TrajNS: Numerical and Semantic Modeling Framework for Realistic and Controllable Trajectory Generation

Dimuthu Lakmal
dimuthu.kariyawasan@unimelb.edu.au
University of Melbourne
Australia

Renata Borovica-Gajic
renata.borovica@unimelb.edu.au
University of Melbourne
Australia

Shanika Karunasekera
karus@unimelb.edu.au
University of Melbourne
Australia

Abstract

Generating realistic and controllable vehicle trajectories remains a key challenge in traffic simulation due to the dynamic and stochastic nature of road traffic. Existing deep learning approaches often rely solely on raw sensory inputs, limiting their ability to capture both fine-grained spatiotemporal dependencies and high-level semantic context. To address this gap, we propose a generalizable framework with two complementary modules. The first performs low-level numerical modeling using a dynamic Spatial-Temporal Graph Neural Network and visual encoder to capture spatial–temporal dynamics and environmental context. The second applies perception-driven semantic modeling via a Multi-Modal Large Language Model pipeline to extract human-like interpretations of traffic scenarios. Outputs from both modules condition a diffusion-based generative model to produce behaviorally feasible, controllable trajectories. Experiments on a real-world dataset show significant improvements over state-of-the-art baselines across multiple metrics, validating the effectiveness of our approach.

CCS Concepts

• **Computing methodologies → Neural networks**.

Keywords

Trajectory Generation, Spatial-Temporal Data, Semantic Reasoning, Diffusion Models, Multi-Modal Large Language Models

ACM Reference Format:
Dimuthu Lakmal, Renata Borovica-Gajic, and Shanika Karunasekera. 2025. TrajNS: Numerical and Semantic Modeling Framework for Realistic and Controllable Trajectory Generation. In *The 33rd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '25), November 3–6, 2025, Minneapolis, MN, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3748636.3762746

1 Introduction

Vehicle trajectory data is crucial for traffic management, autonomous vehicle development, and urban planning [5–7], but real-world data collection is limited by privacy, cost, and technical barriers. Generating realistic and controllable trajectories remains challenging due to traffic's dynamic and stochastic nature and complex multi-agent interactions. Early rule-based and kinematic trajectory generation models offered interpretability but struggled to capture the complexity and variability of real-world traffic. Deep learning approaches improved adaptability, and diffusion models now produce stable and diverse trajectories for complex traffic scenarios. However, most existing methods oversimplify sensory data embeddings, limiting their ability to capture dynamic spatiotemporal dependencies and fail to reveal higher-level intent and behavioral patterns influencing driving behavior.

This study introduces a two-module trajectory generation framework: (1) a numerical modeling module using a dynamic Spatial Temporal Graph Neural Network and visual encoder to capture spatial–temporal dynamics and environmental context, and (2) a semantic modeling module using a Multi-Modal Large Language Model pipeline to produce human-like interpretations of the same data. The dynamic STGNN adapts its graph structure at each timestep to model evolving vehicle interactions more accurately than fixed-graph STGNNs, improving the extraction of complex traffic dynamics and neighbor influence. The semantic module goes beyond low-level sensory features by generating and encoding textual summaries of movements and constraints with an MLLM semantic pipeline. This extracts intent, scene semantics, and behavioral patterns using reasoning capabilities and pretrained world knowledge.

By combining the numerical and semantic modeling components, our framework conditions a diffusion-based generative model on a richer, more comprehensive representation of historical observations. This dual-perspective approach significantly enhances both the realism and controllability of the generated trajectories.

While this paper presents one concrete implementation of our approach, the proposed method can be generalized as a conceptual framework for input modeling in trajectory generation. It encourages analyzing spatiotemporal data from two distinct but complementary perspectives: numerical modeling, which captures precise patterns and interactions, and semantic modeling, which infers intent and context. Human drivers naturally engage in semantic reasoning, considering behaviors, goals, and context, rather than computing exact motion paths. Nonetheless, numerical analysis remains essential for identifying fine-grained interactions that semantic reasoning alone may overlook. By integrating both paradigms, our framework promotes a modular design philosophy, allowing researchers to experiment with different architectures tailored to numerical or semantic processing without being limited to the specific components used in our study. *Source code: https://github.com/DimuthuLakmal/TrajNS.git*

The main contributions of this paper are summarized as follows:

- Dual-perspective input framework combining numerical modeling of vehicle dynamics and interactions with semantic modeling of intent, behavior, and context, improving the

SIGSPATIAL '25, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2086-4/2025/11
https://doi.org/10.1145/3748636.3762746

controllability and realism of diffusion-based trajectory generation.
- MLLM-based pipeline that generates textual summaries from historical spatial data, extracting intent, scene semantics, and behavioral pattern information to facilitate semantic-level reasoning.
- Dynamic STGNN with temporally adaptive graph structures for accurately capturing evolving vehicle interactions in complex traffic.

2 TrajNS: Problem Statement and Architecture

In this section, we first define the key problem statement, followed by the architectural details of TrajNS, our dual-perspective framework. Finally, we detail the training and inference procedures, including rule-based guidance for controllable and realistic trajectory synthesis.

2.1 Problem Statement

Given historical observations over a window of T steps up to time step t, including ego vehicle data $O^e_{t-T+1:t}$, M adjacent vehicles $O^M_{t-T+1:t}$, and the surrounding environment $O^{\text{env}}_{t-T+1:t}$, the goal is to generate the ego vehicle's future trajectory $\tau^e_{t+1:t+H} = \{(s^e_{t+1}, a^e_{t+1}), \ldots, (s^e_{t+H}, a^e_{t+H})\}$ for a horizon of H steps. Each element contains an action $a^e_t = (\dot{v}^e_t, \dot{\psi}^e_t)$ - acceleration and yaw rate, and a state $s^e_t = (v^e_t, \psi^e_t, p^e_t)$ - speed, yaw, and position.

The objective is to learn a trajectory distribution $\pi_\phi(\tau^e_{t+1:t+H} \mid O^e_{t-T+1:t}, O^M_{t-T+1:t}, O^{\text{env}}_{t-T+1:t})$ that approximates the ground truth while satisfying constraints C such as collision avoidance and staying on-road. At each time t, the ego observation is $o^e_t = (v^e_t, \psi^e_t, p^e_t)$, and the m-th adjacent vehicle's observation is $o^m_t = (v^m_t, \psi^m_t, p^m_t)$, with histories $O^e_{t-T+1:t} = \{o^e_{t'}\}^t_{t'=t-T+1}$ and $O^M_{t-T+1:t} = \{(o^1_{t'}, .., o^M_{t'})\}^t_{t'=t-T+1}$. The model generates future trajectories conditioned on this spatiotemporal context to ensure they are feasible, realistic, and compliant with all constraints.

2.2 Architecture

Dual-Perspective Conditioning Architecture: We refer to our architecture as **TrajNS**, which combines numerical and semantic modeling of spatiotemporal data to condition a diffusion-based trajectory generator (Figure 1). The diffusion model adopts a U-Net backbone and is guided by three embeddings from the input pipeline's two modules. The *numerical modeling module* includes a dynamic STGNN and an image encoder. The STGNN extracts spatial–temporal features from historical vehicle observations using a temporally adaptive graph, while the image encoder captures environmental context from visual map data. The *semantic modeling module* uses a Multi-Modal Large Language Model to perform human-like reasoning on historical spatiotemporal data, producing textual descriptions of scene semantics, intent, and behavioral patterns. These descriptions are converted into text embeddings via a text encoder. The three embeddings jointly condition the diffusion model, which outputs an ego-vehicle action sequence $A^e_{t+1:t+H} = \{a^e_{t+1}, \ldots, a^e_{t+H}\}$. Actions are transformed into states $S^e_{t+1:t+H} = \{s^e_{t+1}, \ldots, s^e_{t+H}\}$ using a kinematic model such as the Unicycle model, where $s^e_{t+1} = f(a^e_t, s^e_t)$. The following sections detail the framework's components and diffusion process.

2.2.1 Components of the Numerical Modeling Module. The numerical modeling module components are designed to extract low-level numerical spatiotemporal information from raw sensory data.

Dynamic Spatiotemporal Graph Neural Network: Unlike traditional STGNNs with fixed graph connectivity, our dynamic STGNN adapts to evolving vehicle interactions by combining a temporal transformer encoder with a dynamic Graph Attention Network (GAT). The transformer encoder computes time-dependent node weights $W_t = \{w^1_t, \ldots, w^N_t\}$ from embedded historical features $X \in \mathbb{R}^{T \times N^{(m)} \times F}$, allowing the model to reflect changes in vehicle influence over time. Here, $N^{(m)}$ denotes the maximum number of adjacent vehicles observed in the m^{th} input sequence, which may vary across sequences. These weights are passed to a GAT to capture spatial dependencies, producing latent spatial vectors $G = \{g^1, \ldots, g^N\}$. Finally, Graph Temporal Encoder blocks analyze G across T timesteps to capture temporal patterns, yielding the first set of low-level embeddings.

Image Encoder: Alongside the dynamic STGNN, the numerical modeling module incorporates an image encoder based on a ResNet architecture to capture environmental context from visual maps. Directly converting maps into graph structures is computationally inefficient due to their large node and edge counts. Instead, we generate a sequence of map images for each timestep with vehicle locations superimposed to embed dynamic spatial context. These images are stacked along the channel axis and processed by the ResNet, producing the second type of low-level embedding. This design follows the approach in [10].

2.2.2 Components of the Semantic Modeling Module. The semantic modeling module interprets historical spatiotemporal data through human-like reasoning using an MLLM pipeline to extract high-level intent, behavioral patterns, and contextual insights.

Fine-Tuned Multi-Modal Large Language Model: We employ a fine-tuned QWEN2.5-VL model [1] to capture semantic information from map images and sequences of vehicle locations. Starting from a publicly available pretrained model, we fine-tune it to align with reasoning over historical traffic data. The model outputs textual summaries describing vehicle behaviors and scene-level context.

Text Encoder: The generated summaries, capped at 512 tokens, are tokenized and passed through an Electra encoder [4] to produce compact text embeddings. These high-level semantic embeddings condition the diffusion model alongside the low-level spatiotemporal embeddings from the numerical module.

2.3 Training and Inference

2.3.1 Training. The model is trained via imitation learning to generate trajectories that match those in the training set. We minimize the objective $L(\theta) = \mathbb{E}_{\epsilon, \tau_0, \mathbf{c}} \left[\|\boldsymbol{\tau}_0 - \hat{\boldsymbol{\tau}}_0\|^2 \right]$ where τ_0 is the ground-truth trajectory, $\hat{\tau}_0$ is the reconstructed output, and $\mathbf{c}$ denotes the conditioning features from our dual-perspective input framework.

2.3.2 Inference. For fair comparison with other diffusion-based approaches, we apply rule-based gradient guidance during inference [10] to balance realism and constraint satisfaction. A rule-agnostic trajectory generator is first trained, and during inference it is guided by an iterative gradient optimization process for rule-specific controllability. The reverse diffusion step is approximated

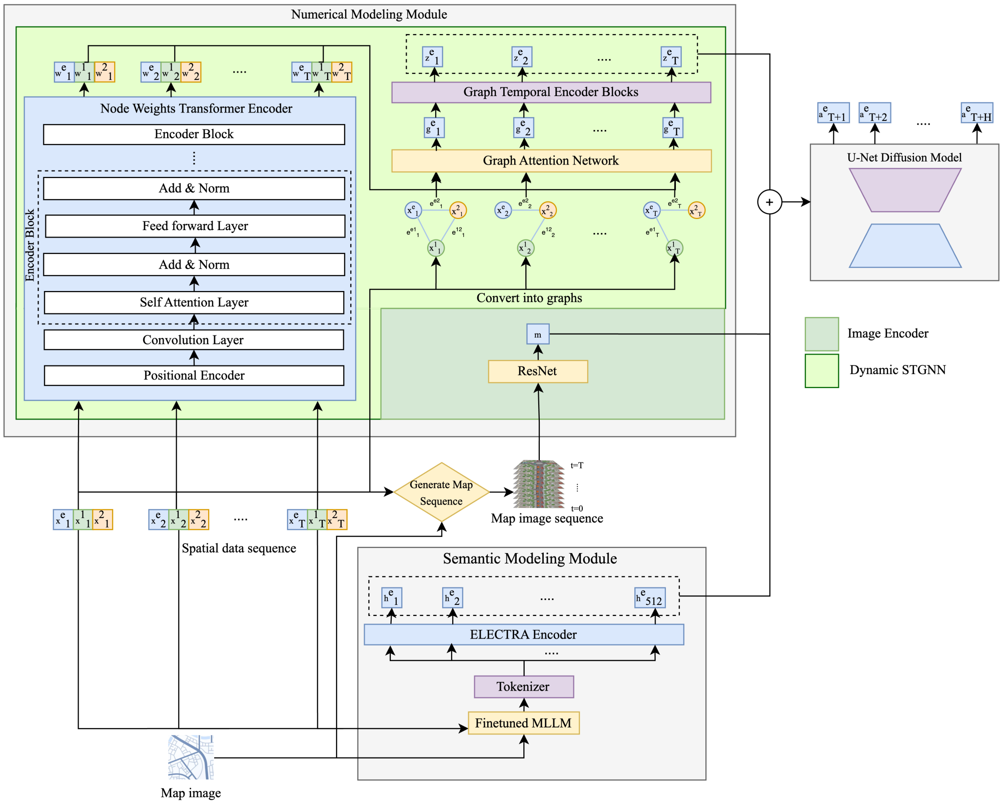

Figure 1: Overall Architecture of TrajNS. Key components are described in Sec. 2.2

Table 1: Comparison of methods for trajectory generation. Lower values indicate better performance for all metrics except *cov*, where higher values are preferable.

	No Collision							No Off-Road						
	ADE	FDE	cov	real	rel-real	rule	fail	ADE	FDE	cov	real	rel-real	rule	fail
SimNet	7.043	17.948	194.70	0.129	0.353	0.375	0.536	7.043	17.948	194.70	0.129	0.353	0.296	0.536
SimNet + opt	6.840	17.543	200.80	0.104	0.346	0.341	0.537	5.124	13.322	239.10	0.135	0.357	0.236	0.522
TrafficSim	5.629	13.902	260.60	0.134	0.356	0.108	0.157	5.629	13.902	260.60	0.134	0.356	0.111	0.157
TrafficSim + opt	5.628	13.901	261.00	0.168	0.348	0.106	0.155	5.640	13.916	258.40	0.143	0.345	0.094	0.163
BITS	2.851	7.004	544.55	0.167	0.386	0.103	0.138	**2.851**	7.004	544.55	0.167	0.386	0.085	**0.138**
BITS + opt	2.933	**6.699**	549.00	0.162	0.381	0.062	0.113	2.902	**6.938**	532.70	0.166	0.378	0.043	0.151
CTG	3.639	9.706	530.10	0.079	0.360	0.084	0.145	3.639	9.706	530.10	**0.079**	0.360	0.084	0.145
CTG + opt	3.604	9.514	536.00	0.094	0.356	0.079	0.134	3.090	8.068	529.90	0.132	0.372	0.037	0.184
TrajNS	3.012	7.775	546.10	0.087	0.343	0.071	0.158	3.012	7.775	546.10	0.087	0.343	0.071	0.158
TrajNS + opt	**2.793**	7.410	**577.10**	**0.075**	**0.321**	**0.059**	**0.078**	2.863	7.573	**555.00**	0.086	**0.333**	**0.031**	0.144

Table 2: Comparison of methods for trajectory generation with inference-time gradient optimization for combination of rules: No Collision + No Off-Road

	No Collision + No Off-Road							
	ADE	FDE	cov	real	rel-real	rule(col)	rule(off-road)	fail
SimNet	5.064	13.223	170.45	0.166	0.340	0.360	0.240	0.521
TrafficSim	5.631	13.904	261.10	0.135	0.341	0.111	0.096	0.160
BITS	**2.250**	**6.036**	587.40	0.158	0.357	0.041	0.037	0.044
CTG	2.783	7.456	**602.65**	0.110	0.347	0.049	0.037	0.047
TrajNS	2.712	7.421	595.60	**0.091**	**0.317**	**0.039**	**0.034**	**0.041**

as $p_\theta(\tau_{t-1} \mid \tau_t, y, c_{\text{inf}}) \approx \mathcal{N}(\tau_{t-1}; \mu_\theta + \Sigma g, \Sigma_\theta)$, where c_{inf} denotes rule-based constraints and Σg is the final gradient from the guidance process. Given initial observations, the model generates future trajectories for subsequent future timesteps, processing all vehicle observations in parallel to enable multi-agent trajectory generation.

3 Experimental Evaluation

We conducted a series of experiments to evaluate how well the proposed framework balances physical feasibility, safety, and generalization across complex driving scenarios, while maintaining controllability over the generated output compared to a set of competitive baselines.

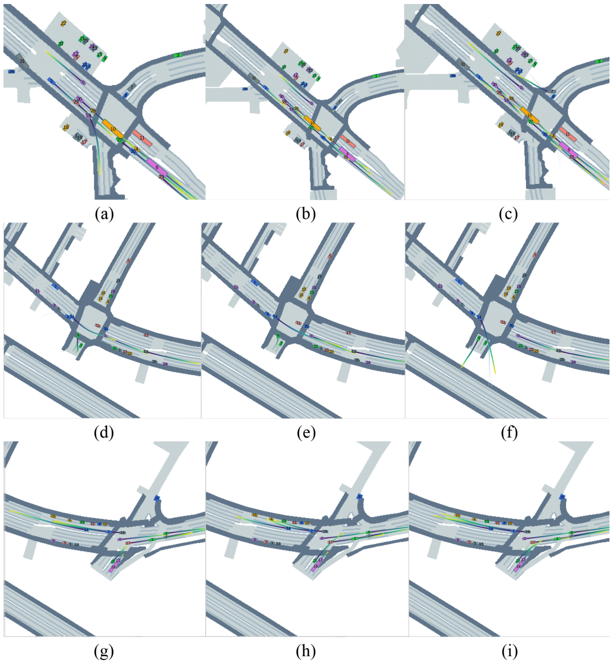

Figure 2: Visual comparison of generated trajectories. (a), (d), (g) represent trajectories generated by TrajNS. (b), (e), (h) represent trajectories generated by CTG. (c), (f), (i) represent trajectories generated by BITS model. All the outputs generated applying inference time optimization for both no-collision and no-off-road rules

3.1 Setup

Dataset: We use the nuScenes dataset [3], containing 1,000 20-second scenes captured in two cities from radar, lidar, and camera sensors installed in autonomous vehicles. For evaluation, we select 50 scenes covering a range of traffic complexities.

Evalution Metrics: We assess realism using the Wasserstein distance between histograms of simulated and ground-truth trajectory profiles, along with the mean of three motion characteristics [9]: longitudinal acceleration, lateral acceleration, and jerk—measured both independently (*real*) and relative to other vehicles (*rel-real*). Controllability is evaluated via rule-specific violations [10]: collision and off-road avoidance, and by the failure rate (*fail*) - the fraction of agents experiencing critical failures. Accuracy is measured using Average Displacement Error (*ADE*) and Final Displacement Error (*FDE*); however, these do not capture the diversity or generalization ability of generative models. We therefore also report *Coverage*, which quantifies the proportion of drivable area occupied by valid trajectories. Together, *ADE*, *FDE*, and *Coverage* provide a holistic evaluation of trajectory accuracy, diversity, and safety.

Baselines: We evaluate our method against four baseline models: CTG [10], BITS [9], SimNet [2], and TrafficSim [8]. All baselines are evaluated both with and without inference-time rule-based optimization.

3.2 Experimental Results

We compare *TrajNS* with competitive baselines under two inference-time settings: (i) single-rule optimization for *No Collision* or *No Off-Road*, and (ii) combined optimization applying both rules [10]. The *+opt* suffix in Table 1 denotes models with rule-based gradient guidance during inference. Every model is applied with inference time optimization in Table 2.

Single-rule evaluation: As shown in Table 1, *TrajNS* achieves strong accuracy and the highest *Coverage*, outperforming CTG and matching or surpassing BITS. While BITS yields slightly better *ADE*, *TrajNS* excels in diversity (*Coverage*) and realism (*real*, *rel-real*) with lower violation and failure rates, indicating better controllability and safety.

Combined-rule evaluation: Under simultaneous *No Collision* and *No Off-Road* constraints (Table 2), *TrajNS* maintains competitive accuracy and coverage while achieving the lowest violation and failure rates and the highest realism scores. Joint optimization improves compliance without degrading accuracy or diversity.

Overall, *TrajNS*- built on dual numerical and semantic modeling - consistently balances accuracy, diversity, realism, controllability, and safety, demonstrating the value of integrating both perspectives in trajectory generation.

4 Conclusion and Future Work

We presented *TrajNS*, a vehicle trajectory generation framework that combines numerical modeling—via a dynamic STGNN and image encoder-with semantic modeling through a fine-tuned MLLM. This dual-perspective design enables a diffusion-based generator to produce accurate, realistic, diverse, and controllable trajectories. Future work will focus on improving robustness in partially observable environments, enhancing performance under limited or sparse observations to increase reliability in safety-critical scenarios.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923* (2025).

[2] Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Błażej Osiński, Hugo Grimmett, and Peter Ondruska. 2021. Simnet: Learning reactive self-driving simulations from real-world observations. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5119–5125.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11621–11631.

[4] K Clark. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555* (2020).

[5] Udesh Gunarathna, Renata Borovica-Gajic, Shanika Karunasekera, and Egemen Tanin. 2022. Dynamic graph combinatorial optimization with multi-attention deep reinforcement learning. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2022*. ACM, 22:1–22:12. doi:10.1145/3557915.3560956

[6] Udesh Gunarathna, Shanika Karunasekera, Renata Borovica-Gajic, and Egemen Tanin. 2022. Real-Time Intelligent Autonomous Intersection Management Using Reinforcement Learning. In *2022 IEEE Intelligent Vehicles Symposium, IV 2022*. IEEE, 135–144. doi:10.1109/IV51971.2022.9827188

[7] Udesh Gunarathna, Hairuo Xie, Egemen Tanin, Shanika Karunasekera, and Renata Borovica-Gajic. 2020. Real-Time Lane Configuration with Coordinated Reinforcement Learning. In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track - European Conference, ECML PKDD 2020*, Vol. 12460. Springer, 291–307. doi:10.1007/978-3-030-67667-4_18

[8] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. 2021. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10400–10409.

[9] Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. 2023. Bits: Bi-level imitation for traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2929–2936.

[10] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. 2023. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3560–3566.