int_{-\pi}^{\pi} T(\omega) X^*(\omega) E(k,\omega) d\omega \right| \cdot |S_j(k)| \tag{73}$$

where $|S_j(k)|$ is given by:

$$|S_j(k)| = \frac{|a_j|}{2\pi |w_j|^2} e^{-\frac{\alpha |k|}{|w_j|}} \cdot |ib_j - \alpha \cdot \mathrm{sgn}(k) \cdot \mathrm{sgn}(w_j)| \tag{74}$$

Let:

$$Q_j(k) \frac{|a_j|}{2\pi |w_j|^2} |ib_j - \alpha \cdot \mathrm{sgn}(k) \cdot \mathrm{sgn}(w_j)| \cdot \frac{1}{4\pi^2} \left| \int_{-\pi}^{\pi} T(\omega) X^*(\omega) E(k,\omega) d\omega \right| \tag{75}$$

Then:

$$|\frac{\partial \mathcal{L}(k)}{\partial w_j}| = e^{-\frac{\alpha|k|}{|w_j|}} \cdot Q_j(k) \tag{76}$$

According to the Cauchy-Schwarz inequality, $I(k) = |\langle TX, E(k,\cdot)\rangle| \leq \|TX\| \cdot \|E(k,\cdot)\|$. Given the condition $\frac{|\langle TX, E(k,\cdot)\rangle|}{\|TX\|\cdot\|E(k,\cdot)\|} \geq \rho > 0$, it implies $I(k) \geq \rho\|TX\| \cdot \|E(k,\cdot)\|$. Based on Parseval's Theorem, $A(k) \leq \|E(k,\cdot)\| \leq \sqrt{T}A(k)$ holds. Thus:

$$Q_j(k) \geq p\rho\|TX\|A(k), \quad Q_j(k) \leq p\|TX\|\sqrt{T}A(k) \tag{77}$$

Comparing $Q_j(k_1)$ and $Q_j(k_2)$ yields:

$$\frac{Q_j(k_1)}{Q_j(k_2)} \geq \frac{p\rho\|TX\|A(k_1)}{p\|TX\|\sqrt{T}A(k_2)} = \frac{\rho A(k_1)}{\sqrt{T}A(k_2)} > \frac{\rho c_1}{\sqrt{T}c_2} \tag{78}$$

Let $C_4 = \frac{\sqrt{T}c_2}{\rho c_1}$. The ratio of gradient magnitudes is:

$$\frac{|\frac{\partial \mathcal{L}(k_1)}{\partial w_j}|}{|\frac{\partial \mathcal{L}(k_2)}{\partial w_j}|} = e^{\frac{\alpha(k_2-k_1)}{|w_j|}} \cdot \frac{Q_j(k_1)}{Q_j(k_2)} > e^{\frac{\alpha(k_2-k_1)}{|w_j|}} \cdot \frac{1}{C_4} \tag{79}$$

The target inequality holds if $e^{\frac{\alpha(k_2-k_1)}{|w_j|}} > C_4$, which simplifies to:

$$|w_j| < \frac{\alpha(k_2-k_1)}{\ln C_4} W \tag{80}$$

Define the set $\mathcal{A} = \{w_j \in \mathbb{R} \setminus \{0\} : |\frac{\partial \mathcal{L}(k_1)}{\partial w_j}| > |\frac{\partial \mathcal{L}(k_2)}{\partial w_j}|\}$. Based on the derivation above, the target inequality is satisfied for all $0 < |w_j| < W$, which implies that the punctured interval $(-W, W) \setminus \{0\}$ is a subset of $\mathcal{A}$. For a symmetric neighborhood $B_\epsilon = (-\epsilon, \epsilon)$ where $0 < \epsilon < W$, the Lebesgue measure is $\mu(B_\epsilon) = 2\epsilon$. The measure of the subset within B_ϵ where the inequality holds is $\mu(\mathcal{A} \cap B_\epsilon) = 2\epsilon$. Finally, the limit of the measure ratio as $w_j \to 0$ is calculated as follows:

$$\lim_{\epsilon \to 0^+} \frac{\mu\left(\left\{w_j \in B_\epsilon : \left|\frac{\partial \mathcal{L}(k_1)}{\partial w_j}\right| > \left|\frac{\partial \mathcal{L}(k_2)}{\partial w_j}\right|\right\}\right)}{\mu(B_\epsilon)} = \lim_{\epsilon \to 0^+} \frac{\mu(\mathcal{A} \cap B_\epsilon)}{\mu(B_\epsilon)} = \lim_{\epsilon \to 0^+} \frac{2\epsilon}{2\epsilon} = 1 \tag{81}$$

Therefore, in the infinitesimal neighborhood as $w_j \to 0$, the target inequality holds almost everywhere. The proofs for b_j and a_j are analogous.

A.3.4 Proof of Theorem 2

Theorem 2 *Consider a SNN that utilizes LIF neurons and a Cauchy distribution $g(x) = \frac{\alpha}{\pi(\alpha^2+x^2)}$ as the surrogate gradient. Let $k_2 > k_1 > 0$ and $|\mathcal{F}[f](k_1)| > 0$. Assume there exists a positive constant $\rho > 0$ such that for any frequency k, the condition $\frac{|\langle TX, E(k,\cdot)\rangle|}{\|TX\|\cdot\|E(k,\cdot)\|} \geq \rho > 0$ holds. If there exists a constant $C_a > 0$ such that $A(k_2) < C_a$, then for any $\epsilon > 0$, there exists $M > 0$ such that for any $w_j \in [-M, M] \setminus \{0\}$, the condition:*

$$\left|\frac{\partial \mathcal{L}(k_1)}{\partial w_j}\right| = \left|\frac{\partial \mathcal{L}(k_2)}{\partial w_j}\right| \neq 0 \tag{82}$$

holds, then the relative error satisfies $\Delta_F(k_1) = \frac{A(k_1)}{|\mathcal{F}[f](k_1)|} < \epsilon$.

Following the derivation of Theorem 1, for any k, the gradient magnitude is:

$$\left|\frac{\partial \mathcal{L}(k)}{\partial w_j}\right| = e^{-\frac{\alpha|k|}{|w_j|}} \cdot Q_j(k) \tag{83}$$

where $Q_j(k) = p_j(k) \cdot I(k)$, $p_j(k) = \frac{|a_j|}{2\pi|w_j|^2}|ib_j - \alpha \cdot \mathrm{sgn}(k) \cdot \mathrm{sgn}(w_j)|$, and $I(k) = |\langle TX, E(k,\cdot)\rangle|$. Since $k_2 > k_1 > 0$, it follows that $\mathrm{sgn}(k_1) = \mathrm{sgn}(k_2) = +1$, hence:

$$p_j(k_1) = p_j(k_2) = \frac{|a_j|}{2\pi|w_j|^2}|ib_j - \alpha \cdot \mathrm{sgn}(w_j)|p \tag{84}$$

The gradient equality condition $\left|\frac{\partial \mathcal{L}(k_1)}{\partial w_j}\right| = \left|\frac{\partial \mathcal{L}(k_2)}{\partial w_j}\right| \neq 0$ implies:

$$\left|\frac{\partial \mathcal{L}(k_1)}{\partial w_j}\right| = \left|\frac{\partial \mathcal{L}(k_2)}{\partial w_j}\right| \neq 0 \tag{85}$$

Similar to Theorem 1, the correlation assumption and Parseval's theorem yield:

$$p\rho\|TX\|A(k) \leq Q_j(k) \leq p\|TX\|\sqrt{T}A(k) \tag{86}$$

Utilizing the lower bound for $Q_j(k_1)$ and the upper bound for $Q_j(k_2)$ leads to:

$$e^{\frac{\alpha(k_2-k_1)}{|w_j|}} \cdot p\rho\|TX\|A(k_1) \leq p\|TX\|\sqrt{T}A(k_2) \tag{87}$$

Dividing by $p\|TX\| > 0$ and rearranging terms gives:

$$A(k_1) \leq \frac{\sqrt{T}}{\rho} e^{-\frac{\alpha(k_2-k_1)}{|w_j|}} A(k_2) \tag{88}$$

Applying the condition $A(k_2) < C_a$ results in:

$$A(k_1) < \frac{\sqrt{T}C_a}{\rho} e^{-\frac{\alpha(k_2-k_1)}{|w_j|}} \tag{89}$$

The relative error $\Delta_F(k_1)$ is thus bounded as:

$$\Delta_F(k_1) = \frac{A(k_1)}{|\mathcal{F}[f](k_1)|} < \frac{\sqrt{T}C_a}{\rho|\mathcal{F}[f](k_1)|} e^{-\frac{\alpha(k_2-k_1)}{|w_j|}} \tag{90}$$

By defining $C = \frac{\sqrt{T}C_a}{\rho|\mathcal{F}[f](k_1)|}$, we have $\Delta_F(k_1) < Ce^{-\frac{\alpha(k_2-k_1)}{|w_j|}}$. To satisfy $\Delta_F(k_1) < \epsilon$: If $\epsilon \geq C$, the inequality holds for any w_j. If $\epsilon < C$, we require $e^{-\frac{\alpha(k_2-k_1)}{|w_j|}} < \frac{\epsilon}{C}$, which simplifies to:

$$|w_j| < \frac{\alpha(k_2-k_1)}{\ln(C/\epsilon)} M \tag{91}$$

Thus, $\Delta_F(k_1) < \epsilon$ holds when $|w_j| < M$. The proofs for b_j and a_j are analogous.

A.3.5 Proof of Theorem 3

Theorem 3 *Consider a SNN that employs LIF neurons and the Cauchy distribution $g(x) = \frac{\alpha}{\pi(\alpha^2+x^2)}$ as the surrogate gradient. Let $k_2 > k_1 > 0$ and $|\mathcal{F}[f](k_1)| > 0$. Assume that there exists a positive constant $\rho > 0$ such that the correlation condition $\frac{|\langle TX, E(k,\cdot)\rangle|}{\|TX\| \cdot \|E(k,\cdot)\|} \geq \rho > 0$ holds. For any $\Lambda > 0$, there exists $M > 0$ such that if $A(k_2) > M$ and the condition:*

$$\left|\frac{\partial \mathcal{L}(k_1)}{\partial w_j}\right| = \left|\frac{\partial \mathcal{L}(k_2)}{\partial w_j}\right| \neq 0 \tag{92}$$

holds, then the relative error $\Delta_F(k_1)$, defined as $\frac{A(k_1)}{|\mathcal{F}[f](k_1)|}$, satisfies $\Delta_F(k_1) > \Lambda$.

Following the derivation of Theorem 1, for any k, the gradient magnitude is:

$$\left|\frac{\partial \mathcal{L}(k)}{\partial w_j}\right| = e^{-\frac{\alpha|k|}{|w_j|}} \cdot Q_j(k) \tag{93}$$

where $Q_j(k) = p_j(k) \cdot I(k)$, $p_j(k) = \frac{|a_j|}{2\pi|w_j|^2}|ib_j - \alpha \cdot \text{sgn}(k) \cdot \text{sgn}(w_j)|$, and $I(k) = |\langle TX, E(k,\cdot)\rangle|$. The gradient equality condition $|\frac{\partial \mathcal{L}(k_1)}{\partial w_j}| = |\frac{\partial \mathcal{L}(k_2)}{\partial w_j}| \neq 0$ implies:

$$e^{\frac{\alpha(k_2-k_1)}{|w_j|}} = \frac{Q_j(k_2)}{Q_j(k_1)} = \frac{I(k_2)}{I(k_1)} \tag{94}$$

By applying the upper bound for $Q_j(k_1)$ and the lower bound for $Q_j(k_2)$ from Theorem 1:

$$e^{\frac{\alpha(k_2-k_1)}{|w_j|}} \cdot p\|TX\|\sqrt{T}A(k_1) \geq p\rho\|TX\|A(k_2) \tag{95}$$

Rearranging the terms yields:

$$A(k_1) \geq \frac{\rho}{\sqrt{T}} e^{-\frac{\alpha(k_2-k_1)}{|w_j|}} A(k_2) \tag{96}$$

The relative error $\Delta_F(k_1)$ is bounded from below as:

$$\Delta_F(k_1) = \frac{A(k_1)}{|\mathcal{F}[f](k_1)|} \geq \frac{\rho}{\sqrt{T}|\mathcal{F}[f](k_1)|} e^{-\frac{\alpha(k_2-k_1)}{|w_j|}} A(k_2) \tag{97}$$

By defining $C_0 = \frac{\rho}{\sqrt{T}|\mathcal{F}[f](k_1)|} e^{-\frac{\alpha(k_2-k_1)}{|w_j|}}$, we obtain $\Delta_F(k_1) \geq C_0 \cdot A(k_2)$. To ensure $\Delta_F(k_1) > \Lambda$, it is sufficient to have $A(k_2) > \Lambda/C_0$. Define M as follows:

$$M = \Lambda/C_0 = \frac{\Lambda\sqrt{T}|\mathcal{F}[f](k_1)|}{\rho e^{-\alpha(k_2-k_1)/|w_j|}} \tag{98}$$

Consequently, the inequality $\Delta_F(k_1) > \Lambda$ holds when $A(k_2) > M$. The proofs for b_j and a_j are analogous.

A.3.6 Proof of Theorem 4

Theorem 4 *Consider a SNN that employs LIF neurons and the Cauchy distribution $g(x) = \frac{\alpha}{\pi(\alpha^2+x^2)}$ as the surrogate gradient. The gradient density at temporal frequency ω decomposes into a product of three factors: $\left|\frac{\partial \mathcal{L}}{\partial w_j}(\omega)\right| = |T(\omega)| \cdot |X^*(\omega)| \cdot |SE_j(\omega)|$. The learning preference of the network for two distinct temporal frequencies $\omega_1 < \omega_2$ is determined by their ratio,* $\mathrm{Ratio}(\omega_1, \omega_2)$*:*

$$\mathrm{Ratio}(\omega_1, \omega_2) = \frac{|T(\omega_1)|}{|T(\omega_2)|} \cdot \frac{|X^*(\omega_1)|}{|X^*(\omega_2)|} \cdot \frac{|SE_j(\omega_1)|}{|SE_j(\omega_2)|} \tag{99}$$

The network prioritizes ω_1 when $\mathrm{Ratio} > 1$ *and ω_2 when* $\mathrm{Ratio} < 1$*, while* $\mathrm{Ratio} = 1$ *indicates equal priority. Thus, the temporal learning trajectory is dynamically governed by three factors—the temporal kernel $T(\omega)$, the input spectrum $X(\omega)$, and the integrated error magnitude $SE_j(\omega)$—precluding any inherent low-frequency bias.*

The gradient contribution from temporal frequency ω is defined as follows:

$$\frac{\partial \mathcal{L}}{\partial w_j}(\omega) = \frac{1}{4\pi^2} T(\omega) X^*(\omega) \int_{-\infty}^{\infty} S_j(k) E(k, \omega)\, dk \tag{100}$$

where $\frac{\partial \mathcal{L}}{\partial w_j}(\omega)$ denotes the density of the gradient contribution at temporal frequency ω. The magnitude is of primary interest:

$$\left|\frac{\partial \mathcal{L}}{\partial w_j}(\omega)\right| = \frac{1}{4\pi^2} |T(\omega)| \cdot |X^*(\omega)| \cdot \left|\int_{-\infty}^{\infty} S_j(k) E(k, \omega)\, dk\right| \tag{101}$$

We define $SE_j(\omega)$ as $|\int_{-\infty}^{\infty} S_j(k) E(k, \omega)\, dk|$. $SE_j(\omega)$ represents the total contribution magnitude at temporal frequency ω, which is obtained by integrating the spatio-temporal error spectrum weighted by the spatial frequency kernel $S_j(k)$ over the spatial frequency domain. Consequently, it follows that:

$$\left|\frac{\partial \mathcal{L}}{\partial w_j}(\omega)\right| \propto |T(\omega)| \cdot |X^*(\omega)| \cdot |SE_j(\omega)| \tag{102}$$

For distinct frequencies $\omega_1 \neq \omega_2$, the ratio of their magnitudes is defined as:

$$\mathrm{Ratio}(\omega_1, \omega_2) \frac{|T(\omega_1)|}{|T(\omega_2)|} \cdot \frac{|X^*(\omega_1)|}{|X^*(\omega_2)|} \cdot \frac{|SE_j(\omega_1)|}{|SE_j(\omega_2)|} \tag{103}$$

If $\mathrm{Ratio} > 1$, the network prioritizes ω_1; if $\mathrm{Ratio} < 1$, the network prioritizes ω_2; and if $\mathrm{Ratio} = 1$, the learning priorities for both frequencies are equivalent. The learning direction in the temporal dimension is dynamically determined by these three factors, implying that no inherent low-frequency bias exists.

Three specific scenarios are analyzed as follows:

1. Leakage factor dominance: As $\beta \to 1$, $|T(\omega_1)|$ diverges at $\omega = 0$. For $\omega_1 < \omega_2$, the ratio $\frac{|T(\omega_1)|}{|T(\omega_2)|}$ approaches infinity. Under these conditions, even if the other factors are small, $\mathrm{Ratio} \to \infty$ holds, which compels the network to prioritize low-frequency information.
2. Input signal dominance: When the input signal energy is concentrated in high-frequency regions (e.g., $|X^*(\omega_1)| \ll |X^*(\omega_2)|$), $\mathrm{Ratio} < 1$ may occur even if the temporal frequency kernel favors low frequencies, thereby forcing the network to prioritize high-frequency components.
3. Error distribution dominance: When the low-frequency error is significantly reduced (i.e., $|SE_j(\omega_1)| \ll |SE_j(\omega_2)|$), Ratio may still be less than 1 even if other factors favor low frequencies, prompting the network to transition toward learning high-frequency details.

A.3.7 Generalization to Other Surrogate Gradients

In the derivations above, we adopt the Cauchy distribution as the surrogate gradient:

$$g(x) = \frac{\alpha}{\pi(\alpha^2 + x^2)}, \qquad \hat{g}(k) = e^{-\alpha|k|}. \tag{104}$$

However, the theoretical framework does not depend on the specific form of the Cauchy distribution, but rather on a common property shared by most surrogate gradients: their Fourier transform decays as the frequency increases. To illustrate this point, we take the Gaussian function as another commonly used surrogate gradient:

$$g_{\mathrm{gauss}}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/(2\sigma^2)}, \qquad \sigma > 0, \tag{105}$$

where σ controls the smoothness. Its Fourier transform is given by:

$$\hat{g}_{\mathrm{gauss}}(k) = e^{-\sigma^2 k^2/2}. \tag{106}$$

The Gaussian surrogate gradient decays exponentially across the entire frequency domain with a constant prefactor of 1, which cleanly satisfies the required structural properties without any low-frequency caveat.

Following the same derivation procedure as in Section A.3.1, we only need to replace the Cauchy-based spatial kernel $S_j(k)$ with its Gaussian-based counterpart:

$$S_j^{\mathrm{gauss}}(k) = \frac{a_j}{2\pi w_j |w_j|} \left(ib_j - \alpha \cdot \mathrm{sgn}(k) \cdot \mathrm{sgn}(w_j)\right) e^{-i\frac{kb_j}{w_j}} \cdot \hat{g}_{\mathrm{gauss}}\left(\frac{k}{w_j}\right) \tag{107}$$

$$\hat{g}_{\mathrm{gauss}}\left(\frac{k}{w_j}\right) = e^{-\sigma^2 (k/w_j)^2/2}. \tag{108}$$

It should be emphasized that all derivations in Section A.3.2 remain unchanged, including the temporal frequency kernel $T(\omega)$, the spatio-temporal error spectrum $E(k, \omega)$, and the BPTT recursion, because these parts depend only on the LIF neuron dynamics and the structure of error backpropagation, rather than the specific form of the surrogate gradient.

Furthermore, the proofs of Theorems A.3.3–A.3.6 rely only on the following two structural properties of the spatial kernel $S_j(k)$:

1. Exponential decay in the spatial frequency k:

$$|S_j(k)| \le C_j \cdot e^{-\eta|k|/|w_j|}, \qquad \eta > 0, \tag{109}$$

   which holds for both the Cauchy and the Gaussian surrogate gradients. For the Cauchy distribution, $\eta = \alpha$; for the Gaussian, the decay is even stronger with $\eta \propto |k|$.
2. Polynomial boundedness of the frequency-domain kernel: In the spatial kernel $S_j(k)$, the exponential decay term is multiplied by at most a polynomial factor $|k|^m$ in k (with finite m), rather than an exponentially growing factor. For both the Cauchy and Gaussian surrogates, the prefactor is constant, trivially satisfying this condition.

In practice, commonly used surrogate gradients satisfy both properties. For example:

- Cauchy distribution: $\hat{g}(k) = e^{-\alpha|k|}$, constant prefactor;
- Gaussian distribution: $\hat{g}(k) = e^{-\sigma^2 k^2/2}$, constant prefactor;
- Derivative of tanh: $\hat{g}(k) = \frac{\pi k/\alpha}{\sinh(\pi k/\alpha)} \sim \frac{2\pi k}{\alpha} e^{-\pi k/\alpha}$, with linear prefactor k, which also satisfies the boundedness condition;
- SLAYER gradientShrestha & Orchard (2018): $\hat{g}(k) = e^{-\alpha|k|}$, constant prefactor;
- Mexican Hat-type gradient: $\hat{g}(k) = |k|^2 e^{-\sigma^2 k^2/2}$, with polynomial prefactor $|k|^2$, which satisfies the boundedness condition.

As long as the Fourier transform of the surrogate gradient exhibits exponential decay and the polynomial prefactor is bounded, the STSBP theoretical framework applies qualitatively and even quantitatively. Therefore, the STSBP theory is not restricted to the Cauchy distribution.

A.3.8 Theoretical Framework Generalization

The theoretical framework proposed in this paper generalizes to more complex SNN architectures. Since the error spectrum $E(k, \omega)$ and input spectrum $X(\omega)$ remain independent of the architecture of the network, we focus on the sensitivity of $S_j(k)$, $T(\omega)$, and the output spectrum $O_j(\omega)$ of the model to modifications of the structure, as analyzed below:

- Number of Neurons: Fourier-domain summation linearly superimposes $S_j(k)$, preserving the spectral decay of individual neurons while maintaining the filtering properties of $T(\omega)$. Consequently, the output spectrum $O_j(\omega)$ undergoes amplitude scaling as the count of neurons increases, while the decay trend of frequency remains invariant.
- Multiple Hidden Layer Structure: As $S_j(k)$ exhibits frequency-dependent decay, multi-layer composition yields a recursive spectral product that preserves the high-frequency decay of the structure. The reset mechanism, representing the repeated multiplication of $T(\omega)$, maintains stable filtering properties. Furthermore, as a product of the temporal responses of successive layers, the filtering features of $O_j(\omega)$ remain invariant to the depth of the network.
- High-Dimensional Input: In high-dimensional spaces, k generalizes to a frequency vector $\mathbf{k}$, where $S_j(\mathbf{k})$ depends on the norm of the frequency $\|\mathbf{k}\|$ while preserving decay across all dimensions. Since $T(\omega)$ operates exclusively in the temporal domain, it remains decoupled from and invariant to the spatial dimensionality of the input. Likewise, the temporal spectrum $O_j(\omega)$ is independent of the dimensionality of the space.
- High-Dimensional Output: In multi-output tasks, the total loss aggregates the contributions of individual nodes, where each $O_j(\omega)$ exhibits identical spectral properties. Internal operators $S_j(k)$ and $T(\omega)$ remain invariant to the scale of the output nodes.

In summary, the analytical conclusions regarding $S_j(k)$, $T(\omega)$, and $O_j(\omega)$ in single-hidden-layer SNNs exhibit strong robustness to neuron scale, hidden layer count, and input/output dimensions, enabling a qualitative generalization to more complex spiking neural network models.

Table 4: Publicly available dataset descriptions.

Dataset	Dim	Dataset Size	Frequency	Information
ETTm1	7	(34465, 11521, 11521)	15 min	Electricity
ETTm2	7	(34465, 11521, 11521)	15 min	Electricity
ETTh1	7	(8545, 2881, 2881)	Hourly	Electricity
ETTh2	7	(8545, 2881, 2881)	Hourly	Electricity
Traffic	862	(12185, 1757, 3509)	Hourly	Transportation
Exchange	8	(5120, 665, 1422)	Daily	Weather
Weather	21	(36792, 5271, 10540)	10 min	Weather
Solar-Energy	137	(36601, 5161, 10417)	10 min	Energy

Algorithm 1 Frequency-LIF Core Algorithm

Require: Base Model $M = \{M_{pre}, M_{post}\}$ with parameters ϑ; Hyper-parameters $\{\beta_0, \xi, \eta, T\}$.
Ensure: Optimized parameters ϑ.

Initialize ϑ and $Switch \leftarrow$ False.
for epoch $= 1$ to $MaxEpochs$ **do**
  for batch i with sequence $\{x[t]\}_{t=0}^{T-1}$ **do**
    // Phase 1: TFDAM
    1. Forward Propagation:
    for $t = 0$ to $T-1$ **do**
      $c[t] = M_{pre}(x[t])$ *// Intermediate encoding*
      Dynamically adjust $\beta[t]$ via Eq. (25)
      Update $v_j[t]$ and $o_j[t]$ via Eq. (26).
      $\hat{y}[t] = M_{post}(o_j[t])$ *// Intermediate decoding*
    end for
    $\hat{y} = \frac{1}{T}\sum_{t=0}^{T-1} \hat{y}[t]$ *// Final prediction aggregation*
    // Phase 2: SFDAM
    2. Critical Point Detection:
    if $Switch$ is False **then**
      Calculate $R_{grad}^{(i)}$ using Fourier Transform $\mathcal{F}$ as per Eq. (21).
      if $\sum_{j=i-\mathcal{N}_{batch}+1}^{i} \mathbb{I}(R_{grad}^{(j)} > 1) > \xi$ **then** $Switch \leftarrow$ True. (Eq. (22))
      $\mathcal{L} = \text{MSE}(\hat{y}, y)$.
    else
      3. High-Frequency Learning Mode:
      Decompose $(\hat{y}, y) \rightarrow (\hat{y}_s, y_s)$ and $(\hat{y}_t, y_t)$.
      Set $\lambda_s = 1 + \eta$ and $\lambda_t = 1 - \eta$ via Eq. (24).
      $\mathcal{L} = \mathcal{L}_{TS}(\hat{y}_s, y_s, \hat{y}_t, y_t)$ via Eq. (23).
    end if
    4. Parameter Update:
    Update M parameters ϑ by minimizing L.
  end for
end for

A.4 Experiments Details

A.4.1 Experimental Setup

Datasets Details Table 4 summarizes the detailed statistical information of these datasetsZhou et al. (2021); Wu et al. (2021); Lai et al. (2018).

Implementation Details The model is trained using the Adam optimizer Kingma & Ba (2014) with an initial learning rate of 10^{-3} and $L2$ loss as the optimization objective. The batch size is set to 32, and the number of training epochs is fixed at 10. The SNN time step is set to $T = 4$Feng et al. (2025). Regarding hyperparameter settings, k_l and k_h are set to 0.1 and 0.5, respectively. The tolerance threshold is $\xi = 1$, and the adjustment factor is $\eta \in \{0.5, 0.8\}$. Within the TFDAM module, the initial leakage factor is $\beta_0 = 0.7$, the minimum leakage factor is $\beta_{\min} = 0.5$, and the exponential decay factor is $t_{decay} = 0.5$. The surrogate gradient utilizes a Cauchy distribution with a shape parameter $\alpha = 0.1$. Following the conventions of previous research Liu et al. (2023), all experiments fix the look-back window length at $L = 96$ and evaluate model performance across prediction horizons $H \in \{96, 192, 336, 720\}$. Evaluation metrics include Mean Squared Error (MSE) and Mean Absolute Error (MAE).

A.4.2 Experimental Results Details

Experimental Results Details Detailed experimental results for the "Comparison with State-of-the-Art Methods", "Generality Analysis of Frequency-LIF", and "Ablation Study" sections are provided in Tables 6, 7, and 8, respectively.

Robustness Analysis of Frequency-LIF To evaluate the robustness of Frequency-LIF under different initialization conditions, we conduct experiments on eight benchmark datasets across four prediction horizons using iTransformer as the base model. For each setting, we run experiments with five different random seeds and report the mean and standard deviation of MSE and MAE to assess the stability of model performance against random initialization perturbations.

As shown in Table 5, Frequency-LIF achieves consistently low prediction errors with remarkably small standard deviations across all datasets and prediction horizons. Taking ETTh1 as an example, the standard deviations of MSE across the four prediction horizons do not exceed 0.004, and those of MAE do not exceed 0.003. On the Weather dataset, the standard deviations of both MSE and MAE are below 0.001 in most settings, demonstrating exceptional stability. Even on the larger and more volatile Traffic and Solar-Energy datasets, Frequency-LIF maintains low variance. These results indicate that Frequency-LIF is insensitive to random initialization and consistently achieves superior prediction performance across diverse data environments, validating its strong robustness.

Table 5: Robustness of Frequency-LIF performance. The results are obtained from five random seeds.

Datasets		ETTm1		ETTm2		ETTh1		ETTh2		Exchange		Weather		Traffic		Solar-energy	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Horizon	96	0.339±0.004	0.372±0.003	0.181±0.001	0.262±0.001	0.387±0.001	0.408±0.001	0.296±0.003	0.345±0.002	0.088±0.001	0.210±0.002	0.171±0.000	0.211±0.001	0.410±0.001	0.279±0.001	0.206±0.001	0.237±0.001
	192	0.379±0.003	0.394±0.002	0.252±0.002	0.310±0.002	0.439±0.002	0.438±0.003	0.373±0.002	0.394±0.001	0.178±0.002	0.303±0.002	0.223±0.001	0.256±0.001	0.432±0.000	0.287±0.000	0.238±0.001	0.263±0.001
	336	0.416±0.002	0.418±0.002	0.311±0.002	0.347±0.002	0.484±0.004	0.456±0.002	0.417±0.005	0.427±0.004	0.326±0.007	0.414±0.004	0.280±0.000	0.299±0.000	0.449±0.001	0.297±0.001	0.254±0.001	0.278±0.001
	720	0.495±0.011	0.462±0.006	0.408±0.003	0.404±0.002	0.497±0.002	0.482±0.001	0.424±0.003	0.442±0.002	0.866±0.007	0.706±0.005	0.356±0.001	0.349±0.001	0.485±0.000	0.320±0.000	0.257±0.001	0.282±0.001
	Avg	0.407±0.005	0.411±0.003	0.288±0.002	0.331±0.002	0.452±0.002	0.446±0.002	0.378±0.003	0.402±0.003	0.365±0.004	0.408±0.003	0.258±0.001	0.279±0.001	0.444±0.001	0.296±0.001	0.239±0.001	0.265±0.001

A.4.3 Theorem Validation Experiments

Validation Experiment for Theorem 1 To validate Theorem 1, this paper retains gradient updates in the spatial frequency domain of SNNs while truncating them in the temporal frequency domain. Without truncating the temporal frequency domain, it would be impossible to distinguish whether the effects of variable changes originate from the spatial frequency domain or the temporal frequency domain. This treatment is consistent with the description in Section A.4.3. Model weights are initialized using a Gaussian distribution with a mean of 0.001 and standard deviation of 0.001, and experiments are conducted on the ETTh1 dataset with a prediction length of 96. Figure 2(a) displays the time-domain waveform of the target sequence $f(x)$, and Figure 2(b) presents the amplitude distribution of the first 50 frequency components of its Fourier transform $\mathcal{F}[f](k)$.

As illustrated in Figure 2(b), the energy of the target sequence concentrates primarily in the low-frequency region, and the amplitudes of high-frequency components are relatively small. As shown by the relative error curves in Figure 2(c), the first four frequency peaks converge in order from low to high frequencies; that is, lower-frequency components converge earlier and faster. This phenomenon is highly consistent with Theorem 1: under sufficiently small weight initialization, low-frequency components possess higher learning priority during gradient training.

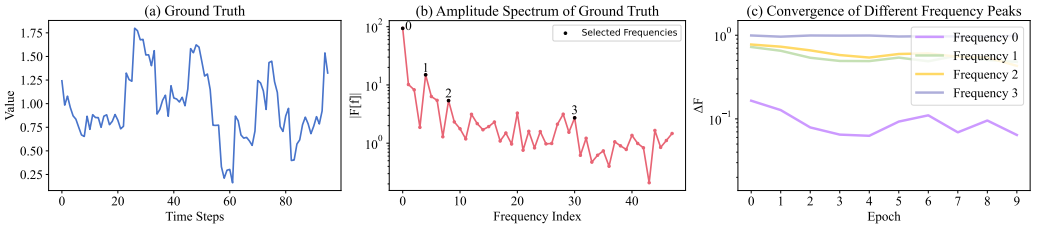

Figure 2: Verification visualization of Theorem 1. From left to right are the ground truth, the target amplitude spectrum, and the learning order of various frequency components.

Validation Experiments for Theorem 2 and Theorem 3 To validate Theorem 2 and Theorem 3, this paper observes the learning behavior of the model for various frequency components by gradually increasing the initialization weight mean while maintaining a constant standard deviation of 0.001. Specifically, three initialization means are defined: 0.001, 0.3, and 0.8. The comparisons between their predicted and ground-truth amplitudes are shown in Figure 3(a), (b), and (c), respectively.

Table 6: Full results of the long-term forecasting task. The look-back window length L is set to 96, and the prediction horizons H take values in $\{96, 192, 336, 720\}$. The best and second-best results are red and blue, respectively.

Methods		Frequency-LIF (Ours)		AMP2 (AAAI2026)		DLIF (ICLR2026)		TS-LIF (ICLR2025)		AGMM (AAAI2025)		AT-LIF (NeurIPS2025)		TC-LIF (AAAI2024)		LM-H (ICLR2024)		LM_HT (NeurIPS2024)		LIF (Vanilla)		GELU (Original)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.339	0.372	0.341	0.375	0.347	0.377	0.368	0.388	0.345	0.377	0.347	0.378	0.343	0.374	0.350	0.382	0.367	0.392	0.342	0.375	0.337	0.372
	192	0.379	0.394	0.390	0.401	0.401	0.404	0.435	0.424	0.390	0.401	0.388	0.401	0.388	0.400	0.393	0.403	0.399	0.406	0.381	0.397	0.384	0.399
	336	0.416	0.418	0.427	0.429	0.426	0.480	0.449	0.472	0.445	0.437	0.432	0.431	0.428	0.433	0.429	0.433	0.427	0.428	0.427	0.460	0.439	
	720	0.495	0.462	0.502	0.465	0.521	0.475	0.519	0.473	0.506	0.469	0.531	0.481	0.533	0.481	0.499	0.465	0.507	0.469	0.512	0.475	0.524	0.475
	Avg	0.407	0.412	0.415	0.417	0.424	0.420	0.450	0.434	0.428	0.423	0.426	0.423	0.424	0.421	0.419	0.420	0.426	0.424	0.416	0.418	0.426	0.421
ETTm2	96	0.181	0.262	0.189	0.267	0.185	0.266	0.193	0.274	0.186	0.268	0.186	0.268	0.184	0.267	0.183	0.265	0.188	0.269	0.192	0.271	0.182	0.267
	192	0.252	0.310	0.249	0.308	0.260	0.318	0.264	0.321	0.260	0.318	0.256	0.314	0.258	0.315	0.264	0.320	0.260	0.316	0.262	0.316	0.256	0.313
	336	0.311	0.347	0.311	0.347	0.321	0.354	0.330	0.363	0.344	0.371	0.347	0.372	0.322	0.357	0.342	0.369	0.329	0.361	0.340	0.366	0.338	0.366
	720	0.408	0.404	0.412	0.404	0.418	0.410	0.423	0.410	0.431	0.417	0.415	0.408	0.427	0.415	0.420	0.408	0.431	0.416	0.420	0.410	0.413	0.406
	Avg	0.288	0.331	0.290	0.332	0.296	0.337	0.302	0.342	0.305	0.344	0.301	0.340	0.298	0.338	0.302	0.340	0.302	0.340	0.304	0.341	0.297	0.338
ETTh1	96	0.387	0.408	0.388	0.410	0.392	0.407	0.392	0.409	0.391	0.408	0.389	0.406	0.389	0.406	0.392	0.407	0.393	0.406	0.390	0.406	0.387	0.405
	192	0.439	0.438	0.447	0.438	0.450	0.440	0.454	0.446	0.446	0.439	0.446	0.441	0.447	0.441	0.447	0.442	0.447	0.438	0.446	0.445	0.439	0.440
	336	0.484	0.456	0.484	0.457	0.481	0.455	0.484	0.458	0.484	0.457	0.489	0.462	0.485	0.459	0.483	0.455	0.486	0.457	0.496	0.467	0.485	0.459
	720	0.497	0.482	0.518	0.495	0.509	0.492	0.488	0.481	0.516	0.495	0.510	0.489	0.495	0.484	0.523	0.499	0.521	0.498	0.512	0.490	0.501	0.490
	Avg	0.452	0.446	0.459	0.450	0.458	0.448	0.454	0.448	0.459	0.450	0.458	0.450	0.454	0.448	0.461	0.451	0.462	0.450	0.461	0.450	0.454	0.448
ETTh2	96	0.296	0.345	0.307	0.354	0.305	0.354	0.303	0.351	0.305	0.351	0.308	0.352	0.307	0.351	0.311	0.355	0.314	0.356	0.310	0.353	0.312	0.356
	192	0.373	0.394	0.376	0.395	0.384	0.402	0.390	0.402	0.386	0.404	0.390	0.405	0.389	0.404	0.391	0.403	0.394	0.404	0.400	0.406	0.383	0.400
	336	0.417	0.427	0.415	0.425	0.448	0.444	0.437	0.437	0.428	0.431	0.430	0.431	0.430	0.431	0.432	0.431	0.438	0.437	0.432	0.431	0.423	0.432
	720	0.424	0.442	0.469	0.468	0.465	0.459	0.453	0.459	0.432	0.446	0.433	0.445	0.431	0.444	0.427	0.442	0.435	0.449	0.431	0.445	0.460	0.459
	Avg	0.378	0.402	0.392	0.410	0.400	0.415	0.396	0.412	0.388	0.408	0.390	0.408	0.389	0.408	0.390	0.408	0.395	0.412	0.393	0.409	0.394	0.412
Exchange	96	0.088	0.210	0.089	0.211	0.086	0.207	0.086	0.206	0.090	0.213	0.091	0.214	0.108	0.233	0.104	0.230	0.096	0.220	0.091	0.213	0.093	0.216
	192	0.178	0.303	0.172	0.298	0.188	0.310	0.181	0.303	0.182	0.305	0.182	0.306	0.181	0.304	0.179	0.302	0.181	0.304	0.182	0.306	0.182	0.305
	336	0.326	0.414	0.340	0.422	0.349	0.426	0.408	0.460	0.336	0.420	0.331	0.416	0.331	0.416	0.332	0.417	0.359	0.434	0.330	0.416	0.332	0.417
	720	0.866	0.706	0.888	0.706	0.870	0.709	1.231	0.819	0.878	0.714	0.922	0.722	0.876	0.710	1.013	0.764	0.851	0.699	0.944	0.734	0.960	0.741
	Avg	0.365	0.408	0.372	0.409	0.373	0.413	0.477	0.447	0.372	0.413	0.382	0.414	0.374	0.416	0.407	0.428	0.372	0.414	0.387	0.417	0.392	0.420
Weather	96	0.171	0.211	0.170	0.208	0.174	0.212	0.178	0.223	0.173	0.213	0.171	0.211	0.172	0.213	0.178	0.219	0.184	0.224	0.172	0.211	0.178	0.218
	192	0.223	0.256	0.221	0.254	0.225	0.258	0.221	0.260	0.224	0.257	0.222	0.256	0.224	0.258	0.226	0.259	0.231	0.263	0.222	0.256	0.223	0.257
	336	0.280	0.299	0.278	0.297	0.283	0.300	0.276	0.301	0.284	0.301	0.280	0.298	0.281	0.299	0.285	0.302	0.287	0.306	0.280	0.298	0.279	0.297
	720	0.356	0.349	0.356	0.350	0.358	0.350	0.351	0.347	0.358	0.352	0.359	0.352	0.357	0.350	0.358	0.351	0.363	0.355	0.356	0.350	0.356	0.350
	Avg	0.258	0.279	0.256	0.277	0.260	0.280	0.256	0.283	0.260	0.281	0.258	0.279	0.258	0.280	0.262	0.283	0.266	0.287	0.258	0.279	0.259	0.280
Traffic	96	0.410	0.279	0.423	0.295	0.426	0.292	0.429	0.298	0.423	0.289	0.420	0.289	0.426	0.291	0.440	0.305	0.451	0.316	0.419	0.285	0.432	0.295
	192	0.432	0.287	0.443	0.298	0.443	0.296	0.449	0.302	0.445	0.297	0.444	0.298	0.446	0.298	0.458	0.310	0.466	0.318	0.441	0.293	0.452	0.303
	336	0.449	0.297	0.460	0.307	0.460	0.306	0.464	0.310	0.464	0.308	0.462	0.308	0.465	0.309	0.476	0.319	0.483	0.327	0.459	0.303	0.472	0.314
	720	0.485	0.320	0.494	0.327	0.496	0.328	0.501	0.331	0.500	0.330	0.496	0.329	0.502	0.331	0.512	0.340	0.520	0.348	0.495	0.325	0.510	0.339
	Avg	0.444	0.296	0.455	0.307	0.456	0.306	0.461	0.310	0.458	0.306	0.456	0.306	0.460	0.307	0.472	0.318	0.480	0.327	0.454	0.302	0.466	0.313
Solar-energy	96	0.206	0.237	0.204	0.243	0.206	0.242	0.217	0.259	0.207	0.245	0.212	0.248	0.207	0.244	0.214	0.257	0.235	0.287	0.208	0.243	0.208	0.246
	192	0.238	0.263	0.240	0.270	0.242	0.274	0.254	0.284	0.238	0.272	0.240	0.271	0.239	0.270	0.246	0.280	0.263	0.307	0.239	0.267	0.240	0.270
	336	0.254	0.278	0.255	0.283	0.264	0.294	0.274	0.300	0.258	0.288	0.258	0.288	0.257	0.285	0.270	0.296	0.286	0.319	0.255	0.282	0.258	0.286
	720	0.257	0.282	0.258	0.287	0.263	0.294	0.278	0.302	0.260	0.291	0.260	0.290	0.262	0.290	0.274	0.300	0.286	0.320	0.258	0.286	0.261	0.289
	Avg	0.239	0.265	0.239	0.271	0.244	0.276	0.256	0.286	0.241	0.274	0.242	0.274	0.241	0.272	0.251	0.283	0.268	0.308	0.240	0.270	0.242	0.273
1st Count		26	27	9	10	2	1	6	3	1	0	0	0	0	0	0	2	1	2	0	0	2	3

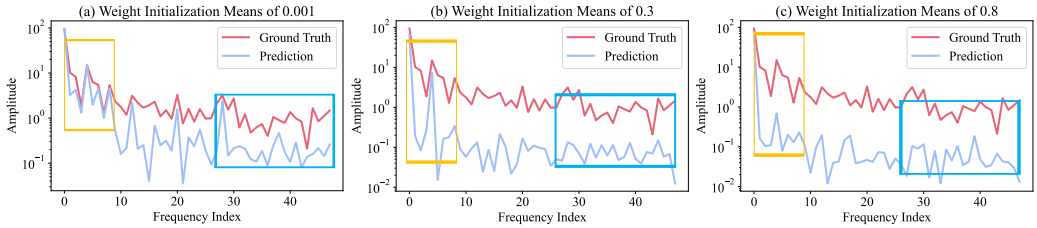

Figure 3: Validation visualization for Theorems 2 and 3. From left to right are frequency-domain comparison plots of predicted and ground-truth values for weight initialization means of 0.001, 0.3, and 0.8, respectively.

Table 7: Full results of generality of Frequency-LIF. Original denotes model performance with the original activation function. Frequency-LIF represents the performance when applied to the aforementioned baseline models without altering the original architecture or execution environment.

Dataset		Transformer				TimeXer				MoFo				DeepBooTS				ChannelTokenFormer			
		Original		Frequency-LIF		Original		Frequency-LIF		Original		Frequency-LIF		Original		Frequency-LIF		Original		Frequency-LIF	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.980	0.806	0.804	0.687	0.394	0.412	0.387	0.402	0.380	0.407	0.373	0.402	0.388	0.406	0.387	0.406	0.378	0.402	0.369	0.393
	192	0.928	0.778	0.858	0.755	0.446	0.443	0.431	0.430	0.426	0.430	0.419	0.430	0.455	0.447	0.444	0.443	0.431	0.436	0.417	0.423
	336	1.444	0.987	1.029	0.836	0.474	0.450	0.472	0.450	0.477	0.452	0.452	0.444	0.495	0.469	0.484	0.463	0.480	0.462	0.461	0.443
	720	1.153	0.871	1.086	0.847	0.473	0.476	0.480	0.469	0.486	0.472	0.491	0.472	0.505	0.498	0.488	0.483	0.499	0.493	0.468	0.463
	Avg	1.126	0.860	0.944	0.781	0.447	0.445	0.442	0.438	0.442	0.440	0.434	0.437	0.461	0.455	0.451	0.449	0.447	0.448	0.429	0.431
ETTh2	96	2.360	1.269	2.056	1.182	0.297	0.344	0.286	0.340	0.308	0.354	0.296	0.346	0.340	0.363	0.289	0.339	0.302	0.346	0.286	0.337
	192	5.264	1.896	5.098	1.818	0.380	0.400	0.368	0.393	0.370	0.394	0.365	0.393	0.392	0.402	0.370	0.393	0.383	0.400	0.368	0.388
	336	4.316	1.728	4.012	1.720	0.409	0.423	0.403	0.423	0.444	0.445	0.420	0.436	0.429	0.433	0.459	0.452	0.429	0.430	0.430	0.432
	720	2.970	1.464	3.209	1.532	0.437	0.448	0.426	0.445	0.458	0.468	0.424	0.446	0.472	0.461	0.483	0.468	0.437	0.450	0.418	0.437
	Avg	3.728	1.589	3.594	1.563	0.381	0.404	0.371	0.400	0.395	0.415	0.376	0.405	0.408	0.415	0.400	0.413	0.388	0.406	0.375	0.398
ETTm1	96	0.579	0.540	0.598	0.553	0.334	0.368	0.334	0.367	0.317	0.361	0.304	0.353	0.332	0.367	0.331	0.366	0.336	0.367	0.323	0.360
	192	0.751	0.651	0.649	0.591	0.375	0.393	0.371	0.392	0.351	0.380	0.354	0.380	0.391	0.399	0.382	0.395	0.361	0.383	0.362	0.384
	336	0.983	0.759	0.913	0.731	0.405	0.414	0.403	0.414	0.374	0.399	0.368	0.398	0.468	0.441	0.422	0.420	0.392	0.406	0.389	0.403
	720	1.126	0.828	0.916	0.739	0.462	0.449	0.461	0.448	0.434	0.441	0.439	0.445	0.497	0.465	0.491	0.464	0.454	0.443	0.448	0.441
	Avg	0.860	0.694	0.769	0.654	0.394	0.406	0.392	0.405	0.369	0.395	0.366	0.394	0.422	0.418	0.406	0.411	0.386	0.400	0.380	0.397
ETTm2	96	0.394	0.462	0.324	0.413	0.176	0.258	0.176	0.260	0.173	0.259	0.171	0.255	0.180	0.261	0.178	0.260	0.177	0.259	0.177	0.260
	192	1.372	0.863	0.913	0.705	0.250	0.308	0.240	0.303	0.247	0.306	0.240	0.304	0.246	0.306	0.244	0.305	0.244	0.303	0.238	0.299
	336	1.353	0.889	1.218	0.849	0.306	0.435	0.304	0.343	0.306	0.344	0.308	0.347	0.308	0.345	0.306	0.344	0.303	0.342	0.299	0.340
	720	3.143	1.360	3.179	1.385	0.417	0.408	0.411	0.408	0.398	0.401	0.399	0.402	0.405	0.400	0.402	0.399	0.395	0.396	0.395	0.398
	Avg	1.566	0.894	1.408	0.838	0.287	0.330	0.283	0.328	0.281	0.327	0.280	0.327	0.285	0.328	0.282	0.327	0.280	0.325	0.277	0.324

Table 8: Full results of ablation study on four datasets.

Methods		Frequency-LIF		w/o TFDAM		w/o SFDAM		w/o TFDAM_SFDAM	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.339**	**0.372**	0.342	0.375	0.340	0.373	0.344	0.377
	192	**0.379**	**0.394**	0.390	0.401	0.380	0.395	0.392	0.403
	336	**0.416**	**0.418**	0.436	0.430	0.426	0.423	0.440	0.431
	720	**0.495**	**0.462**	0.500	0.465	0.502	0.467	0.501	0.466
	Avg	**0.407**	**0.412**	0.417	0.418	0.412	0.415	0.419	0.419
ETTm2	96	**0.181**	**0.262**	0.182	0.263	**0.181**	0.263	0.182	0.263
	192	**0.252**	**0.310**	0.260	0.315	0.255	0.313	0.258	0.314
	336	**0.311**	**0.347**	0.314	0.349	**0.310**	0.348	0.326	0.359
	720	**0.408**	**0.404**	0.413	0.406	0.412	0.407	0.415	0.407
	Avg	**0.288**	**0.331**	0.292	0.333	0.290	0.333	0.295	0.336
ETTh1	96	**0.387**	**0.408**	0.396	0.411	0.388	**0.408**	0.396	0.412
	192	**0.439**	**0.438**	0.447	0.440	0.440	**0.438**	0.447	0.440
	336	**0.484**	**0.456**	0.489	0.462	0.487	0.464	0.490	0.462
	720	**0.497**	**0.482**	0.501	0.488	0.505	0.486	0.502	0.488
	Avg	**0.452**	**0.446**	0.458	0.450	0.455	0.449	0.459	0.450
ETTh2	96	**0.296**	**0.345**	0.306	0.351	0.299	0.346	0.305	0.351
	192	**0.373**	**0.394**	0.383	0.401	0.377	0.397	0.384	0.402
	336	**0.417**	**0.427**	0.434	0.436	0.424	0.429	0.434	0.436
	720	**0.424**	**0.442**	0.442	0.452	0.439	0.451	0.452	0.457
	Avg	**0.378**	**0.402**	0.391	0.410	0.385	0.406	0.394	0.412

When the initialization weight mean is extremely small (0.001), as shown in Figure 3(a), the predicted amplitudes of low-frequency components match the ground-truth amplitudes closely, whereas the predicted amplitudes of high-frequency components are significantly lower. This indicates that the model prioritizes convergence on low-frequency components, which remain dominant. This result validates Theorem 2: at sufficiently small weights, the relative error of low-frequency components remains extremely small, and low-frequency components maintain convergence even when high-frequency components begin to be learned.

When the mean increases to 0.3, as shown in Figure 3(b), low frequencies still possess an advantage, but their leading position weakens as some high-frequency components begin to obtain similar priority. When the mean further increases to 0.8, as shown in Figure 3(c), the predicted amplitudes of low and high frequencies are almost identical, which indicates similar priorities. Consequently, low frequencies are significantly interfered with by high frequencies before fully converging. This phenomenon conforms to Theorem 3: when $A(k_2)$ is large, low-frequency components are significantly affected by high-frequency interference even before they converge.

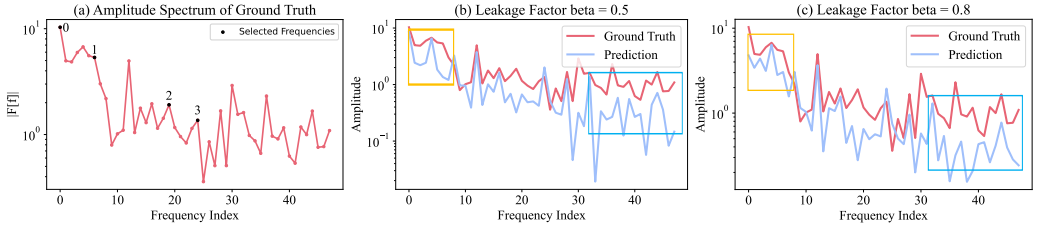

Figure 4: Validation visualization of the temporal frequency kernel dominance in Theorem 4. Subfigures (b) and (c) present frequency-domain comparison plots between predicted and ground-truth values for $\beta = 0.5$ and 0.8, respectively.

Validation Experiment for the Dominance of the Temporal Frequency Kernel in Theorem 4 To validate the dominance of the temporal frequency kernel in Theorem 4, this paper retains gradient updates in the temporal frequency domain of SNNs while truncating those in the spatial frequency domain. Without truncating the spatial frequency domain, it would be impossible to distinguish whether the effects of variable changes originate from the spatial frequency domain or the temporal frequency domain. This is consistent with the descriptions in Sections A.4.3 and A.4.3. By incrementally increasing the β value (0.5 and 0.8), this paper investigates the learning behavior of various temporal frequency components. The results are shown in Figure 4(b) and (c).

When $\beta = 0.5$ (see Figure 4(b)), the fitting degrees of low-frequency and high-frequency components are similar, and the error distribution is relatively uniform. When β increases to 0.8 (see Figure 4(c)), the error of low-frequency components decreases significantly, the fitting effect improves markedly, and the error of high-frequency components begins to increase. This phenomenon is consistent with Theorem 4: when $\beta \to 1$, the temporal frequency kernel dominates, and the network prioritizes temporal low-frequency information, leading to high low-frequency fitting and a rise in high-frequency error.

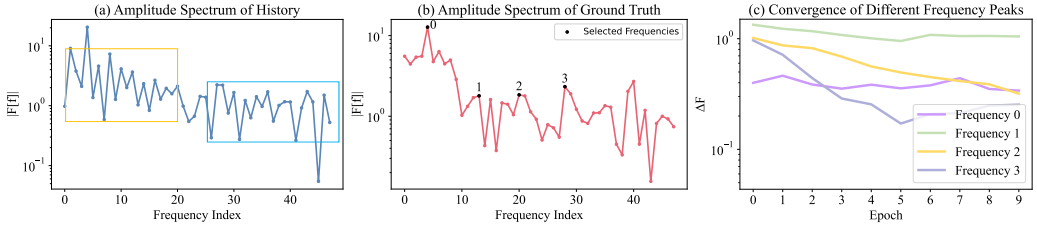

Figure 5: Validation of input spectrum dominance in Theorem 4. From left to right, the figure illustrates the amplitude spectra of historical and ground-truth values and the frequency learning orders

Validation Experiment for the Dominance of the Input Signal Spectrum in Theorem 4 To validate the case where the input signal spectrum dominates in Theorem 4, this paper selects an input sequence where the low-frequency energy is not higher than that of the high-frequency components (see Figure 5(a)). The frequency peak distribution of the target sequence is shown in Figure 5(b), with significant components existing across the range from low to high frequencies.

In this setting, the learning order of the model for various frequency components is shown in Figure 5(c). It is evident that the learning priority of low-frequency components is not significantly higher than that of high-frequency components, and some high-frequency components even converge prior to low-frequency ones. This validates Theorem 4: when the input signal spectrum concentrates in high-frequency regions, the input spectrum dominates, and the network no longer prioritizes temporal low-frequency information but instead adjusts the learning order according to the spectral distribution of the input signal.

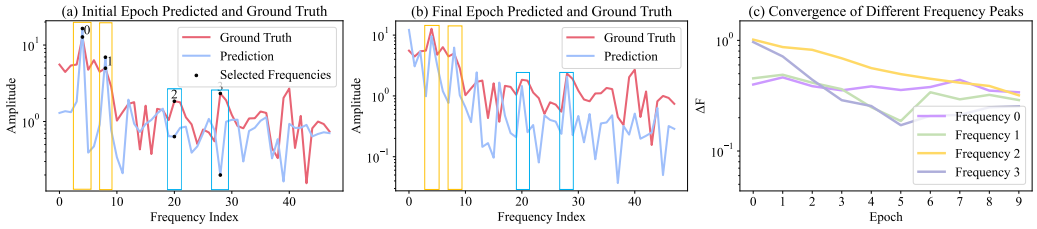

Figure 6: Validation visualization of the dominance of the spatio-temporal error spectrum amplitude integrated over spatial frequency in Theorem 4. Left to right: frequency-domain comparison plots between predicted and ground-truth values for the initial and final epochs, and the final-epoch frequency learning orders.

Validation Experiment for the Dominance of the Integral Amplitude of the Error Spatio-Temporal Spectrum in Theorem 4 To validate the case where the integral amplitude of the error spatio-temporal spectrum over spatial frequency dominates in Theorem 4, four frequency components are selected at the early stage of training (see Figure 6(a)). Among them, Frequency 0 and 1 are low-frequency components with small initial errors, while Frequency 2 and 3 are high-frequency components with large initial errors.

Experimental results are shown in Figure 6(c), where the convergence speed of high-frequency components is significantly faster than that of low-frequency components. Specifically, Frequency 3, which has the largest initial error, converges first, while low-frequency learning lags significantly. As illustrated in Figure 6(b), the high-frequency components of the predicted and ground-truth values exhibit a more pronounced decrease in the final epoch than the low-frequency components relative to the initial epoch. This validates Theorem 4: when temporal high-frequency error is much larger than low-frequency error, the error term dominates and changes the learning priority, prompting the network to prioritize temporal high-frequency information with larger errors.

Visualization Experiment of Correlation Coefficients To validate the assumption of the lower bound of the correlation coefficient $\rho > 0$ in the theorem, we conduct experiments on four datasets (ETTh1, ETTh2, ETTm1, ETTm2) with iTransformer as the backbone architecture and LIF as the activation function, where the prediction length is uniformly set to 96. It calculates the distribution of correlation coefficients between spatial frequency components and targets in each training epoch (see Figure 7(a)). The results show that the correlation coefficients for each dataset remain positive and concentrated, which indicates a stable positive correlation between the spatio-temporal spectrum and the error.

Furthermore, this paper records the trend of the minimum correlation coefficient in each training epoch under different prediction lengths and calculates its mean and variance (see Figure 7(b)). Even in the extreme case of the minimum correlation coefficient, its value remains positive and stabilizes as the number of epochs increases. This validates the core assumption of the theorem: there exists a positive constant $\rho > 0$ such that for all frequency components k, the inequality $\frac{|\langle TX, E(k,\cdot)\rangle|}{\|TX\| \cdot \|E(k,\cdot)\|} \geq \rho$ holds.

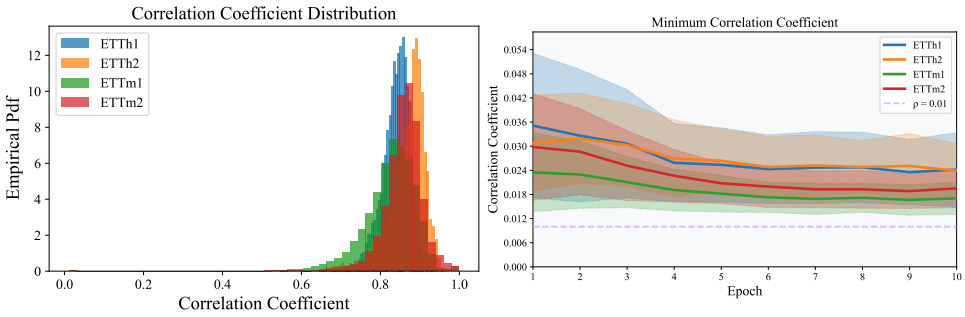

Figure 7: Visualization of correlation coefficients across different datasets. From left to right: distribution plots of correlation coefficients and the mean and standard deviation of minimum correlation coefficients versus rounds.

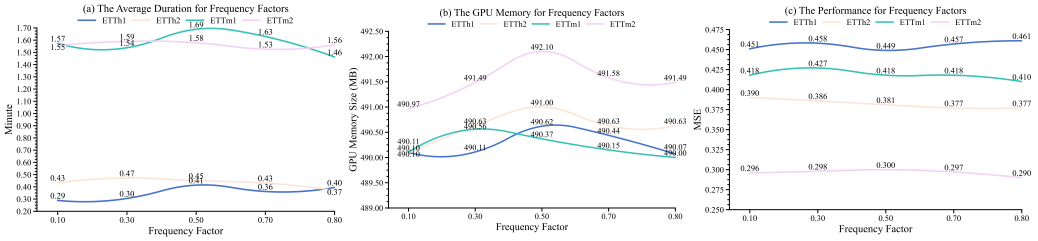

Figure 8: The impact of variations in hyperparameters—frequency factor η—on three metrics: training time, GPU memory, and performance.

A.4.4 Visualization Experiments

Hyperparameter Sensitivity Analysis We evaluate the hyperparameter sensitivity of Frequency-LIF on four benchmark datasets (ETTh1, ETTh2, ETTm1, ETTm2): the spatial frequency factor η, initial leakage factor β_0, time decay factor t_{decay}, high-frequency cutoff k_h, and tolerance threshold ξ.

1) Frequency Factor. η controls the weight allocation between seasonal and trend components in the decomposition loss, determining the transition of the network from low- to high-frequency learning. As shown in Figures 8(a)-8(b), training time and GPU memory remain stable across η, indicating the insensitivity of SFDAM to this factor. As shown in Figure 8(c), small η causes insufficient high-frequency learning and high error; $\eta \approx 0.8$ minimizes MSE; excessive η overemphasizes high-frequency details, slightly increasing MSE. We recommend setting η to an intermediate value.

2) Initial Leakage Factor and Time Decay Factor. β_0 determines the initial memory length for low-frequency trends. As shown in Figure 9(a), small β_0 limits low-frequency representation; $\beta_0 \approx 0.7$ achieves peak performance; larger β_0 slightly degrades performance. As shown in Figure 9(b), t_{decay} controls the exponential decay rate of β: large t_{decay} shifts to high-frequency learning too early; small t_{decay} over-biases toward low frequencies. An appropriate t_{decay} balances both and requires dataset-specific tuning.

3) High-Frequency Cutoff and Tolerance Threshold. As shown in Figure 9(c)- 9(d), k_h adjusts the enhancement amplitude of high-frequency learning after the critical point. ξ determines how many consecutive times the gradient spectrum ratio must satisfy the switching condition to trigger high-frequency mode. Both are insensitive: prediction performance degrades only slightly outside their optimal ranges, and the model remains stable, reducing tuning difficulty.

Visualization Analysis of Prediction Results To qualitatively evaluate the effectiveness of Frequency-LIF, this study selects four baseline models on the ETTh1 dataset (prediction horizons $H = 96$) and compares the visualization of their original versions with those integrated with

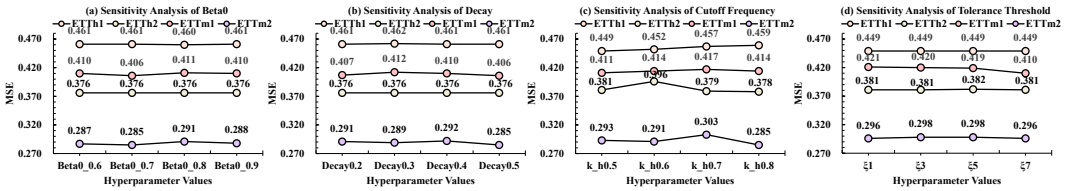

Figure 9: Hyperparameter sensitivity analysis on the four datasets. Subplots show MSE variations across key hyperparameters (β_0, t_{decay},k_h,ξ) with other settings held constant.

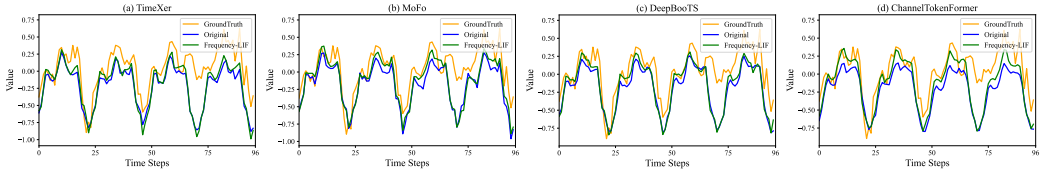

Figure 10: Visualization of prediction results. From left to right, comparisons of prediction results for four models in their original versions and those integrated with Frequency-LIF are presented.

Frequency-LIF. As shown in Figure 10, each subplot contains three curves: Ground Truth, the original model prediction (Origin), and the Frequency-LIF prediction.

Visualization results show that, compared to the original models, Frequency-LIF prediction trajectories match the ground truth more closely and demonstrate more precise trend-capturing capabilities, especially in highly volatile regions. For instance, at the local extrema of ChannelTokenFormer in Figure 10(d), the MSE of Frequency-LIF is significantly lower than that of the original model, which demonstrates a more sensitive response to high-frequency transient events.

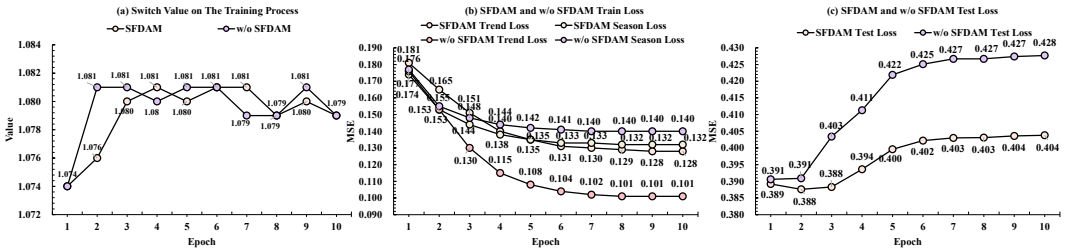

Figure 11: Visualization of the SFDAM training process. The subfigures from left to right depict the switch value over training epochs, the trend and seasonal loss components, and the corresponding test MSE curve.

Visualization Analysis of SFDAM To validate the effectiveness of SFDAM, this paper conducts a visualization analysis of the training dynamics of SFDAM on the ETTh1 dataset with a prediction length of 96. The experiment records the variation of the SFDAM switch value throughout the training process, the evolution of trend and seasonal losses, and the test MSE curve across training epochs, aiming to reveal the dynamic adjustment behavior of SFDAM in the spatial frequency domain and its impact on model performance.

As shown in Figure 11, the three subfigures from left to right depict the switch value during training, the trend/seasonal loss curves, and the test MSE curves, respectively. As illustrated in the left subfigure, after the training reaches the critical point, the switch value consistently exceeds the preset threshold, triggering the high-frequency learning mode of SFDAM and shifting the model toward high-frequency detail learning. As shown in the middle subfigure, prior to the critical point, the trend loss decreases steadily and dominates, while the seasonal loss remains relatively high. After the activation of SFDAM, the seasonal loss decreases significantly, while the trend loss exhibits a slight increase, demonstrating that the model successfully shifts its learning focus from

low-frequency trends to high-frequency seasonal details. As shown in the right subfigure, compared with the version without SFDAM (w/o SFDAM), the model with SFDAM achieves a lower test MSE convergence value in the later training stages, verifying that SFDAM effectively improves prediction accuracy by dynamically regulating the learning priority of spatial frequency components.