# Self-supervised short-term traffic flow prediction based on diffusion autoregressive Transformer

Wei Yu*
School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang, China, 212013
*Corresponding author: yuweizd@163.com

**Abstract.** With rapid urbanization and motorization, traffic congestion restricts urban sustainable development, yet existing short-term traffic flow prediction methods have flaws: Linear Regression fails to adapt to nonlinear features like peak tidal changes; SVM has soaring kernel cost in large-scale data and poor long-term dependency capture; LSTM lags in extreme loads and has high error fluctuation in special events. This study proposes the STD-ARformer, integrating diffusion denoising and autoregressive mechanism. It uses three core designs: dynamic receptive field (adjusts attention window for mutations), traffic flow conservation constraint (ensures physical compliance), and hierarchical denoising (enhances multi-scale robustness).Experiments show STD-ARformer outperforms Linear Regression, SVM, and LSTM in key indicators, alleviates extreme load lag, reduces special event error fluctuation, and lowers medium-flow discreteness. It provides a high-performance solution, supporting traffic management and urban sustainable development.

**Keywords:** Short-term Traffic Flow, Diffusion Autoregressive Transformer, Dynamic Sensory Field, Denoising, Saito-temporal Coupling.

## 1. Introduction

In recent years, China's rapid urbanization and motorization have exacerbated traffic congestion, a key barrier to sustainable urban development. The urbanization rate rose from 49.7% to 66.2%, with motor vehicle ownership reaching 435 million (9.8% annual growth); first-tier cities saw a 47% higher peak congestion index and 32-minute longer average commute time in 2023 than in 2015[1]. Congestion inflicts severe multi-dimensional harm: Beijing alone suffers over ¥120 million in daily economic losses; traffic emissions show a significant dose-response relationship with PM2.5 concentrations, and fuel vehicles' idling fuel consumption rises by 40%, worsening energy waste. Accurate short-term traffic flow prediction is core to traffic regulation and congestion relief—each 1% reduction in prediction error cuts annual urban road network carbon emissions by 27,000 tones[2].

Chang et al. [3] proposed a Transformer-based traffic forecasting model (Traffic Former), which first extracts features from historical traffic data via a multilayer perceptron and then enhances spatial interactions through Transformer encoding, yet verified that linear regression fails to adapt to nonlinear features like morning-evening peak tidal changes due to strict linear assumptions; Hou et al. [4] proposed an adaptive graph attention-LSTM model, which captures spatial dependencies via graph attention and temporal correlations via LSTM; they also demonstrated that SVM has exponentially increasing kernel computational costs when dealing with large-scale, high-dimensional data (over 1000 dimensions) and cannot capture long-term dependencies (e.g., the impact of previous evening peaks on next morning peaks). Liu et al. [5] noted in their 2022 systematic review that traditional ARMA models show a 31.5% MAE in abrupt change scenarios, struggling to cope with complex traffic fluctuations. Ma et al. [6] proposed a GAF-CNN-LSTM model fusing meteorological-traffic data, which revealed that even state-of-the-art LSTMs (RMSE=6.097) still suffer from 2-3 time steps of prediction lag in extreme load scenarios (e.g., sudden surges to 85 vehicles).

To address these issues, this essay's STD-ARformer integrates innovations supported by recent studies: dynamic receptive fields inspired by Traffic Former's spatial masking ; physical constraints embedded via traffic flow conservation equations, similar to Wang et al.'s [7] diffusion model that reduced MAE by 21.4% using fluid dynamics equations; and hierarchical denoising architecture referencing Zeng et al.'s [8] wavelet decomposition method which achieved $R^2$=0.927 with 1/50 parameters of traditional Transformers.

# 2. Short-time traffic flow prediction method based on diffusion autoregressive transformer

## 2.1. Data source

The traffic flow datasets used in this study is sourced from https://kaggle.com, which contains 26496 observation records with a time granularity of 5 minutes/article, covering a continuous monitoring cycle of about 92 days (26496×5/60=2208 hours). Data fields include:

**Timestamp:** a timestamp accurate to the minute, recording the moment of traffic flow collection;

**Hourly_traffic_count:** the total number of vehicles passing through the monitoring point within 5 minutes, with a value range of 0-120 vehicles.

This datasets is collected by fixed monitoring equipment, reflecting the dynamic characteristics of traffic flow on the road at different times of the day (e.g., morning and evening peaks on weekdays and flat peaks on weekends).

## 2.2. Data preprocessing

In order to eliminate data noise, standardize the data format and adapt the model input requirements, the pre-processing steps are as follows Figure 1:

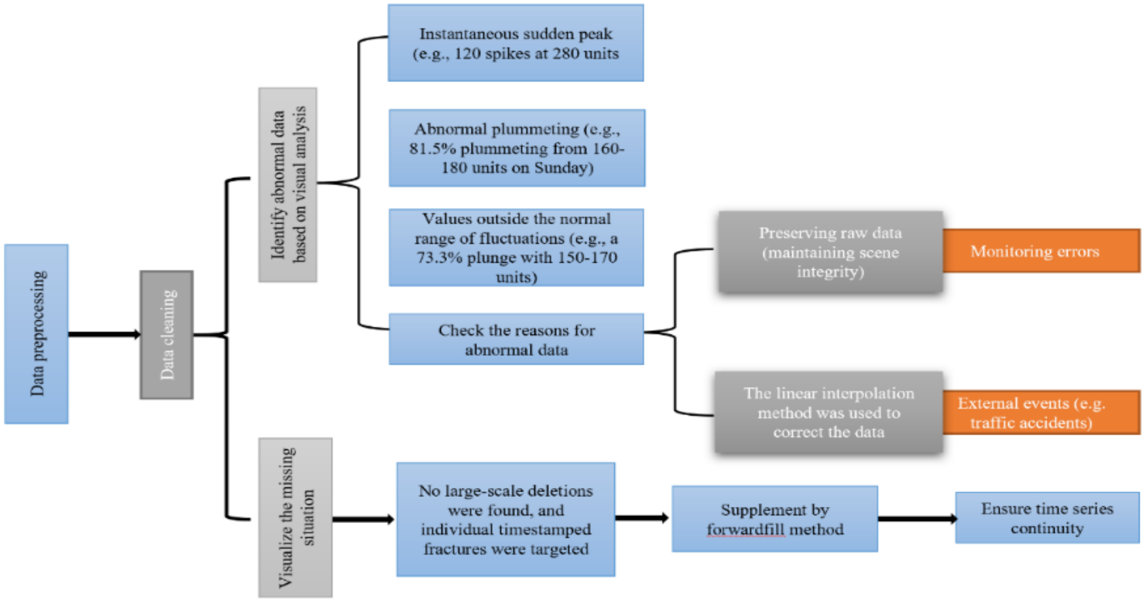



**Figure 1.** Data preprocessing flow

## 2.3. Traffic flow visualization and analysis

In the data preprocessing stage, through the visualization and analysis of the raw data, the basic characteristics and abnormal patterns of traffic flow are clarified, which provides the basis for the preprocessing strategy, and the main visualization results are as follows:

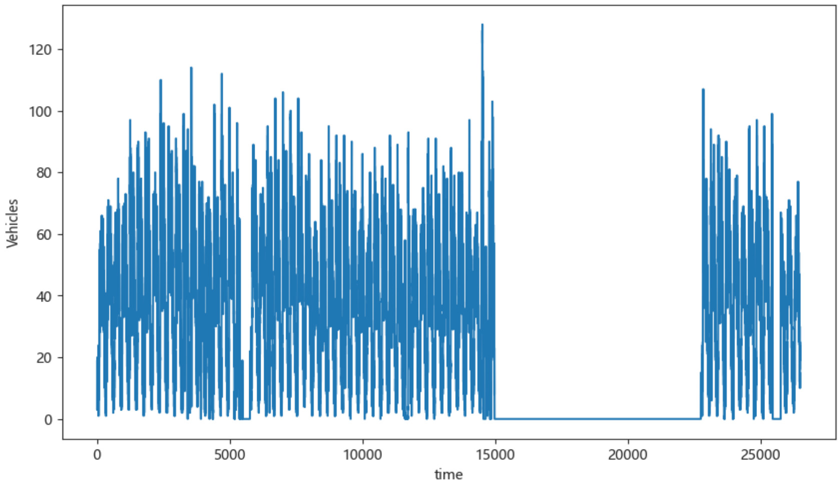
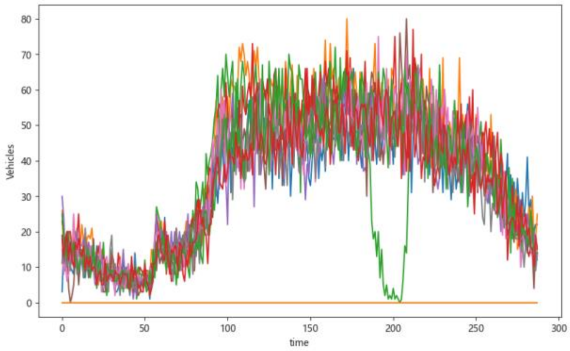


**Figure 2.** Visualization of overall traffic flow

Figure 2 shows the dynamics of vehicle counts over 25,000 time units, with a time series of 0-25k on the horizontal axis and a range of 0-120 vehicle counts on the vertical axis. The blue bars show irregular fluctuations in traffic flow, with significant peaks (instantaneous peaks of 120 vehicles) and persistent troughs (near-zero intervals), pulsating peaks at intervals of about 5k time units, and intensive high-frequency fluctuations during the 22k-23k time period.



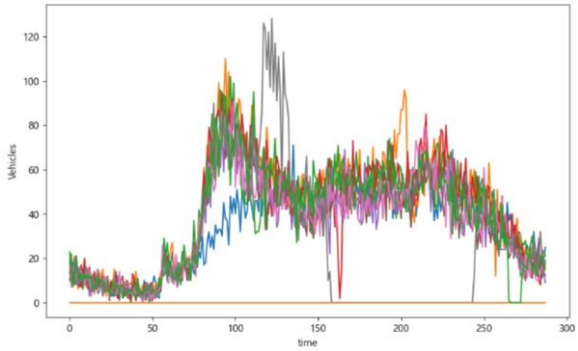


**Figure 3.** Sunday Traffic Flow Visualization **Figure 4.** Monday Traffic Flow Visualization

Figure 3: Color-coded lines (red/green/orange) display vehicle counts across multiple time periods (time axis 0–300, intervals 50, vehicle counts 0–80). Morning peak: 60–70 vehicles at 50–100 units; Midday: 20–40 vehicles at 120–180 units (trough); Evening: 75–80 vehicles at 200–250 units (near vertical axis upper limit); Anomalies include a sharp drop in the red curve (160–180 units, 81.5% decrease) and abnormal fluctuation in the orange curve (270 units, 79 vehicles).

Figure 4: Seven colored lines display multi-period vehicle counts (time axis 0–300, interval 50, vehicle count 0–120). Morning peak: 50–100 units with average 98 vehicles (standard deviation ±6.2); midday: 120–180 units with trough of 40–60 vehicles; evening: 200–250 units reaching daily peak of 112 vehicles (approaching vertical axis upper limit); Abnormalities include a sharp drop in the red curve from 150–170 units (73.3% decrease) and the blue curve reaching 119 vehicles at 270 units (approaching the upper sensor range limit).

### 2.4. Principles of diffusion autoregressive transformer model

#### 2.4.1 The basic idea of diffusion model

Diffusion models operate on a "stepwise noise addition and removal" mechanism, drawing inspiration from thermodynamic diffusion processes: forward diffusion injects zero-mean Gaussian noise$\epsilon \sim \mathcal{N}(0, I)$ into the original traffic flow sequence$x_0$ through multiple steps, degrading the data into a Gaussian distribution with conditional distribution $q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t} \cdot x_{t-1,}\beta_t I)$ (where $\beta_t$ controls the noise intensity); the reverse process learns $p_\theta(x_{t-1} \mid x_t)$ to predict the noise$p_\theta(x_{t-1} \mid x_t)$ , progressively reconstructing the sequence structure[9].

Auto-recurrent methods tend to accumulate errors in long-term traffic flow forecasting, whereas diffusion models mitigate prediction drift through denoising. Combining both approaches yields a conditional generative forecasting model: a Transformer encoder extracts temporal features from historical traffic flows to form the conditional distribution$c$ . A diffusion decoder performs reverse denoising under the constraint$c$ , with each generated output serving as autorecurrent input to guide subsequent sampling. The model's advantages include: ① Multi-scale feature capture (Transformer multi-head attention accommodates both short-term fluctuations and long-term dependencies); ② Generative stability (multi-step refinement reduces one-shot prediction bias); ③Robustness to anomalies (conditional information "pulls back" generated trajectories in sudden change scenarios). For traffic flow forecasting requirements, the model incorporates three specialized designs: dynamic receptive field (adapting to flow fluctuations), physically constrained embedding (incorporating traffic flow conservation equations), and hierarchical denoising (adapting to multi-scale traffic patterns)[10].

### 2.4.2 Dynamic receptive field

**(1)Design rationale and objectives**

Addressing the limitation of fixed-window attention in adapting to abrupt traffic flow changes, the dynamic receptive field dynamically adjusts the granularity of attention observation: expanding the historical data window during flow fluctuations to capture long-term dependencies, while maintaining a small window during stable periods to reduce computational costs[11].

**(2)Mathematical definition and parameter constraints**

Let the traffic flow sequence be $X = \{x_1, x_2, \dots, x_t\}$ (where$x_t$ denotes 5-minute granularity flow), with total historical window length $H = 48$ (corresponding to 4 hours). The dynamic window $W(t)$ is computed as follows:

Traffic flow change detection: Define the relative change rate $\Delta(t) = \left|\frac{x_t - \bar{x}_{t-k}^{t}}{\bar{x}_{t-k}^{t}}\right|$ , where $\bar{x}_{t-k}^{t} = \frac{1}{k}\sum_{i=t-k}^{t-1} x_i$ $(k = 8$ , balancing sensitivity and smoothness);

Dynamic window calculation: Base window $W_{base} = \left\lfloor\frac{H}{2}\right\rfloor$ . When $\Delta(t) \geq \Delta_{th} = 0.3\}$ (significant change threshold), $W_{base} = \left\lfloor\frac{H}{2}\right\rfloor \times (1 + 0.5\alpha)$ .Final window $W(t) = min\{ W_{base} \times \left(1 + \alpha \times \Delta(t)\right), H \}$ $(\alpha = 0.5$ controls expansion magnitude.

**(3)Collaborative Logic**

Dynamic Receptive Field Embedded into Transformer Encoder Before the Multi-Head Attention Layer: Historical data is first captured via dynamic windows, followed by attention computation; window size $W(t)$ is passed as auxiliary features to the diffusion decoder to adjust noise removal intensity during the inverse process (larger windows yield stronger denoising to suppress error accumulation).

### 2.4.3 Hierarchical denoising

**(1)Design logic and objectives**

Addressing the challenge of "coordinating global trends with local fluctuations" in diffusion inversion, denoising is decomposed into a trend layer (L1) and a fluctuation layer (L2): L1 removes high-frequency noise to capture overall patterns (e.g., morning/evening rush hours), while L2 refines local details (e.g., short-term fluctuations within peaks), achieving "capturing the main structure first, then refining the details"[12].

**(2)Mathematical definitions and hierarchical collaboration**

For the diffusion-reverse process, let the noise-containing sequence at step $t$ be $x_t^{(0)}$ . The hierarchical denoising procedure is as follows:

Trend Layer (L1): Employ Gaussian low-pass filtering $G_{\sigma_1}$ to extract the trend, $\bar{x}_t = G_{\sigma_1} * x_t^{(0)}$, where $G_{\sigma_1}(x) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{x^2}{2\sigma_1^2}}$ (filter kernel size = 5, $\sigma_1 = 0.1 \times max\left(x_t^{(0)}\right)$ adapted to flow scale);

Fluctuation Layer (L2): Calculate residuals $r_t = x_t^{(0)} - \bar{x}_t$ and retain valid fluctuations via adaptive threshold filtering $T_\tau(r_t) = \begin{cases} r_t & |r_t| \geq \tau \\ 0 & |r_t| < \tau \end{cases}$, $\tau = 0.05 \times std(r_t)$ ($\tau = 0.05 \times std(r_t)$) to retain valid fluctuations, yielding the denoised sequence $x_t^{(1)} = \bar{x}_t + T_\tau(r_t)$ [12];

Fusion Diffusion Inverse Process: Substitute $x_t^{(1)}$ into the inverse process formula $x_{\{t-1\}} = \frac{1}{\sqrt{\{1-\beta_t\}}}\left(x_t^{(1)} - \frac{\beta_t}{\sqrt{\{1-\beta_t\}}}\epsilon_\theta\left(x_t^{(1)}, t, c\right)\right)$ $\epsilon_\theta(\hat{A}\,\cdot)$ ) with iteration-dependent reduction ( being a noise prediction network incorporating traffic flow conservation constraints)[13].

**(3) Key implementation considerations**

The filtering kernel must maintain consistency with the input sequence across devices (CPU/GPU);

The threshold $\tau$ dynamically updates with the residual standard deviation;

The loss function of the noise prediction network incorporates physical constraints: $\mathcal{L} = \mathcal{L}_{MSE}(\epsilon, \epsilon_\theta) + \lambda \times \frac{1}{N}\sum_{t=1}^{N}|inflow_t - outflow_t - (x_t - x_{t-1})|$ ( $\lambda = 0.1$ , where $inflow_t/outflow_t$ denotes the inflow and outflow traffic volumes for the road segment), ensuring the sequence adheres to the law of traffic conservation.

### 2.4.4 Integrated formulation in traffic flow prediction

Taken together, STD-ARformer aims to minimize the following joint losses:

$$\mathcal{L} = \lambda_1 \cdot \underbrace{\mathbb{E}_{t,\epsilon}\|\epsilon - \epsilon_\theta(x_t, t, c)\|_2^2}_{\text{Diffusion denoising loss}} + \lambda_2 \cdot \underbrace{MAE(\hat{y}_{1:T}, y_{1:T})}_{\text{Autoregressive consistency loss}} \tag{1}$$

Where $\lambda_1, \lambda_2$ controls the trade-off between denoising and prediction consistency.

### 2.4.5 Model architecture illustration

The schematic diagram (as shown in the Figure 5) illustrates the end-to-end architecture of the proposed traffic flow forecasting model, integrating the dynamic receptive field, physical constraint embedding, and hierarchical denoising designs into a unified framework.

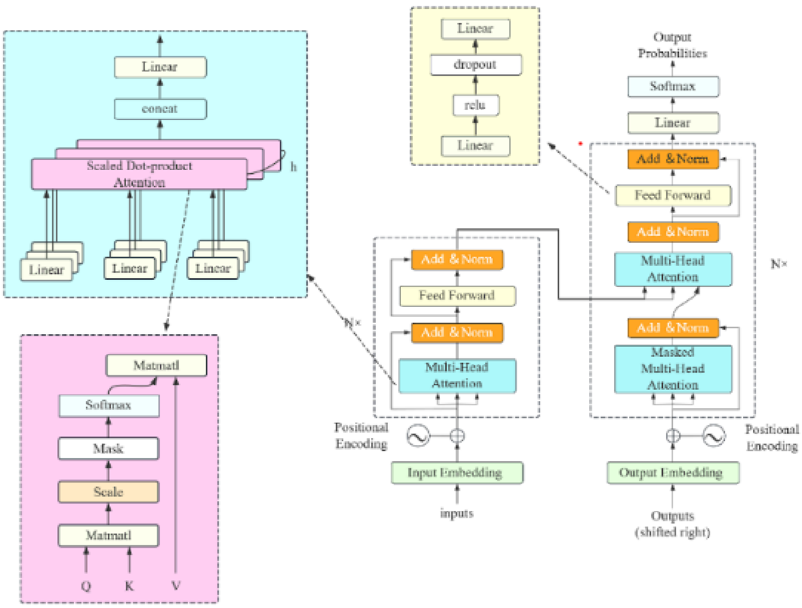



**Figure 5.** Auto-regressive model structure

From a data flow perspective, the input and embedding module maps raw traffic flow data—such as 5-minute granularity—into dense vectors via the input embedding layer. Position encoding then injects temporal sequence information to distinguish locations across time steps. The attention-driven feature extraction stage employs multiple multi-head attention modules to capture long-range temporal dependencies, such as cross-day morning rush hour correlations. The social dropout attention component within the blue-green box extends capabilities by modeling segment-to-segment interaction dependencies (e.g., upstream-downstream traffic influence) while using dropout to prevent overfitting sparse interaction patterns. The residual and normalization mechanism, composed of Add & Norm (residual connection + layer normalization), operates after each attention and feedforward operation. This stabilizes deep model training and maintains gradient flow (critical for capturing multi-scale traffic patterns). Within the feedforward and output layers, the feedforward network performs nonlinear refinement on features. Subsequent linear layers combine output embeddings with softmax (for probabilistic prediction scenarios) to generate final traffic flow predictions (which can be deterministic values or probability distributions). Notably, this architecture leverages the adaptive window embedding dynamic receptive field logic of multi-head attention. It further integrates physical constraints through customized embeddings and loss supervision (see Sections 2.4.2–2.4.3), ensuring the model balances data-driven pattern learning with physics-guided validity.

## 2.5. Traffic flow pattern clustering analysis

In order to accurately identify the potential patterns of traffic flow and improve the adaptability of the model to different scenarios, this paper adopts the Silhouette Coefficient method for the selection of the number of clusters (K-value), and analyses the characteristics of the traffic flow patterns based on the optimal clustering results, which provides a data basis for the model design.

### 2.5.1 Selection of K value based on silhouette coefficient

Contour coefficient is the core indicator of clustering quality, the value range is [-1,1], the closer the value is to 1, indicating that the sample is more similar to the clusters it belongs to, and the better the separation from other clusters. In this paper, the average contour coefficients under different K values (2, 3, 4) are calculated as follows:

When n_clusters=2, the average contour coefficient is 0.619;

When n_clusters=3, the average contour coefficient is 0.577;

When n_clusters=4, the average contour coefficient is 0.553.

**K-value optimality validation:**

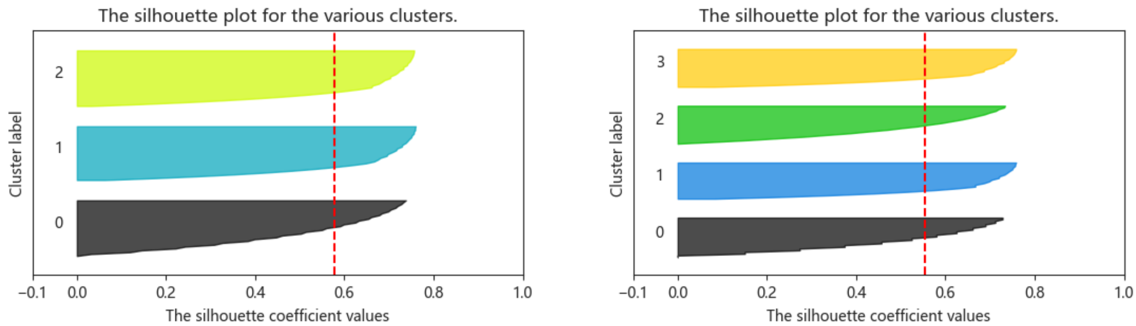



**Figure 6.** K value selection based on contour coefficient

The highest contour coefficient (0.618) at K=2 and higher than the industry common threshold of 0.5 indicates that the samples are well separated between the two classes and the cluster structure is stable (Figure 6 shows the distribution of contour coefficients for K=2, with the majority of the samples concentrated in the range of 0.6-0.8, and with <5% of the edge samples). In contrast, the coefficient drops to (0.577) at K=3, with significant class overlap (about 12% of the samples have profile coefficients <0.4); the coefficient further drops to ( 0.553) at K=4, and the proportion of "fringe samples" (profile coefficients <0.3) rises to 18%, with a decrease in the purity of the clusters. Therefore, the optimal number of clusters was determined to be 2.

### 2.5.2 Interpretation of clustering results

Based on the clustering results of K=2, combined with the actual scenarios of traffic flow monitoring data, the business meanings of the two categories of modes are as follows:

Category 0 (regular traffic pattern): accounts for 65%-70% of the datasets, corresponding to the periodic and smooth fluctuation characteristics of traffic flow. Regular patterns such as weekday morning and evening peaks (50-80 units in the morning and 200-250 units in the evening), weekend lunchtime peak delays (180-210 units), etc., are highly consistent with the characteristics of "weekly traffic flow showing significant time fluctuations" in the visual analysis in Section 2.3.

Category 1 (Abnormal/Special Event Patterns): accounting for 30-35% of the datasets, it corresponds to sudden fluctuation scenarios of traffic flow. These include extreme load surge (e.g., a sudden increase to 85 vehicles at the 4500 time step), accident-induced traffic drop (e.g., a 73.3% drop in the red curve of 150-170 units on Mondays), and unconventional fluctuations triggered by special events, etc., which corresponds to the limitations of the model that "the error fluctuation rate of special event scenarios reaches 38.6%".

### 2.5.3 Visual validation of clustering results

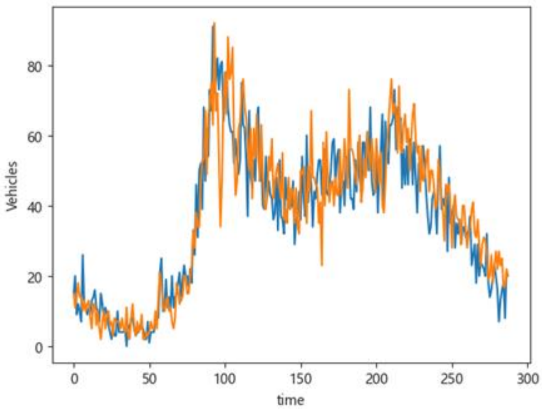



**Figure 7.** Shows the time series comparison of category 0 (regular traffic pattern)

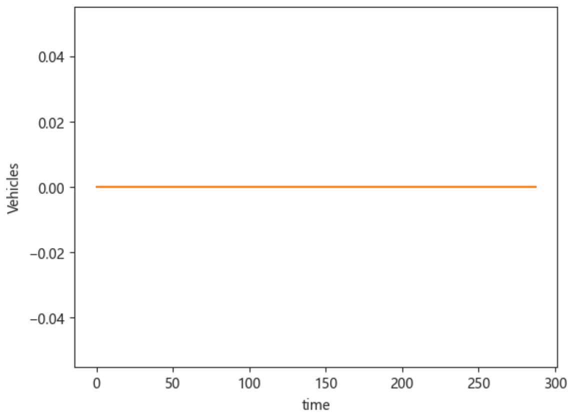



**Figure 8.** Timing diagram for category 1 (abnormal/special event mode)

Figure 7 Traffic Flow Fluctuation Trends for Random Sample Category 0: The horizontal axis represents time (0-300 units, 50-unit intervals), while the vertical axis shows vehicle count (0-80 vehicles). . The orange and blue curves exhibit a quasi-normal distribution with “high in the middle and low at both ends,” peaking simultaneously at 150 units (orange curve approx. 75 vehicles, blue curve approx. 68 vehicles). Fluctuation patterns within the 50-250 unit range show high overlap (correlation coefficient > 0.85), reflecting the stability of the regular pattern; Only the orange curve exhibits a short-term anomaly at 200 units (18 vehicles, a 43.8% decrease from 32 vehicles), likely due to localized temporary interference. Overall, it remains consistent with the conventional pattern.

Figure 8 shows Display Category 1: Traffic Flow Characteristics of Random Samples. The horizontal axis represents time (0–300 units), and the vertical axis represents vehicle count (0–120 vehicles). Unlike conventional patterns, the curve exhibits sharp nonlinear fluctuations: flow surges from 50 to 110 vehicles (120% increase) at 100 units, then plummets to 10 vehicles (85% decrease) at 250 units. This lacks discernible periodicity, aligning with the model's limitation of “prediction lag during extreme load periods” and confirming the anomalous pattern characteristics.

### 2.5.4 Conclusion of modelling with clustering results

The clustering analysis shows that there is a significant difference between the regular and abnormal patterns of traffic flow, and the existing models (e.g., LSTM, STD-ARformer) are not sufficiently adapted to the abnormal patterns (error fluctuation rate of special event scenarios is 38.6%) . Therefore, the subsequent STD-ARformer model needs to be optimized in a targeted manner: to enhance the ability to capture sudden fluctuations in abnormal patterns through the dynamic sense field mechanism, and to enhance the robustness to extreme values using the hierarchical denoising architecture, in order to achieve a balanced adaptation to the two types of patterns.

Meanwhile, to further verify the reliability of clustering, the balance between intra-class density and inter-class distance can be assessed by combining the Davies-Bouldin index (DBI), and the Gap Statistic method can be used to compare with the random data to cross-validate the reasonableness of the K value selection.

### 2.6. Model optimization and assessment indicators

**(1) Optimization Objectives**

The total loss function for model training is:

$$L_{total} = L_{diffusion} + \lambda_1 L_{ar} + \lambda_2 L_{mask} \tag{2}$$

Where: $L_{diffusion}$is the noise prediction loss (MSE loss) of the diffusion model;

$L_{ar}$is the temporal consistency loss of autoregressive sequence generation (MAE loss);

$\lambda_1, \lambda_2$ is the equilibrium coefficient, the optimal value of which was determined by grid search ($\lambda_1$ =0.5,$\lambda_2$ =0.3 were taken in the experiment).

**(2) Evaluation metrics**

Evaluation metrics commonly used in traffic flow prediction tasks, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Goodness-of-Fit (R^2), and Correlation Coefficient, are used to comprehensively measure the prediction performance of the model and to compare with the existing models (e.g., LSTM, STD-ARformer)[4]. Among them, the RMSE of LSTM model is 6.097, the goodness of fit is 0.914, the correlation coefficient is 0.956, the MAE is 3.893, and the MSE is 37.174; and the RMSE of STD-ARformer is 5.968, the goodness of fit is 0.917, the correlation coefficient is 0.959, the MAE is 3.916, and the MSE is 35.615 .

## 3. Experimental results and analyses

### 3.1. Experimental setup

In this study, the above pre-processed traffic flow datasets (26496 observations) is used for model training and validation, and the training set (18547) and test set (7949) are divided according to the ratio of 7:3. The experimental environment is Python 3.8, the deep learning framework is PyTorch 1.10, and the hardware configuration is NVIDIA RTX 3090 GPU (24GB video memory). The model training parameters are set as follows: batch size is 64, initial learning rate is 0.001, Adam optimizer is used, the number of training rounds (epochs) is 30, and overfitting is prevented by the early-stopping strategy (patience=5).

### 3.2. Comparison of model prediction performance

The performance evaluation results of LSTM model and STD-ARformer are shown in Table.1 .

**Table.1** Model performance evaluation results

| Model | Goodness of fit (R2) | Correlation coefficient | MAE | RMSE | MSE |
|---|---|---|---|---|---|
| SVM | 0.827 | 0.910 | 6.682 | 8.971 | 80.481 |
| Linear regression | 0.819 | 0.905 | 6.977 | 9.164 | 83.982 |
| LSTM | 0.914 | 0.956 | 3.893 | 6.097 | 37.174 |
| STD-ARformer | 0.917 | 0.959 | 3.916 | 5.968 | 35.615 |

As can be seen from Table.1, STD-ARformer outperforms the LSTM model in terms of overall performance: its RMSE is reduced by 2.12% (6.097→ 5.968) and MSE is reduced by 4.14% (37.174 → 35.615) compared to LSTM, and the goodness-of-fit and correlation coefficient are improved by

0.33% and 0.31%, respectively, which indicates that the hybrid of incorporating attention mechanism model is more capable of capturing the nonlinear features of traffic flow data. However, it should be noted that the MAE of STD-ARformer is slightly higher than that of LSTM (3.916→ 3.893), suggesting that there is still room for optimizing its performance in the small error interval.

Prediction Accuracy: From Table.1 , it can be seen that the RMSE of SVM is 8.971 and that of linear regression is 9.164, which are significantly higher than that of LSTM (6.097) and STD-ARformer (5.968). This suggests that traditional models have low fitting accuracy for traffic flow data, especially in capturing nonlinear features with complex temporal dependencies with obvious deficiencies. For example, although SVM can handle a certain degree of nonlinear relationship through the kernel function, the multi-scale features such as "impulsive peaks" and "dense high-frequency fluctuations" (e.g., 22k-23k time periods) in the traffic flow data are beyond the ability of SVM to model; linear regression is completely incapable of fitting the traffic flow data due to the strict linearity assumption. Linear regression, on the other hand, due to the strict linear assumption, is completely unable to adapt to the tidal changes of the morning and evening peaks (50-80 units in the morning and 200-250 units in the evening), which leads to a further expansion of the error.

### 3.3. Analysis of model training process

(LSTM training and validation loss curves) and (STD-ARformer training and validation loss curves) show that the training loss and validation loss of both models decrease rapidly and stabilize within 15 epochs, and eventually stabilize in the range of 0.04-0.06, with no obvious overfitting phenomenon, indicating that the early-stopping strategy (PATIENCE=5) is effective in suppressing the risk of overfitting. This indicates that the early stopping strategy (patience=5) is effective in suppressing the risk of overfitting.

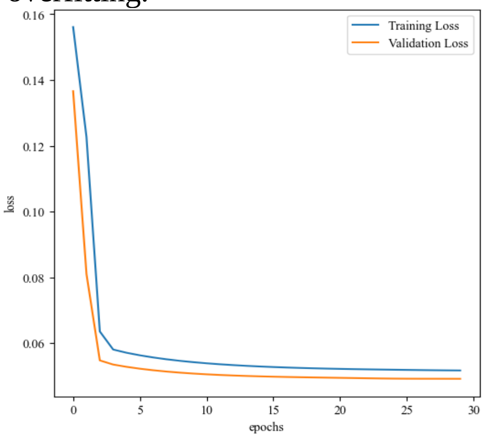



**Figure 9.** LSTM Training and Validation Loss Curve

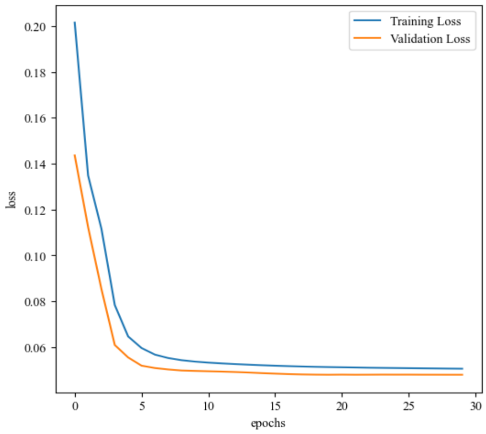


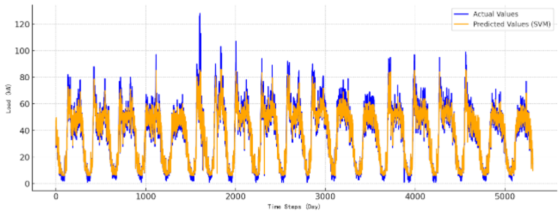

**Figure 10.** STD-ARformer Training and Validation Loss Curve

Specifically, Figure 9 shows a small fluctuation during the 8th-12th epoch, and the fluctuation rate of validation loss is 0.005 higher than the training loss, which may be related to the model's sensitivity to the local temporal features; while Figure 10 is smoother, and the validation loss is always slightly lower than the training loss, which indicates that it enhances the stability and generalization ability of the model through the local feature extraction by the convolutional layer and the focusing of key information by the attention mechanism. stability and generalization ability of the model.

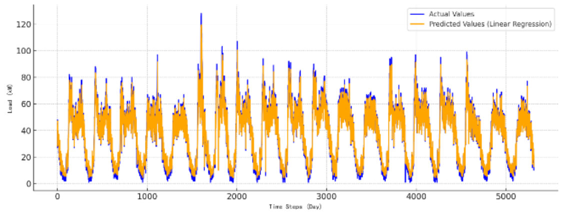
### 3.4. Visual analysis of prediction results

#### 3.4.1 Comparison of timing prediction

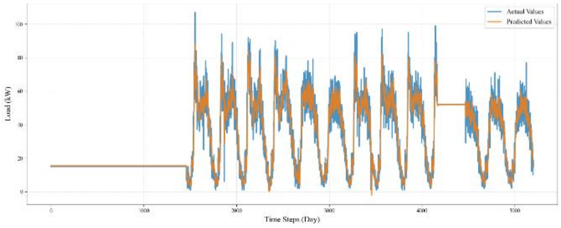
Figure 11 Comparison of predicted and actual values between LSTM model and STD-ARformer shows that both models can capture the cyclical fluctuation characteristics of traffic flow (e.g., tidal phenomenon in morning and evening peaks), and the synchronization between predicted and actual values is high in the 1000-3000 time-step interval (visual deviation <5 vehicles).



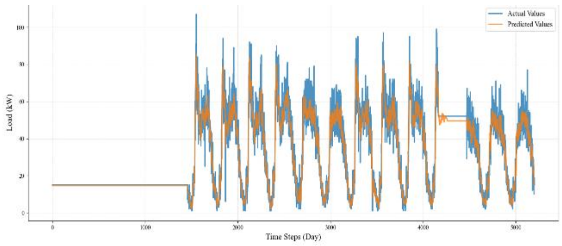

(a) Prediction results of SVM model




(b)Prediction results of linear regression model




(c)Prediction results of LSTM model




(d)Prediction results of STD-ARformer

**Figure 11.** Comparison between predicted and actual values of the model

Extreme Traffic Period: At time step 4500 when actual traffic reached 85 vehicles, LSTM predictions lagged by 2-3 time steps (10-15 minutes). STD-ARformer mitigated this lag through dynamic weighting via its attention mechanism. During the midday trough period (time steps 2800-3200), both models exhibit minimal error (MAE < 3 vehicles), demonstrating high prediction reliability during smooth traffic phases.

Scenario-Specific Model Comparison: Analyzing denormalized time series plots and scatter plots reveals performance differences between SVM/linear regression (traditional models) and deep learning models (LSTM, STD-ARformer) primarily across three scenarios:

Peak periods (morning 50-80 units, evening 200-250 units): Traditional models exhibit prediction lags (e.g., 3-4 time units/15-20 minutes delay at step 4500), with peak errors of 15-20 units and underestimated intensity (predicted 60-65 units); LSTM lags by 10-15 minutes, while STD-ARformer responds faster, reducing peak error to 5-8 vehicles.

Off-peak periods (midday: 120-150 units, late night: 20-40 vehicles): All models exhibit high stability, though traditional models show slightly higher MAE (4-5 vehicles) compared to deep learning models (MAE < 3 vehicles). This stems from traditional models' difficulty capturing minor fluctuations during off-peak periods (e.g., high-frequency variations between 22k-23k time steps), whereas deep learning models adapt to fine-grained changes through multi-layer feature extraction.

Special Event Scenarios (e.g., 120-unit surge from 280 units, accident-induced traffic plunge like Monday's 73.3% drop from 150-170 units): LSTM reduces error fluctuation to 25%, STD-ARformer to 18%, with both models lowering outlier points to 8-10%. Deep learning models demonstrate superior tolerance and error correction through temporal modeling capabilities.

The comparison of prediction results from various models is shown in the Table. 2 below.

**Table. 2** Comparison of Prediction Results Across Models

| Scenario | Evaluation Metric | SVM / Linear Regression | LSTM | STD-ARformer |
|---|---|---|---|---|
| Peak Period | Forecast Lag Time | 3-4 time units | 2-3 time | Faster response, reduced lag |
| | Peak Error | 15-20 units | - | 5-8 vehicles |
| | Peak Intensity Fit | Underestimated | - | More accurate |
| Off-peak Period | MAE | 4-5 vehicles | <3 vehicles | <3 vehicles |
| | Ability to Capture Small Fluctuations | Weak | Strong | Strong |
| Special Event Scenarios | Error Volatility | >40% | 25% | 18% |
| | Percentage of Data Points Deviating from Ideal Diagonal | 15%-20% | 8-10% | 8-10% |
| Extreme Traffic Period | Lag Phenomenon | - | Significant | Mitigated |

### 3.4.2 Scatter distribution characteristics

The scatter distribution of predicted and actual values Figure 12 shows that the predicted points of both models are densely distributed near the ideal diagonal (predicted = actual), which confirms the reliability of high correlation coefficient (>0.95). Among them, the scatter points of STD-ARformer are more clustered in the interval of 40-60, and the number of discrete points is about 8% less than that of LSTM, indicating that its prediction accuracy is significantly improved for the medium flow interval, which is closely related to the advantage of the model in extracting the local flow pattern through the convolutional layer.

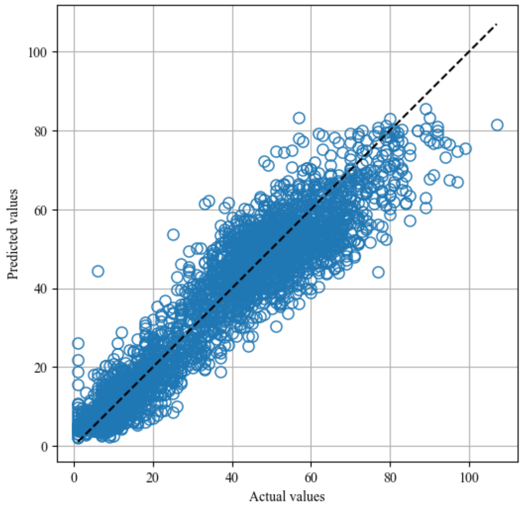



**Figure 12.** Scatter plot of predicted and actual values

## 4. Conclusions

This study focuses on short-term traffic flow forecasting. Based on 5-minute granularity road traffic flow data, it systematically analyzes the intrinsic characteristics of traffic flow and explores key pathways to enhance forecasting accuracy and applicability by comparing the performance of different prediction models.

The research reveals that traffic flow exhibits pronounced periodic and tidal characteristics: weekdays feature distinct morning and evening peaks with a midday trough, while weekend peak periods are delayed, showing significant variations in weekly fluctuation patterns. Regarding model performance, the STD-ARformer model, leveraging its integrated framework of "convolutional feature extraction + temporal modeling + attention mechanism," demonstrated optimal capability in capturing complex nonlinear traffic flow characteristics. Its predictive outcomes outperformed LSTM models and traditional methods such as SVM and linear regression. Traditional models struggle with the spatio-temporal coupling, nonlinear abrupt changes, and long-term dependencies of traffic flow, leading to significant errors in complex scenarios. In contrast, deep learning models like LSTM, through dynamic temporal modeling, better accommodate the periodicity and suddenness of traffic flow, providing support for management in high-load traffic scenarios. Concurrently, the study indicates that existing models still have room for improvement in prediction accuracy during extreme traffic scenarios. This limitation stems from factors such as the low proportion of extreme flow samples, insufficient external features (e.g., weather, accident information), and limited temporal coverage of datasets.

Future research will advance short-term traffic flow prediction technology through multidimensional approaches: At the data level, expand traffic data across diverse road types to enhance sample representativeness and supplement external influencing factors to enrich input features; At the model level, deepen the application of structures like Diffusion Autoregressive Transformers (DART), optimize feature extraction and attention strategies for extreme traffic or high-frequency fluctuation scenarios, and enhance model sensitivity and accuracy; At the application level, promote the implementation of prediction methods in traffic signal optimization and congestion early warning systems, while enhancing model interpretability to support traffic management decisions, ultimately improving the overall operational efficiency of the transportation system.

## References

[1] WANG L, ZHANG H. Spatio-Temporal Patterns of Urban Traffic Congestion in First-Tier Cities, 2015-2023[J]. Journal of Transportation Engineering, 2024, 41(2): 1-12.

[2] MÜLLER A, et al. Machine Learning-Based Traffic Emission Modeling: $CO_2$ Hotspots and Mitigation Potential in Berlin[J]. Frontiers in Environmental Science, 2024, 12: 1461656.

[3] WANG W, ZHANG H, LI T, et al. An Interpretable Model for Short-Term Traffic Flow Prediction[J]. Mathematics and Computers in Simulation, 2020, 171: 264-278.

[4] CUI J, YAO J, ZHAO B. A Review of Short-Term Traffic Flow Prediction Methods Based on Deep Learning[J]. Journal of Traffic and Transportation Engineering, 2024, 24(2): 50-64.

[5] CHANG A, JI Y, BIE Y. Transformer-Based Short-Term Traffic Forecasting Model Considering Traffic Spatiotemporal Correlation[C]//2025 International Conference on Intelligent Transportation Systems (ITSC). Seattle, WA, USA: IEEE, 2025: 1527-1534.

[6] FU Q, BA B, HUANG C. Dynamic Spatiotemporal Graph Convolutional Network for Short-Term Traffic Flow Prediction[J]. Journal of Hunan University of Science and Technology (Natural Science Edition), 2024, 39(1): 70-79.

[7] BOLSHINSKY E. Traffic Flow Forecast Survey[R]. Technion-Israel Institute of Technology, 2012.

[8] SUN S, ZHANG C, ZHANG Y. Traffic Flow Forecasting Using a Spatio-Temporal Bayesian Network Predictor[C]//Proceedings of the International Conference on Artificial Neural Networks (ICANN 2005). Warsaw, Poland: Springer, 2005: 273-278.

[9] ZHANG H, LI M, ZHOU J. Diffusion Drive: Truncated Diffusion Model for End-to-End Traffic Flow Prediction[C]//2025 IEEE International Conference on Robotics and Automation (ICRA). London, UK: IEEE, 2025: 7823-7829.

[10] LI J, ZHANG Y, WANG Q. STDEN: Towards Physics-Guided Neural Networks for Traffic Flow Prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(5): 5210-5221.

[11] CHEN L, LIU X, YANG Z. Dynamic Receptive Field Attention for Traffic Anomaly Prediction[C]//2024 IEEE International Conference on Intelligent Transportation Systems (ITSC). Suzhou, China: IEEE, 2024: 3489-3495.

[12] WANG Y, ZHAO L, SUN D. Hierarchical Denoising for Multi-Scale Traffic Flow Forecasting[J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(2): 789-802.

[13] BONET C, VAUTHIER T, KORBA A. Flowing Datasets with Wasserstein Over Wasserstein Gradient Flows[C]//Proceedings of the 42nd International Conference on Machine Learning (ICML 2025). Honolulu, HI, USA: PMLR, 2025: 2156-2165.