INVESTIGATING PATTERN NEURONS IN URBAN TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Urban time series forecasting is crucial for smart city development and is key to sustainable urban management. Although urban time series models (UTSMs) are effective in general forecasting, they often overlook low-frequency events, such as emergencies and holidays, leading to degraded performance in practical applications. In this paper, we first investigate how UTSMs handle these infrequent patterns from a neural perspective. Based on our findings, we propose **P**attern **N**euron guided **Train**ing (`PN-Train`), a novel training method that features (i) a *perturbation-based detector* to identify neurons responsible for low-frequency patterns in UTSMs, and (ii) a *fine-tuning mechanism* that enhances these neurons without compromising representation learning on high-frequency patterns. Empirical results demonstrate that `PN-Train` considerably improves forecasting accuracy for low-frequency events while maintaining high performance for high-frequency events.

1 INTRODUCTION

Recent advancements in urban time series models (UTSMs) have significantly improved forecasting accuracy, facilitating smart city applications such as optimizing metropolitan transit, managing pedestrian flow, and enhancing resource allocation for ride-hailing services (Yao et al., 2018; Wu et al., 2020; Ji et al., 2022). While deep learning models (Bai et al., 2020; Wu et al., 2019; Gao et al., 2023) have shown great promise in urban time series forecasting, existing models focus on capturing cross-variable and temporal dependencies to enhance overall accuracy. However, their performance degrades in many real-world scenarios, especially when forecasting low-frequency events such as extreme weather, emergencies, holidays and etc (Lee et al., 2022; Lee & Ko, 2024). Accurate forecasting of these events is critical for effective resource management, enabling ride-hailing companies to adjust fleet sizes and service frequencies, and allowing transit systems to modify schedules, thereby optimizing operations and reducing costs during periods of fluctuating demand (Geng et al., 2019; Park et al., 2020; Jiang et al., 2023).

Urban time series data exhibits distinct patterns for high- and low-frequency events (Lee et al., 2019; Wang et al., 2019). As the example illustrated in Figure 1, while patterns within each category remain consistent, significant differences exist between holiday, weekday, and weekend patterns. Specifically, weekdays and weekends represent high-frequency patterns, occurring regularly throughout the year, whereas holidays are low-frequency, spanning fewer than 15 days, or approximately 4% of the year. Deep learning models often struggle to predict low-frequency events, such as holidays in the example, largely due to their bias toward majority patterns and the scarcity of data for these rare occurrences. Several studies have attempted to improve holiday forecasting, e.g., using exponential-growth models (Wang et al., 2019) and support vector regression (Luo et al., 2019). More recently, deep learning models have in-

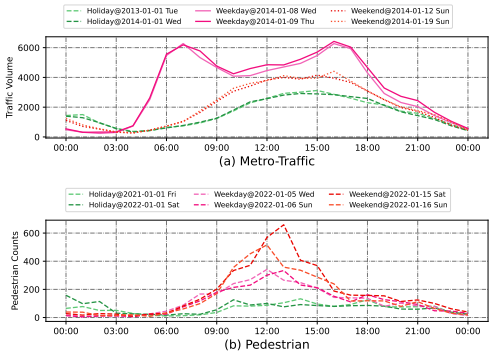


Figure 1: Examples of high-frequency patterns during weekdays and weekends, contrasted with low-frequency patterns on holidays.

troduced memory architectures for retrieving patterns from a pattern bank (Lee et al., 2022; Li et al., 2022) and dynamic positional embeddings to implicitly capture various patterns (Shao et al., 2022; Liu et al., 2023), leading to enhanced forecasting accuracy. However, despite improved forecasting accuracy, understanding the underlying mechanisms through which these models capture low-frequency patterns at the neuron level remains unexplored.

In this paper, inspired by the *knowledge neurons* in large language models (LLMs) (Dai et al., 2022; Zhao et al., 2024), we investigate two fundamental questions: (1) Do neurons associated with low-frequency patterns exist in UTSMs? (2) If so, how can we enhance the representation learning of these neurons to improve urban time series forecasting?

To answer these questions, we perform an in-depth analysis of UTSMs at the neuron level. First, we introduce a Pattern Neuron Detector (PND), which identifies *pattern neurons*, i.e., neurons strongly correlated with low-frequency patterns, using a perturbation-based approach. This method evaluates neuron importance by measuring the impact of perturbations on the model's output features. Next, we employ a Pattern Neuron Verifier (PNV) to quantify how these neurons impact forecasting performance by deactivating them, so as to confirm that neurons specifically tied to certain patterns indeed exist in UTSMs. Based on our findings, we propose **P**attern **N**euron Guided **Train**ing (`PN-Train`), a novel training method that detects these pattern neurons and fine-tunes them using a Pattern Neuron Optimizer (PNO) to improve forecasting for low-frequency patterns while maintaining performance for high-frequency patterns. We summarize our main contributions as follows:

- We conduct the first investigation into neurons associated with low-frequency patterns in urban time series models (UTSMs) and confirm their existence.
- We introduce `PN-Train`, a pattern neuron-guided training method for urban time series forecasting, which effectively detects these neurons using a perturbation-based detector.
- We propose a fine-tuning mechanism that enhances the representation learning of detected pattern neurons, significantly improving forecasting accuracy.
- Extensive experiments demonstrate that `PN-Train` significantly improves the forecasting accuracy of state-of-the-art methods across real-world datasets.

2 PRELIMINARIES

Urban Time Series Forecasting (UTSF) UTSF aims to forecast future time series data using sensor readings collected from urban environments. The objective is to predict H future values $\mathbf{x}_{\tau:\tau+H}$ at each time step τ, using a learnable model, UTSM, which leverages a look-back window of L past observations $\mathbf{x}_{\tau-L:\tau}$. Additionally, auxiliary features E, such as the time of day, day of the week, holiday indicators, and etc., are incorporated to enhance the forecasting process. Formally, the prediction task can be formulated as $\hat{\mathbf{x}}_{\tau:\tau+H} = \text{UTSM}(\mathbf{x}_{\tau-L:\tau}, E)$. To ensure accurate forecasting, the UTSM model is typically trained using the Mean Absolute Error (MAE) loss function, defined as $\mathcal{L} = \frac{1}{H}\sum_{h=1}^{H} \|\hat{\mathbf{x}}_{\tau+h} - \mathbf{x}_{\tau+h}\|_1$.

Pattern Neurons In a neural network, individual neurons contribute differently to the representation and memorization of various patterns. Let $I(h_i, p)$ denote the influence of neuron h_i on pattern p. A *pattern neuron* can then be defined as a neuron with a strong influence on a specific pattern, namely $I(h_i, p)$ is high for the particular pattern p.

3 PN-TRAIN

In this section, we introduce `PN-Train`, a training method designed to enhance urban time series forecasting for low-frequency patterns by identifying and fine-tuning the pattern neurons associated with these patterns in the UTSM. The overall architecture of `PN-Train` is illustrated in Figure 2.

3.1 PATTERN NEURON DETECTOR FOR URBAN TIME SERIES MODELS

Typically, Urban Time Series Models (UTSMs) are designed to capture patterns from historical data (Zhou et al., 2020; Liu et al., 2020; 2022). While they can effectively learn frequent patterns

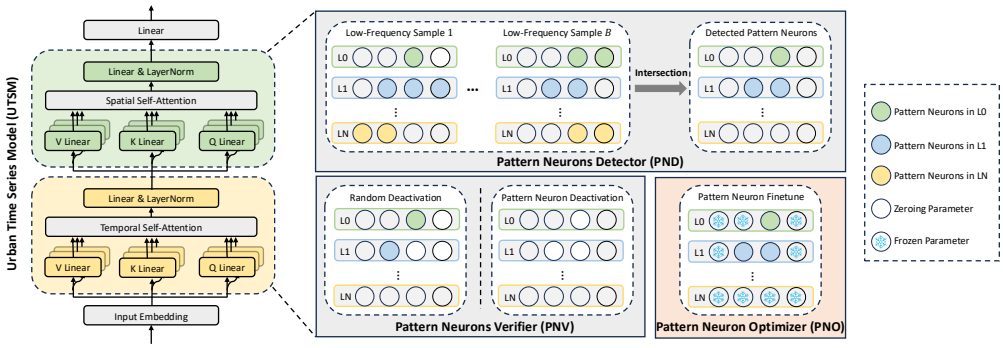


Figure 2: The architecture of `PN-Train`, which consists of four components: Urban Time Series Model (UTSM) captures time series patterns from historical data; Pattern Neuron Detector (PND) identifies neurons associated with specific patterns, such as low-frequency samples; Pattern Neuron Verifier (PNV) validates the detected neurons; and Pattern Neuron Optimizer (PNO) fine-tunes the UTSM at the neuron level. LX represents the X-th linear layer in the UTSM.

thanks to sufficient training data, the distribution of patterns is often imbalanced in practice (Luo et al., 2019; Lee & Ko, 2024). In particular, high-frequency events like weekdays and weekends are well-represented, making them easier to learn. In contrast, low-frequency events, such as holidays, have fewer training samples, which results in reduced forecasting performance (Krawczyk, 2016; Smyl et al., 2023). We hypothesize that some neurons in UTSMs are already tuned to capture low-frequency patterns based on past encounters with these events. To test this hypothesis, we introduce a Pattern Neuron Detector (PND) to identify neurons linked to low-frequency patterns.

Neurons in UTSMs UTSMs nowadays are effective at learning patterns from historical data, and a key component of these models is the linear layer, which is central to pattern learning and memorization (Geva et al., 2021; Dai et al., 2022). In this work, we focus on a transformer-based UTSM (Liu et al., 2023), which employs both linear layers and self-attention layers to model temporal correlations (patterns over time) and spatial correlations (relationships across urban locations). The detection of pattern neurons in the UTSM is carried out using our proposed Pattern Neuron Detector (PND), which can be applied to both linear layers and self-attention layers as introduced below.

Pattern Neuron Detector (PND) Inspired by *Knowledge Neurons* (Tang et al., 2024; Zhao et al., 2024) in large language models (LLMs), which identify neurons with high activation values, we define *pattern neurons* in Urban Time Series Models (UTSMs) whose contributions to forecasting targets are significant in the perturbation assessment. Specifically, the influence h_i^k of the k-th neuron at the i-th layer can be quantified by comparing the model outputs when the neuron is deactivated:

$$I(h_i^k|\mathbf{x}^p) = \left\| UTSM(\mathbf{x}^p, \boldsymbol{W}) - UTSM(\mathbf{x}^p, \boldsymbol{W} \backslash \boldsymbol{w}_i^k) \right\|_1, \tag{1}$$

where $\mathbf{x}^p$ represents an input with the pattern p, $\boldsymbol{W}$ and $\boldsymbol{w}_i^k$ denote the weights of the UTSM and the weights of the neurons respectively, and $UTSM(\mathbf{x}^p, \boldsymbol{W} \backslash \boldsymbol{w}_i^k)$ represents the model output with only neuron h_i^k deactivated.

As UTSMs contain a large number of neurons, deactivating each neuron individually is impractical. Previous study (Zhao et al., 2024) has shown that neurons linked to specific patterns often exhibit high feature activation values. This suggests that the neuron activation values can serve as strong indicators of their importance in capturing corresponding patterns. Therefore, we devise an attribute score Attr_p to quantify the influence of the k-th neuron for a specific pattern p given an input $\mathbf{x}^p$ with this pattern:

$$\text{Attr}_p(h_i^k \mid \mathbf{x}^p) = \left\| \sum_{s,t} f(\mathbf{x}^p, \boldsymbol{w}_i^k)_{s,t,:} \right\|_1, \tag{2}$$

Algorithm 1: Pattern Neuron Guided Training Method

Input: The urban time series model $UTSM$; the training dataset $\mathcal{D}_{\text{train}}$ and validation dataset $\mathcal{D}_{\text{val}}$; the size of the detection sample B, and the size of the fine-tuning sample R; and the learning rates for training, α_1, and fine-tuning, α_2.
Output: The fine-tuned urban time series model $UTSM$

```
// Process fine-tuning samples and training samples
```
$\mathcal{D}_{\text{finetune}} \leftarrow \text{RandomSample}(\{\mathbf{x} \in \mathcal{D}_{\text{train}} \mid \mathbf{x} \text{ is a low-frequency sample}\}, R)$
$\mathcal{D}_{\text{train'}} \leftarrow \mathcal{D}_{\text{train}} \setminus \mathcal{D}_{\text{finetune}}$
```
// Train the urban time series model
```
repeat
  Randomly select a batch of instances $\mathcal{S}$ from $\mathcal{D}_{\text{train'}}$
  Optimize $UTSM$ using AdamW with a learning rate of α_1 on batch $\mathcal{S}$.
until *met the stopping criteria*;
```
// Select detection samples and detect the pattern neurons
```
$\mathcal{D}_{\text{detect}} \leftarrow \text{RandomSample}(\{\mathbf{x} \in \mathcal{D}_{\text{train}} \mid \mathbf{x} \text{ is a low-frequency sample}\}, B)$
$\mathcal{N}^{p_l} \leftarrow \text{PND}(UTSM, \mathcal{D}_{\text{detect}})$
```
// Fine-tune the detected pattern neurons
```
$\hat{\mathbf{y}} \leftarrow UTSM(\mathcal{D}_{\text{finetune}}, \mathcal{N}^p)$
$\mathcal{L} \leftarrow \text{MAE}(\hat{\mathbf{y}}, \mathbf{y})$
Optimize pattern neurons $\mathcal{N}^p$ using AdamW with a learning rate α_2.
```
// Return the fine-tuned UTSM
```
return $UTSM$

where s and t represent spatial and temporal dimensions respectively, and $f(\mathbf{x}^p, \boldsymbol{w}_i^k)$ is the function to generate the activation values for the k-th neuron at the i-th layer.

To detect pattern neurons, we focus on samples that exhibit the patterns of interest. Specifically, for identifying pattern neurons associated with low-frequency patterns p_l, e.g., holidays, we use a set of samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_B\}$, where B is the number of samples used for detection, and define pattern neurons as neurons whose attribute scores are high across *all* the B samples:

$$\mathcal{N}^{p_l} = \bigcap_{b=1}^{B} \left\{ n_i^k \mid \text{rank}(\text{Attr}_p(h_i^k \mid \mathbf{x}_b^{p_l})) \leq \epsilon N, \quad \forall i, k \right\} \tag{3}$$

where $\text{rank}(\cdot)$ gives the rank of the attribution score in descending order for the k-th neuron at the i-th layer, ϵ is a predefined threshold that determines the fraction of candidate pattern neurons among all the N neurons in the UTSM given a sample $\mathbf{x}_b^{p_l}$.

Notably, such a detection process can be easily applied to self-attention layers, where the query Q, key K, and value V are the weights of the attention function:

$$\begin{aligned} \text{Attention}(\mathbf{x}) &= \text{softmax}\left(\frac{Q(\mathbf{x})K(\mathbf{x})^\top}{\sqrt{d}}\right)V(\mathbf{x}), \\ Q(\mathbf{x}) = f(\mathbf{x}, W_Q), \quad K(\mathbf{x}) &= f(\mathbf{x}, W_K), \quad V(\mathbf{x}) = f(\mathbf{x}, W_V). \end{aligned} \tag{4}$$

In particular, the attribution scores for these layers can be obtained via Equation 2 using $f(\mathbf{x}, W_Q)$, $f(\mathbf{x}, W_K)$, and $f(\mathbf{x}, W_V)$ respectively as the function $f(\mathbf{x}^p, \boldsymbol{w}_i^k)$.

3.2 Pattern Neuron Verification and Optimization

In this section, we answer the two key research questions: (1) Do pattern neurons exist for low-frequency patterns? (2) Can optimizing these pattern neurons improve the performance of UTSMs? To answer these, we employ a Pattern Neuron Verifier (PNV) to validate the existence of pattern neurons and devise a Pattern Neuron Optimizer (PNO) to enhance UTSM performance by fine-tuning the detected pattern neurons.

Pattern Neuron Verifier (PNV) To validate the existence of pattern neurons associated with low-frequency patterns, we deactivate the neurons identified by the PND and observe the effect on UTSM predictions. For comparison, we also deactivate a set of randomly selected neurons while ensuring that the number of randomly deactivated neurons matches that of the identified pattern neurons. By measuring the difference in forecasting accuracy, we can then confirm the importance of pattern neurons. Particularly, if the prediction error increases significantly without pattern neurons, the importance of these neurons to forecasting can be validated:

$$\sum_{d=1}^{D} \|\mathbf{y}_d - \text{UTSM}(\mathbf{x}_d, \boldsymbol{W} \backslash \boldsymbol{w}_{\text{pattern}})\|_1 \gg \sum_{d=1}^{D} \|\mathbf{y}_d - \text{UTSM}(\mathbf{x}_d, \boldsymbol{W} \backslash \boldsymbol{w}_{\text{random}})\|_1 , \tag{5}$$

where $\mathbf{y}_d$ represents the ground truth for the low-frequency sample $\mathbf{x}_d$, and D is the number of verification samples.

Pattern Neuron Optimizer (PNO). If pattern neurons are confirmed to exist, the next step is to determine whether optimizing these neurons can enhance urban time series forecasting. To achieve this, we propose a fine-tuning mechanism designed specifically to optimize the detected pattern neurons. The objective of PNO is to minimize this loss while improving forecasting accuracy for low-frequency events, and the loss function is defined as:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y} \mid \theta_{\boldsymbol{w}_{\text{pattern}}}) = \frac{1}{R} \sum_{r=1}^{R} \|\hat{\mathbf{y}}_r - \mathbf{y}_r\|_1 , \tag{6}$$

where $\theta_{\boldsymbol{w}_{\text{pattern}}}$ represents the parameters associated with the pattern neurons, $\hat{\mathbf{y}_r}$ and $\mathbf{y}_r$ denote the prediction and ground truth for the fine-tuning sample $\mathbf{x}_r$ respectively, and R is the total number of samples used for fine-tuning. The `PN-Train` training algorithm is outlined in Algorithm 1.

4 EXPERIMENTS

In this section, we evaluate the capability of our proposed `PN-Train` by designing experiments to address the following questions: **RQ1**: Does `PN-Train` successfully detect the *Pattern Neurons*? **RQ2**: How does `PN-Train` perform in comparison to baseline methods across various urban scenarios by optimizing the detected *Pattern Neurons*? **RQ3**: How does the pattern neuron detector perform compared to existing neuron detection methods? **RQ4**: How do the *Pattern Neurons* in different UTSM components affect forecasting results? **RQ5**: How does `PN-Train` perform under various hyperparameters?

4.1 EXPERIMENT SETTINGS

Datasets We perform experiments on two real-world datasets from two urban scenarios: Metro-Traffic (Hogue, 2019) and Pedestrian (Fang et al., 2024). Metro-Traffic contains hourly westbound traffic volumes on Interstate 94 between Minneapolis and St. Paul, MN from 2012 to 2018, including 63 holidays. Pedestrian comprises hourly pedestrian counts from 48 sensors in Melbourne from 2019 to 2022, covering 52 holidays. Detailed dataset statistics are provided in Appendix A.1.

Baselines We evaluate `PN-Train` against seven commonly used URSMs, categorized as follows: the traditional time series model Historical Average (`HA`), graph-based models including `STGCN` (Yu et al., 2018), `GWNET` (Wu et al., 2019), `AGCRN` (Bai et al., 2020), and `TESTAM` (Lee & Ko, 2024), and graph-free models including `STID` (Shao et al., 2022), `STWA` (Cirstea et al., 2022), and `STAEformer` (Liu et al., 2023). Detailed baseline descriptions are in Appendix A.2.

Implementation Details All experiments are conducted using PyTorch (Paszke et al., 2019) on a single NVIDIA A100 80GB GPU. The look-back window L and forecasting horizon H are both set to 12. The selective ratio ϵ is 0.5, with a pattern neuron detection sample length B of 30 and a fine-tuning sample length R of 10. We split the dataset chronologically into training, validation, and test sets in a 6:2:2 ratio. Fine-tuning samples are randomly selected from the holiday data in the training set and are excluded from training. Detection samples are randomly selected from the validation set,

while test samples are used for verification. We employ STAEformer (Liu et al., 2023) as our UTSM. During training, the UTSM is optimized using the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate α_1 of 0.001. Early stopping is applied with a patience of 20 epochs, and the maximum number of epochs is set to 300. For pattern neuron optimization, the UTSM is fine-tuned using the same optimizer with a learning rate α_2 of 0.002 for one epoch. Further implementation details can be found in Appendix A.3, while important notations and their parameter settings are in Appendix A.4. The model is evaluated using MAE, RMSE, and WMAPE, with more details in Appendix A.5.

4.2 Main Results

Validation of Pattern Neurons To address **RQ1** and validate the existence of *Pattern Neurons*, we use PND to detect them and PNV to evaluate `PN-Train`'s performance under neuron deactivation 1. `Original` leaves all neurons active, `D-PN` deactivates pattern neurons associated with holidays as identified by PND, and `D-Random` deactivates the same number of neurons randomly.

Table 1: Pattern neuron verification via neuron deactivation. Lower MAE, RMSE, and WMAPE values indicate better prediction accuracy. † denotes statistically worse results.

Model	**Metro-Traffic (Deactivate ratio 8.77%)**								
	Holiday			Non-Holiday			Overall		
	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE
`Original`	446.04	846.75	16.36%	208.84	339.77	6.19%	220.00	379.14	6.58%
`D-Random`	492.29	833.99	18.05%	263.34	380.46	7.80%	274.11	413.12	8.19%
`D-PN`	663.46†	1046.40†	24.33%†	474.02†	586.01†	14.04%†	482.93†	615.44†	14.43%†
Model	**Pedestrian (Deactivate ratio 9.77%)**								
	Holiday			Non-Holiday			Overall		
	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE
`Original`	109.01	259.79	29.31%	78.82	196.39	21.70%	80.45	200.33	22.12%
`D-Random`	116.99	264.03	31.46%	91.75	210.79	25.26%	93.12	214.01	25.60%
`D-PN`	194.53†	370.80†	52.31%†	174.92†	321.45†	48.15%†	175.98†	324.31†	48.38%†

The results confirm the existence of *Pattern Neurons*, and PND successfully detects them. Deactivating the neurons identified by PND (`D-PN`) leads to a significant performance drop compared to the `Original`, with MAE increasing by 48.75% for the Metro-Traffic dataset and 78.46% for the Pedestrian dataset for holiday samples. In contrast, randomly deactivating an equivalent number of neurons (`D-Random`) causes much smaller degradation: 10.37% for Metro-Traffic and 7.32% for Pedestrian. This stark difference in performance suggests that the neurons detected by PND are indeed closely associated with the patterns of interest, i.e., holidays.

The findings also show that holiday pattern neurons constitute a small fraction of the entire UTSM, comprising 8.77% in the Metro-Traffic dataset and 9.77% in the Pedestrian dataset. Despite their small number, deactivating these pattern neurons significantly degrades performance. Notably, deactivating neurons associated with low-frequency patterns also negatively impacts the performance of non-holiday patterns. This occurs because the pattern neurons include those that capture general time series knowledge, as they were selected based on their high influence on overall forecasting accuracy. The variation in deactivation ratios between the two datasets demonstrates that our PND can dynamically select neurons based on the data, as it identifies pattern neurons by focusing on those with consistently high attribution scores across all detection samples.

Overall Performance We report the results of `PN-Train` with baselines in Table 2 to answer the **RQ2**. The findings confirm that optimizing the *Pattern Neurons* improves urban time series forecasting. `PN-Train` achieves the best overall performance across both the Metro-Traffic and Pedestrian datasets.

By fine-tuning the holiday pattern neurons, `PN-Train` consistently outperforms `PN-Train` * on both datasets, as it enhances the model's ability to capture holiday patterns. While excluding holiday samples during training causes `PN-Train` * to underperform its base UTSM (`STAEformer`) in the Metro-Traffic dataset, fine-tuning the holiday pattern neurons offsets this and improves forecast-

Table 2: Comparison with baselines on Metro-Traffic and Pedestrian datasets. Lower MAE, RMSE, and WMAPE indicate better prediction accuracy. * denotes PN-Train without PNO. **Best results** are in bold, and second-best are underlined. PN-Train employs STAEformer as its UTSM.

	Method	Holiday			Non-Holiday			Overall		
		MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE
Metro-Traffic	HA	1652.11	2027.66	59.50%	2301.77	2762.69	65.15%	2271.20	2732.54	64.88%
	STGCN (Yu et al., 2018)	460.97	739.63	16.91%	289.85	501.33	8.58%	297.90	515.02	8.90%
	GWNet (Wu et al., 2019)	534.76	832.13	19.61%	347.50	582.90	10.29%	356.31	596.97	10.64%
	AGCRN (Bai et al., 2020)	453.23	738.60	16.62%	280.41	496.75	8.31%	288.54	510.71	8.62%
	STID (Shao et al., 2022)	586.90	1031.50	21.52%	216.09	346.81	6.40%	233.54	405.81	6.98%
	STWA (Cirstea et al., 2022)	521.02	820.57	19.11%	355.63	619.28	10.53%	364.36	630.61	10.89%
	STAEformer (Liu et al., 2023)	443.23	821.42	16.25%	210.41	343.01	6.23%	221.37	379.29	6.62%
	TESTAM (Lee & Ko, 2024)	486.89	857.99	17.86%	335.05	555.09	9.92%	342.19	572.94	10.22%
	PN-Train *	446.04	846.75	16.35%	208.84	339.77	6.19%	220.00	379.14	6.58%
	PN-Train	**430.40**	**816.50**	**15.78%**	**203.62**	**332.15**	**6.03%**	**214.29**	**369.46**	**6.40%**
Pedestrian	HA	208.49	388.17	64.48%	255.12	471.08	83.46%	253.24	468.01	82.69%
	STGCN (Yu et al., 2018)	120.75	258.53	32.47%	101.61	214.32	27.97%	102.65	216.95	28.22%
	GWNet (Wu et al., 2019)	119.77	267.48	32.21%	113.69	245.87	31.30%	114.02	247.09	31.35%
	AGCRN (Bai et al., 2020)	118.48	267.32	31.86%	108.22	245.55	29.79%	108.78	246.78	29.91%
	STID (Shao et al., 2022)	116.42	263.79	31.31%	85.32	206.36	23.49%	87.00	209.87	23.92%
	STWA (Cirstea et al., 2022)	114.18	261.03	30.70%	106.62	234.88	29.35%	106.90	236.13	29.39%
	STAEformer (Liu et al., 2023)	115.24	273.64	30.99%	82.23	202.73	22.64%	84.02	207.19	23.10%
	TESTAM (Lee & Ko, 2024)	**103.79**	**257.10**	**27.91%**	94.04	219.46	25.89%	94.57	221.67	26.00%
	PN-Train *	109.01	259.79	29.31%	78.82	196.39	21.70%	80.45	200.33	22.12%
	PN-Train	106.11	253.86	28.54%	**78.35**	**194.72**	**21.57%**	**79.85**	**198.38**	**21.95%**

ing performance. This is because optimizing the holiday neurons helps the network better represent holidays than training on a mix of low-frequency holiday and high-frequency non-holiday samples. In contrast, with more frequent holidays in the Pedestrian dataset, excluding some holiday samples can actually improve accuracy by removing noisy outliers. Nevertheless, fine-tuning the pattern neurons further enhances PN-Train *, as holiday events, though more frequent, are still low-frequency overall and may not be fully captured during initial training.

Additionally, fine-tuning the *Pattern Neurons* not only improves performance on holiday samples but also enhances non-holiday and overall performance. This is because these neurons also memorize general time series knowledge, such as level and trend (Brockwell et al., 2016), and optimizing them strengthens the model's representation learning of general time series. Although TESTAM performs well on holiday samples in the Pedestrian dataset by leveraging different experts, its overall performance is limited by the routing mechanism. In contrast, PN-Train addresses low holiday performance at the neuron level without degrading non-holiday performance, leading to better results across all scenarios.

4.3 Model Analysis

Study on Pattern Neuron Detector We further assess our proposed PND by comparing it with recent neuron detection techniques to address **RQ3**. Specifically, we evaluate the following variants of PN-Train, including: **w/o** PND: excludes the PND; **w** GD: replaces PND with the gradient-based detector from (Chen et al., 2024). **w** FD: replaces PND with the perturbation-based detector from (Zhao et al., 2024). The results are shown in Table 3.

The results confirm the importance of neuron detection. Performance drops significantly without it, as fine-tuning all parameters in the UTSM based on low-frequency patterns leads to overfitting. In contrast, UTSMs with neuron detection, i.e., **w GD**, **w FD**, and PN-Train, effectively identify and fine-tune only the pattern neurons, preventing overfitting and preserving the model's generalization capability.

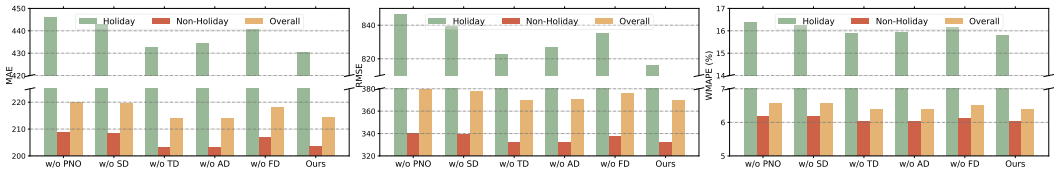
The results also reveal that perturbation-based detectors outperform gradient-based methods in urban time series forecasting, as they directly measure how changes impact predictions, offering clearer insights into neuron importance. While gradient-based methods capture sensitivity to parameter changes, they fall short in demonstrating a neuron's overall impact on forecasting accuracy. Our proposed PND is a finer-grained perturbation-based detector that evaluates how changes affect pre-

Table 3: Results of PN-Train with different neuron detection techniques.

Model	Metro-Traffic								
	Holiday			Non-Holiday			Overall		
	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE
w/o PND	1082.46	1444.20	60.80%	1294.44	1664.49	38.34%	1284.47	1654.78	38.39%
w GD	438.32	826.83	16.07%	206.84	335.71	6.13%	217.73	373.58	6.51%
w FD	434.76	825.71	15.94%	204.52	333.76	6.05%	215.36	371.80	6.44%
PN-Train	**430.40**	**816.50**	**15.78%**	**203.62**	**332.15**	**6.03%**	**214.29**	**369.46**	**6.40%**
Model	**Pedestrian**								
	Holiday			Non-Holiday			Overall		
	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE
w/o PND	238.24	426.90	64.06%	225.54	378.80	62.09%	226.23	381.56	62.20%
w GD	108.27	256.64	29.12%	79.94	196.20	22.01%	81.48	199.94	22.40%
w FD	107.55	256.26	28.92%	78.93	195.39	21.73%	80.48	199.16	22.13%
PN-Train	**106.11**	**253.86**	**28.54%**	**78.35**	**194.72**	**21.57%**	**79.85**	**198.38**	**21.95%**

dictions at each linear layer in the UTSM, rather than focusing only on attention scores and feed-forward layers as in **w FD**. This allows PND to achieve the best performance.

Ablation Study We design the following variants to answer **RQ4** by evaluating the effectiveness of the pattern neuron optimizer (PNO) on different transformer-based UTSM components, including: **w/o** PNO: excludes the PNO in PN-Train; **w/o** SD: omits optimization of pattern neurons in the spatial transformer; **w/o** TD: omits optimization of pattern neurons in temporal transformer; **w/o** AD: omits optimization of pattern neurons in self-attention mechanism; **w/o** FD: omits optimization of pattern neurons in the feed-forward layer.



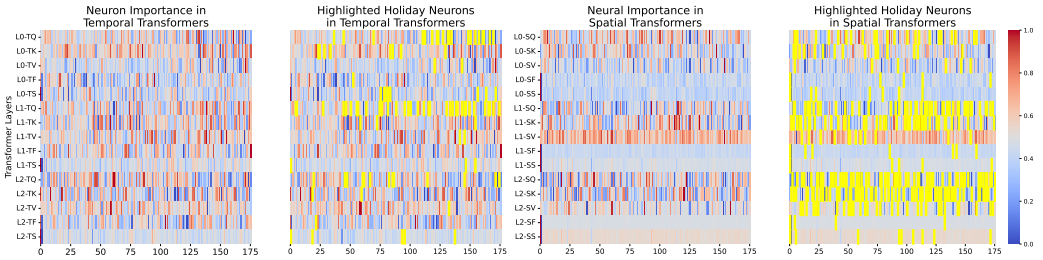
(a) Ablation study on Metro-Traffic dataset.

(b) Ablation study on Pedestrian dataset.

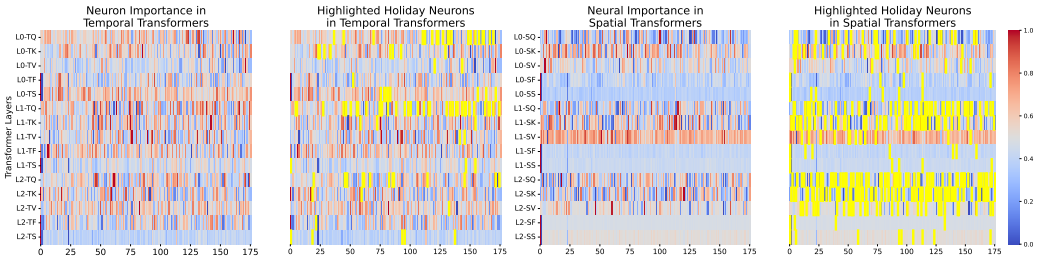
Figure 3: Ablation study results.

The results presented in Figure 3 confirm that the proposed PNO significantly enhances forecasting performance. Across both datasets, the absence of PNO leads to a notable decline in accuracy, particularly in holiday scenarios. Fine-tuning the *Pattern Neurons* across all UTSM components proves crucial, as each component addresses a distinct aspect of the data: the spatial transformer learns spatial correlations, the temporal transformer captures temporal patterns, the attention mechanism refines short-term dependencies, and the feed-forward layer enhances long-term memory. PN-Train fine-tunes *Pattern Neurons* in all components, consistently outperforming its variants and highlighting the importance of identifying and fine-tuning *Pattern Neurons* across the entire model.

Pattern Neuron Visualization To address **RQ3**, we conducted a deeper analysis of neurons across various UTSM layers by visualizing attribution scores for holiday patterns in the Traffic dataset, as shown in Figure 4.



(a) Visualization of detection sample #3 for Metro-Traffic dataset.

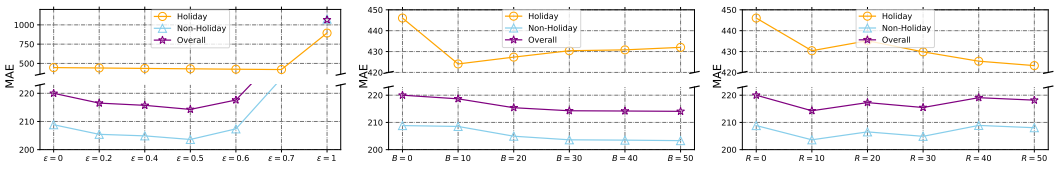


(b) Visualization of detection sample #13 for Metro-Traffic dataset.

Figure 4: Visualization of neuron importance, i.e., normalized attribution scores, across USTM layers. LX represents the X-th linear layer in the transformer. TQ, TK, TV, TF, and TS represent neurons in the temporal transformer's query, key, value, first linear layer, and second linear layer, respectively, while SQ, SK, SV, SL, and SF denote the same in the spatial transformer. *Pattern Neurons* are highlighted in yellow.

The results show that neurons with high attribution scores consistently appear in similar positions, identifying specific holiday neurons that can be detected with a small number of samples. The number of holiday neurons varies between the temporal and spatial transformers, with more concentrated in the query and key components, emphasizing the role of attention mechanisms in detecting low-frequency patterns like holidays. Furthermore, the distribution of pattern neurons across layers reflects a hierarchical structure, where shallow layers capture general patterns and middle layers refine lower-level features. The visualization of neuron importance for low- and high-frequency patterns, as well as for the Pedestrian dataset, can be found in Appendix A.6.

Hyperparameter Study We investigate the effects of hyperparameters in `PN-Train` to address **RQ5**. Specifically, we examine three key hyperparameters: the selection ratio (ϵ), the number of detection samples (B), and the number of fine-tuning samples (R). From Figure 5, we observe:



(a) Selective Ratio (b) No. of Detection Samples (c) No. of Fine-tuning Samples

Figure 5: Hyperparameter study on Metro-Traffic dataset.

There is a trade-off between holiday and non-holiday performance. When $\epsilon = 1$, all neurons in the UTSM are fine-tuned. Increasing ϵ from 0 to 0.7 improves holiday performance, but non-holiday performance declines as ϵ increases from 0.5 to 0.7. This occurs because, with a larger ϵ, too many *Pattern Neurons*, including those responsible for general time series knowledge, are detected and fine-tuned, leading to overfitting the UTSM to holiday patterns. We opt for $\epsilon = 0.5$ as it provides the best balance between holiday and non-holiday performance.

A detection sample size of $B = 30$ is sufficient to identify the *Pattern Neurons*. Increasing the number of detection samples reduces the number of neurons associated with low-frequency patterns being selected, as we only detect neurons with high attribution scores across all detection samples. Consequently, using a larger number of detection samples may cause certain pattern neurons to go undetected due to slight variations in holiday patterns.

`PN-Train` achieves the best performance when $R = 10$, indicating that fine-tuning specific neurons associated with low-frequency patterns is low-cost, requiring only a few samples to boost performance for both low- and high-frequency patterns.

5 Related Work

Urban Time Series Forecasting Urban time series forecasting is a crucial aspect of smart city development, with various urban time series models (UTSMs) designed to support a wide range of applications. Initial efforts relied on conventional time series models, such as Autoregressive Integrated Moving Average (ARIMA) (Williams & Hoel, 2003; Tran et al., 2015) and Holt-Winters methods (de Assis et al., 2013; Brügner, 2017). However, these approaches often fail to capture the complex patterns inherent in urban data. Recently, deep learning-based models, including graph-based (Zheng et al., 2020; Wu et al., 2019; Bai et al., 2020; Wu et al., 2020) and graph-free approaches (Deng et al., 2021; Shao et al., 2022; Liu et al., 2023), have gained prominence due to their ability to learn non-linear relationships more effectively. While these methods improve overall performance, they often overlook low-frequency patterns in urban time series, particularly when the sample size is insufficient to train deep models Krawczyk (2016); Lee & Ko (2024). Furthermore, existing research has not examined UTSMs at the component level. In this study, we focus on investigating UTSMs, and based on our findings, we design `PN-Train` to address low-frequency patterns at the neuron level to improve forecasting accuracy.

Neuron Interpretability Neuron interpretability has gained significant attention for explaining neural networks across various applications, from visual (Bau et al., 2017; Mu & Andreas, 2020) to language models (Bau et al., 2019; Xin et al., 2019; Dalvi et al., 2020). Recent studies (Dai et al., 2022; Wang et al., 2022) have demonstrated that certain neurons in large language models (LLMs) capture knowledge-specific contexts. To detect these knowledge neurons, techniques can be grouped into three categories: gradient-based methods (Dai et al., 2022; Chen et al., 2024), which identify the neurons with high attribution scores from the integrated gradients as knowledge neurons; entropy-based activation analysis (Tang et al., 2024), which identifies neurons with low activation probability entropy as knowledge neurons; and perturbation-based difference evaluation (Zhao et al., 2024), which detects knowledge neurons by measuring differences in feature representations between activated and deactivated states. Recent works (Tang et al., 2024; Zhao et al., 2024) also demonstrate that neuron-level manipulation can enhance model capabilities. However, neuron interpretability remains largely unexplored in the context of urban time series data. In this work, we interpret neurons in UTSMs using a finer-grained perturbation-based technique, revealing how these models capture low-frequency patterns in urban time series forecasting.

6 Conclusion

We introduced `PN-Train`, a novel training method that incorporates a perturbation-based neuron detector to confirm the existence of pattern neurons associated with distinct patterns in urban time series data. Building on this, we proposed a pattern neuron optimizer that focuses on fine-tuning these neurons to enhance forecasting for low-frequency patterns, such as holidays. Our experiments demonstrate that fine-tuning a small subset of neurons, i.e., less than 10% of the total parameters, can significantly improve forecasting accuracy for low-frequency patterns. We also observed that in transformer-based urban time series models, the key and query components play a crucial role in capturing patterns. Additionally, optimizing these pattern neurons enhances forecasting for high-frequency patterns, such as non-holidays, as they capture essential underlying time series knowledge. `PN-Train` surpasses previous baselines in forecasting accuracy for both low- and high-frequency events, contributing to overall performance improvements. Given that neuron-level investigations have received limited attention in time series analysis, we hope our findings provide a fresh perspective and inspire further exploration of time series models at the neuron level.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James R. Glass. Identifying and controlling important neurons in neural machine translation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Peter J Brockwell, Peter J Brockwell, Richard A Davis, and Richard A Davis. *Introduction to time series and forecasting*. Springer, 2016.

Helge Brügner. Holt-winters traffic prediction on aggregated flow data. *Future Internet (FI) and Innovative Internet Technologies and Mobile Communication (IITM) Focal Topic: Advanced Persistent Threats*, 25:25–32, 2017.

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 17817–17825. AAAI Press, 2024.

Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. Towards spatio-temporal aware traffic time series forecasting. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2900–2913. IEEE, 2022.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 8493–8502. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.581. URL https://doi.org/10.18653/v1/2022.acl-long.581.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. Analyzing redundancy in pre-trained transformer models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 4908–4926. Association for Computational Linguistics, 2020.

Marcos VO de Assis, Luiz F Carvalho, Joel JPC Rodrigues, and Mario Lemes Proença. Holt-winters statistical forecasting and aco metaheuristic for traffic characterization. In *2013 IEEE International Conference on Communications (ICC)*, pp. 2524–2528. IEEE, 2013.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 269–278, 2021.

Jiangyi Fang, Liyue Chen, Di Chai, Yayao Hong, Xiuhuai Xie, Longbiao Chen, and Leye Wang. Uctb: An urban computing tool box for building spatiotemporal prediction services. In *2024 IEEE International Conference on Software Services Engineering (SSE)*, pp. 54–65. IEEE, 2024.

Haotian Gao, Renhe Jiang, Zheng Dong, Jinliang Deng, and Xuan Song. Spatio-temporal-decoupled masked pre-training for traffic forecasting. *arXiv preprint arXiv:2312.00516*, 2023.

Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3656–3663, 2019.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 5484–5495. Association for Computational Linguistics, 2021.

John Hogue. Metro interstate traffic volume. `https://doi.org/10.24432/C5X60B`, May 2019. URL `https://doi.org/10.24432/C5X60B`.

Jiahao Ji, Jingyuan Wang, Zhe Jiang, Jiawei Jiang, and Hu Zhang. STDEN: towards physics-guided neural networks for traffic flow prediction. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 4048–4056. AAAI Press, 2022.

Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 8078–8086, 2023.

Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in artificial intelligence*, 5(4):221–232, 2016.

Chunggi Lee, Yeonjun Kim, Seungmin Jin, Dongmin Kim, Ross Maciejewski, David Ebert, and Sungahn Ko. A visual analytics system for exploring, monitoring, and forecasting road traffic congestion. *IEEE transactions on visualization and computer graphics*, 26(11):3133–3146, 2019.

Hyunwook Lee and Sungahn Ko. TESTAM: A time-enhanced spatio-temporal attention model with mixture of experts. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=N0nTk5BSvO`.

Hyunwook Lee, Seungmin Jin, Hyeshin Chu, Hongkyu Lim, and Sungahn Ko. Learning to remember patterns: Pattern matching memory networks for traffic forecasting. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Daifeng Li, Kaixin Lin, Xuting Li, Jianbin Liao, Ruo Du, Dingquan Chen, and Andrew Madden. Improved sales time series predictions using deep neural networks with spatiotemporal dynamic pattern acquisition mechanism. *Information Processing & Management*, 59(4):102987, 2022.

Dachuan Liu, Jin Wang, Shuo Shang, and Peng Han. Msdr: Multi-step dependency relation networks for spatial temporal forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1042–1050, 2022.

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

Lingbo Liu, Jiajie Zhen, Guanbin Li, Geng Zhan, Zhaocheng He, Bowen Du, and Liang Lin. Dynamic spatial-temporal representation learning for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):7169–7183, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Xianglong Luo, Danyang Li, and Shengrui Zhang. Traffic flow prediction during the holidays based on DFT and SVR. *J. Sensors*, 2019:6461450:1–6461450:10, 2019.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Cheonbok Park, Chunggi Lee, Hyojin Bahng, Yunwon Tae, Seungmin Jin, Kihwan Kim, Sungahn Ko, and Jaegul Choo. St-grat: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 1215–1224, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4454–4458, 2022.

Slawek Smyl, Grzegorz Dudek, and Paweł Pełka. Es-drnn: a hybrid exponential smoothing and dilated recurrent neural network model for short-term load forecasting. *IEEE transactions on neural networks and learning systems*, 2023.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. Language-specific neurons: The key to multilingual capabilities in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 5701–5715. Association for Computational Linguistics, 2024.

Quang Thanh Tran, Zhihua Ma, Hengchao Li, Li Hao, and Quang Khai Trinh. A multiplicative seasonal arima/garch model in evn traffic prediction. *International Journal of Communications, Network and System Sciences*, 8(4), 2015.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill neurons in pre-trained transformer-based language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 11132–11152. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.765.

Zhenzhu Wang, Yishuai Chen, Jian Su, Yuchun Guo, Yongxiang Zhao, Weikang Tang, Chao Zeng, and Jingwei Chen. Measurement and prediction of regional traffic volume in holidays. In *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, October 27-30, 2019*, pp. 486–491. IEEE, 2019.

Billy M Williams and Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 1907–1913. ijcai.org, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Ji Xin, Jimmy Lin, and Yaoliang Yu. What part of the neural network does this? understanding lstms by measuring and dissecting neurons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5823–5830, 2019.

Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 3634–3640. ijcai.org, 2018.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. How do large language models handle multilingualism? *arXiv preprint arXiv:2402.18815*, 2024.

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

Yirong Zhou, Jun Li, Hao Chen, Ye Wu, Jiangjiang Wu, and Luo Chen. A spatiotemporal attention mechanism-based model for multi-step citywide passenger demand prediction. *Information Sciences*, 513:372–385, 2020.

A Appendix

A.1 Dataset Details

Table A.1 provides a summary of the statistical information for the two real-world datasets, Metro-Traffic (Hogue, 2019) and Pedestrian (Fang et al., 2024). This includes the time span of each dataset, the selected frequency and sensor size, as well as the number of weekdays, weekends, and holidays within the time span.

Table A.1: Statistics of the datasets.

Dataset	Time Span	Frequency	Sensor Size	Weekdays	Weekends	Holidays
Metro-Traffic	10/02/2012 - 30/09/2018	1 hour	1	1,731	694	63
Pedestrian	11/02/2019 - 31/10/2022	1 hour	48	971	388	52

A.2 Baselines

To thoroughly evaluate our model, we compare `PN-Train` with several widely used urban time series models (UTSMs), including the following:

- `HA` is a traditional time series model that forecasts future values by averaging historical data for corresponding time slots.
- `STGCN` (Yu et al., 2018) is a graph-based UTSM that employs graph convolution networks to capture spatial dependencies among citywide sensors and uses a 1D convolution network to model temporal dependencies.
- `GWNET` (Wu et al., 2019) enhances `STGCN` by introducing a self-adaptive graph neural network to learn dynamic spatial dependencies and uses stacked dilated causal convolutions to model temporal patterns.
- `AGCRN` (Bai et al., 2020) is a graph-based model that captures region-specific spatio-temporal correlations through an adaptive graph convolutional recurrent network.
- `STID` (Shao et al., 2022) is a graph-free UTSM that encodes spatial and temporal identities using an embedding layer and applies Multi-Layer Perceptrons to learn spatio-temporal correlations in urban time series data.
- `STWA` (Cirstea et al., 2022) is a graph-free urban traffic series model (UTSM) that employs location- and time-specific parameters to enable a spatio-temporal aware attention mechanism.

- `STAEformer` (Liu et al., 2023) improves upon `STID` by introducing spatio-temporal adaptive embeddings, allowing the vanilla transformer to learn dynamic spatio-temporal correlations more effectively.
- `TESTAM` (Lee & Ko, 2024) is a graph-based UTSM that captures dynamic spatial relationships through an adaptive graph-based attention mechanism and employs a mixture of experts to capture both regular and irregular patterns in urban time series.

A.3 Experimental Setup

All experiments were conducted on an NVIDIA A100 80GB GPU and repeated three times. We used the AdamW optimizer Loshchilov & Hutter (2019) with a 0.001 learning rate, early stopping with a patience of 20 epochs, and a maximum of 300 epochs. The batch size was 32, with a look-back window (L) of 12 and a forecast horizon (H) of 12. Implemented in PyTorch (Paszke et al., 2019), our method used the official code for all baselines. STAEformer (Liu et al., 2023) served as our UTSM, with all other parameters the same as the original model.

A.4 Notations

In this section, we present a table of important notations in Table A.2.

Table A.2: Table of important notations in `PN-Train`.

Notation	Description	Parameter
L	Look-back window	12
H	Forecast horizon	12
$Attr_p$	Attribution score for pattern p	-
ϵ	Selective ratio for neurons with high attribution scores	0.5
B	Sample sizes for pattern neuron detection	30
D	Sample sizes for pattern neuron verification	-
R	Sample sizes for pattern neuron optimization	10

A.5 Metric Details

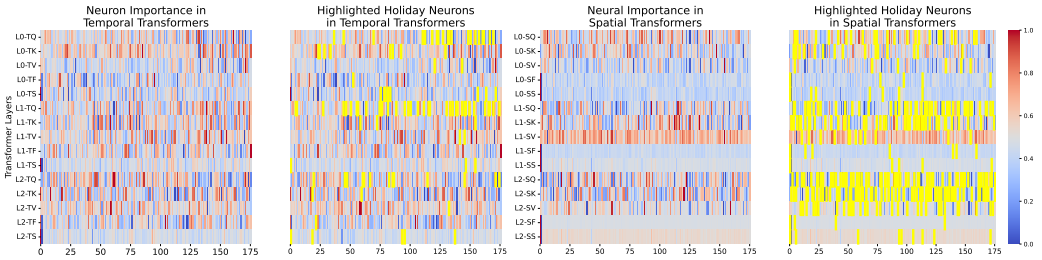
We evaluate performance using three metrics, each assessing the model from a different perspective: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Weighted Mean Absolute Percentage Error (WMAPE). MAE measures the average L1 distance between predicted values and the ground truth, making it less sensitive to outliers. RMSE, as the square root of the average L2 distance, gives more weight to outliers. WMAPE evaluates accuracy based on percentage errors, which is scale-independent.

$$\text{MAE} = \frac{1}{\xi}\sum_{i=1}^{\xi}\left|\hat{\mathbf{y}}^i - \mathbf{y}^i\right|, \text{RMSE} = \sqrt{\frac{1}{\xi}\sum_{i=1}^{\xi}\left(\hat{\mathbf{y}}^i - \mathbf{y}^i\right)^2}, \text{WMAPE} = \frac{\sum_{i=1}^{\xi}\left|\hat{\mathbf{y}}^i - \mathbf{y}^i\right|}{\sum_{i=1}^{\xi}\left|\mathbf{y}^i\right|} \tag{7}$$

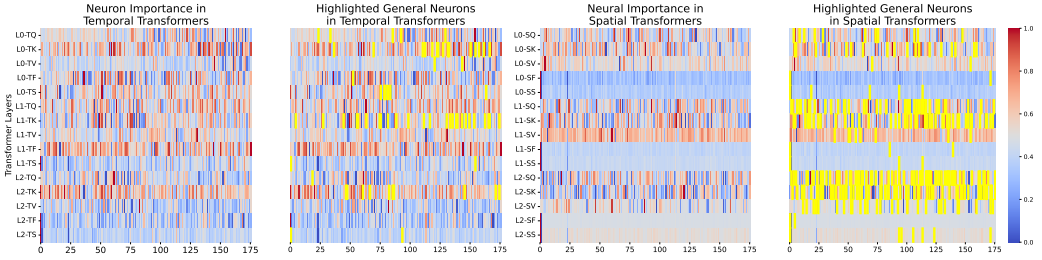
where $\hat{\mathbf{y}}^i$ and $\mathbf{y}^i$ denote the predicted values and ground truth, and ξ is the total number of samples.

A.6 Pattern Neuron Visualization

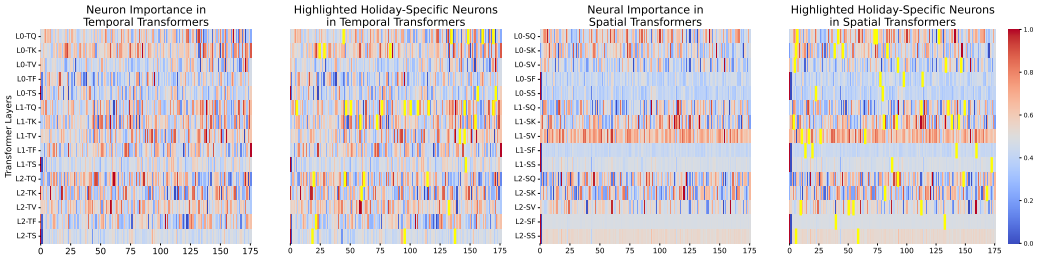
We further visualize pattern neurons for both low- and high-frequency patterns. Specifically, we visualize holiday, non-holiday, and holiday-specific neurons on the Metro-Traffic dataset in Figure A.1. The results confirm our assumption that holiday-related neurons include those specific to low-frequency events, which do not contribute to high-frequency events, as well as those that learn general time series features useful for both low- and high-frequency patterns. Additionally, pattern neurons are primarily located in the transformer's query and key components, which are responsible for capturing patterns.



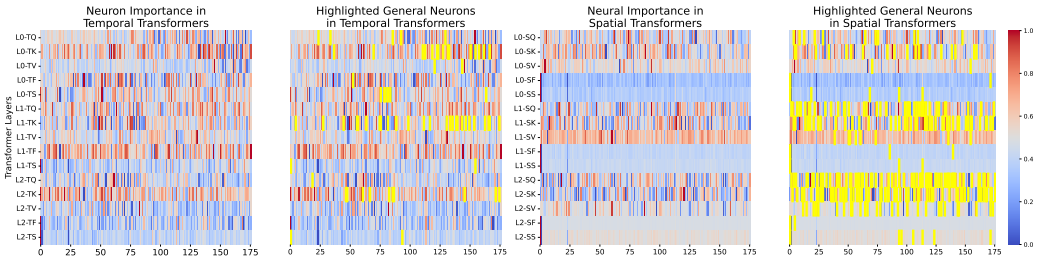
(a) Visualization of Holiday Neurons for Metro-Traffic dataset.



(b) Visualization of Non-Holiday Neurons for Metro-Traffic dataset.



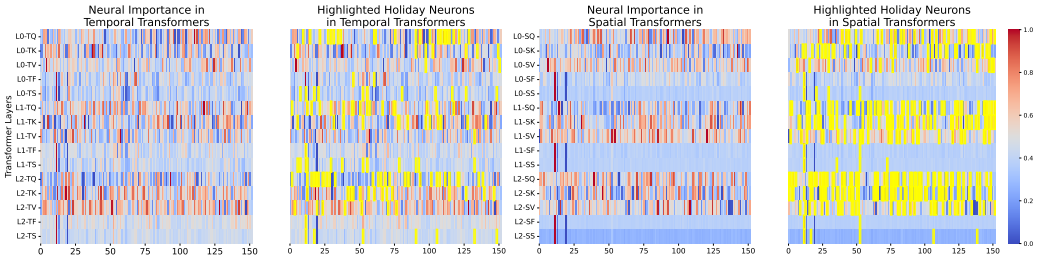
(c) Visualization of Holiday-Specific Neurons for Metro-Traffic dataset.



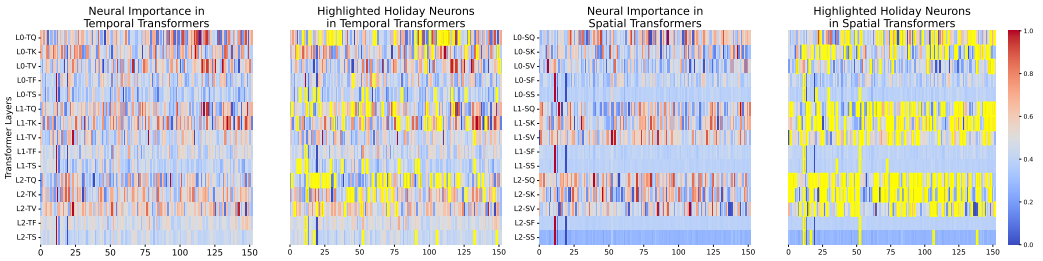
(d) Visualization of General Neurons for Metro-Traffic dataset.

Figure A.1: Visualization of neuron importance, i.e., normalized attribution scores, across USTM layers for Traffic dataset. TQ, TK, TV, TF, and TS represent neurons in the temporal transformer's query, key, value, first linear layer, and second linear layer, respectively, while SQ, SK, SV, SL, and SF denote the same in the spatial transformer. *Pattern Neurons* are highlighted in yellow.

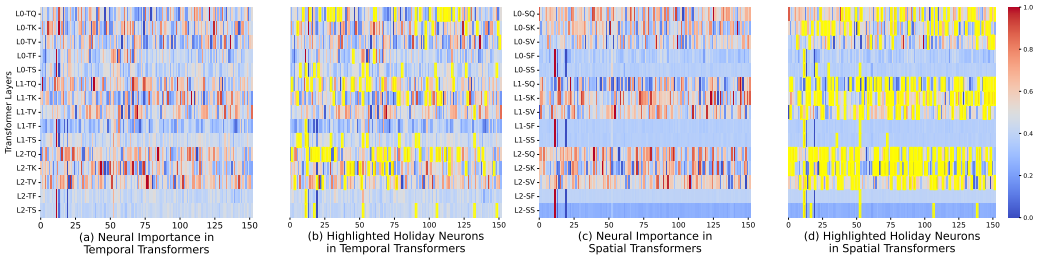
Furthermore, in Figure A.2, we visualize pattern neurons for the holiday pattern on the Pedestrian dataset. Similar to the holiday neurons in the Metro-Traffic dataset discussed in the main paper, we observe that high attribution scores consistently appear in similar positions, with the query and key components playing a crucial role in emphasizing holiday patterns.



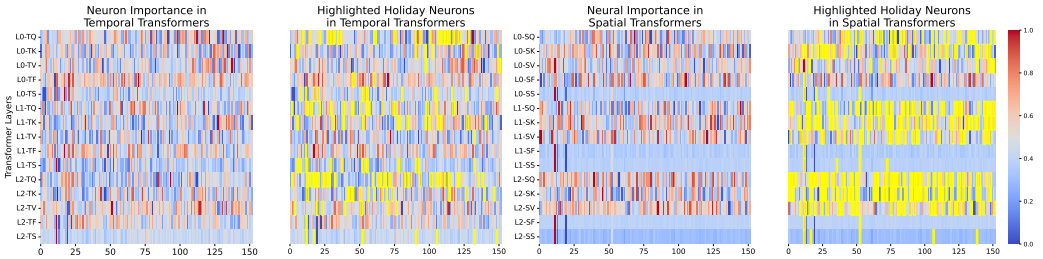
(a) Visualization of detection sample #1 for Pedestrian dataset.



(b) Visualization of detection sample #2 for Pedestrian dataset.



(c) Visualization of detection sample #3 for Pedestrian dataset.



(d) Visualization of detection sample #4 for Pedestrian dataset.

Figure A.2: Visualization of neuron importance, i.e., normalized attribution scores, across USTM layers for Pedestrian dataset. *Pattern Neurons* are highlighted in yellow.