Preprints.org

Article

Not peer-reviewed version

State-Space and Multi-Scale Convolutional Generative Adversarial Network for Traffic Flow Forecasting

Wenxie Lin , Zhe Zhang , Yangzhen Zhao , Jinyu Zhang , Gang Ren *

Posted Date: 6 October 2025

doi: 10.20944/preprints202510.0402.v1

Keywords: state-space models; multi-scale convolution; generative adversarial networks; traffic flow forecasting

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

State-Space and Multi-Scale Convolutional Generative Adversarial Network for Traffic Flow Forecasting

Wenxie Lin [1,2], Zhe Zhang [1,2], Yangzhen Zhao [1,2], Jinyu Zhang [1], Gang Ren [1,2,*]

[1] School of Transportation, Southeast University, Jiulonghu Campus, Nanjing, 211100, China
[2] Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Nanjing, 211100, China
* Correspondence: rengang@seu.edu.cn

Abstract

Long-sequence traffic flow forecasting plays a crucial role in intelligent transportation systems. However, existing Transformer-based approaches face a quadratic complexity bottleneck in computation and are prone to over-smoothing in deep architectures. This results in overly averaged predictions that fail to capture the peaks and troughs of traffic flow. To address these issues, we propose a State-Space Generative Adversarial Network (SSGAN) with a state-space generator and a multi-scale convolutional discriminator. Specifically, a bidirectional Mamba-2 model is designed as the generator to leverage the linear complexity and efficient forecasting capability of state-space models for long-sequence modeling. Meanwhile, the discriminator incorporates a multi-scale convolutional structure to extract traffic features from the frequency domain, thereby capturing flow patterns across different scales, alleviating the over-smoothing issue, and enhancing discriminative ability. Through adversarial training, the model is able to better approximate the true distribution of traffic flow. Experiments conducted on four real-world public traffic flow datasets demonstrate that the proposed method outperforms baselines in both forecasting accuracy and computational efficiency.

Keywords: state-space models; multi-scale convolution; generative adversarial networks; traffic flow forecasting

1. Introduction

With the acceleration of urbanization and the continuous growth of traffic demand, accurate and efficient traffic flow forecasting has become one of the core issues in intelligent transportation systems. High-precision traffic forecasting not only helps alleviate congestion and optimize travel routes but also provides crucial decision-making support for urban planning and traffic management [1–5]. However, traffic flow data typically exhibit strong temporal dependencies and non-stationarity, which makes modeling particularly challenging in long-term forecasting tasks.

Deep learning methods have achieved remarkable progress in traffic forecasting. Among them, Transformer-based approaches [6] have attracted widespread attention due to their strong capability in temporal modeling [7–13]. Nevertheless, in long-sequence forecasting scenarios, the quadratic complexity of the Transformer attention mechanism results in substantial computational overhead, limiting the efficiency of such models on large-scale datasets [14]. Recently, state space models (SSMs) have emerged as a powerful alternative for long-sequence modeling. Unlike attention-based Transformers, SSMs model long-range dependencies through recursive hidden states and convolutional kernel expansion, achieving linear time complexity while effectively capturing long-term dependencies. In particular, a series of improved SSMs [15–18] have demonstrated superior

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

performance and strong long-sequence modeling ability in time series forecasting tasks. This provides a new perspective for traffic flow forecasting, namely leveraging state space models to enhance both predictive accuracy and computational efficiency.

In traffic flow forecasting tasks, traditional methods commonly adopt Mean Squared Error (MSE) or Mean Absolute Error (MAE) as optimization objectives. However, such regression losses inherently encourage models to produce "averaged" predictions to minimize overall error, leading to pronounced over-smoothing at peaks and abrupt changes [19]. Consequently, the predicted sequences fail to capture fine-grained temporal dynamics and overlook the true peaks and troughs in traffic flow, thereby degrading forecasting accuracy, as illustrated in Figure 1. Over-smoothing not only weakens the model's ability to detect sharp fluctuations in traffic flow but also limits the practical applicability of forecasting results in real-world traffic management scenarios.

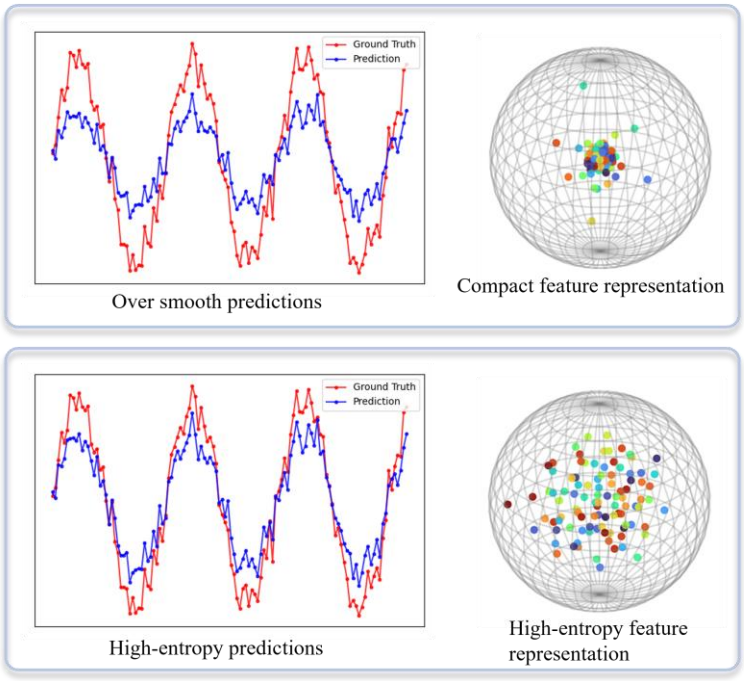

Figure 1. Over-smoothing problem.

To mitigate this issue, we introduce a Generative Adversarial Network (GAN) framework [20]. By training a discriminator to distinguish between predicted and real sequences, the generator is required not only to approximate the overall trend of traffic flow but also to learn the high-frequency fluctuations and abnormal patterns present in real data, thus avoiding averaged predictions. Compared with training solely based on MSE or MAE, adversarial learning more effectively drives the generator to recover the fine-grained dynamic features of traffic flow, alleviating the over-smoothing phenomenon. Furthermore, to enhance the discriminative capability of the discriminator, we design a multi-scale convolutional structure in the frequency domain. Frequency-domain convolution enables the extraction of both high- and low-frequency components of sequences and is particularly sensitive to local patterns such as spikes and abrupt changes. This design allows the discriminator to comprehensively perceive the differences between predicted and real sequences across multiple scales, thereby imposing stronger constraints on the generator and effectively mitigating over-smoothing in forecasting results.

Therefore, we propose a State-Space Generative Adversarial Network (SSGAN) with a state-space generator and a multi-scale convolutional discriminator. In the generator design, we introduce a bidirectional dual state-space model that leverages the linear complexity of state-space modeling to enhance the efficiency and stability of long-sequence forecasting. For the discriminator, we design a multi-scale convolutional structure capable of capturing traffic patterns across different scales and extracting features from the frequency domain. This design alleviates the over-smoothing problem

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

and strengthens the model's ability to discriminate complex traffic flow patterns. Through adversarial training, the proposed model learns to approximate the true distribution of traffic flow, thereby improving both the realism and accuracy of predictions. The main contributions of this work are summarized as follows:

- We propose a novel adversarial learning framework for long-sequence traffic flow forecasting that integrates state-space modeling with multi-scale convolution.
- We design a bidirectional dual state-space generator that maintains linear complexity while enhancing long-sequence forecasting performance.
- We introduce a multi-scale convolutional structure in the discriminator, enabling joint extraction of traffic features from the frequency domain to effectively mitigate over-smoothing and improve discriminative capability.

2. Related Work

2.1. Transformer-Based Methods

The Transformer architecture has demonstrated strong capabilities in modeling long-range dependencies within sequential data, where its self-attention mechanism effectively captures global dependencies across time steps. Compared with traditional Recurrent Neural Networks (RNNs), the parallel computation in Transformers significantly improves training efficiency. Particularly in traffic flow forecasting, the self-attention mechanism enables direct modeling of relationships between different time steps, thereby better capturing complex temporal patterns and periodic fluctuations. For instance, Zhou et al. [7] proposed FEDformer, which integrates seasonal-trend decomposition with frequency-domain enhancement to alleviate inefficiencies and global modeling limitations in long-sequence forecasting, resulting in significantly improved predictive accuracy. Fang et al. [13] introduced the PatchSTG framework, which employs irregular spatial partitioning and alternating attention to efficiently model spatial dependencies in large-scale traffic datasets, achieving both faster training and memory efficiency while maintaining competitive forecasting performance.

However, a major drawback of Transformers is their quadratic computational complexity. Specifically, the self-attention mechanism computes pairwise correlations among all time steps, leading to a complexity of $O(n^2)$, where n is the sequence length. This makes Transformers computationally expensive and memory-intensive when handling long sequences [21]. Furthermore, Transformers inherently do not account for the ordering of input sequences. Although positional encodings or segment embeddings are introduced to supplement temporal information, these mechanisms are insufficient to fully preserve sequential order, which can compromise the stability of temporal relationship modeling in long-sequence forecasting tasks [22].

2.2. Convolution-Based Methods

Convolutional Neural Networks (CNNs), as powerful feature extractors, demonstrate significant advantages in time series analysis. By applying local convolutional kernels, CNNs effectively capture local dependencies within sequences, enabling efficient identification of short-term patterns and fluctuations [23–27].

In traffic flow forecasting, CNNs can extract spatiotemporal features through multi-layer convolutions, with deeper layers progressively capturing higher-level representations. For example, Eldele et al. [24] proposed TSLANet, a lightweight and general convolutional model that leverages adaptive spectral blocks combined with Fourier analysis to enhance feature representation and mitigate noise interference. Additionally, TSLANet introduces an interaction convolution module and self-supervised learning to improve the decoding of complex temporal patterns. Lee et al. [27] introduced CASA, which combines CNN autoencoders with a scoring attention mechanism, effectively reducing both time and computational costs for spatiotemporal sequence modeling.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

Nevertheless, CNNs face inherent limitations due to their restricted receptive field. The receptive field is determined by kernel size and network depth, making CNNs less effective in modeling long-range dependencies within long sequences. When temporal dependencies span extended horizons, convolutional layers often fail to capture these relationships adequately. Moreover, as CNNs become deeper, they risk losing global information, which hampers their ability to model long-term trends and periodic variations in time series data.

2.3. State Space Model-Based Methods

In recent years, SSMs have attracted increasing attention in time series modeling tasks. Unlike Transformers and CNNs, SSMs leverage dynamic update mechanisms and selective processing of hidden states to effectively capture temporal dependencies in long sequences. In the context of traffic flow forecasting, SSMs are capable of maintaining long-term memory while automatically filtering out irrelevant information during updates, thereby enhancing their ability to handle long sequential data [28–32]. For instance, Lin et al. [29] combined Mamba with Graph Convolutional Networks (GCNs) for long-sequence traffic forecasting, where GCNs capture the spatial dependencies of road networks while bidirectional Mamba models long-range temporal dependencies.

Compared with CNNs, SSMs possess a global receptive field, enabling them to effectively model long-range dependencies without being constrained by the locality of convolutional kernels. While CNNs excel at extracting local features and modeling short-term dependencies, their limited receptive fields make it difficult to capture global trends in long sequences. In contrast, SSMs iteratively update hidden states, allowing them to effectively model long-distance temporal dependencies and overcome the limitations of convolution-based approaches.

Compared with Transformers, SSMs also offer advantages in terms of computational efficiency. Although the self-attention mechanism in Transformers captures global dependencies, its quadratic complexity leads to rapidly increasing computational and memory costs as sequence length grows. SSMs, by contrast, employ optimized algorithms and selective state update mechanisms to reduce time complexity to linear, thereby significantly improving efficiency in processing long-sequence data.

2.4. Generative Adversarial Network-Based Methods

Conventional forecasting approaches typically optimize point-wise regression losses. Although this strategy can reduce overall error, it often results in overly smoothed predictions that fail to capture abrupt fluctuations in traffic flow. In contrast, GANs leverage the adversarial training mechanism between the generator and discriminator, compelling the model not only to approximate the true traffic values numerically but also to capture the complex characteristics of the data distribution. This distribution-learning capability endows GANs with significant advantages in modeling the diversity and uncertainty inherent in traffic flow [33–35]. For instance, Khaled et al. [34] combined GANs with multi-graph convolutional networks to model complex spatial relationships from multiple perspectives, while integrating gated recurrent unit and self-attention mechanisms to capture dynamic temporal dependencies. Moreover, through the feedback provided by the discriminator, the generator is encouraged to produce sequences that more closely resemble real traffic data, effectively mitigating the over-smoothing problem and enhancing both the realism and robustness of the predictions.

Therefore, by combining the advantages of state-space models in long-sequence modeling with the feature extraction capabilities of CNNs, we design a generator that incorporates a bidirectional Mamba-2 model. By leveraging the linear-time complexity of state-space models, this design effectively improves both the efficiency and stability of long-sequence modeling. The bidirectional modeling enables the generator to simultaneously consider historical and future information, thereby enhancing its ability to capture complex traffic flow patterns. In the discriminator, we design a frequency-domain multi-scale convolutional architecture capable of extracting traffic patterns at multiple scales. The multi-scale convolution enhances the discriminator's ability to recognize

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

complex traffic patterns while mitigating the over-smoothing issue, enabling the model to better capture both local variations and global trends in traffic flow. Through adversarial training, the proposed framework effectively learns the true distribution of traffic flow, thereby improving both the accuracy and realism of the predictions.

3. Preliminaries

3.1. Traffic Flow Forecasting

Traffic flow forecasting aims to predict the traffic volume at multiple detector points in the road network at future time steps, based on historical traffic data. We use historical data from S time steps, denoted as $X = \{x_1, \ldots, x_S\} \in \mathbb{R}^{S \times N}$, to predict the traffic flow for the next T time steps, denoted as $\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_T\} \in \mathbb{R}^{T \times N}$, where N represents the number of detectors.

3.2. State Space Models

The core idea of SSMs is to introduce a latent low-dimensional state vector that maps the dynamic changes of the observation sequence into the state transition equation and the observation equation for description. Formally, a state space model consists of the following two components:

$$h_t' = Ah_t + Bx_t \tag{1}$$

$$y_t = Ch_t \tag{2}$$

where $A \in \mathbb{R}^{D \times D}$ is the state transition matrix, and $B \in \mathbb{R}^{D \times 1}, C \in \mathbb{R}^{1 \times D}$ are the mapping matrices. The input sequence $x_t \in \mathbb{R}$ is processed through calculations with the mapping matrix and the state transition matrix to obtain the output sequence $y_t \in \mathbb{R}$.

The discretized state space model is as follows:

$$h_t = \bar{A}h_{t-1} + \bar{B}x_t \tag{3}$$

$$\bar{A} = \exp(\Delta A), \bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B \tag{4}$$

$$y_t = Ch_t \tag{5}$$

$$y_t = \sum_{k=0}^{s-1} K_k x_{t-k}, K_k = CA^k B \tag{6}$$

where S is the length of the sequence. The discretized state space model can be computed using the convolution method in equation (6).

In traditional SSMs, directly computing the matrix K_k has high computational complexity, making it difficult to apply for modeling long sequences. Gu et al. introduced the High-Order Polynomial Projection Operator (HiPPO) matrix [15] to parameterize A in a special way. The HiPPO matrix efficiently handles long sequence data and prevents the vanishing or exploding gradients commonly encountered in traditional RNNs. To reduce the high computational cost associated with the HiPPO matrix, Gu et al. proposed the Structured State Space Model (S4) [16], which approximates the HiPPO matrix in a low-rank form via matrix decomposition, thereby improving computational efficiency. However, in both traditional SSMs and the S4 model, the mapping matrix and state transition matrix are updated only during training. Once the model is trained, both matrices are fixed, preventing content-based inference and limiting the model's adaptability.

To address this, Gu and Dao [17] proposed replacing the parameter matrix (Δ, B, C) with a data-driven parameterized matrix. This matrix changes with the input, allowing the model to

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

selectively propagate or forget information along the sequence dimension. Although this data-driven parameterized matrix cannot be computed using convolution, the Mamba model employs hardware-aware parallel algorithms to enhance the efficiency of recursive operations. Recently, Dao and Gu introduced the Mamba-2 model [18], which reveals the theoretical connection between state space models and attention mechanisms through the decomposition of structured semi-separable matrices. Based on this framework, the Mamba-2 model was designed to improve the computational efficiency of state space models.

3.3. Generative Adversarial Network

The GAN framework consists of a generator G and a discriminator D, as shown in Figure 2. The generator predicts future traffic flow data $\hat{Y} \in \mathbb{R}^{T\times N}$ based on historical traffic flow data $X \in \mathbb{R}^{S\times N}$. The discriminator is used to distinguish between the real traffic flow $Y \in \mathbb{R}^{T\times N}$ and the generated prediction $\hat{Y} \in \mathbb{R}^{T\times N}$.

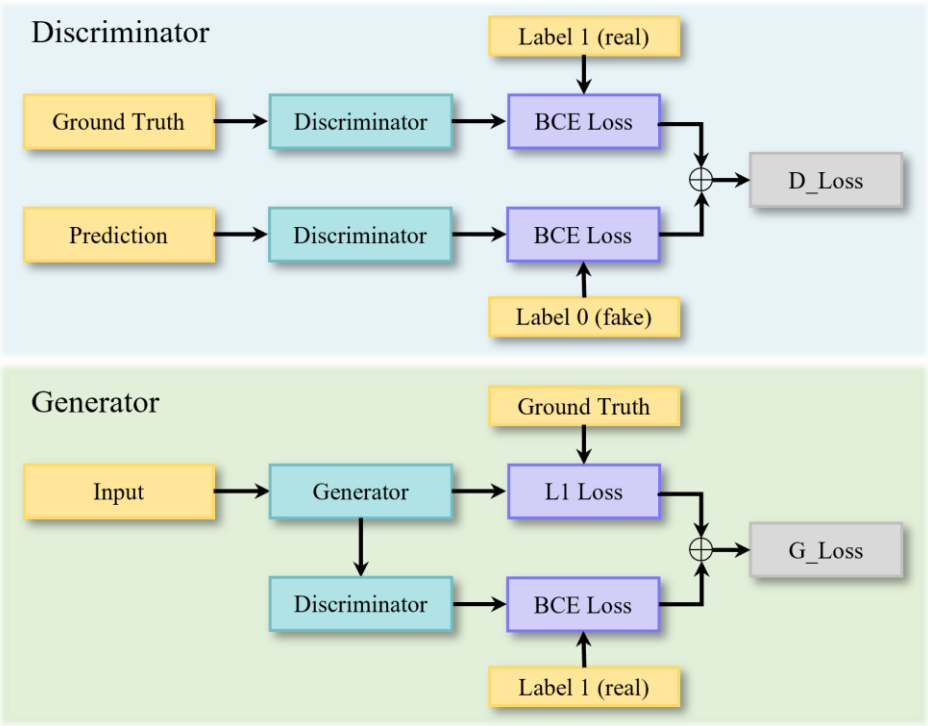

Figure 2. The Structure of GAN.

During the discriminator training phase, the generator first creates a counterfeit prediction sequence without gradient updates. Subsequently, the discriminator distinguishes between the real label sequence and the generated sequence, outputting the probability of authenticity. To enhance the discriminator's ability to distinguish, we use Binary Cross-Entropy (BCE) as the optimization objective. Specifically, the discriminator aims to maximize the recognition probability of real samples while minimizing the recognition probability of generated samples. Its loss function consists of two parts: the loss for real samples and the loss for generated samples:

$$L_D = L_{real} + L_{fake} = L_{BCE}\left(D(Y),1\right) + L_{BCE}\left(D\left(\hat{Y}\right),0\right) \tag{7}$$

The BCE loss function is defined as:

$$L_{BCE}\left(y,\hat{y}\right) = -\left[y\log\left(\hat{y}\right) + \left(1-y\right)\log\left(1-\hat{y}\right)\right] \tag{8}$$

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

where $y \in \{0,1\}$ represents the real label, and $\hat{y} \in (0,1)$ represents the predicted probability, which can be either $D(Y)$ or $D(\hat{Y})$.

During the training phase of the generator, its goal is not only to minimize the prediction error with respect to the real sequence but also to deceive the discriminator, thereby generating sequences that are closer to the real distribution. Specifically, the generator's loss function consists of two parts: first, the prediction loss, which ensures the consistency between the generated sequence and the real sequence in terms of numerical values; second, the adversarial loss, which measures the discriminator's judgment on the generated sequence. The overall loss is defined as:

$$L_G = L_{pred} + \lambda L_{adv} = L_{MAE}\left(G(X), Y\right) + \lambda L_{BCE}\left(D(G(X)), 1\right) \tag{9}$$

where L_{pred} represents the prediction error loss function, for which we use L1 Loss, and L_{adv} denotes the adversarial loss. $\lambda \in [10^{-5}, 1.0]$ is a weight factor used to balance the trade-off between prediction accuracy and generation quality. By minimizing L_G, the generator ensures that the predicted values are close to the real values while improving the distribution fitting ability of the generated sequence.

Through alternating training of the discriminator and the generator, the discriminator continuously enhances its ability to distinguish between real and generated samples, while the generator gradually learns to produce more realistic and reliable traffic flow prediction sequences. This process effectively improves the model's prediction performance in complex traffic environments.

4. Methodology

The structure of the proposed SSGAN model consists of a generator and a discriminator. These two components interact through an adversarial training objective to mutually enhance each other: the generator is responsible for producing future traffic flow sequences based on historical traffic data and temporal features, while the discriminator aims to distinguish whether an input sequence corresponds to real traffic observations or synthetic predictions generated by the generator. Through this adversarial mechanism, the generator continuously improves the realism and plausibility of its outputs, whereas the discriminator progressively strengthens its discriminative capability, jointly driving the overall model to approximate the true distribution of traffic flow. To provide a clearer exposition of the model design, the following sections detail the architectures and optimization objectives of the generator and discriminator, respectively.

4.1. Generator

The overall architecture of the generator in the proposed SSGAN model is illustrated in Figure 3. First, the input sequence is standardized to eliminate differences in numerical ranges across traffic flow series and thereby improve training stability. The standardized input is then passed through an embedding layer, which maps both traffic flow and temporal features into a high-dimensional representation space.

Subsequently, the embedded representations are fed into a Mamba-2 Block, which consists of bidirectional Mamba-2 units, feed-forward layer, and linear layers, designed to efficiently capture long-term temporal dependencies. In this process, a residual connection is introduced by adding the output of the embedding layer to the output of the Mamba-2 Block, thereby enhancing feature transmission and stabilizing gradient propagation.

On this basis, the output is further transformed through a feed-forward layer for nonlinear feature learning, followed by an interactive convolution block that extracts local spatial features and fuses information from multiple receptive fields. Finally, the fused representations are processed by

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

layer normalization and linear projection to map the high-dimensional features into the target prediction space, producing the final traffic flow forecasting results.

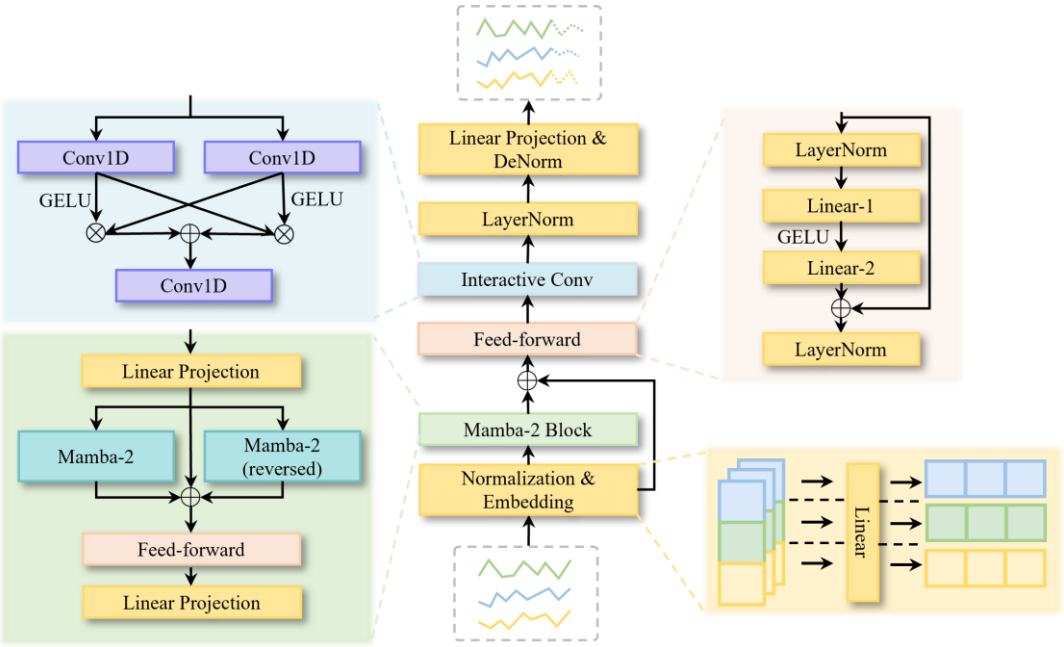

Figure 3. Framework of generator.

Embedding Layer. In traffic flow forecasting tasks, the raw input data are typically represented as a three-dimensional tensor $X \in \mathbb{R}^{B\times N\times S}$, where B denotes the batch size, N represents the number of detectors in the road network, and S is the length of the time series. This input directly contains traffic flow values; however, due to its low dimensionality, it is insufficient to effectively capture complex temporal and spatial dependencies. To address this limitation, we introduce an embedding layer at the front end of the model, which projects the input features from the original space into a high-dimensional representation space through a linear transformation. Specifically, a fully connected layer is employed to perform the following mapping:

$$X = \text{Dropout}\left(XW + b\right) \tag{10}$$

where $W \in \mathbb{R}^{T\times d}$ denotes the learnable projection matrix, d is the embedding dimension, and b represents the bias term.

Through this embedding layer, the raw traffic sequences are projected into a high-dimensional space, which not only strengthens the feature representation capacity but also provides a unified and trainable input representation for the subsequent spatiotemporal modeling modules.

Mamba-2 Module. The embedded representations are fed into the Mamba-2 block, which is built upon Structured State Space Duality (SSD). This design enables the model to capture both local and long-term dependencies with approximately linear computational complexity.

We consider a time-varying discrete SSM defined as follows:

$$h_t = \exp\left(A_t\right) \odot h_{t-1} + B_t x_t \tag{11}$$

$$y_t = C_t^{\mathrm{T}} h_t \tag{12}$$

where $h_t \in \mathbb{R}^{B\times H\times N}$ denotes the hidden state, H represents the number of heads, and N is the state dimension; $x_t, y_t \in \mathbb{R}^{B\times H\times P}$, P refers to the dimensionality of each head; $A_t \in \mathbb{R}^{B\times H}$; $B_t, C_t \in \mathbb{R}^{B\times H\times N}$ correspond to the input and output projections, respectively, $\odot$ denotes the element-wise product.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

Direct temporal recursion suffers from low parallelism and numerical instability when modeling long sequences. To address this issue, Mamba-2 reformulates long-sequence recurrence into a combination of intra-block parallelism and inter-block lightweight recurrence through block partitioning and low-rank decomposition [18]. Specifically, a sequence of length L is divided into $C = L/l$ blocks, and the input is rearranged into $x \in \mathbb{R}^{B\times C\times l\times H\times P}$. Within each block c, a decay kernel (a lower-triangular structure) is defined as follows:

$$L_c[\ell, s] = \exp\left(\sum_{k=s}^{\ell-1} A_{c,k}\right), \ell > s \tag{13}$$

where ℓ, s indicates the position within the block, and $A_{c,k}$ represents the decay parameter at position k of block c.

The computation of the diagonal block is formulated as follows, which corresponds to block-diagonal operations that can be executed fully in parallel within each block:

$$Y_{c,\ell}^{diag} = \sum_{s<\ell} C_{c,\ell} \odot B_{c,s} \odot L_c[\ell, s] \odot x_{c,s} \tag{14}$$

The compressed state at the end of each block is then computed as:

$$states[c] = \sum_{\ell=1}^{l} B_{c,\ell} \odot \exp\left(\sum_{k=\ell+1}^{l} A_{c,k}\right) \odot x_{c,\ell} \tag{15}$$

where $states[c] \in \mathbb{R}^{B\times H\times N}$ aggregates the inputs at all positions ℓ within block c, decayed to the end of the block. $states[0]$ serves as an optional initial sequence state, which defaults to zero if not specified.

The inter-block recurrence is computed as follows:

$$\tilde{h}_c = \sum_{k=1}^{c-1} \exp\left(\sum_{\ell=1}^{l} A_{k,\ell}\right) \odot states[k] \tag{16}$$

The computation from state to output is defined as:

$$Y_{c,\ell}^{off} = C_{c,\ell} \odot \tilde{h}_c \odot \exp\left(\sum_{k=1}^{\ell} A_{c,k}\right) \tag{17}$$

Finally, the overall output is obtained as the sum of two components:

$$Y = Y^{diag} + Y^{off} \tag{18}$$

Through the above mechanism, Mamba-2 achieves linear complexity when modeling long sequences. However, if relying solely on unidirectional accumulation, information transmission may suffer from latency. To address this issue, we employ a bidirectional Mamba-2 structure in the generator, where the forward and backward state evolutions are computed separately and subsequently fused along the feature dimension. By introducing this bidirectional design, the model can simultaneously account for the influence propagated from the past to the present as well as the reverse constraints traced back from the future to the present, thereby capturing more comprehensive global semantics. This design enables the model to more effectively learn bidirectional temporal dependencies in traffic flow, offering significant advantages for long-horizon forecasting and the modeling of complex traffic patterns.

Feed-Forward Layer. This layer is designed to perform nonlinear transformation and feature recombination on the high-dimensional representations of nodes, thereby enhancing the expressive power of the model. The structure consists of two linear transformations, with a nonlinear activation function applied in between to introduce nonlinearity:

$$X_1 = \text{Dropout}\left(\sigma\left(\text{LayerNorm}(X)\cdot W_1 + b_1\right)\right) \tag{19}$$

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

$$X_2 = \text{Dropout}\left(X_1 \cdot W_2 + b_2\right) \tag{20}$$

where $W_1 \in \mathbb{R}^{d\times 4d}$ and $W_2 \in \mathbb{R}^{4d\times d}$ denote the linear transformation matrices for dimensionality expansion and reduction, respectively; σ is the GELU activation function; and LayerNorm represents the layer normalization operation.

Finally, residual connection and normalization are applied as follows:

$$X = \text{LayerNorm}\left(X + X_2\right) \tag{21}$$

Interactive Convolution Block. In traffic flow forecasting, spatial correlations among nodes also play a crucial role. To this end, we adopt the interactive convolution block [24], which is specifically designed to capture spatial features across traffic network nodes. Given an input tensor $X \in \mathbb{R}^{B\times S\times N}$, the temporal dimension is treated as the convolutional channel, while the node dimension is regarded as the convolutional sequence, thereby enabling convolution operations along the spatial dimension of nodes.

In terms of structure, the interactive convolution block first applies two one-dimensional convolutions with kernel sizes of 1 and 3, respectively, to extract spatial features under different receptive fields:

$$X_1 = \text{Conv1d}_1\left(X\right) \tag{22}$$

$$X_2 = \text{Conv1d}_3\left(X\right) \tag{23}$$

To enhance the interaction between multi-scale convolutional features, an interaction mechanism is further introduced:

$$X_3 = X_1 \odot \text{Dropout}\left(\sigma\left(X_2\right)\right) + X_2 \odot \text{Dropout}\left(\sigma\left(X_1\right)\right) \tag{24}$$

where σ is the GELU activation function. This mechanism effectively fuses multi-scale spatial features, thereby enabling the model to better capture the complex dependencies among nodes in the road network.

Subsequently, the fused representations are processed through layer normalization and a one-dimensional convolution to project them back to the original channel dimension:

$$Y = \text{Conv1d}_1\left(\text{LayerNorm}\left(X_3\right)\right) \tag{25}$$

Finally, a linear layer is applied to adjust the output dimensionality, followed by de-normalization, yielding the predicted traffic flow values.

4.2. Discriminator

The objective of the discriminator is to distinguish between real traffic flow data and the predictions generated by the generator. Unlike conventional discriminators that operate directly on time-domain sequences, we propose a frequency-domain-based discriminator to enhance its ability to capture periodic patterns in traffic flow. In real-world operations, traffic flow often exhibits clear periodic patterns (e.g., morning and evening peaks), which are difficult to explicitly capture in the time domain. By mapping the data to the frequency domain via the Fourier transform, these periodic characteristics can be more clearly represented, enabling the discriminator to more effectively distinguish real samples from generated ones.

Specifically, given an input tensor $X \in \mathbb{R}^{B\times N\times T}$, the temporal dimension is first transformed using the Fast Fourier Transform (FFT):

$$X_{fft} = \text{FFT}\left(X\right) \tag{26}$$

from which the real part X_{real} and the imaginary part X_{imag} are extracted. These components are then concatenated along the channel dimension to form the composite input feature:

$$X_{concat} = \text{Concat}\left(X_{real}, X_{imag}\right) \in \mathbb{R}^{B\times 2N\times T} \tag{27}$$

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

Building on this, to further enhance the model's ability to extract features at different temporal scales, a Multi-Scale Convolution module is introduced, as illustrated in Figure 4. This module applies one-dimensional convolutions with kernel sizes of 3, 5, and 7 to extract features at multiple receptive fields:

$$H_k = \sigma\left(\text{Conv1d}_k\left(X_{concat}\right)\right), k \in \{3,5,7\} \quad (28)$$

where σ is the LeakyReLU activation function. The multi-scale features are then concatenated to obtain a combined representation:

$$H = \text{Concat}\left(H_3, H_5, H_7\right) \quad (29)$$

To remove redundancy and restore the original number of node channels, a one-dimensional convolution with a kernel size of 1 is applied for feature compression:

$$Z = \text{Conv1d}_1\left(H\right) \quad (30)$$

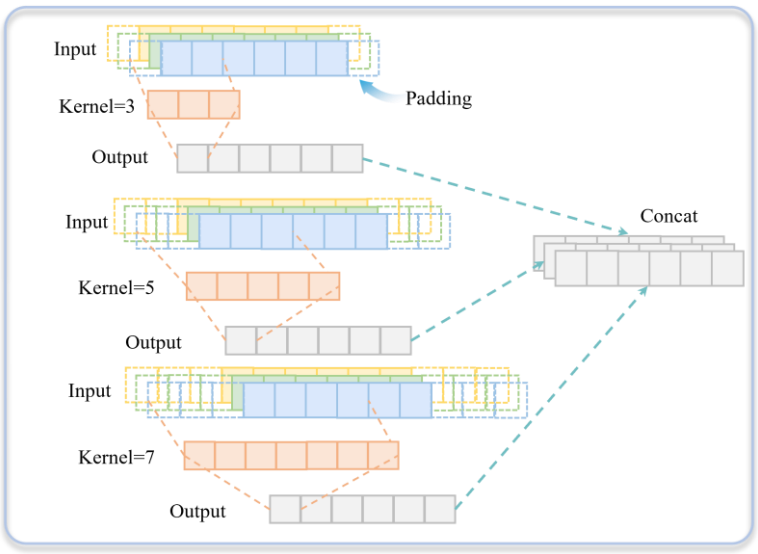

Figure 4. Framework of discriminator.

5. Experiments

5.1. Datasets and Baselines

Experiments were conducted on four publicly available traffic datasets to evaluate the performance of the proposed SSGAN model. The PEMS datasets [36] collect traffic flow data from various regions of California highways and include four subsets: PEMS03, PEMS04, PEMS07, and PEMS08. These datasets cover selected months between 2017 and 2018. The statistical details of the datasets are summarized in Table 1, where Nodes denotes the number of detectors in each dataset, Dataset Size indicates the split proportions for training, validation, and testing sets, and Frequency represents the sampling interval. These datasets provide a comprehensive benchmark for assessing the generalization ability and modeling effectiveness of the proposed model across different traffic prediction scenarios.

Table 1. Statistical description of datasets.

Dataset	Nodes	Time Steps	Dataset size	Frequency
PEMS03	358	26209	6:2:2	5 min
PEMS04	307	16992	6:2:2	5 min
PEMS07	883	28224	6:2:2	5 min
PEMS08	170	17856	6:2:2	5 min

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

We selected seven baseline models, along with GL-SSD, for comparison:

- Mamba-based model. S-Mamba is built upon the Mamba model and efficiently captures sequential dependencies through linear encoding and feed-forward networks [28].
- Transformer-based models. iTransformer applies attention mechanisms along the variable dimension by transposing the input, modeling multivariate correlations while leveraging feed-forward networks for nonlinear representation learning [10]. Crossformer employs a dimension-segment-wise embedding to preserve temporal and variable information and integrates a two-stage attention mechanism to simultaneously model cross-time and cross-dimension dependencies [8].
- Linear models. TiDE combines an MLP-based encoder–decoder architecture with nonlinear mapping and covariate handling to achieve high-accuracy long-term time series forecasting with fast inference [37]. RLinear efficiently captures periodic features using linear mapping, invertible normalization, and channel-independent mechanisms [38]. DLinear, with its simple linear structure, reveals the temporal information loss issue inherent in Transformer-based models for long-term forecasting [22].
- Convolution-based model. TimesNet transforms one-dimensional time series into a set of two-dimensional tensors based on multiple periodicities, decomposing complex temporal variations into intra-period and inter-period changes, which are then modeled efficiently using 2D convolution kernels [39].

5.2. Forecasting Results

All prediction experiments were conducted using the PyTorch framework on an NVIDIA GeForce RTX 4090 GPU. The Adam optimizer was employed, with the generator parameters and trainable adversarial weights optimized jointly, while the discriminator was optimized separately. A dynamic learning rate schedule was applied, decaying according to the training epoch. The loss function was the L1 Loss, the batch size was set to 32, and an early stopping strategy with a patience of 3 epochs was adopted. For all baseline models, hyperparameters were configured according to the original papers or official implementations. The input sequence length for all prediction models was set to 96 time steps, and the prediction horizons were set to $\{12, 24, 48, 96\}$ time steps.

The prediction results across all forecast horizons for the PEMS datasets are presented in Table 2, where red indicates the best-performing results, and avg represents the average over four prediction horizons. The proposed SSGAN achieves the best performance in most settings for both MSE and MAE, and consistently outperforms other models in the avg metrics across all datasets, demonstrating its robustness and consistency across different horizons and datasets. As the prediction horizon increases, errors for all methods rise; however, the error growth of SSGAN is notably smaller. This advantage is particularly pronounced in long-horizon settings (e.g., 48 or 96 time steps), indicating the model's superior ability to capture long-term dependencies and complex spatiotemporal patterns.

Table 2. Prediction results on the PEMS datasets.

Models		SSGAN (ours)		S-Mamba (2025)		iTransformer (2024)		RLinear (2023)		Crossformer (2023)		TiDE (2023)		TimesNet (2023)		DLinear (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	12	0.072	0.174	0.065	0.169	0.071	0.174	0.126	0.236	0.090	0.203	0.178	0.305	0.085	0.192	0.122	0.243
	24	0.086	0.191	0.087	0.196	0.093	0.201	0.246	0.334	0.121	0.240	0.257	0.371	0.118	0.223	0.201	0.317
	48	0.120	0.225	0.133	0.243	0.125	0.236	0.551	0.529	0.202	0.317	0.379	0.463	0.155	0.260	0.333	0.425
	96	0.147	0.248	0.201	0.305	0.164	0.275	1.057	0.787	0.262	0.367	0.490	0.539	0.228	0.317	0.457	0.515
	Avg	0.106	0.210	0.122	0.228	0.113	0.221	0.495	0.472	0.169	0.281	0.326	0.419	0.147	0.248	0.278	0.375
PEMS0	12	0.070	0.169	0.076	0.180	0.078	0.183	0.138	0.252	0.098	0.218	0.219	0.340	0.087	0.195	0.148	0.272
	24	0.079	0.179	0.084	0.193	0.095	0.205	0.258	0.348	0.131	0.256	0.292	0.398	0.103	0.215	0.224	0.340

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

	48	0.091	0.193	0.115	0.224	0.120	0.233	0.572	0.544	0.205	0.326	0.409	0.478	0.136	0.250	0.355	0.437
	96	0.103	0.207	0.137	0.248	0.150	0.262	1.137	0.820	0.402	0.457	0.492	0.532	0.190	0.303	0.452	0.504
	Avg	0.086	0.187	0.103	0.211	0.111	0.221	0.526	0.491	0.209	0.314	0.353	0.437	0.129	0.241	0.295	0.388
PEMS07	12	0.068	0.152	0.063	0.159	0.067	0.165	0.118	0.235	0.094	0.200	0.173	0.304	0.082	0.181	0.115	0.242
	24	0.076	0.167	0.081	0.183	0.088	0.190	0.242	0.341	0.139	0.247	0.271	0.383	0.101	0.204	0.210	0.329
	48	0.096	0.185	0.093	0.192	0.110	0.215	0.562	0.541	0.311	0.369	0.446	0.495	0.134	0.238	0.398	0.458
	96	0.122	0.203	0.117	0.217	0.139	0.245	1.096	0.795	0.396	0.442	0.628	0.577	0.181	0.279	0.594	0.553
	Avg	0.091	0.177	0.089	0.188	0.101	0.204	0.504	0.478	0.235	0.315	0.380	0.440	0.124	0.225	0.329	0.395
PEMS08	12	0.139	0.182	0.076	0.178	0.079	0.182	0.133	0.247	0.165	0.214	0.227	0.343	0.112	0.212	0.154	0.276
	24	0.162	0.200	0.104	0.209	0.115	0.219	0.249	0.343	0.215	0.260	0.318	0.409	0.141	0.238	0.248	0.353
	48	0.189	0.218	0.167	0.228	0.186	0.235	0.569	0.544	0.315	0.355	0.497	0.510	0.198	0.283	0.440	0.470
	96	0.217	0.237	0.245	0.280	0.221	0.267	1.166	0.814	0.377	0.397	0.721	0.592	0.320	0.351	0.674	0.565
	Avg	0.177	0.209	0.148	0.224	0.150	0.226	0.529	0.487	0.268	0.307	0.441	0.464	0.193	0.271	0.379	0.416
1st Count		11	18	9	2	0	0	0	0	0	0	0	0	0	0	0	0

Compared with strong baselines such as S-Mamba and iTransformer, SSGAN demonstrates competitive performance for short-term horizons and maintains a stable advantage over longer horizons. Other baselines, including DLinear, RLinear, Crossformer, TiDE, and TimesNet, generally underperform, with their disadvantages becoming more pronounced for long-term predictions. Moreover, the rankings of MSE and MAE are largely consistent, indicating that the improvement brought by SSGAN is not merely due to the reduction of a few extreme errors, but reflects an overall improvement in the error distribution. Overall, SSGAN achieves state-of-the-art performance and exhibits stronger generalization capability for traffic flow forecasting tasks.

To provide a more intuitive illustration of the model's predictive performance, we plotted the forecasted traffic flows of the three best-performing models over a 96-step prediction horizon. It can be observed that all methods generally follow the overall trend of the ground truth; however, significant differences exist in fitting accuracy at traffic peaks and troughs. Compared with iTransformer and S-Mamba, the proposed SSGAN more closely tracks the true curve during abrupt traffic fluctuations, exhibiting smaller prediction delays and deviations. Moreover, SSGAN maintains high prediction accuracy in stable periods, demonstrating superior stability and robustness.

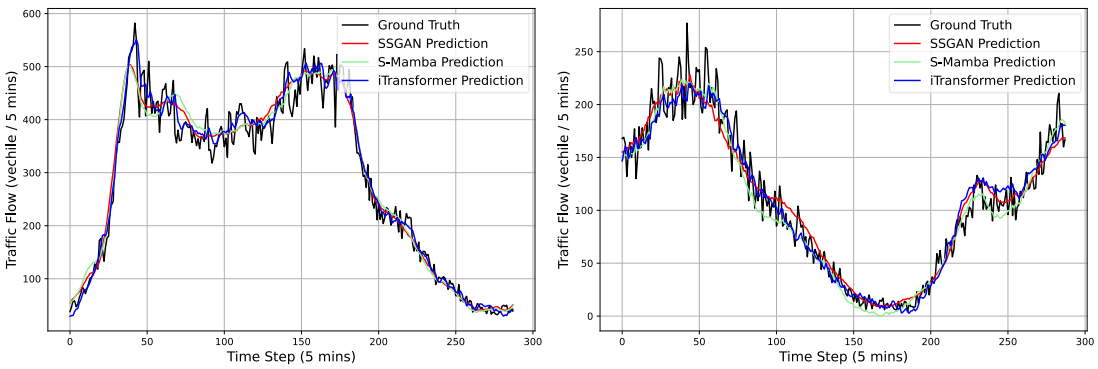

Figure 5. 96-step traffic flow predictions on the PEMS03 (left) and PEMS04 (right) datasets.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

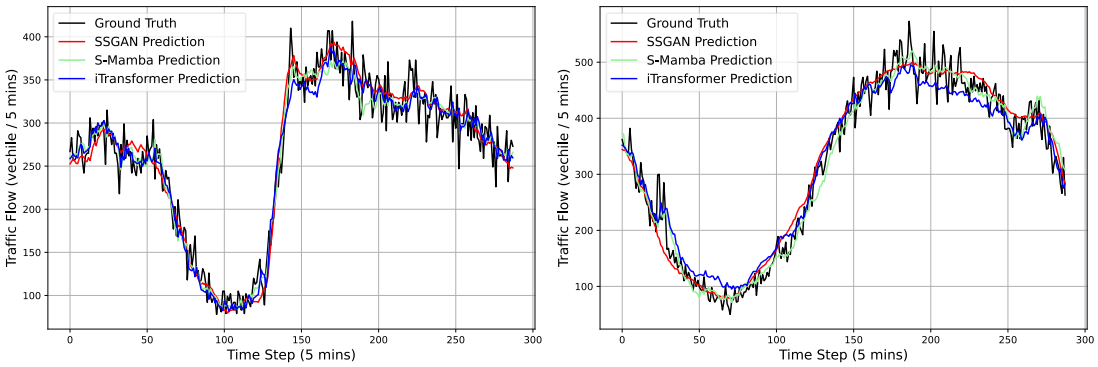

Figure 6. 96-step traffic flow predictions on the PEMS07 (left) and PEMS08 (right) datasets.

5.3. Hyperparameter Experiments

We employed the Optuna framework for systematic hyperparameter optimization, using the tree-structured parzen estimator as the sampler. The optimization objective was to minimize the validation set loss, with MAE selected as the evaluation metric. For each prediction horizon, 100 trials were conducted, and the hyperparameter configuration corresponding to the best performance on the validation set was selected.

In designing the hyperparameter search space, both model complexity and computational feasibility were considered. The main parameters optimized include: embedding dimension selected from $\{32, 64, 128\}$; hidden layer dimension, state dimension, and the number of channels in the interactive convolution block chosen from $\{64, 128, 256\}$; multi-scale convolution channels selected from $\{32, 64, 128\}$, with kernel size combinations including $\{[3], [5], [7], [3, 5], [3, 7], [5, 7], [3, 5, 7]\}$.

Additionally, to ensure reproducibility, all experiments used fixed random seeds, guaranteeing stability during model training and comparability of results.

5.4. Ablation study

To validate the effectiveness of key components in the proposed model, three sets of ablation experiments were designed to assess the impact of different modules on overall performance:

- Bidirectional Mamba-2 replaced with unidirectional: The bidirectional Mamba-2 module in the generator was replaced with a unidirectional structure to evaluate the contribution of bidirectional temporal dependency modeling to long-sequence prediction performance.
- Removal of the interactive convolution block: The interactive convolution block in the generator was removed to examine its role in feature fusion.
- Removal of the adversarial framework (generator-only): The discriminator in the GAN framework was completely removed, leaving only the generator for prediction, in order to analyze the importance of the adversarial mechanism in alleviating over-smoothing and improving the capture of fine-grained temporal patterns.

Table 3 presents the results of the ablation experiments. Replacing the bidirectional Mamba-2 with a unidirectional variant led to increases in both MSE and MAE across all prediction horizons on the PEMS04, PEMS07, and PEMS08 datasets. The most notable increase occurred on PEMS08 for the 96-step prediction horizon, where MSE and MAE increased by 3.69% and 2.95%, respectively. This demonstrates that the bidirectional structure provides an advantage in capturing both past and future temporal dependencies, contributing to improved accuracy for long-sequence predictions.

Removing the interactive convolution block resulted in performance comparable to the full model for short-term predictions, but significantly degraded performance for long-term horizons.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

For example, in the 96-step prediction on PEMS03, MSE increased from 0.147 to 0.160. This indicates that the interactive convolution block plays a critical role in fusing multi-scale features along the spatial dimension, and its absence weakens the model's ability to capture spatial correlations.

When the GAN framework was removed and only the generator was used for prediction, the average prediction errors increased across all four datasets. In particular, for the 96-step prediction on PEMS08, MSE rose from 0.217 to 0.227, and MAE increased from 0.237 to 0.245. This highlights the indispensable role of adversarial learning in mitigating over-smoothing and restoring fine-grained fluctuations in traffic flow.

In summary, these three ablation studies fully validate the importance of the bidirectional state-space structure, the interactive convolution block, and the adversarial framework in the proposed model.

Table 3. Ablation study results.

Models	Horizon	PEMS03		PEMS04		PEMS07		PEMS08	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
SSGAN	12	0.072	0.174	0.070	0.169	0.068	0.152	0.139	0.182
	24	0.086	0.191	0.079	0.179	0.076	0.167	0.162	0.200
	48	0.120	0.225	0.091	0.193	0.096	0.185	0.189	0.218
	96	0.147	0.248	0.103	0.207	0.122	0.203	0.217	0.237
	Avg	0.106	0.210	0.086	0.187	0.091	0.177	0.177	0.209
Bidirectional Mamba-2 replaced with unidirectional	12	0.072	0.174	0.070	0.169	0.070	0.154	0.140	0.186
	24	0.086	0.192	0.079	0.180	0.077	0.167	0.159	0.200
	48	0.117	0.223	0.093	0.196	0.099	0.191	0.190	0.220
	96	0.149	0.252	0.104	0.208	0.124	0.207	0.225	0.244
	Avg	0.106	0.210	0.087	0.188	0.093	0.180	0.179	0.213
Removal of the interactive convolution block	12	0.071	0.173	0.070	0.169	0.064	0.153	0.129	0.184
	24	0.087	0.194	0.078	0.179	0.078	0.169	0.162	0.202
	48	0.118	0.223	0.090	0.192	0.094	0.186	0.189	0.222
	96	0.160	0.256	0.102	0.207	0.126	0.208	0.214	0.241
	Avg	0.109	0.212	0.085	0.187	0.091	0.179	0.174	0.212
Removal of the adversarial framework	12	0.074	0.176	0.070	0.169	0.071	0.154	0.133	0.182
	24	0.085	0.190	0.080	0.180	0.079	0.167	0.162	0.201
	48	0.120	0.223	0.093	0.196	0.097	0.189	0.188	0.219
	96	0.152	0.252	0.104	0.207	0.121	0.203	0.227	0.245
	Avg	0.108	0.210	0.087	0.188	0.092	0.178	0.178	0.212

To evaluate whether the proposed model can alleviate the over-smoothing problem in traffic flow prediction, we conducted a visualization analysis of the 96-step predictions on the PEMS03 and PEMS08 datasets. Specifically, the prediction matrices were treated as node representations in the prediction space, and t-distributed stochastic neighbor embedding was applied to reduce the high-dimensional data to three dimensions. A unit-sphere projection was then used to preserve the overall distribution structure. As shown in Figure 7 and Figure 8, in the generator-only model (right panels), most node features are highly concentrated, exhibiting significant over-smoothing. In contrast, in the SSGAN model (left panels), the nodes are more widely distributed on the sphere. These results indicate that the SSGAN model can effectively mitigate over-smoothing in long-term sequence prediction, thereby capturing the complex spatiotemporal variations of traffic flow more accurately.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

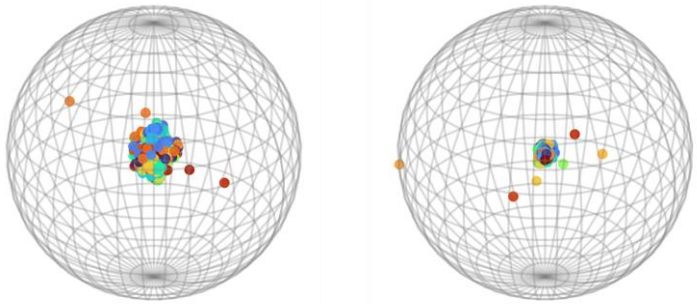

Figure 7. Visualization of over-smoothing alleviation by SSGAN on the PEMS03 dataset.

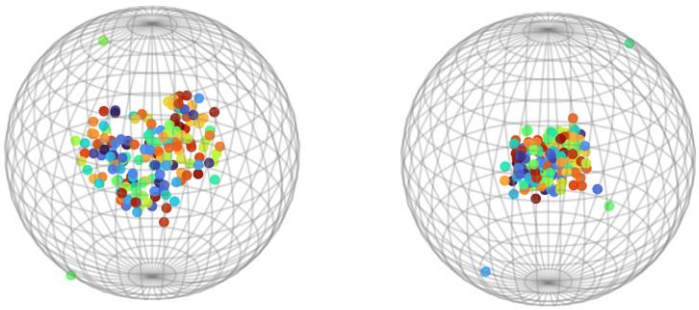

Figure 8. Visualization of over-smoothing alleviation by SSGAN on the PEMS08 dataset.

5.5. Model Efficiency

On the PEMS04 dataset, we conducted an efficiency evaluation of different models under the setting of a prediction horizon of 96 and a batch size of 32, measuring their prediction errors, average training time, and memory usage. As shown in Table 4, SSGAN achieves the lowest MSE and MAE, while also delivering the fastest training speed (60 ms/iter) with memory consumption only slightly higher than that of DLinear, demonstrating significant advantages in both accuracy and efficiency. S-Mamba attains prediction accuracy comparable to SSGAN, but its training speed is substantially slower and its memory usage is higher, making it less efficient overall. iTransformer exhibits further degradation in accuracy, with training speed and memory consumption at moderate levels. Overall, SSGAN achieves superior prediction performance while substantially reducing computational and memory overhead, highlighting its potential for practical deployment.

Table 4. Efficiency results.

Model	MAE	MSE	Training speed (ms/iter)	Memory usage (GB)
SSGAN	0.207	0.103	60	0.48
S-Mamba	0.248	0.137	486	8.75
iTransformer	0.262	0.150	161	6.07
Crossformer	0.457	0.402	543	41.07
TimesNet	0.303	0.190	1146	10.53
DLinear	0.504	0.452	92	0.13

6. Conclusions

we proposed SSGAN, a novel generative adversarial framework for long-term traffic flow forecasting, which integrates a state-space-based generator with multi-scale convolutional discriminators. By jointly modeling temporal dependencies and spatial correlations, the proposed

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

method effectively alleviates the over-smoothing issue commonly observed in long-horizon time series prediction and enhances the model's ability to capture fine-grained spatiotemporal dynamics.

Extensive experiments on four real-world PEMS datasets demonstrate that SSGAN consistently outperforms a wide range of strong baselines, including Mamba-, Transformer-, linear-, and convolution-based models. In particular, SSGAN achieves state-of-the-art accuracy across different horizons, with performance advantages becoming more pronounced under long prediction lengths. Visualization analyses further confirm that SSGAN preserves node-level heterogeneity and mitigates over-smoothing in comparison to generator-only models. Moreover, efficiency experiments highlight that SSGAN not only delivers superior predictive accuracy but also reduces computational and memory costs, underscoring its potential for large-scale deployment in intelligent transportation systems.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (Grant No.52432010 and 52372314).

Data Availability Statement: Data will be made available on request. The code is available at https://github.com/Vincent665/SSGAN-for-traffic-prediction.git.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Wang, R., Lin, W., Ren, G., Cao, Q., Zhang, Z., & Deng, Y. (2025). Interaction-Aware Vehicle Trajectory Prediction Using Spatial-Temporal Dynamic Graph Neural Network. Knowledge-Based Systems, 114187.
2. Zhang, J., Huang, D., Liu, Z., Zheng, Y., Han, Y., Liu, P., Huang, W. A data-driven optimization-based approach for freeway traffic state estimation based on heterogeneous sensor data fusion. Transportation Research Part E: Logistics and Transportation Review, 189 (2024), Article 103656.
3. Huang, D., Zhang, J., Liu, Z., & Liu, R. (2025). Prescriptive analytics for freeway traffic state estimation by multi-source data fusion. Transportation Research Part E: Logistics and Transportation Review, 198, 104105.
4. Zhao, Y., Hua, X., Yu, W., Lin, W., Wang, W., & Zhou, Q. (2025). Safety and efficiency-oriented adaptive strategy controls for connected and automated vehicles in unstable communication environment. Accident Analysis & Prevention, 220, 108121.
5. Huang, D., Zhang, J., Liu, Z., He, Y., & Liu, P. (2024). A novel ranking method based on semi-SPO for battery swap** allocation optimization in a hybrid electric transit system. Transportation Research Part E: Logistics and Transportation Review, 188, 103611.
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention is all you need. Advances in Neural Information Processing Systems, 30 (2017).
7. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In International Conference on Machine Learning, PMLR (2022), pp. 27268-27286.
8. Zhang, Y., Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In The Eleventh International Conference on Learning Representations, 2023.
9. Nie, Y., Nguyen, N. H., Sinthong, P., Kalagnanam, J. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In The Eleventh International Conference on Learning Representations, 2023.
10. Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In The Twelfth International Conference on Learning Representations, 2024.
11. Luo, Q., He, S., Han, X., Wang, Y., & Li, H. (2024). LSTTN: A long-short term transformer-based spatiotemporal neural network for traffic flow forecasting. Knowledge-Based Systems, 293, 111637.
12. Xiao, J., & Long, B. (2024). A multi-channel spatial-temporal transformer model for traffic flow forecasting. Information Sciences, 671, 120648.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

13. Fang, Y., Liang, Y., Hui, B., Shao, Z., Deng, L., Liu, X., ... & Zheng, K. (2025, July). Efficient large-scale traffic forecasting with transformers: A spatial data management perspective. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1 (pp. 307-317).
14. Lu, J., Yao, J., Zhang, J., Zhu, X., Xu, H., Gao, W., ... & Zhang, L. (2021). Soft: Softmax-free transformer with linear complexity. Advances in Neural Information Processing Systems, 34, 21297-21309.
15. Gu, A., Dao, T., Ermon, S., Rudra, A., Ré, C. Hippo: Recurrent memory with optimal polynomial projections. Advances in Neural Information Processing Systems, 33 (2020), pp. 1474-1487.
16. Gu, A., Goel, K., Ré, C. Efficiently modeling long sequences with structured state spaces. (2021), arXiv preprint arXiv:2111.00396.
17. Gu, A., Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. (2023), arXiv preprint arXiv:2312.00752.
18. Dao, T., Gu, A. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. (2024), arXiv preprint arXiv:2405.21060.
19. Sun, Y., Xie, Z., Chen, D., Eldele, E., & Hu, Q. (2025, April). Hierarchical classification auxiliary network for time series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 19, pp. 20743-20751).
20. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.
21. Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L. Transformers in time series: a survey. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 2023, pp. 6778-6786.
22. Zeng, A., Chen, M., Zhang, L., Xu, Q. Are transformers effective for time series forecasting?. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37 (9) (2023), pp. 11121-11128.
23. Bi, J., Zhang, X., Yuan, H., Zhang, J., & Zhou, M. (2021). A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM. IEEE Transactions on Automation Science and Engineering, 19(3), 1869-1879.
24. Eldele, E., Ragab, M., Chen, Z., Wu, M., & Li, X. (2024). Tslanet: Rethinking transformers for time series representation learning. arxiv preprint arxiv:2404.08472.
25. Su, J., Xie, D., Duan, Y., Zhou, Y., Hu, X., & Duan, S. (2024). MDCNet: Long-term time series forecasting with mode decomposition and 2D convolution. Knowledge-Based Systems, 299, 111986.
26. Reza, S., Ferreira, M. C., Machado, J. J. M., & Tavares, J. M. R. (2025). Enhancing intelligent transportation systems with a more efficient model for long-term traffic predictions based on an attention mechanism and a residual temporal convolutional network. Neural Networks, 107897.
27. Lee, M., Yoon, H., & Kang, M. (2025). CASA: CNN Autoencoder-based Score Attention for Efficient Multivariate Long-term Time-series Forecasting. arxiv preprint arxiv:2505.02011.
28. Wang, Z., Kong, F., Feng, S., Wang, M., Yang, X., Zhao, H., Wang, D., Zhang, Y. Is mamba effective for time series forecasting?. Neurocomputing, 619 (2025), Article 129178.
29. Lin, W., Zhang, Z., Ren, G., Zhao, Y., Ma, J., Cao, Q. MGCN: Mamba-integrated spatiotemporal graph convolutional network for long-term traffic forecasting. Knowledge-Based Systems, 309 (2025), Article 112875.
30. Cao, J., Sheng, X., Zhang, J., & Duan, Z. (2025). iTransMamba: A lightweight spatio-temporal network based on long-term traffic flow forecasting. Knowledge-Based Systems, 317, 113416.
31. Hamad, M., Mabrok, M., & Zorba, N. (2025). MCST-Mamba: Multivariate Mamba-Based Model for Traffic Prediction. arxiv preprint arxiv:2507.03927.
32. Wang, X., Cao, J., Zhao, T., Zhang, B., Chen, G., Li, Z., ... & Li, Q. (2025). ST-Camba: A decoupled-free spatiotemporal graph fusion state space model with linear complexity for efficient traffic forecasting. Information Fusion, 103495.
33. Zhang, L., Wu, J., Shen, J., Chen, M., Wang, R., Zhou, X., ... & Wu, Q. (2021). SATP-GAN: Self-attention based generative adversarial network for traffic flow prediction. Transportmetrica B: Transport Dynamics, 9(1), 552-568.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 October 2025 doi:10.20944/preprints202510.0402.v1

34. Khaled, A., Elsir, A. M. T., & Shen, Y. (2022). TFGAN: Traffic forecasting using generative adversarial network with multi-graph convolutional network. Knowledge-Based Systems, 249, 108990.
35. Zhang, Z., Cao, Q., Lin, W., Song, J., Chen, W., & Ren, G. (2024). Estimation of Arterial Path Flow Considering Flow Distribution Consistency: A Data-Driven Semi-Supervised Method. Systems, 12(11), 507.
36. Chen, C., Petty, K., Skabardonis, A., Varaiya, P., Jia, Z. Freeway performance measurement system: mining loop detector data. Transportation research record, 1748 (1) (2001), pp. 96-102.
37. [37]Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., Yu, R. Long-term forecasting with tide: Time-series dense encoder. (2023), arXiv preprint arXiv:2304.08424.
38. Li, Z., Qi, S., Li, Y., Xu, Z. Revisiting long-term time series forecasting: An investigation on linear mapping. (2023), arXiv preprint arXiv:2305.10721.
39. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In The Eleventh International Conference on Learning Representations, 2023.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.