

## A APPENDIX

### A.1 EXPERIMENT SETTINGS FOR MAIN RESULTS

In this section, we outline the experimental setup used to evaluate the performance of the proposed TD-LIF model. We conducted experiments on several benchmark time series datasets, including Metr-LA (Li et al., 2017), which records average traffic speed on highways in Los Angeles County; Pems-Bay (Li et al., 2017), capturing traffic speed data in the Bay Area; Electricity (Lai et al., 2018), which tracks hourly electricity consumption in kWh; and Solar (Lai et al., 2018), detailing solar power production. Preprocessing steps were applied to ensure consistency across datasets, standardizing input dimensions, sampling rates, and data normalization.

The TS-LIF model framework was implemented in line with the approach in Lv et al. (2024), incorporating CNN-based TCNs (Bai et al., 2018), RNN-based GRUs (Cho, 2014), and Transformer-based models such as Autoformer (Wu et al., 2021) and iTransformer (Liu et al., 2024). As for the SNN-based structure, we introduce the TCN, RNN, GRUs, and Transformer models of the SNN format from Lv et al. (2024). Hyperparameters, including learning rate, timestep intervals, and feature-mixing weights, were optimized through cross-validation. We employ two statistical metrics: the Root Relative Squared Error (RSE) and the coefficient of determination ( $R^2$ ) followed by the Lv et al. (2024) settings. Detailed descriptions of the experimental settings and hyperparameter configurations are provided in the appendix.

### A.2 DATASET AND METRIC DETAILS

**Datasets.** The details of the datasets used in the main experiment are shown in Table 3. In the experimental partitioning of datasets Metr-la and Pems-bay, we adopted a train-validation-test ratio of 0.7, 0.2, and 0.1, respectively, while for datasets Solar and Electricity, we used ratios of 0.6, 0.2, and 0.2. The settings for history and prediction lengths in the experiments followed those in the paper by Lv et al. (2024), except that for the history length in datasets Metr-la and Pems-bay, we added a setting of 168 to further improve experimental performance.

Table 3: Properties of the datasets in experiments

DATASET	Dimension	Domain	Freq	Samples	Context Length	Pred Length
Metr-la	207	$\mathbb{R}^+$	30-min	34,272	{12, 168}	{6, 24, 48, 96}
Pems-bay	325	$\mathbb{R}^+$	30-min	52,116	{12, 168}	{6, 24, 48, 96}
Solar	137	$\mathbb{R}^+$	Hourly	52,560	168	{6, 24, 48, 96}
Electricity	321	$\mathbb{R}^+$	Hourly	26,304	168	{6, 24, 48, 96}

**Metrics.** To comprehensively evaluate our model’s performance, we employ two statistical metrics: the Root Relative Squared Error (RSE) and the coefficient of determination ( $R^2$ ). The RSE measures the relative discrepancy between the predicted and actual values, while the  $R^2$  indicates the proportion of variance in the dependent variable that is predictable from the independent variables. These metrics are calculated as follows:

$$\text{RSE} = \sqrt{\frac{\sum_{m=1}^M \|\mathbf{Y}^m - \hat{\mathbf{Y}}^m\|^2}{\sum_{m=1}^M \|\mathbf{Y}^m - \bar{\mathbf{Y}}\|^2}}, \quad (22)$$

$$R^2 = \frac{1}{MCL} \sum_{m=1}^M \sum_{c=1}^C \sum_{l=1}^L \left[ 1 - \frac{(Y_{c,l}^m - \hat{Y}_{c,l}^m)^2}{(Y_{c,l}^m - \bar{Y}_{c,l})^2} \right],$$

where  $M$  denotes the number of samples in the test set,  $C$  represents the number of channels or variables, and  $L$  is the prediction horizon. The true values for the  $m$ -th sample are denoted by  $\mathbf{Y}^m$ , and their average over all samples is  $\bar{\mathbf{Y}}$ . Specifically,  $Y_{c,l}^m$  represents the  $l$ -th future value of the  $c$ -th variable for the  $m$ -th sample, with its mean across all samples given by  $\bar{Y}_{c,l}$ . The predicted values corresponding to these true values are denoted by  $\hat{\mathbf{Y}}^m$  and  $\hat{Y}_{c,l}^m$ , respectively.

### A.3 IMPLEMENTATION DETAILS

In this section, we summarize the detailed experiment setup of our TS-LIF. Table 4 and 5 show the hyperparameters of our overall structure in three types of backbones (TCN, GRU, Transformer). As for the timesteps in the SNN structures, we align them with the history length in each setting. The threshold of the TS-LIF is set to 1.0.

Table 4: Hyperparameters of different backbones (TCN, GRU, Transformer) used for each dataset

Datasets	TCN Layers	TCN Kernels	GRU Layers	Transformer Layers	Attention Heads	Attention Dim
Metr-la	3	3	1	2	8	256
Pems-bay	3	16	1	2	8	512
Solar	3	16	1	2	8	512
Electricity	3	3	1	2	8	256

Table 5: Training details of different backbones (TCN, GRU, Transformer) used for each dataset

Datasets	TCN Hidden	TCN Dilation	GRU Hidden	Transformer d_ff	Learning Rate	Batch Size
Metr-la	64	2	128	1024	{.0001, .0005}	{32, 64}
Pems-bay	64	2	128	2048	{.0001, .0005}	{32, 64}
Solar	64	2	128	2048	.0001	64
Electricity	64	2	128	1024	.0001	64

### A.4 EXPERIMENT SETTINGS FOR TEMPORAL ANALYSIS

The injected current,  $I(t)$ , consists of two sinusoidal components: a low-frequency component,  $I_{\text{low\_freq}}$ , with an amplitude of 3 and a frequency of 0.5 Hz, and a high-frequency component,  $I_{\text{high\_freq}}$ , with an amplitude of 5 and a frequency of 4 Hz. This combination represents a complex input environment with both slow and rapid variations, simulating mixed-frequency stimuli.

For the TCLIF model, we adopted parameters  $\alpha_1 = \alpha_2 = 1$ ,  $\beta_1 = -0.5$ , and  $\beta_2 = 0.5$ , as suggested in Zhang et al. (2024a). In contrast, the TS-LIF model was set with  $\alpha_1 = 0.95$ ,  $\alpha_2 = 0.05$ ,  $\beta_1 = 0$ , and  $\beta_2 = -0.9$ , which corresponds to the parameter settings used in the frequency response analysis in the previous subsection. These values were selected to facilitate low-pass filtering in the dendritic compartment ( $v_d[t]$ ) and high-pass filtering in the somatic compartment ( $v_s[t]$ ).

### A.5 THEORETICAL ENERGY CONSUMPTION CALCULATION

The theoretical energy consumption for each layer during inference is determined based on the operations performed by spiking neural networks (SNNs) and artificial neural networks (ANNs) (Yao et al., 2023).

For SNNs, the energy required by layer  $l$  is calculated as:

$$\text{Energy}(l) = E_{AC} \times \text{SOPs}(l),$$

where  $\text{SOPs}(l)$  is the number of spike-based accumulate (AC) operations, and  $E_{AC}$  represents the energy per AC operation.

For ANNs, the energy consumption for layer  $b$  is:

$$\text{Energy}(b) = E_{MAC} \times \text{FLOPs}(b),$$

where  $\text{FLOPs}(b)$  refers to the number of floating-point multiply-and-accumulate (MAC) operations, and  $E_{MAC}$  is the energy per MAC operation. The constants are set as  $E_{MAC} = 4.6 \text{ pJ}$  and  $E_{AC} = 0.9 \text{ pJ}$ , assuming operations are performed on 45nm hardware.

For SNNs, the number of synaptic operations in layer  $l$  is further estimated as:

$$\text{SOPs}(l) = T \times \gamma \times \text{FLOPs}(l),$$

where  $T$  is the number of timesteps required in the simulation, and  $\gamma$  is the firing rate of the input spike train for layer  $l$ .

Table 6: Energy consumption per sample of the Electricity dataset during inference. "OPs" includes SOPs for SNNs and FLOPs for ANNs. "SOPs" refers to synaptic operations in SNNs, and "FLOPs" denotes floating-point operations in ANNs.

Model	Param(M)	OPs (G)	Energy (mJ)	Energy Reduction	Train/Infer Time (s)	R <sup>2</sup>
TCN	0.460	0.14	0.64	-	21.34/11.47	.973
Spike-TCN	0.461	0.15	0.23	63.60% ↓	306.91/27.85	.963
<b>TS-TCN</b>	0.465	0.19	0.25	60.93% ↓	308.26/28.14	.971
GRU	1.288	1.32	6.07	-	37.73/7.35	.972
Spike-GRU	1.289	1.63	1.51	75.05% ↓	235.46/10.05	.964
<b>TS-GRU</b>	1.291	1.67	1.58	73.80% ↓	246.23/9.78	<b>.981</b>
iTransformer	1.634	2.05	9.47	-	7.24/6.38	.977
iSpikformer	1.634	3.55	3.19	66.30% ↓	49.84/8.69	.974
<b>TS-former</b>	1.640	3.59	3.22	65.99% ↓	50.36/8.72	<b>.985</b>

Table 7: Performance of our TS-former with SparseTSF and SAMformer of 3 prediction lengths (24, 48, 96) on the Metr-la and Electricity datasets. SparseTSF\*: replace the ReLU function of SparseTSF with our TS-LIF. **Bold** numbers represent the best outcomes.

Datasets	Metr-la						Electricity					
	24		48		96		24		48		96	
Lengths												
Metrics	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓
SparseTSF	0.576	0.681	0.427	0.792	0.253	0.916	<b>0.991</b>	<b>0.167</b>	<b>0.986</b>	<b>0.195</b>	<b>0.982</b>	<b>0.232</b>
SparseTSF*	0.580	0.692	0.426	0.801	0.247	0.924	0.990	0.177	0.985	0.201	<b>0.982</b>	0.234
SAMformer	0.549	0.739	0.401	0.863	0.219	0.965	0.983	0.218	0.980	0.239	0.978	0.257
<b>TS-former</b>	<b>0.620</b>	<b>0.655</b>	<b>0.445</b>	<b>0.763</b>	<b>0.283</b>	<b>0.874</b>	0.985	0.215	0.981	0.234	0.977	0.261

#### A.6 PERFORMANCE COMPARISON ON SOTA TIME SERIES FORECASTING METHODS

Table 7 compares the performance of our TS-former with SparseTSF (Lin et al., 2024), SparseTSF\* (where ReLU is replaced by TS-LIF), and SAMformer (Ilbert et al., 2024a) on the Metr-la and Electricity datasets for prediction lengths of 24, 48, and 96.

On the Metr-la dataset, TS-former achieves the best results across all metrics and prediction lengths, demonstrating its ability to effectively capture complex temporal dependencies. For example, at a prediction length of 24, our TS-former achieves an  $R^2$  of 0.620 and RSE of 0.655, outperforming both SparseTSF and SAMformer. On the Electricity dataset, SparseTSF achieves slightly better performance in some cases, such as an  $R^2$  of 0.991 and RSE of 0.167 at a prediction length of 24. However, TS-former remains competitive, delivering consistent and robust results across different prediction lengths. These results highlight the effectiveness of TS-LIF in SparseTSF\* and the overall robustness of TS-former in time series forecasting tasks.

#### A.7 COMPARISON OF TS-LIF WITH OTHER LIF NEURONS

Table 8 compares the performance of our TS-LIF with TC-LIF, LM-H, and CLIF in the GRU backbone on the Metr-la and Electricity datasets for prediction lengths of 6, 24, and 96. TS-LIF consistently outperforms the baseline methods across all metrics and prediction lengths. For example, on the Metr-la dataset, TS-LIF achieves the highest  $R^2$  of 0.848 and the lowest RSE of 0.412 at a prediction length of 6. Similarly, on the Electricity dataset, TS-LIF achieves an  $R^2$  of 0.991 and RSE of 0.216 at the same prediction length, demonstrating its robustness and effectiveness in modeling temporal dependencies. These results highlight the superiority of TS-LIF over existing LIF structures, making it a strong choice for time series forecasting tasks.

#### A.8 ROBUSTNESS ANALYSIS ON THE METR-LA DATASET

To further verify the robustness of the proposed TS-LIF model, we evaluated its performance on the Metr-la dataset under different ratios of missing values in the historical inputs, comparing it to the

Table 8: Performance of our TS-LIF with other LIF neurons (TC-LIF, LM-H, and CLIF) in the GRU backbone. **Bold** numbers represent the best outcomes.

Datasets	Metr-la						Electricity					
	6		24		96		6		24		96	
Lengths												
Metrics	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓
TC-LIF	0.828	0.453	0.594	0.673	0.259	0.956	0.978	0.263	0.976	0.257	0.941	0.503
LM-H	0.812	0.464	0.570	0.719	0.246	0.973	0.971	0.269	0.969	0.280	0.936	0.512
CLIF	0.837	0.429	0.606	0.667	0.271	0.930	0.973	0.259	0.972	0.276	0.954	0.376
<b>TS-LIF</b>	<b>0.848</b>	<b>0.412</b>	<b>0.618</b>	<b>0.651</b>	<b>0.329</b>	<b>0.853</b>	<b>0.991</b>	<b>0.216</b>	<b>0.981</b>	<b>0.240</b>	<b>0.976</b>	<b>0.271</b>

vanilla LIF-based models. The models were assessed under missing data ratios of 10%, 20%, 40%, 60%, and 80%. The results are presented in Table 9.

Table 9: Experimental performance of the TS-LIF model compared to the vanilla LIF on the Metr-la dataset, evaluated under different ratios of missing values in the historical inputs. Model\_\* indicates a backbone model with a prediction length of \*, and Transformer\_6 represents the Transformer architecture with a prediction length of 6.

Missing Ratio		10%		20%		40%		60%		80%	
Metric		R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓
Transformer_6	iSpikformer	0.815	0.479	0.813	0.486	0.809	0.488	0.804	0.492	0.802	0.496
	<b>TS-former</b>	<b>0.843</b>	<b>0.419</b>	<b>0.842</b>	<b>0.421</b>	<b>0.838</b>	<b>0.430</b>	<b>0.835</b>	<b>0.436</b>	<b>0.831</b>	<b>0.440</b>
	Promotion	3.43%	14.3%	3.56%	15.4%	3.58%	13.4%	3.86%	12.8%	3.61%	12.7%
Transformer_96	iSpikformer	0.270	0.915	0.254	0.926	0.230	0.935	0.204	0.948	0.194	0.959
	<b>TS-former</b>	<b>0.277</b>	<b>0.907</b>	<b>0.263</b>	<b>0.911</b>	<b>0.238</b>	<b>0.927</b>	<b>0.212</b>	<b>0.936</b>	<b>0.205</b>	<b>0.945</b>
	Promotion	2.59%	0.88%	3.54%	1.65%	3.47%	0.86%	3.92%	1.28%	5.67%	1.48%
GRU_6	Spike-GRU	0.830	0.429	0.819	0.440	0.771	0.497	0.746	0.522	0.743	0.530
	<b>TS-GRU</b>	<b>0.843</b>	<b>0.417</b>	<b>0.839</b>	<b>0.414</b>	<b>0.834</b>	<b>0.425</b>	<b>0.823</b>	<b>0.435</b>	<b>0.792</b>	<b>0.473</b>
	Promotion	1.50%	2.40%	2.50%	5.90%	8.10%	16.9%	10.3%	20.0%	6.50%	12.1%
GRU_96	Spike-GRU	0.243	0.924	0.240	0.919	0.213	0.932	0.191	0.944	0.171	0.970
	<b>TS-GRU</b>	<b>0.342</b>	<b>0.857</b>	<b>0.341</b>	<b>0.860</b>	<b>0.338</b>	<b>0.863</b>	<b>0.319</b>	<b>0.869</b>	<b>0.294</b>	<b>0.878</b>
	Promotion	40.7%	7.80%	42.0%	6.86%	58.7%	7.99%	67.0%	8.63%	71.9%	10.4%
TCN_6	Spike-TCN	0.774	0.509	0.765	0.521	0.757	0.549	0.742	0.570	0.731	0.596
	<b>TS-TCN</b>	<b>0.792</b>	<b>0.469</b>	<b>0.781</b>	<b>0.493</b>	<b>0.773</b>	<b>0.512</b>	<b>0.756</b>	<b>0.538</b>	<b>0.744</b>	<b>0.562</b>
	Promotion	2.33%	8.53%	2.12%	5.84%	2.11%	7.23%	1.89%	5.94%	1.78%	6.05%

#### A.9 STANDARD DEVIATION ANALYSIS

Table 10 shows the standard deviation of  $R^2$  and RSE metrics over three runs with different random seeds for TS-GRU and TS-former on the Metr-la and Electricity datasets, across prediction lengths of 6, 24, 48, and 96. Both models exhibit low standard deviations, demonstrating their stability and robustness. These results confirm the reliability of TS-GRU and TS-former under varying random seeds.

#### A.10 AVERAGE POWER SPECTRUM ANALYSIS

To gain a deeper understanding of how TS-LIF processes temporal features, we analyze the average power spectrum of dendritic and somatic voltages after the first encoder layer of TS-former. Figure 5 illustrates the power distribution of voltage signals from dendrites and soma across different frequency ranges.

The analysis reveals distinct frequency response characteristics between dendritic and somatic compartments. These different roles allow TS-LIF to encode diverse temporal features effectively, contributing to its superior performance on time series forecasting tasks.

Table 10: The standard deviation of 3 runs with different random seeds with our TS-former and TS-GRU on Metr-1a and Electricity datasets.

Datasets	Metr-1a								Electricity							
Lengths	6		24		48		96		6		24		48		96	
Metrics	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓	R <sup>2</sup> ↑	RSE↓
TS-GRU (ours)	.002	.005	.004	.006	.002	.011	.004	.009	.001	.002	.001	.005	.002	.004	.001	.008
TS-former (ours)	.001	.008	.002	.006	.005	.013	.002	.010	.001	.003	.001	.003	.001	.005	.001	.007

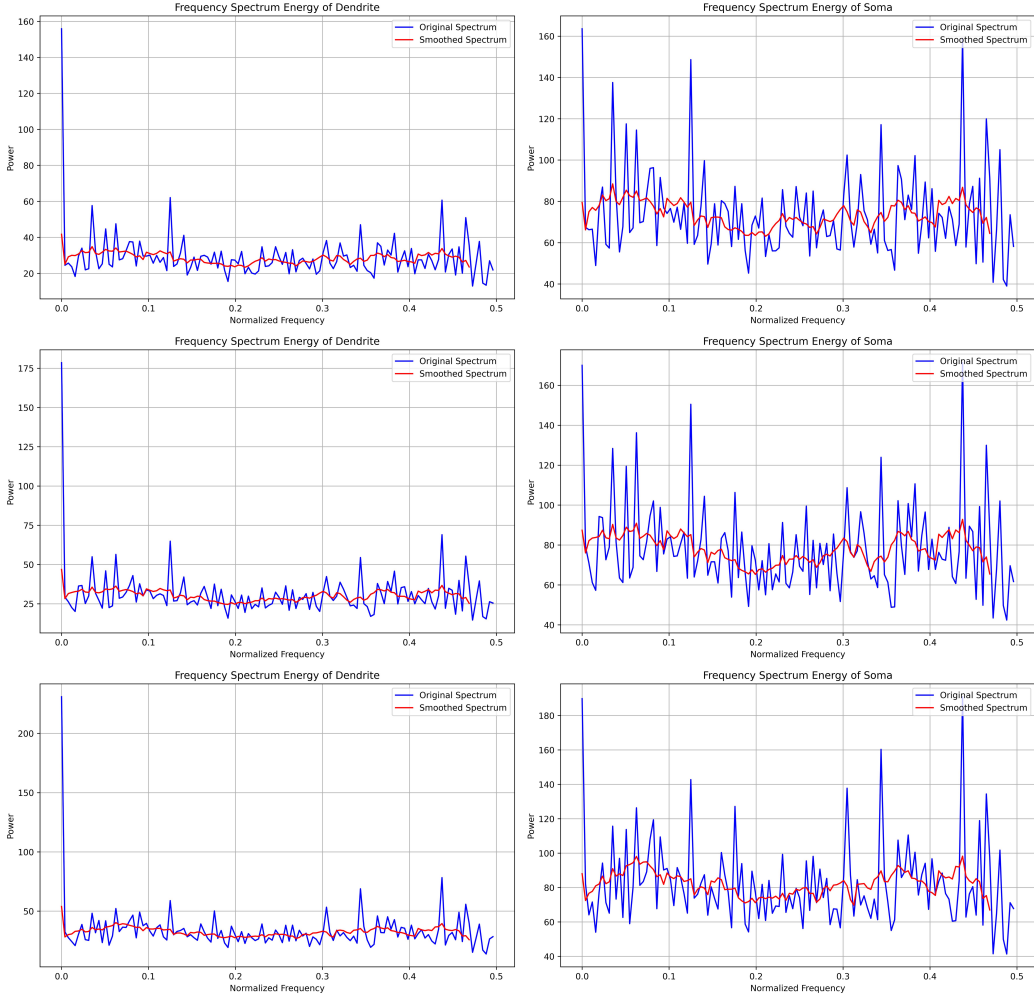


Figure 5: Average power spectrum analysis of dendritic and somatic voltages. The figure illustrates the power distribution of voltage signals from dendrites and soma across different frequency ranges, providing insights into neural signal processing mechanisms.

#### A.11 EVALUATING LONG-TERM DEPENDENCY CAPTURE WITH DELAYED SPIKING XOR PROBLEM

To further evaluate the ability of spiking neuron models to capture long-term dependencies, we conducted experiments using the delayed spiking XOR problem. This task tests the model’s capacity to retain and process information over extended periods. The task involves three stages: an initial spike input, a delay period with noisy spikes, and a second spike input. The model computes an XOR operation between the first and final inputs based on their firing rates.

Our experimental setup follows the parameters described in (Zheng et al., 2024). Specifically, we employed a two-layer MLP with only 20 hidden neurons, where the input feature size was set to 20. The results, measuring prediction accuracy under different delay timesteps and activation functions (ReLU, LIF, and TS-LIF), are summarized in Table 11.

Table 11: Prediction accuracy of the delayed spiking XOR problem under different delay timesteps and activation functions.

Delay timesteps	ReLU	LIF	TS-LIF
10	0.5	0.748	0.994
20	0.5	0.504	0.977
30	0.5	0.504	0.791
40	0.5	0.500	0.585
50	0.5	0.500	0.585

The results demonstrate that TS-LIF significantly outperforms both ReLU and standard LIF across all tested delay timesteps, particularly excelling at shorter and moderate delays. While the performance of LIF degrades as delay increases, TS-LIF retains higher accuracy, showcasing its enhanced capability for capturing long-term dependencies. This improvement can be attributed to the distinct processing mechanisms of dendritic and somatic compartments, which allow TS-LIF to maintain a more robust temporal memory compared to traditional spiking and non-spiking activation functions.

#### A.12 LIMITATIONS AND FUTURE WORK

**Limitations.** In multivariate time series forecasting, modeling the correlations between variables is crucial for improving prediction accuracy. Current SNN-based models for time series forecasting primarily focus on temporal modeling and lack explicit mechanisms for capturing inter-variable correlations. For instance, explicitly computing cross-variable correlations, as shown in works like Ilbert et al. (2024a) and Zhang & Yan (2023), can effectively model multivariate relationships. We intend to explore how SNN filtering mechanisms can efficiently model cross-variable relationships to further enhance predictive performance.

**Future Work.** Future research directions include: (1) designing an efficient and effective SNN mechanism for capturing cross-variable correlations in multivariate time series, and (2) developing more generalized SNN structures for comprehensive time series analysis tasks, such as anomaly detection, time series generation, and classification.