ARCHIVES OF TRANSPORT *Volume 74, Issue 2, 2025*

ISSN (print): *0866-9546*
e-ISSN (online): *2300-8830*
DOI: *10.61089/aot2025.13gyse14*

RESEARCH ON HIGH-FIDELITY RECONSTRUCTION ALGORITHM FOR VEHICLE TRAJECTORY BASED ON CENTRALIZED SPATIOTEMPORAL FEATURE FUSION

Rijun CHEN[1], Xian REN[2], Qihua MA[3]

[1, 2] *Shanghai Technology and Innovation Vocational College, Shanghai, China*

[3] *Shanghai University of Engineering Science, Shanghai, China*

Abstract:

Vehicle trajectory data, primarily characterized by time-series features, is the key information carrier for vehicle movement. It documents dynamic characteristics such as position, velocity, and acceleration across spatiotemporal dimensions. This data supports critical tasks, including microscopic traffic flow modeling, driving state analysis, and traffic safety assessment. To address two key challenges — low-sampling-rate trajectory data losing high-frequency motion details in intelligent transportation systems, and traditional methods failing to simultaneously meet requirements for high precision and motion plausibility — this study proposes a high-fidelity vehicle trajectory reconstruction algorithm based on centralized spatiotemporal feature fusion. Based on the SOFTS (Series-cOre Fused Time Series) framework, the algorithm constructs a global-local collaborative spatiotemporal feature representation mechanism via a multidimensional enhanced STAR (Spatio-Temporal Aggregation and Representation) module. Specifically, it incorporates a spatiotemporal embedding layer to capture interdependencies between timestamps and spatial coordinates. Residual connections preserve original trajectory details, while a spatial proximity weighting mechanism optimizes core feature aggregation. A dynamic weight matrix is constructed to adaptively focus on velocity-position correlations among neighboring vehicles within a 20-meter radius. Kinematic constraints are integrated to ensure the physical plausibility of reconstructed trajectories, including a kinematic loss function. This function uses acceleration smoothness regularization terms and trajectory curvature continuity regularization to guide the model toward physically feasible solutions that comply with vehicle dynamics. To validate effectiveness, the proposed method was extensively tested on public datasets (NGSIM, HighD, and CQSkyeyeX) and systematically compared with traditional approaches (e.g., linear interpolation) and deep learning models. Experimental results demonstrate that the improved algorithm significantly outperforms baseline methods in interpolation accuracy, spatiotemporal smoothness, and computational efficiency. The findings can be applied to high-frequency trajectory generation for autonomous driving, microscopic traffic flow simulation, and other domains, providing critical technical support for upgrading intelligent transportation systems from data collection to decision-making optimization.

Keywords: *vehicle trajectory, data reconstruction, improved SOFTS algorithm, spatiotemporal features*

To cite this article:

Chen, R., Ren, X., Ma, Q., (2025). Research on high-fidelity reconstruction algorithm for vehicle trajectory based on centralized spatiotemporal feature fusion. Archives of Transport, 74(2), 81-97. https://doi.org/10.61089/aot2025.13gyse14

Contact:
1) 122079566@qq.com [https://orcid.org/0009-0001-6163-378X] – corresponding author; 2) 181669264@qq.com [https://orcid.org/0009-0005-6178-9601]; 3) mqh0386@sues.edu.cn [https://orcid.org/0000-0003-0049-9190]

1. Introduction

Vehicle trajectory data serves as a core foundation for intelligent transportation systems and autonomous driving research. It underpins key tasks such as traffic behavior pattern analysis, driving decision optimization, and accident risk prediction (H. Li & Yu, 2025; Biszko et al., 2024). However, existing vehicle trajectory data is constrained by acquisition and storage costs, resulting in widespread insufficient sampling frequency. This limitation causes the loss of fine-grained motion characteristics in high-dynamic behaviors — such as abrupt acceleration and emergency lane changes — severely restricting the accuracy of traffic flow simulation and driving scene reconstruction (Quddus & Washington, 2015). In complex interactive scenarios, low-frequency sampling may obscure critical frames of vehicle-vehicle conflicts, directly impacting the robustness verification of autonomous driving algorithms in edge cases (Waddell et al., 2020). For interpolating and reconstructing low-frequency trajectory data, existing research falls broadly into two categories. First, kinematic model-based methods (e.g., cubic spline interpolation, constant acceleration models) ensure physical plausibility but struggle to model nonlinear spatiotemporal interactions in complex traffic scenarios (Wang et al., 2024). Second, deep learning-based end-to-end methods (e.g., Trajectory-Transformer) capture data-driven spatiotemporal patterns but often generate trajectories that violate vehicle dynamics. This is due to insufficient explicit constraints, such as acceleration abruptions and curvature discontinuities (Liu et al., 2020). Fundamentally, the inability of existing methods to couple spatiotemporal feature representation with physical law optimization has become a core bottleneck for high-frequency trajectory reconstruction.

To address these gaps, this study aims to develop a spatiotemporal feature-fused high-fidelity vehicle trajectory reconstruction method (ST-SOFTS, Spatio-Temporal Enhanced SOFTS). Built upon the SOFTS algorithm (Han et al., 2024), ST-SOFTS introduces three innovations: (1) a spatiotemporal-aware embedding layer to capture temporal-spatial dependencies in trajectory timestamps and coordinates; (2) a spatial-weighted core aggregation module to enhance modeling of inter-vehicle spatiotemporal dependencies; (3) a spatiotemporal constraint loss function to ensure reconstructed trajectories comply with kinematic constraints (e.g., velocity smoothness, safe distance).

2. Literature Review

Vehicle trajectory reconstruction remains a key research topic in intelligent transportation, with its technological evolution broadly categorized into two branches: traditional methods and deep learning methods. Traditional trajectory interpolation approaches, predominantly rooted in vehicle kinematics priors, fit local motion patterns through parameterized models. Representative techniques include cubic spline interpolation (He et al., 2024), constant acceleration models (Yang et al., 2018), and Kalman filtering (Zhang et al., 2023). These methods generate smooth trajectories by integrating physical constraints (e.g., acceleration continuity) and exhibit stability in low-complexity scenarios. However, their limitations include reliance on strong assumptions (e.g., constant velocity or acceleration). These assumptions hinder their ability to characterize nonlinear interactions in complex traffic flows — such as emergency obstacle avoidance and multi-vehicle interactions (Reis et al., 2023). In highly dynamic or dense traffic scenarios, such approaches often deviate from real-world trajectories due to model mismatch (Yang et al., 2025). Concurrently, deep learning-based trajectory generation methods have emerged as the mainstream with the rapid advancement of neural network technologies.

Recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) (Lin et al., 2024) capture long-term dependencies in vehicle motion through temporal modeling (Ip et al., 2021). However, they encounter challenges in sequence alignment when modeling multi-vehicle interactions. Graph neural networks (GNNs) have significantly improved trajectory prediction accuracy in interactive scenarios (Youssef et al., 2023) by explicitly modeling vehicle spatial topological relationships (Singh & Srivastava, 2022). For example, the Spatio-Temporal Dynamic Graph Neural Network (Li et al., 2023) leverages spatiotemporal graph convolutions to extract local traffic flow features, while Trajectory-Transformer (Amin et al., 2024) employs self-attention mechanisms to model global spatiotemporal dependencies. Notably, most existing approaches focus predominantly on trajectory prediction rather than low-sampled data interpolation. They often neglect kinematic constraints, generating trajectories with abrupt acceleration changes or curvature discontinuities (Rathore et al., 2019). Wang et al. (2025) employed a two-layer LSTM network to extract contextual information from target vehicles and their neighbors, enhancing trajectory prediction accuracy. Additionally, vehicle-road cooperative technologies have provided critical inputs for trajectory modeling. For instance, Lei et al. (2024) utilized dynamic environmental data from surrounding vehicles and roadside units to forecast neighboring vehicle trajectories.

To balance the flexibility and physical plausibility of data-driven models, attempts have been made to integrate kinematic constraints into deep learning frameworks. Gu et al. (2024) proposed incorporating vehicle stability, lane-changing duration, terminal velocity, and target lane distance into the loss function to enforce trajectory smoothness. Jiang et al. (2023) corrected neural network outputs through dynamic model post-processing. However, these approaches typically use loosely coupled frameworks, failing to achieve end-to-end collaborative optimization of physical laws and spatiotemporal features. Regarding spatiotemporal feature fusion, centralized modeling has gained traction (Jiang et al., 2024). For example, Li et al. (2023) aggregated multi-vehicle features via global pooling layers, but their coarse-grained spatial interaction modeling struggled to capture heterogeneous influences in local neighborhoods.

Overall, existing research exhibits three critical limitations: (1) spatiotemporal modeling fragmentation: Traditional methods process temporal and spatial dimensions in isolation, hindering the characterization of dynamic interactions in complex traffic scenarios; (2) inadequate physical constraints: Data-driven methods often generate trajectories that violate vehicle dynamics, compromising their utility in high-fidelity applications (e.g., autonomous driving simulation); (3) sparse local interaction modeling: Most centralized approaches overlook differentiated weight allocation in spatial neighborhoods during global feature aggregation, leading to the loss of critical interaction information.

Therefore, this study proposes a high-fidelity vehicle trajectory reconstruction algorithm based on centralized spatiotemporal feature fusion (CSTFF). The innovations include: (1) developing a global spatiotemporal core aggregation network, which achieves collaborative modeling of long-range dependencies and local interactions in traffic scenarios through multi-scale spatiotemporal embedding and a proximity-weighting mechanism; (2) designing a kinematic-constraint-guided joint optimization framework, which integrates physical priors (e.g., acceleration smoothness, trajectory curvature continuity) in a differentiable form to ensure the physical feasibility of reconstructed trajectories. Experimental results demonstrate that the proposed method outperforms mainstream baseline models in both reconstruction accuracy and motion plausibility. It offers a novel approach for high-frequency restoration of low-sampled trajectory data, balancing data-driven learning and physical constraints.

3. Methodology

Vehicle trajectory data reconstruction entails dense inference of sparse, low-frequency observation data across spatiotemporal dimensions. This task requires algorithms to mine spatiotemporal dependency patterns among vehicles — such as temporal sequence continuity and spatial positional correlations — from partial observations of vehicle positions and velocities. By integrating vehicle motion constraints, the process aims to estimate missing or undersampled spatiotemporal points and generate high-frequency continuous trajectories that reflect real-world traffic flow characteristics. This reconstruction process involves two key challenges: modeling temporal dependencies within individual vehicle trajectories and capturing spatial interactions across multiple vehicles. The core challenge lies in balancing data-driven feature learning with domain knowledge to achieve high-precision, physically plausible trajectory completion.

3.1. Model Overview

The ST-SOFTS (Spatio-Temporal Enhanced Series-cOre Fused Time Series) model aims to address the challenge of high-frequency reconstruction for low-frequency vehicle trajectory data. Its core architecture builds upon the SOFTS algorithm, enhanced by three innovations: a spatiotemporal-aware embedding layer, a spatial-weighted core aggregation module, and a spatiotemporal constraint optimization layer. These innovations enable efficient modeling of spatiotemporal dependencies across multi-vehicle trajectories. As illustrated in Fig. 1, the model takes as input a low-frequency trajectory matrix with missing values, $X \in \mathbb{R}^{C\times \mathrm{L}\times \mathrm{D}}$(where C is the number of vehicles, L is the time step, and D is the trajectory dimension) along with corresponding timestamp sequences. First, the spatiotemporal embedding layer encodes temporal intervals and spatial coordinates into high-dimensional feature vectors. Next, the improved STAR (Spatio-Temporal Aggregation and Representation) module dynamically generates global core features using spatial proximity weights, facilitating centralized spatiotemporal feature fusion across vehicles. Finally, the spatiotemporal constraint optimization layer enforces temporal smoothness (e.g., velocity continuity) and spatial safety (e.g., minimum distance) to produce high-frequency continuous trajectories, $X \in \mathbb{R}^{C \times L' \times D}$, where $L' > L$ denotes the upsampled time steps.

3.2. Spatial-time embedding layer

Spatiotemporal reconstruction of vehicle trajectories requires simultaneous capture of interactions between temporal dynamics and spatial dependencies. The spacetime embedding layer of the ST-SOFTS framework transforms raw trajectory data into high-dimensional feature vectors that encapsulate spatiotemporal semantics. This is achieved through joint modeling of temporal and spatial feature encodings, establishing the foundational input for subsequent core aggregation modules.

To address temporal discontinuities inherent in low-frequency trajectory data, ST-SOFTS integrates relative time interval encoding with sinusoidal position encoding to accentuate sequential and periodic patterns. For each time step t, the relative time interval $\Delta t_t = t - t_1$ is computed to delineate the temporal positional relationship of trajectory points. Temporal embeddings are then generated using trigonometric functions:

$$E_t^{time} = [\sin\left(\frac{\Delta t_t}{T_{period}}\right), \cos\,(\frac{\Delta t_t}{T_{period}})] \in \mathbb{R}^{d_{time}} \quad (1)$$

Among them, T_{period} is a periodic parameter, and d_{time} denotes the temporal embedding dimension. This encoding method mitigates the curse of dimensionality associated with absolute timestamps and exhibits robustness to nonlinear variations in time intervals. For spatial features, ST-SOFTS captures inter-vehicle interactions by generating dynamic spatial proximity weights via an euclidean distance matrix, replacing the fixed-radius neighborhood used in traditional methods. For the coordinates of vehicle c at time t $(x_{c,t}, y_{c,t})$, the distance to each vehicle i is calculated as:

$$D_{c,t,i} = \sqrt{\left(x_{c,t} - x_{i,t}\right)^2 + \left(y_{c,t} - y_{i,t}\right)^2} \quad (2)$$

Generate dynamic weights with SoftMax normalization, focusing on adjacent vehicles:

$$w_{c,t,i} = \frac{\exp\,(-D_{c,t,i}/\sigma)}{\sum_{j=1}^{C} \exp\,(-D_{c,t,j}/\sigma)} \quad (i, j \in [1, C]) \quad (3)$$

Where σ is a parameter controlling the concentration of weight distribution. For spatiotemporal feature fusion, the raw trajectory features (position, velocity), temporal embeddings, and spatial weights are concatenated and mapped into a unified spatiotemporal embedding via a linear layer $S_{c,t} \in \mathbb{R}^d$:

$$S_{c,t} = \text{Linear}([X_{c,t}; E_t^{time}; w_{c,t}]) \quad (4)$$

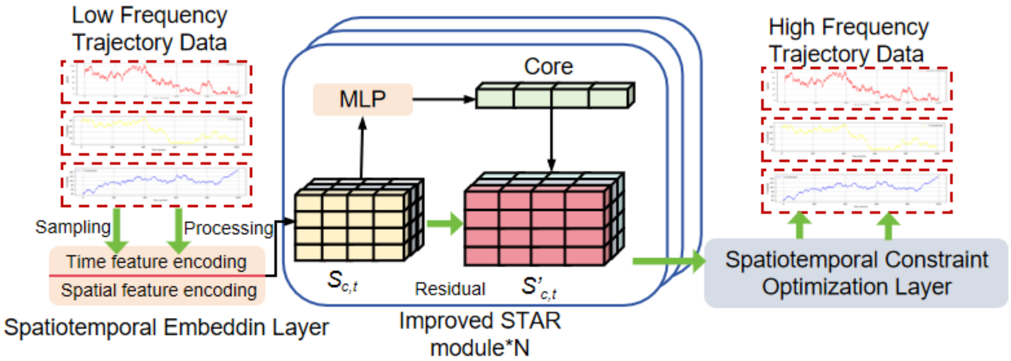
Low Frequency Trajectory Data
Sampling
Processing
Time feature encoding
Spatial feature encoding
Spatiotemporal Embeddin Layer
MLP
Core
$S_{c,t}$
Residual
$S'_{c,t}$
Improved STAR module*N
High Frequency Trajectory Data
Spatiotemporal Constraint Optimization Layer

Fig. 1. Schematic diagram of ST-SOFTS algorithm structure

Among them, $X_{c,t} \in \mathbb{R}^D$the original trajectory feature. This method breaks the independent processing of limitation of space-time features in traditional methods , provides composite features containing interactive information for subsequent core aggregation.

3.3. Space-weighted core aggregation module

The core innovation of ST-SOFTS lies in enhancing the original STAR module of SOFTS. By integrating spatial proximity weights with spatiotemporal features, ST-SOFTS upgrades the time-series aggregation of the original algorithm to spatiotemporal joint aggregation, enabling centralized and efficient interaction among multiple vehicle trajectories. In contrast to the stochastic pooling mechanism in vanilla SOFTS, ST-SOFTS employs spatial proximity weights $w_{c,t,i}$ to perform weighted aggregation on individual vehicles' spatiotemporal embeddings, generating a global core feature $o_t \in \mathbb{R}_d$ that encapsulates spatiotemporal dependencies:

$$o_t = \sum_{c=1}^{C} w_{c,t,i} \cdot \mathrm{MLP}_1(S_{c,t}) \tag{5}$$

Among them, MLP_1 is a two-layer perceptron that maps the spatiotemporal embedding$S_{c,t}$to the core feature space. This component offers two key advantages: First, the weights$w_{c,t,i}$enable the core features to prioritize neighboring vehicles. For example, in car-following scenarios, the influential weight of the preceding vehicle's trajectory on the current vehicle is significantly higher than that of distant vehicles. Second, the aggregation process requires only $O(C)$ computational complexity, avoiding the $O(C^2)$ complexity of attention mechanisms and thus proving suitable for large-scale scenarios involving tens of thousands of vehicles.

To prevent the dilution of individual vehicle details by global core features, ST-SOFTS achieves complementary fusion of local features and global dependencies through residual connections. The single-vehicle spatiotemporal embedding $S_{c,t}$ is concatenated with the global core o_t along the channel dimension, forming$[S_{c,t}; o_t] \in \mathbb{R}^{2d}$, which is then projected via a multi-layer perceptron (MLP) before residual connections are added to retain the original local embedding information:

$$S'_t = \mathrm{MLP}_2([S_{c,t}; o_t]) + S_{c,t} \tag{6}$$

MLP₂ restores the dimension to d, where residual connections preserve original feature details (e.g., abrupt acceleration changes of individual vehicles) while integrating global spatiotemporal dependencies. The spatial-weighted core aggregation module employs a three-layer stacked architecture, with dynamically updated spatial weights in each layer: shallow modules focus on local spatiotemporal details (e.g., vehicle acceleration/deceleration patterns), while deep modules capture the overall velocity trends of global traffic flow.

3.4. Spatiotemporal Constraint Optimization Layer

To address the challenge of deep learning models generating physically infeasible trajectories, ST-SOFTS introduces a joint spatiotemporal constraint loss that incorporates kinematic prior knowledge into the optimization process, ensuring reconstructed trajectories adhere to real-world traffic flow dynamics. The temporal continuity of vehicle motion requires that velocity changes between adjacent timesteps remain within plausible bounds (e.g., acceleration constraint $a_{max} = 5m/s^2$), enforced via L₁ regularization:

$$\mathcal{L}_{smooth} = \frac{1}{CL'} \sum_{c=1}^{C} \sum_{t=2}^{L'} \left| \hat{v}_{c,t} - \hat{v}_{c,t-1} \right| \tag{7}$$

Here, $\hat{v}_{c,t}$ denotes the reconstructed velocity. This loss term compels the model to learn smooth velocity variation patterns, thereby preventing interpolation results from exhibiting implausible sudden accelerations or decelerations.

Additionally, in dense traffic scenarios, vehicles must maintain a minimum safety distance $d_{min} = 2m$, which is enforced through the following loss constraint:

$$\mathcal{L}_{safe} = \frac{1}{C^2 L'} \sum_{t=1}^{L'} \sum_{i<j} \max\left(0, d_{min} - \left\| \hat{p}_{i,t} - \hat{p}_{j,t} \right\|_2\right) \tag{8}$$

Here, $\hat{p}_{i,t} = (\hat{x}_{c,t}, \hat{y}_{c,t})$ denotes the reconstructed vehicle position. This loss term penalizes vehicle pairs with excessively close distances, thereby enhancing the safety and plausibility of reconstructed trajectories.

The total loss function fuses the data fitting loss and the constraint loss:

$$\mathcal{L} = \mathcal{L}_{MSE} + \lambda_1 \mathcal{L}_{smooth} + \lambda_2 \mathcal{L}_{safe} \tag{9}$$

$$\mathcal{L}_{MSE} = \frac{1}{CL'D} \sum_{c,t,d} (\hat{X}_{c,t,d} - X^{gt}_{c,t,d})^2 \tag{10}$$

Here, $\mathcal{L}_{MSE}$ denotes the mean squared error (MSE) between reconstructed trajectories and ground-truth trajectories, where λ_1, λ_2 are constraint weights tuned via a validation set.

The pseudo-code of the ST-SOFTS framework is shown in Table 1.

Table. 1. The pseudo-code of the ST-SOFTS framework

```
Algorithm: Spatio-Temporal Enhanced Series-cOre Fused Time Series(ST-SOFTS)
Input: Low-frequency trajectory matrix $X \in \mathbb{R}^{C \times L \times D}$ (C: number of vehicles, L: time steps, D: features including position/velocity); Timestamp sequence $T$; Other hyperparameters.
Output: Reconstructed high-frequency trajectory $X \in \mathbb{R}^{C \times L' \times D}$.

1. Preprocessing
    Normalize spatial coordinates, velocity, and acceleration

2. Initialize Parameters
    Linear layer weights for spatiotemporal embedding
    MLP parameters for core aggregation ($\mathrm{MLP}_1$, $\mathrm{MLP}_2$)
    Optimizer (Adam)

3. Spatiotemporal Embedding Layer
    For each vehicle $c \in [1, C]$ and time step $t \in [1, L]$:
    a. Compute temporal interval: $\Delta t_t = t - t_1$
    b. Temporal embedding: $E_t^{time} = [\sin\left(\frac{\Delta t_t}{T_{period}}\right), \cos\left(\frac{\Delta t_t}{T_{period}}\right)] \in \mathbb{R}^{d_{time}}$
    c. Spatial weight for neighboring vehicles $i \in [1, C]$ : $D_{c,t,i} = \sqrt{(x_{c,t} - x_{i,t})^2 + (y_{c,t} - y_{i,t})^2}$; $w_{c,t,i} = SoftMax\left(\frac{\exp(-D_{c,t,i}/\sigma)}{\sum_{j=1}^{C} \exp(-D_{c,t,j}/\sigma)}\right)$ $(i, j \in [1, C])$
    d. Fusion embedding: $S_{c,t} = \mathrm{Linear}([X_{c,t}; E_t^{time}; w_{c,t}])$
    e.
4. Spatial-Weighted Core Aggregation
    For each vehicle l ∈ [1,3]:
    a. Compute global core feature: $\mathrm{o_t} = \sum_{c=1}^{C} w_{c,t,i} \cdot \mathrm{MLP}_1(S_{c,t})$
    b. Residual fusion: $S_t' = \mathrm{MLP}_2([S_{c,t}; o_t]) + S_{c,t}$

5. High-Frequency Trajectory Generation
    Upsample $S_{c,t}'$ to $L'$ time steps via linear interpolation
    Map features back to position/velocity via output layer

6. Loss Calculation & Optimization
    a. MSE loss: $\mathcal{L}_{MSE} = \frac{1}{CL'D} \sum_{c,t,d} (\hat{X}_{c,t,d} - X^{gt}_{c,t,d})^2$
    b. Smoothness loss (acceleration constraint): $\mathcal{L}_{\mathrm{smooth}} = \frac{1}{\mathrm{CL'}} \sum_{\mathrm{c=1}}^{\mathrm{C}} \sum_{\mathrm{t=2}}^{\mathrm{L'}} |\hat{\mathrm{v}}_{\mathrm{c,t}} - \hat{\mathrm{v}}_{\mathrm{c,t-1}}|$
    c. Safety loss (minimum distance): $\mathcal{L}_{\mathrm{safe}} = \frac{1}{\mathrm{C^2L'}} \sum_{\mathrm{t=1}}^{\mathrm{L'}} \sum_{\mathrm{i<j}} \max(0, \mathrm{d_{min}} - \|\hat{\mathrm{p}}_{\mathrm{i,t}} - \hat{\mathrm{p}}_{\mathrm{j,t}}\|_2)$
    d. Total loss: $\mathcal{L} = \mathcal{L}_{\mathrm{MSE}} + \lambda_1 \mathcal{L}_{\mathrm{smooth}} + \lambda_2 \mathcal{L}_{\mathrm{safe}}$
    e. Update parameters via Adam optimizer
Return $\hat{X}$
```

4. Experimental verification

4.1. Experimental Datasets

To validate the data reconstruction performance of ST-SOFTS, we evaluate the algorithm using three publicly available vehicle trajectory datasets: NGSIM, HighD, and CQSkyeyeX.

NGSIM: Collected by the U.S. Federal Highway Administration (FHWA), this dataset contains highway vehicle trajectories from the US-101 and I-80 road segments, serving as benchmark data for high-density traffic flow research. Recorded at a 10 Hz sampling frequency, each scenario lasts approximately 15 minutes and includes typical highway behaviors such as car-following, lane-changing, and overtaking.

HighD: Released by the Institute for Automotive Engineering at RWTH Aachen University (Germany), this large-scale naturalistic driving dataset captures vehicle trajectories on German highways. Collected at 25 Hz, it includes 11.5 hours of measurements, trajectories of 110,000 vehicles, and 5,600 annotated lane-change maneuvers.

CQSkyeyeX: This dataset uses drones to collect 5.1k high-resolution videos of highway traffic operations. Vehicle trajectories are extracted via advanced computer vision techniques, providing 650 minutes of measurement data across eight locations on Chinese highways. Scenarios include straight segments, weaving areas, and merge/diverge sections.

4.2. Experimental Setup

Given the high-frequency nature of the original datasets and the low-frequency requirements of real-world applications, the experiments first involve downsampling and missing data simulation on the raw data. Since the three original datasets have non-uniform sampling frequencies, their raw data are uniformly downsampled to 5 Hz as model input (as shown in Fig. 2), with the reconstruction target set to 20 Hz — forming a typical "low-frequency input — high-frequency output" application scenario.

The selection of 5 Hz (input) and 20 Hz (target) is based on three key considerations: (1) Field consistency: 5 Hz is a typical low-sampling-rate scenario in vehicle trajectory research, which simulates practical challenges such as sensor transmission delays or low-cost hardware limitations; (2) Data adaptability: The original sampling rates of the NGSIM (10 Hz), HighD (25 Hz), and CQSkyeyeX (15 Hz) datasets allow 2:1, 5:1 or 3:1 downsampling to 5 Hz, ensuring no loss of key motion features while reflecting data sparsity; (3) Application and fairness: 20 Hz complies with the high-precision trajectory requirements of relevant standards for intelligent vehicles and matches the target frequency of baseline methods, ensuring both practical applicability and experimental fairness.

During the data preprocessing stage, raw data were normalized as follows: spatial coordinates were first translated to the scene centroid (i.e., repositioned to the origin at the scene centroid) and then mapped to the interval [-1, 1] using standard min-max scaling.

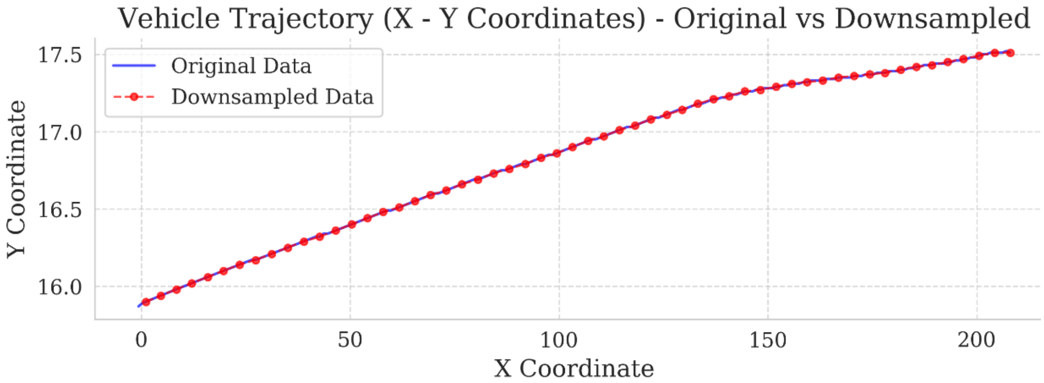

Fig. 2 Schematic diagram of data downsampling and reconstruction. The input (5 Hz) and target (20 Hz) frequencies are selected based on field consistency, data adaptability, and application requirements

Dynamic features (e.g., velocity, acceleration) were normalized using the mean and standard deviation calculated from the training set.

To ensure fair and reproducible model training and evaluation, the trajectory data of each dataset were randomly split into training, validation, and test sets at a fixed ratio of 7:1:2. A random seed (seed = 123) was used to fix the splitting result-this avoids randomness in data partitioning and ensures that other researchers can replicate the same data subsets for comparison.

All experiments were conducted on a server with the following hardware configuration: CPU: Intel Core i7-14700K(2.0 GHz, 16 cores); GPU: NVIDIA RTX 4090 24GB; memory: 64GB DDR5. The software environment included: deep learning framework: PyTorch 2.0.1; CUDA version: 11.8; and data processing libraries: Pandas 1.5.3, NumPy 1.15.1, and Matplotlib 3.7.1.

4.3. Model Parameter Configuration

The core parameters of ST-SOFTS are determined through grid search combined with validation set tuning, detailed as follows: In the spatiotemporal embedding layer, temporal features adopt sinusoidal positional encoding with an embedding dimension set to $d_{time} = 32$, capturing the periodicity and sequential dependencies of time series; the spatial proximity weight $\sigma = 5$ focuses the weight distribution on neighboring vehicles within 20 meters, balancing the interaction effects between nearby and distant vehicles. The core aggregation module consists of three stacked layers of improved STAR structures, where each MLP layer employs a two-layer GELU activation function with a hidden layer dimension $d = 128$. Residual connections are used to preserve original feature details and mitigate the vanishing gradient problem in deep networks. The weights of constraint losses are optimized via the validation set: the temporal smoothness loss weight $\lambda_1 = 0.3$ and the spatial safety loss weight $\lambda_2 = 0.2$, balancing data fitting accuracy and the intensity of physical constraints. During training, the Adam optimizer is used with an initial learning rate of 1×10^{-4}, a batch size of 128, and 200 training epochs. An early stopping mechanism is introduced (training terminates if the validation loss does not decrease for 10 consecutive epochs) to prevent overfitting and enhance model generalization.

4.4. Comparison Methods

To compare the performance differences of various methods in the experiments, representative traditional methods and deep learning models are selected as baselines, covering three technical routes: single-vehicle interpolation, spatiotemporal graph models, and attention models.

Linear Interpolation (LI): A linear fitting method based on single-vehicle time series, assuming linear changes in motion states between adjacent timesteps.

Cubic Spline Interpolation (CSI): Employs piecewise cubic polynomial fitting to ensure second-derivative continuity of trajectories and improve curve smoothness.

Kalman Filter (KF): Establishes a state-space model for single-vehicle constant-velocity motion and estimates missing trajectory points through recursive filtering.

STS-DGNN (Li et al., 2023): A dynamic graph convolutional network that constructs interaction graphs from vehicle positions and captures spatiotemporal dependencies via graph convolution and temporal convolution.

Vanilla SOFTS (Yang et al., 2018): A baseline model without spatiotemporal constraints, used to validate the independent contributions of the spatial weighting mechanism and constraint module.

TrajectoFormer (Amin et al., 2024): A spatiotemporal joint attention model that leverages multi-head self-attention to model global spatiotemporal dependencies, serving as a high-complexity comparative method.

TCN-SA (Li et al., 2023): A hybrid model combining TCN for temporal feature extraction and self-attention for long-range dependency modeling, which has shown strong performance in trajectory interpolation tasks.

4.5. Evaluation Metrics

This study designs a multi-metric evaluation framework from two dimensions: reconstruction accuracy and spatiotemporal plausibility. For reconstruction accuracy, the position mean squared error (MSE_{pos}) quantifies the overall deviation between reconstructed positions and ground-truth trajectories, while the velocity mean absolute error (MAE_{vel}) assesses the absolute precision of velocity reconstruction. Regarding spatiotemporal plausibility, temporal smoothness is measured by the standard deviation of reconstructed accelerations—denoted as σ_v, the standard deviation of velocity change rates:

$$\sigma_v = \sqrt{\frac{1}{CL'}\sum_{c=1}^{C}\sum_{t=2}^{L'}(\hat{a}_{c,t} - \bar{a}_c)^2}, \quad \hat{a}_{c,t} = \frac{\hat{v}_{c,t} - \hat{v}_{c,t-1}}{\Delta t} \tag{11}$$

Where $\bar{a}_c$ denotes the average acceleration of vehicle c, and Δt represents the time interval of 0.2s. Additionally, we calculate the proportion of vehicles that maintain the minimum safety distance ($d_{min} = 2m$):

$$SCR = \frac{1}{N_{total}}\sum_{t=1}^{L'}\sum_{1<i<j<C} S\left(\left\|\hat{p}_{i,t} - \hat{p}_{j,t}\right\| \geq d_{min}\right) \times 100\% \tag{12}$$

Where N_{total} denotes the total number of vehicles, and $S(\cdot)$ represents the indicator function.

4.6. Experimental Results

Under identical experimental conditions, the trajectory data reconstruction results of different algorithms on the three datasets are shown in Table 2.

Across the three datasets, traditional methods (LI/CSI/KF) exhibit significantly higher position mean squared error (MSEpos) than deep learning models. This is because they rely solely on single-vehicle time series and fail to account for inter-vehicle spatial interactions — shortcomings that amplify errors in dense car-following or lane-changing scenarios. While cubic spline interpolation (CSI) enhances trajectory smoothness via piecewise polynomial fitting, its single-vehicle modeling framework fails to effectively capture global traffic flow dynamics. The Kalman filter (KF), based on a constant-velocity motion assumption, achieves a velocity mean absolute error (MAE$_{vel)}$ of 1.38 m/s in HighD's stable car-following scenarios — slightly outperforming CSI — but exhibits significantly larger errors in NGSIM's lane-changing and CQSkyeyeX's hard-braking scenarios. This highlights the limited adaptability of physical models to complex motions.

ST-SOFTS reduces MSE$_{pos}$ by 15.2% (NGSIM), 10.6% (HighD), and 9.0% (CQSkyeyeX) compared to the baseline Vanilla SOFTS, demonstrating the synergistic effects of its spatial weighting mechanism and spatiotemporal constraints. Relative to STS-GNN, it achieves 27.0% (NGSIM), 23.0% (HighD), and 17.1% (CQSkyeyeX) reductions in MSE$_{pos}$, demonstrating the efficiency advantages of centralized aggregation over graph-based architectures.

Table 2. Trajectory Data Reconstruction Results of Different Algorithms on Three Datasets

Dataset	Method	MSE$_{pos}$ (m²)	MAE$_{vel}$ (m/s)	SCR （%）
NGSIM	ST-SOFTS	0.89	0.75	91.1
	SOFTS	1.05	0.82	80.5
	STS-DGNN	1.22	0.89	83.2
	TrajectoFormer	1.10	0.85	82.9
	TCN-SA	0.98	0.81	87.7
	KF	1.79	1.08	66.6
	CSI	1.87	1.12	62.1
	LI	2.15	1.23	63.7
HighD	ST-SOFTS	1.27	0.88	92.3
	SOFTS	1.42	0.95	78.5
	STS-DGNN	1.50	0.98	85.2
	TrajectoFormer	1.65	1.02	83.0
	TCN-SA	1.44	0.90	88.8
	KF	2.78	1.38	68.7
	CSI	2.89	1.45	65.2
	LI	3.21	1.56	62.3
CQSkyeyeX	ST-SOFTS	1.92	1.03	88.7
	SOFTS	2.11	1.15	75.2
	STS-DGNN	2.34	1.21	81.3
	TrajectoFormer	2.22	1.18	79.5
	TCN-SA	2.03	1.12	82.7
	KF	3.81	1.69	63.8
	CSI	3.95	1.76	61.5
	LI	4.32	1.89	59.7

Regarding the safety distance compliance rate (SCR), traditional methods exhibit SCR values below 70% (HighD) and 65% (CQSkyeyeX) due to their disregard for inter-vehicle spatial relationships. In contrast, ST-SOFTS increases compliance to 92.3% and 88.7% through its spatial safety loss function. Consider CQSkyeyeX's intersection scenario: Linear Interpolation (LI) achieves an SCR of only 62.1%, with 38% of vehicle pairs violating the minimum safety distance (spacing < 2 m), while ST-SOFTS reduces this proportion to 11.3%. This stark contrast highlights the limitations of traditional methods in high-density interaction environments, where capturing spatial dependencies is critical for realistic trajectory reconstruction.

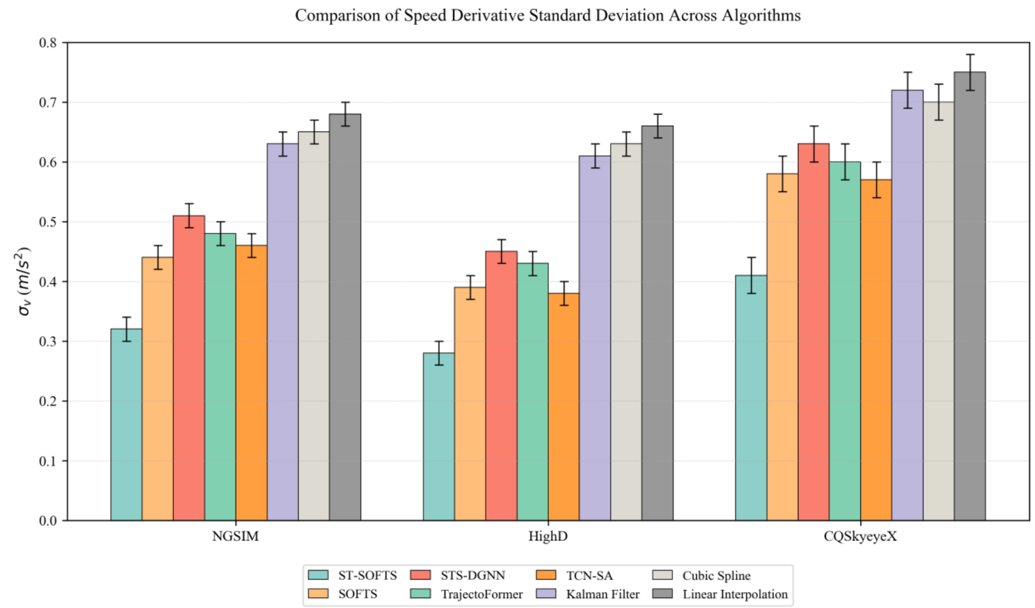

Fig. 3. Comparison of the Standard Deviation of Velocity Change Rates σ_v

Fig. 3 compares the standard deviation of velocity change rates σ_v across all methods. Traditional approaches exhibit σ_v values consistently exceeding 0.6 m/s² (NGSIM: LI=0.68, CSI=0.65, KF=0.63)—significantly higher than deep learning models (ST-SOFTS=0.32). This indicates their inability to capture the temporal continuity of vehicle motion, leading to unrealistic acceleration fluctuations. For instance, in HighD's acceleration lane scenarios, LI yields a σ_v of 0.85 m/s² — three times that of ST-SOFTS — directly contradicting the smooth acceleration dynamics inherent in real-world traffic flow. A validation experiment using a real lane-changing trajectory from CQSkyeyeX is conducted. Fig. 4 visualizes the ground-truth trajectory alongside reconstruction results from ST-SOFTS, Vanilla SOFTS, STS-DGNN, and LI. Notably, the coordinate distribution of ST-SOFTS aligns more closely with the original trajectory than competing methods. This empirically demonstrates its superiority in preserving both spatial accuracy and motion smoothness during complex maneuvers.

The computational efficiency of various methods is presented in Table 3. Traditional approaches, leveraging simplistic computational logic, exhibit batch processing times (for 1,000 vehicles) below 50 ms — significantly faster than deep learning counterparts. Though their parameter counts are less relevant to practical performance evaluation, this speed advantage is notable. As a lightweight deep learning framework, ST-SOFTS incurs a processing latency of 72 ms per batch. While this exceeds the speed of traditional methods, it remains substantially lower than STS-DGNN (125 ms) and TrajectoFormer (210 ms), demonstrating a robust balance between reconstruction accuracy and computational efficiency.

Notably, the low latency of traditional methods comes at the expense of reconstruction fidelity. Their trajectories lack physical plausibility in complex scenarios, failing to meet the requirements of real-world applications — a trade-off effectively addressed by ST-SOFTS through its optimized architectural design.

As shown in Table 4, the experimental results indicate that the three innovative modules of ST-SOFTS synergistically drive performance improvements: removing the spatial weighting mechanism (-SW) leads to a 14.6% increase in NGSIM's MSE$_{pos}$ and a 17.0% increase in HighD's MAE$_{vel}$, demonstrating that dynamic weights of neighboring vehicles are indispensable for spatiotemporal dependency modeling; removing spatiotemporal constraints (-SC) causes an SCR decrease of 9.1% and a σ_v increase of 29.3%, indicating that explicit integration of kinematic constraints significantly enhances trajectory rationality; when simplifying the spatiotemporal embedding layer to a single temporal feature (-STF), CQSkyeye's $\mathrm{MAE}_{\mathrm{vel}}$ increases by 17.5%, validating the necessity of joint spatiotemporal encoding for complex scenarios. When both the spatial weighting mechanism (SW) and spatiotemporal constraints (SC) are removed (-SW & TC), the performance degradation is more pronounced than that of single-module removals. This confirms the synergistic effect between SW and TC, where their joint action is critical to maintaining the model's reconstruction accuracy and spatial consistency.

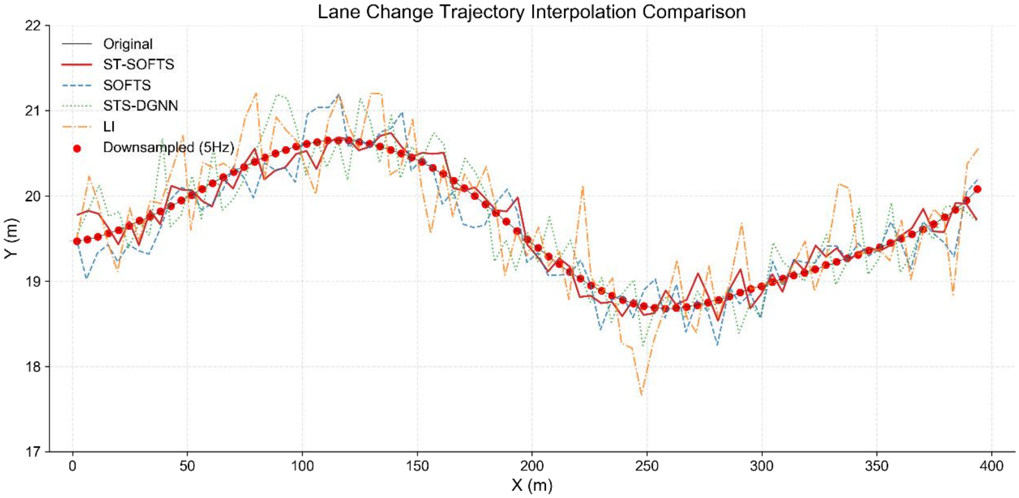

Fig. 4. Schematic Illustration of Lane-Changing Trajectory Reconstruction by Different Methods

Table 3. Parameter Counts and Per-Batch Computational Times of Different Methods

Method	Parameter Count (Million)	Per-Batch Time (ms, 1000 Vehicles)
ST-SOFTS	3.8	72
SOFTS	3.5	68
STS-DGNN	8.7	125
TrajectoFormer	15.2	210
TCN-SA	2.0	61
KF	-	45
CSI	-	38
LI	-	29

Table 4. Ablation Study Configurations and Results

Ablation Configuration	NGSIM $\mathrm{MSE}_{\mathrm{pos}}$	HighD $\mathrm{MAE}_{\mathrm{vel}}$	CQSkyeyeX SCR (%)
Full Model	0.89	0.88	88.7
-SW	1.02 (+14.6%)	1.03 (+17.0%)	83.2 (-6.2%)
-SC	0.95 (+6.7%)	0.97 (+10.2%)	80.5 (-9.3%)
-STF	1.08 (+21.3%)	1.01 (+14.8%)	81.9 (-7.7%)
-SW & TC	1.22 (+37.1%)	1.16(+31.8%)	74.3(-16.2%)

Experimental results demonstrate that although traditional methods are computationally efficient, their reliance on single-vehicle modeling and unconstrained optimization limits performance in complex traffic scenarios. They fail to meet requirements for both accuracy and physical plausibility. By integrating multi-vehicle spatiotemporal interaction modeling (via spatial-weighted core aggregation) and explicit constraint optimization (via spatiotemporal joint loss), ST-SOFTS achieves position error reductions of 27%, 23%, and 17% on NGSIM, HighD, and CQSkyeyeX, respectively, while increasing safety compliance rates by 14%- 18%. These findings validate the effectiveness of the centralized spatiotemporal feature fusion framework.

The linear complexity design of ST-SOFTS results in a processing time one-third that of Transformer-based models for large-scale data, offering a viable solution for real-time applications in intelligent transportation systems. Ablation experiments further confirm that the three innovative modules exhibit synergistic effects — each is indispensable to the algorithm's robustness and generalization capability, laying a solid foundation for its practical utility.

5. Conclusion

To address the core challenges of low-frequency sampling and spatiotemporal sparsity in existing vehicle trajectory datasets, this study proposes ST-SOFTS — a centralized spatiotemporal-enhanced trajectory reconstruction algorithm. By improving the SOFTS framework, it achieves high-frequency reconstruction of multi-vehicle trajectories. The algorithm constructs a three-level architecture: spatiotemporal embedding — weighted aggregation — constraint optimization. Key components include:

Spatiotemporal embedding layer: Encodes inter-vehicle spatiotemporal dependencies via dynamic proximity weighting;

Spatial-weighted core aggregation module: Enables efficient spatiotemporal feature fusion with linear complexity;

Spatiotemporal constraint optimization layer: Enhances trajectory physical plausibility by embedding velocity smoothness and safety distance constraints.

Experiments on three representative datasets — NGSIM, HighD, and CQSkyeyeX — demonstrate that ST-SOFTS significantly outperforms competitors in reconstruction accuracy, spatiotemporal rationality, and computational efficiency: position mean squared error $\mathrm{MSE}_{\mathrm{pos}}$ is reduced by 41%–58% compared to traditional interpolation methods and 17%–27% relative to state-of-the-art deep learning models; safety compliance rates increase by 14%–18%, the standard deviation of velocity change rates σ_v decreases by 27%–38%; parameter count is reduced by over 50%, and the algorithm supports real-time processing of tens of thousands of vehicle trajectories.

Ablation studies validate the necessity of the spatiotemporal weighting mechanism, constraint loss, and joint embedding layer, confirm the synergistic effects of the algorithm's innovative modules. This establishes a robust and efficient solution for high-fidelity trajectory reconstruction in intelligent transportation systems that addresses critical challenges in sparse trajectory data processing.

This study introduces an accuracy-efficient solution for high-frequency reconstruction of low-frequency vehicle trajectories, with engineering value manifest in supplying high-quality data inputs for autonomous driving simulation, enabling microscopic traffic flow modeling, and facilitating safety assessment in transportation systems.

This study introduces an accurate and efficient solution for high-frequency reconstruction of low-frequency vehicle trajectories, with engineering value manifested in three aspects: supplying high-quality data inputs for autonomous driving simulation, enabling microscopic traffic flow modeling, and facilitating safety assessment in transportation systems. In real-world intelligent transportation systems (ITS), these applications translate to tangible improvements across critical operational workflows.

For instance, in connected vehicle (CV) networks, data transmission latency and bandwidth constraints frequently result in sparse trajectory updates, creating “temporal blind spots” of 0.5-1.0 seconds between consecutive data points. These gaps pose significant risks for autonomous vehicle (AV) decision systems, as sudden lane changes or hard braking by adjacent vehicles may go undetected. ST-SOFTS addresses this by functioning as a real-time data augmentation module with sub-50 ms processing latency — its spatial-weighted core aggregation dynamically captures inter-vehicle proximity and temporal continuity. By filling in missing spatiotemporal points with physically plausible trajectories, the model ensures RSUs maintain situational awareness horizon, reducing AV emergency braking events compared to systems relying on raw sparse data. In traffic management centers, congestion prediction models traditionally struggle with low-frequency input data, which blurs the distinction between gradual speed reductions (normal flow) and abrupt decelerations (incipient congestion). ST-SOFTS' ability to generate physically consistent high-frequency trajectories (20-50 Hz) resolves this ambiguity: its spatiotemporal constraint optimization layer enforces velocity smoothness and captures microscopic interactions. Additionally, for vehicle-to-everything (V2X) communication testbeds, ST-SOFTS-derived trajectories provide a cost-effective alternative to high-frequency sensor deployments, enabling realistic simulation of multi-vehicle interaction scenarios without relying on expensive LiDAR or radar arrays.

Despite its advantages, ST-SOFTS has three key limitations that guide future research directions: (1)Limited adaptability to mixed vehicle types: The current physical constraints are calibrated for passenger vehicles, which are the sole vehicle type in the datasets. For heavy-duty vehicles or large commercial vehicles, these fixed thresholds may lead to suboptimal reconstruction accuracy or an increased proportion of physically implausible trajectories. (2) Sensitivity to ultra-low sampling rates: While the model is optimized for 5 Hz input, its performance degrades when the input sampling rate falls below 5 Hz. Because the spatiotemporal fusion layer relies on at least 3-5 consecutive input points to model temporal trends, insufficient data breaks this continuity assumption, leading to increased interpolation uncertainty. (3) Lack of road network constraint integration: ST-SOFTS currently does not incorporate road structural information (e.g., lane width, curve radius, intersection geometry) or traffic rules. This gap can result in reconstructed trajectories that are physically plausible but violate real-world road rules.

Future research may focus on three targeted directions, directly addressing the above limitations while expanding application scenarios: (1) Developing a vehicle-type-aware dynamic constraint framework to replace fixed thresholds, adapting acceleration limits and safety distances based on vehicle category, enhancing adaptability to mixed-vehicle traffic; (2) Designing a multi-scale temporal interpolation module to improve robustness to ultra-low sampling rates, which supplements sparse input points with historical motion trend features to maintain temporal continuity and reduce interpolation uncertainty; (3) Integrating HD map and road rule encoding into the spatiotemporal fusion layer. Incorporating lane geometry, curve radius, and traffic restrictions to ensure reconstructed trajectories comply with both physical laws and real-world road rules, and further extending to complex urban scenarios by fusing multimodal data. These advancements would foster a full-chain upgrade of intelligent transportation systems by addressing unmet challenges in mixed-vehicle, ultra-low-sampling, and complex urban environments.

Acknowledgements

We would like to thank Professor Xu Jin's team from Chongqing Jiaotong University for providing the CQskyeyeX dataset for algorithm validation

References

1. Amin, F., Gharami, K., & Sen, B. (2024). TrajectoFormer: Transformer-based trajectory prediction of autonomous vehicles with spatio-temporal neighborhood considerations. *International Journal of Computational Intelligence Systems*, *17*(1), 87. https://doi.org/10.1007/s44196-024-00410-1
2. Biszko, K., Oskarbski, J., & arski, K. (2024). Impact of different car-following models on estimating safety and emissions on signal-controlled intersections using microscopic simulations. *Archives of Transport, 72*(4), 43-73. https://doi.org/10.61089/aot2024.eb2hfe27
3. Gu, M., Su, Y., Wang, C., & Guo, Y. (2024). Trajectory planning for automated merging vehicles on freeway acceleration lane. *IEEE Transactions on Vehicular Technology*, *73*(11), 16108–16124. https://doi.org/10.1109/TVT.2024.3430291
4. Han, L., Chen, X.-Y., Ye, H.-J., & Zhan, D.-C. (2024). *SOFTS: Efficient multivariate time series forecasting with series-core fusion* (No. arXiv:2404.14197). arXiv. https://doi.org/10.48550/arXiv.2404.14197
5. He, S., Shen, L., Wu, Q., & Yuan, C. (2024). A certified cubic B-spline interpolation method with tangential direction constraints. *Journal of Systems Science and Complexity*, *37*(3), 1271–1294.

https://doi.org/10.1007/s11424-024-2420-0

6. Ip, A., Irio, L., & Oliveira, R. (2021). Vehicle trajectory prediction based on LSTM recurrent neural networks. *2021 IEEE 93rd Vehicular Technology Conference (Vtc2021-Spring)*, 1–5. https://doi.org/10.1109/VTC2021-Spring51267.2021.9449038
7. Jiang, Y., Zhu, B., Yang, S., Zhao, J., & Deng, W. (2023). Vehicle trajectory prediction considering driver uncertainty and vehicle dynamics based on dynamic bayesian network. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(2), 689–703. https://doi.org/10.1109/TSMC.2022.3186639
8. Jiang, J., Zuo, Y., Xiao, Y., Zhang, W., & Li, T. (2024). STMGF-Net: a spatiotemporal multi-graph fusion network for vessel trajectory forecasting in intelligent maritime navigation. *IEEE Transactions on Intelligent Transportation Systems,* *25*(12), 21367-21379. https://doi.org/10.1109/TITS.2024.3465234
9. Lei, J., Xun, Y., He, Y., Liu, J., Mao, B., & Guo, H. (2024). Flexible Multi-Channel Vehicle Trajectory Prediction Based on Vehicle-Road Collaboration.Proceedings - *IEEE Global Communications Conference, GLOBECOM* 2024 IEEE Global Communications Conference, GLOBECOM 2024, December 8, 2024 - December 12, 2024, Cape Town, South africa.
10. Li, F.-J., Zhang, C.-Y., & Chen, C. L. P. (2023). STS-DGNN: Vehicle trajectory prediction via dynamic graph neural network with spatial–temporal synchronization. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–13. https://doi.org/10.1109/TIM.2023.3307179
11. Li, H., & Yu, L. (2025). Prediction of traffic accident risk based on vehicle trajectory data. Traffic Injury Prevention, *26*(2), 164–171. https://www.tandfonline.com/doi/abs/10.1080/15389588.2024.2402936
12. Li, Q., Ou, B., Liang, Y., Wang, Y., Yang, X., & Li, L. (2023). TCN-SA: A social attention network based on temporal convolutional network for vehicle trajectory prediction. *Journal of Advanced Transportation*, *2023*, 1–12. https://doi.org/10.1155/2023/1286977
13. Liu, Z., Fang, J., Tong, Y., & Xu, M. (2020). Deep learning enabled vehicle trajectory map-matching method with advanced spatial–temporal analysis. *IET Intelligent Transport Systems*, *14*(14), 2052–2063. https://doi.org/10.1049/iet-its.2020.0486
14. Lin, X., Tan, C., & Lin, X. (2024). Short term road network Macroscopic Fundamental Diagram parameters and traffic state prediction based on LSTM. *Archives of Transport, 71*(3), 107-126. https://doi.org/10.61089/aot2024.qxx9rh86
15. Quddus, M., & Washington, S. (2015). Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transportation Research Part C: Emerging Technologies*, *55*, 328–339. https://doi.org/10.1016/j.trc.2015.02.017
16. Rathore, P., Kumar, D., Rajasegarar, S., Palaniswami, M., & Bezdek, J. C. (2019). A scalable framework for trajectory prediction. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3860–3874. https://doi.org/10.1109/TITS.2019.2899179
17. Reis, J., Yu, G., & Silvestre, C. (2023). Kalman-based velocity-free trajectory tracking control of an underactuated aerial vehicle with unknown system dynamics. *Automatica*, *155*, 111148. https://doi.org/10.1016/j.automatica.2023.111148
18. Singh, D., & Srivastava, R. (2022). Graph neural network with RNNs based trajectory prediction of dynamic agents for autonomous vehicle. *Applied Intelligence*, *52*(11), 12801–12816. https://doi.org/10.1007/s10489-021-03120-9
19. Waddell, J. M., Remias, S. M., Kirsch, J. N., & Trepanier, T. (2020). Utilizing low-ping frequency vehicle trajectory data to characterize delay at traffic signals. *Journal of Transportation Engineering, Part A: Systems*, *146*(8), 4020069. https://doi.org/10.1061/JTEPBS.0000382
20. Wang, J., Liang, Y., Tang, J., & Wu, Z. (2024). Vehicle trajectory reconstruction using lagrange-interpolation-based framework. *Applied Sciences*, *14*(3), 1173. https://doi.org/10.3390/app14031173
21. Wang, T., Fu, Y., Cheng, X., Li, L., He, Z., & Xiao, Y. (2025). Vehicle Trajectory Prediction Algorithm Based on Hybrid Prediction Model with Multiple Influencing Factors. *Sensors, 25*(4). https://doi.org/10.3390/s25041024
22. Wang, Y., Gao, S., Wang, Y., Wang, P., Zhou, Y., & Xu, Y. (2021). Robust trajectory tracking control

for autonomous vehicle subject to velocity-varying and uncertain lateral disturbance. *Archives of Transport, 57*(1), 7-23. https://doi.org/10.5604/01.3001.0014.7480

23. Yang, X., Liang, G., Chen, Y., Liu, Y., Chen, Z., Yang, X., & Wang, Z. (2025). Research on the Spatiotemporal Evolution of Disturbed Through Vehicles at Signalized Intersections Based on Trajectory Data. *Journal of Transportation Engineering, Part A: Systems, 151*(7), 04025040. https://doi.org/10.1061/JTEPBS.TEENG-9029
24. Yang, G., Wang, Z., Xu, H., & Tian, Z. (2018). Feasibility of using a constant acceleration rate for freeway entrance ramp acceleration lane length design. *Journal of Transportation Engineering, Part A: Systems*, *144*(3), 6017001. https://doi.org/10.1061/JTEPBS.0000122
25. Youssef, T., Zemmouri, E., & Bouzid, A. (2023). STM-GCN: a spatiotemporal multi-graph convolutional network for pedestrian trajectory prediction. *Journal of Supercomputing, 79*(18). https://doi.org/10.1007/s11227-023-05467-x
26. Zhang, Q., Bhattarai, N., Chen, H., Xu, H., & Liu, H. (2023). Vehicle trajectory tracking using adaptive kalman filter from roadside lidar. *Journal of Transportation Engineering, Part A: Systems*, *149*(6), 4023043. https://doi.org/10.1061/JTEPBS.TEENG-7535

Appendix A: Parameter Settings of ST-SOFTS

Parameter Category	Parameter Name	Theoretical Basis	Tuned Value Range	Final Selected Value
Core Model Parameters	Embedding Dimension	Feature representation capability vs. efficiency	32, 64, 128, 256, 512	64
	Spatial Proximity Range	Average inter-vehicle distance in datasets	10 m, 15 m, 20 m, 25 m	20 m
	Number of STAR Layers	Deep network gradient preservation (residual links)	2, 3, 4	3
	MLP Hidden Layer Dimension	Matching embedding dimension for feature transformation	64, 128	64
Constraint Loss Parameters	Temporal Smoothness Loss Weight (w_t)	Prioritizing physical plausibility	0.3, 0.4, 0.5, 0.6	0.4
	Spatial Safety Loss Weight (w_s)		0.4, 0.5, 0.6, 0.7	0.6
Training Parameters	Initial Learning Rate	Adam optimizer standard setting	1e-4, 5e-4, 1e-3, 5e-3	1e-3
	Batch Size	Memory efficiency vs. gradient stability	64, 128, 256	128
	Max Epochs	Avoiding overfitting (early stopping trigger)	150, 200, 250	200
	Early Stopping Patience	Stabilizing validation loss	5, 10, 15	10