UniTSG: Unified Modeling Token Scenarios Generating for Autonomous Driving Task

Jianrong Xu
Tongji University
jyfootprint@foxmail.com

Baicang Guo
Yanshan University
guobaicang@ysu.edu.cn

Xingchen Liu
Yanshan University
liuxc999@163.com

Wei Hong
Xi'an Technological University
hongwei@st.xatu.edu.cn

Liangliang Li
Xi'an Technological University
liangliangli_xatu@st.xatu.edu.cn

Chenyun Xi
Tongji University
2331672@tongji.edu.cn

Yewei Shi
Yanshan University
shiyw@stumail.ysu.edu.cn

Peng Wang
Xi'an Technological University
wp_xatu@163.com

Ruohai Di
Xi'an Technological University
wp_xatu@163.com

Abstract

Autonomous driving simulation demands both high-fidelity scene generation and scalable, efficient rollout for robust evaluation. Recent advances demonstrate that continuous generative diffusion models yield high realism and controllability but face challenges in efficiency and generalization, while token-based autoregressive models excel in scalability and transferability, yet lack fine-grained editability and strong controllability. In this work, we propose UniTSG—a unified framework that integrates tokenization-based scenario representation with diffusion generative modeling. UniTSG encodes agent trajectories and vectorized maps into discrete token sequences, leverages discrete/embedding-space diffusion for scene inpainting and rollout, and supports efficient, controllable, and scalable simulation. Experiments on Waymo Open Motion Dataset and NuPlan demonstrate UniTSG's effectiveness in multi-agent closed-loop simulation, scenario editing, and cross-domain generalization.

1. Introduction

Driven by the wave of digitalization and intelligence, autonomous driving technology has become the most revolutionary force in the transportation field. As a key link in this domain, autonomous driving scene generation attracts significant attention from researchers, engineers, and traffic planners worldwide due to its complexity, diversity, and professionalism. It serves not only as the core foundation for the development and testing of autonomous driving systems but also as the key support for the comprehensive implementation of intelligent transportation systems, with its importance being self-evident. This process involves not only the fine-tuning of the sensing module but also the rigorous testing of the planning and control modules in simulation environments to ensure their stable operation in complex and dynamic traffic scenarios [1, 21, 27].

Traditionally, the evaluation of the perception module has largely depended on standardized annotation data from large-scale real-world datasets [9]. These meticulously annotated data provide a robust basis for algorithm training and performance evaluation. However, testing the planning and control modules has predominantly taken place within simulators, relying on manually crafted scenes. Although these hand-made scenes can simulate specific traffic situations to some extent, they are often overly simplified and fail to replicate the intricate map structures, diverse behaviors of traffic participants, and rapidly changing road conditions found in the real world [16, 26]. Moreover, the manual construction of test scenarios is both time-consuming and labor-intensive, and it demands a high level of expertise from the testers, especially when dealing with complex scenarios involving numerous traffic participants. This makes the large-scale expansion of comprehensive evaluation and training of autonomous driving decision-making across diverse scenarios extremely challenging. Meanwhile, mining traffic scenes from real-world driving data offers a novel approach to addressing data accessibility issues. By replaying these real-world scenarios in simulation environments, we can provide a more realistic benchmark for evaluating the decision-making capabilities of autonomous driving systems. This, in turn, enhances the

generalization ability of data-driven planning and control components, enabling them to make sound decisions even when confronted with unfamiliar road conditions. However, numerous critical challenges must be overcome in the development of large-scale data-driven simulation platforms [13, 25]. Firstly, well-annotated driving data are extremely scarce and valuable. There is a pressing need to maximize the utility of such data by extracting as much effective information as possible for training and evaluation purposes. However, achieving this goal is highly challenging due to the diverse origins and formats of publicly available driving data. This diversity creates a significant bottleneck in data aggregation, severely limiting the integration and utilization of multi-source data. Consequently, this impacts the training effectiveness and evaluation accuracy of autonomous driving systems. Secondly, existing datasets are often tightly coupled with specific simulators or toolkits, which greatly restricts data sharing. For instance, some datasets are specifically designed for imitation learning involving a single agent, while others are tailored for multi-agent reinforcement learning (RL) scenarios [29] under partial observability. Still, others focus on scene generation tasks. This specificity makes cross-dataset fusion training extremely difficult, with compatibility issues between different datasets emerging as a key constraint on the development of autonomous driving technology. Additionally, although some driving datasets provide abundant raw sensor data, such as high-resolution images and high-precision point clouds, existing two-dimensional data-driven simulators are unable to fully leverage these three-dimensional data [15, 20]. In summary, the advancement of autonomous driving technology urgently requires a novel scene generation method. This method should break down the barriers between existing datasets and simulators, efficiently integrate multi-source heterogeneous data, and fully utilize three-dimensional sensor data. By doing so, it can provide rich, realistic, and diverse test scenarios for the planning and control modules of autonomous driving systems. This will promote the continuous progress of autonomous driving technology towards safety, reliability, and efficiency, and accelerate its transition from laboratory research to practical application.

However,Simulation is crucial for the development and safety evaluation of autonomous driving systems, supporting the generation of diverse and challenging traffic scenarios. Traditional continuous-space generative models achieve high realism and controllability through iterative denoising but often suffer from high computational cost and limited generalization. Conversely, token-based autoregressive (LLM-style) models show superior scalability and zero-shot transfer by discretizing trajectories and maps, but their editability and precise control are restricted by the coarseness of tokenization and lack of explicit constraint mechanisms [11, 14, 19, 28].

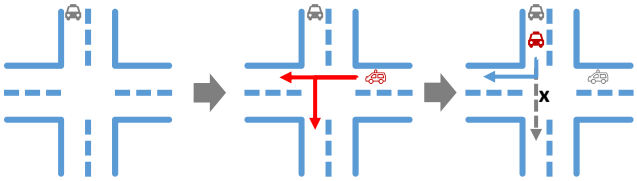

Figure 1. The solution process is illustrated in the figure, which shows the iterative procedure of UniTSG in the scene generation task. Gray vehicles represent the one-second pose history of the ego agent on the provided map. Red vehicles indicate all attributes of newly injected agents within the scene context. Blue trajectories denote future trajectories, while gray trajectories represent discarded future trajectories.

To address these limitations, **UniTSG** addresses these challenges by unifying the strengths of both paradigms: we tokenize agent motions and road topology into discrete codebooks, and perform conditional scenario generation and rollout via discrete or embedding-space diffusion processes. This hybridization achieves (1) high-fidelity and controllable scene synthesis, (2) fast and scalable simulation, and (3) robust transferability to novel domains. The solution process is illustrated in the Fig. 1. Our contributions:

- A unified tokenization scheme for agent trajectories and vectorized maps.
- A token/embedding-space diffusion model for efficient scenario generation and editing.
- Strong experimental results on public datasets and ablation on key design choices.

2. Related Work

2.1. Scene Generation Methods

Cao Y [2] proposed a virtual-real fusion testing method that integrates Graph Theory and Artificial Potential Fields (APF) for autonomous vehicle testing. Conducted using SUMO software, the method outperformed traditional Rapidly-exploring Random Tree (RRT) methods in handling vehicle dynamics and environmental interactions, with a 41% faster operation completion time in simulations and 55% faster in real-world tests. Field experiments in Suzhou High-Speed Railway New Town validated its practicality and robustness. This approach enhances the authenticity and efficiency of testing, promoting the development of reliable autonomous driving systems and improving testing processes. A-A Fime [6] analyzed methods such as image-to-3D conversion, text-to-3D generation, UI/layout design, graph-based methods, and interactive scene generation. The article also discussed evaluation metrics like FID,

KL divergence, IS, IoU, and mAP. It highlighted challenges in the field, including maintaining scene realism, handling complex scenes with multiple objects, and ensuring consistency in object relationships and spatial arrangements. Fremont [7] introduced Scenic, a new probabilistic programming language for designing and analyzing machine learning-based perception systems, especially for dealing with rare events, testing performance, and debugging faults. Scenic specifies distributions of input types and generates training and testing sets through sampling to address these issues. ChatScene employs an agent based on a large language model (LLM) to generate safety-critical scenarios for autonomous vehicles. Given unstructured instructions, the agent uses the LLM to generate textual descriptions of traffic scenarios, breaks them down into specific details, and converts them into domain-specific languages, ultimately generating actual code for the CARLA simulation environment. ChatScene includes a knowledge retrieval component that efficiently translates textual descriptions into code snippets by training a database of scenario descriptions and code pairs. Experiments showed that ChatScene-generated scenarios increased the collision rate of reinforcement learning ego vehicles by 15% compared to the baseline, and fine-tuning with the generated safety-critical scenarios reduced the collision rate by 9%, outperforming existing methods. ChatScene effectively bridges the gap between textual descriptions and CARLA simulations, providing a unified method for safety testing and improvement of autonomous vehicles [30]. Lu [17] proposed SceneControl, a controllable traffic scene generation framework addressing the limitations of manual creation being non-scalable and automatic generation lacking realism. SceneControl captures the complexity of real traffic by learning expressive diffusion models and uses guided sampling to flexibly generate scenes with desired characteristics. Experiments demonstrated its superior realism and controllability compared to existing technologies, and it can also serve as an interactive scene generation tool.

2.2. Motion Prediction

The key to the motion prediction problem lies in modeling the future motion of agents based on their current state and recent motion trajectories. Significant progress has been made in this field through improvements in input modeling [8, 12], output modeling [18], agent interaction [4, 22], and multimodal modeling [3, 5]. These improvements have significantly enhanced the accuracy and robustness of motion prediction models, enabling them to more reliably predict the behavior of agents in a variety of complex scenarios. However, these existing methods cannot be directly applied to the task of scene generation. The reason is that these methods essentially rely on the current state and historical motion trajectory information of all agents in the scene. In other words, they require a comprehensive understanding of the agents' historical motion and current positions to generate accurate predictions. This dependence on detailed agent-specific data limits their applicability in scenarios where such information is unavailable or impractical to obtain. For example, in large-scale traffic scenarios, obtaining the complete historical trajectories of each agent can be both time-consuming and resource-intensive, posing challenges for the practical application of existing methods.

2.3. Autoregressive Generative Models

Currently, comprehensive generative models have been widely used in the problem of multi-agent motion prediction, covering continuous motion distribution regression models, diffusion models, and discrete autoregressive models. Among them, MotionDiffuser [10], with its diffusion-based representation method and leveraging a simple predictor design and PCA compression, can efficiently and high-performantly simulate the joint distribution of future trajectories of multiple agents, thereby achieving multi-agent motion prediction. However, although diffusion-based models can generate multimodal future trajectories for individual agents, they can only capture possible agent motions and are unable to simulate the interactions between the future motions of agents. Typical continuous motion distribution regression models generally use parametric continuous distributions such as Gaussian or Laplace distributions to simulate future motion distributions, but these models have limitations, mainly in the uncertainty of whether Gaussian or Laplace mixture distributions are flexible enough to represent future state distributions. In addition, in order to generate multimodal future motions, these models usually need to incorporate candidate motion targets or learnable latent embeddings as multimodal queries into the decoder module, which not only occupies a large amount of memory but also increases inference time. In contrast, MotionLM [24] regards multi-agent motion prediction in autonomous driving scenarios as a language modeling task, generating interactive trajectories through a simplified autoregressive process without the need for complex optimization processes and latent anchor embeddings. Building on this, Trajeglish [23] has further been specifically designed and optimized for multi-agent offline closed-loop simulation. Neural autoregressive models have achieved success in various fields, such as text and 3D indoor scene generation. In terms of traffic scene generation, SceneGen employs an autoregressive method to insert agents into a scene one at a time. Based on this setup, new agents can be adjusted according to the initial state and future trajectories of existing agents, thereby enhancing the realism and consistency of the scene while reducing the conflict rate.

3. Methodology

3.1. Tokenization of Agent Trajectories and Road Maps

3.1.1 Agent Motion Tokenization

Given continuous multi-agent trajectories $\mathbf{X} = x_{a,t} \in \mathbb{R}^d \mid a = 1..A, t = 1..T$, we segment each trajectory into fixed-length windows of size L, e.g., $L = 5$ frames. Each segment $s_{a,i} = [x_{a,i}, x_{a,i+1}, ..., x_{a,i+L-1}]$ is flattened and aggregated across the dataset.

Using k-means clustering, we construct a codebook $\mathcal{V}\text{motion} = c_j j = 1^K$:

$$c_j = \arg\min_c \sum_{s \in \text{cluster}_j} \|s - c\|_2^2, \quad j = 1..K \tag{1}$$

Each segment is assigned to its closest cluster center, i.e., token id:

$$\tau_{a,i} = \arg\min_j \|s_{a,i} - c_j\|_2^2 \tag{2}$$

The full trajectory becomes a token sequence $\tau a = [\tau a, 1, ..., \tau_{a,N_L}]$.

3.1.2 Road Map Tokenization

Similarly, the vectorized road graph is partitioned into a series of polyline segments, each with a fixed and uniform length (e.g., no greater than 5 meters). This segmentation process ensures that the road network is represented in a consistent and manageable format. For each resulting segment, a comprehensive set of features is extracted, including geometric properties such as curvature, orientation, and spatial coordinates, as well as semantic attributes like road type (e.g., highway, residential street, or alley). These features are then subjected to a clustering procedure, which groups similar segments based on their shared characteristics. The outcome of this clustering process is a compact and structured representation known as the road codebook, denoted as $\mathcal{V}_{\text{road}}$. This codebook serves as a foundational resource for efficiently encoding, analyzing, and modeling the road network, enabling downstream applications.

3.1.3 Rolling Matching and Noised Tokenization

To enhance the robustness of the model, the tokenization process employs a rolling window mechanism. Specifically, the token assigned at step i is determined based on the reconstructed end state of the token from the previous step. This approach ensures that the tokenization process dynamically adapts to the evolving context, thereby improving the model's ability to handle sequential dependencies and maintain consistency across steps. During the training phase, an additional layer of complexity is introduced to simulate real-world challenges such as compounding errors and distributional shifts. With a probability p_{noise}, instead of selecting the closest matching token, a random token from the top-k nearest tokens is chosen. This deliberate injection of noise serves two key purposes: first, it forces the model to learn more resilient representations by exposing it to suboptimal or noisy inputs; second, it mimics the effects of cumulative errors that might occur in practical applications, where predictions based on prior states may not always align perfectly with the ground truth. By incorporating this stochastic element, the model becomes more adept at generalizing to unseen data and handling scenarios where the input distribution deviates from the training distribution.

3.2. Token-level Diffusion Model

We represent the scenario as a set of token sequences: $\mathcal{T} = \tau_{a,i}$ for motion, $\mathcal{R} = \rho_j$ for road segments.

3.2.1 Discrete Diffusion Process

Forward Process: At each timestep t, with noise schedule β_t, each token in sequence is randomly replaced as follow: Eq 5.

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \prod_i [(1-\beta_t)\delta(z_{t,i} = z_{t-1,i}) + \beta_t \pi(z_{t,i})] \tag{3}$$

Where π is uniform or learned replacement distribution.

Reverse Process: A neural network parameterized by θ predicts the posterior:

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \text{context}) = \prod_i p_\theta(z_{t-1,i}|\mathbf{z}_t, \text{context}) \tag{4}$$

With cross-entropy loss against the original tokens.

3.2.2 Embedding-space Diffusion

Alternatively, tokens are embedded into $\mathbb{R}^d$ and diffusion/denoising operates in continuous space as in DDPM. Let $e_i = \text{Embed}(z_i)$. The noisy embedding is:

$$q(\mathbf{e}_t|\mathbf{e}_0) = \mathcal{N}(\alpha_t \mathbf{e}_0, \sigma_t^2 I) \tag{5}$$

The model is trained to predict the clean embedding or noise.

3.2.3 Conditional Inpainting and Masking

For scenario generation/editing, a mask $m \in 0, 1^N$ indicates observed (fixed) vs. unobserved (to generate) tokens. During training and inference, masked positions are replaced with noise/unknown tokens.

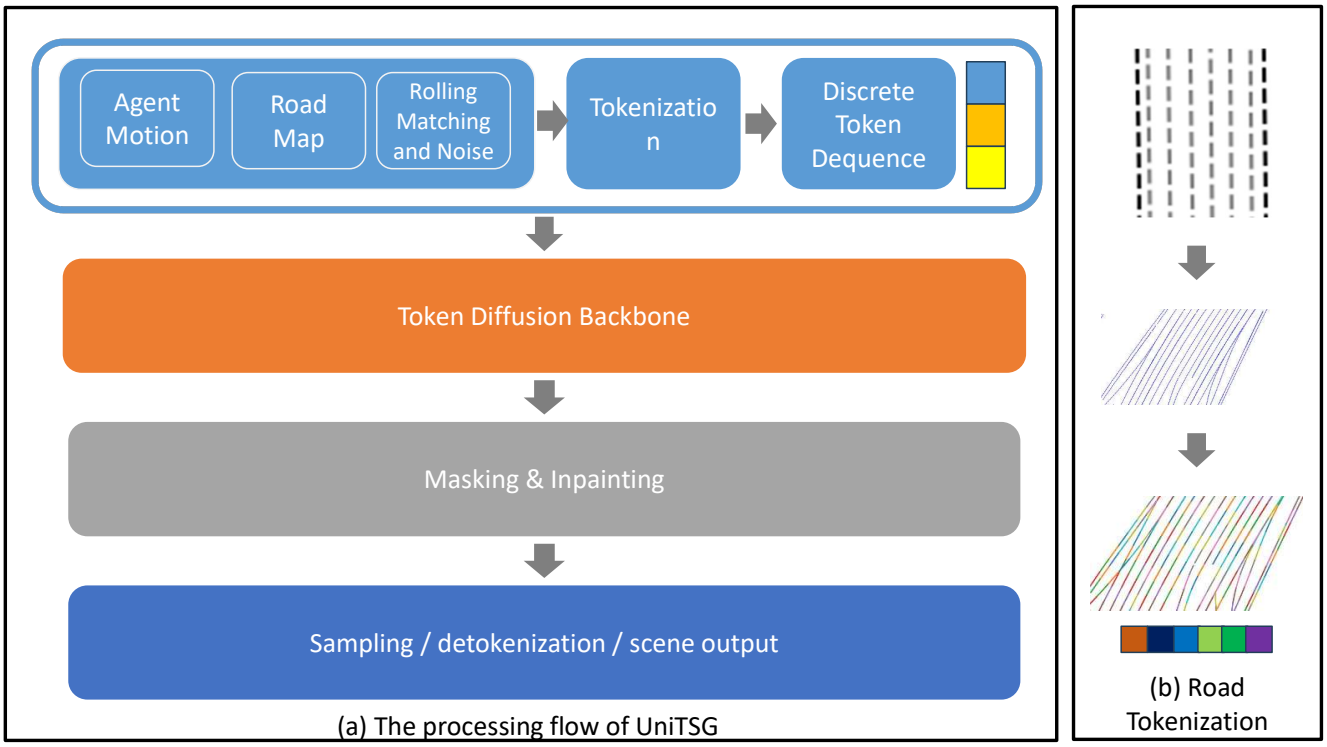

Figure 2. The figure provides a detailed illustration of the overall workflow of the UniTSG algorithm. Part (a) presents the core design and modular architecture of UniTSG, demonstrating how the various functional modules are organically integrated to achieve a complete processing pipeline from input to output. Part (b) illustrates the transformation logic of how map data is converted from its raw form into a discretized representation suitable for model processing.

3.2.4 Architecture

The architecture of the model is designed to handle complex spatio-temporal interactions between agents (e.g., vehicles, pedestrians, cyclists) and their environment (e.g., road topology). It leverages a modular structure that integrates embedding layers, condition encoding, a diffusion backbone, and an output head to generate meaningful predictions. Fig 2 is a detailed breakdown of each component.

Embedding Layer: Both motion and road tokens are mapped to learnable embedding representations, enabling flexible and expressive token encoding.

Condition Encoder: Contextual information, including road topology and agent types, is encoded using a Transformer or Perceiver architecture to capture complex dependencies and structural relationships.

Diffusion Backbone: A series of Transformer layers, augmented with cross-agent and spatio-temporal attention mechanisms, are employed to iteratively predict the distribution of tokens at each diffusion step.

Output Head: For each position, a softmax operation is applied over the codebook to generate probability distributions, facilitating precise token selection during decoding.

4. Experiments

4.1. Dataset

The Waymo Open Motion Dataset (WOMD), released by Waymo, an autonomous driving subsidiary of Alphabet, Google's parent company, is a large - scale, multi - modal open dataset. Against this backdrop, the WOD Motion Dataset emerged. It records agent data at a frequency of 10 Hz. Samples in the training and validation sets contain 10 steps of historical data, 1 step of current data, and 80 steps of future data, totaling 91 time steps. The sequence is $t-10, t-9, \ldots, t-1, t, t+1, \ldots, t+79, t+80$, and the presence status of agents is identified by valid data attributes. The dataset attributes cover agent positions (x, y, z), headings (bbox_yaw), bounding box dimensions (length, width, height), and types (vehicles, pedestrians, cyclists).

4.2. Metrics

4.2.1 Realism Meta Metric

It is a comprehensive measurement method used to evaluate the similarity between the distribution of simulation trajectories and the distribution of real data. Its core goal is to judge the modeling accuracy and credibility of the simula-

Method Name	Realism Meta metric	Kinematic metrics	Interactive metrics	Map-based metrics
SimFormer	0.6623	0.5416	0.7417	0.6293
UniTSG	**0.6604**	**0.5415**	**0.7378**	**0.6288**
OffReg-IDM	0.6185	0.4815	0.7197	0.5668
infgen-full-large	0.6030	0.5044	0.6774	0.5638

Table 1. Performance Comparison on WOMD

tion system by quantifying the difference between simulation data and real data. This indicator uses the approximate negative log-likelihood (NLL) as the calculation basis, aiming to capture the global consistency of the statistical characteristics of the simulation trajectory.

4.2.2 Kinematic Metrics

Used to quantify the motion characteristics of agents in dynamic environments, aiming to evaluate the physical rationality and authenticity of their behavior from the perspective of speed and acceleration. These indicators can not only reflect the instantaneous motion state of the agent, but also reveal its dynamic change rules in time series, thereby ensuring that its behavior conforms to the physical constraints of real traffic participants.

4.2.3 Interactive Metrics

It is a class of key performance indicators used to evaluate the interactive behavior of intelligent agents in a dynamic multi-participant environment. These indicators measure the rationality, safety and adaptability of the behavior of intelligent agents in complex traffic scenarios by quantifying the interactive characteristics between the intelligent agents and the surrounding environment and other traffic participants.

4.2.4 Map-based Metrics

A class of key performance metrics used to evaluate the relationship between an autonomous driving agent and a road map. These metrics are designed to ensure that the agent's behavior complies with road rules and map constraints, thereby improving driving safety and compliance.

5. Results

We systematically compare our proposed UniTSG method with several recent state-of-the-art baselines on the official Waymo competition platform. The evaluation metrics include Realism Meta metric, Kinematic metrics, Interactive metrics, and Map-based metrics. The experimental results are summarized in the table.

As shown in the table, **UniTSG** achieves performance very close to the best method (SimFormer) across all four major metrics, and significantly outperforms other baseline methods. Specifically, **UniTSG** achieves a Realism Meta metric of 0.6604, a Kinematic metrics score of 0.5415, an Interactive metrics score of 0.7378, and a Map-based metrics score of 0.6288, all of which are markedly higher than OffReg-IDM, infgen-full-large, and the official baseline. Detailed analysis is as follows:

For the Realism Meta metric and Kinematic metrics, UniTSG's scores are almost identical to those of SimFormer, with differences as small as 0.002 and 0.0001, respectively. This demonstrates that our model can effectively capture the realistic distribution and kinematic rationality of the scenes.

In terms of Interactive metrics and Map-based metrics, UniTSG also achieves performance nearly on par with the best baseline, further demonstrating the advantage of our approach in modeling multi-agent interactions and scenario feasibility.

Compared to OffReg-IDM and infgen-full-large, UniTSG shows substantial improvements in all evaluation metrics, especially in Kinematic metrics and Interactive metrics, with increases of approximately 6 percentage points, highlighting the effectiveness of the proposed method in high-level scene modeling and agent behavior interaction.

In summary, the experimental results fully demonstrate the effectiveness and superiority of UniTSG in multi-agent traffic scene generation tasks, achieving state-of-the-art performance on multiple major metrics and showing strong practical potential.

6. Conclusion

We present UniTSG, a unified token-diffusion framework for scenario generation in autonomous driving. By combining discrete tokenization with diffusion-based inpainting and rollout, UniTSG achieves high realism, efficiency, and strong controllability, with broad applicability to large-scale AV simulation and safety evaluation.

References

[1] Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Błazej Osinski, Hugo Grimmet, and Peter Ondruska. Simnet: Learning reactive self-driving simulations from real-world observations. In *ICRA*, 2021. 1

[2] Y Cao, H Sun, G Li, et al. Multi-environment vehicle trajectory automatic driving scene generation method based on simulation and real vehicle testing. *Electronics*, 14(5):1000, 2025. 2

[3] Y Chai, B Sapp, M Bansal, and D Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 3

[4] M-F Chang, J Lambert, P Sangkloy, J Singh, S Bak, A Hartnett, D Wang, P Carr, S Lucey, D Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8748–8757, 2019. 3

[5] H Cui, V Radosavljevic, F-C Chou, T-H Lin, T Nguyen, T-K Huang, J Schneider, and N Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019. 3

[6] A A Fime, S Mahmud, A Das, et al. Automatic scene generation: State-of-the-art techniques, models, datasets, challenges, and future prospects. *arXiv preprint arXiv:2410.01816*, 2024. 2

[7] D J Fremont, T Dreossi, S Ghosh, et al. Scenic: a language for scenario specification and scene generation. In *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, pages 63–78, 2019. 3

[8] J Gao, C Sun, H Zhao, Y Shen, D Anguelov, C Li, and C Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. 3

[9] K Gupta, A Singhal, A Kumar, et al. Enhancing sensor perception: Integrating multi-sensor data for robust sensor perception. In *2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS)*, pages 78–83. IEEE, 2025. 1

[10] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9644–9653, 2023. 3

[11] Chiyu Max Jiang, Yijing Bai, Andre Cornman, Christopher Davis, Xiukun Huang, Hong Jeon, Sakshum Kulshrestha, John Lambert, Shuangyu Li, Xuanyu Zhou, Carlos Fuertes, Chang Yuan, Mingxing Tan, Yin Zhou, and Dragomir Anguelov. Scenediffuser: Efficient and controllable driving simulation initialization and rollout. In *NeurIPS 2024: Conference on Neural Information Processing Systems*, 2024. arXiv:2412.12129. 2

[12] J Kim, R Mahjourian, S Ettinger, M Bansal, B White, B Sapp, and D Anguelov. Stopnet: Scalable trajectory and occupancy prediction for urban autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8957–8963. IEEE, 2022. 3

[13] T Kurc, U Catalyurek, X Zhang, et al. A simulation and data analysis system for large-scale, data-driven oil reservoir simulation studies. *Concurrency and Computation: Practice and Experience*, 17(11):1441–1467, 2005. 2

[14] Quanyi Li, Zhenghao Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. In *NeurIPS 2023: Conference on Neural Information Processing Systems, Track on Datasets and Benchmarks*, 2023. arXiv:2306.12241. 2

[15] W Li, C W Pan, R Zhang, et al. Aads: Augmented autonomous driving simulation using data-driven algorithms. *Science robotics*, 4(28):eaaw0863, 2019. 2

[16] Y Liu and F Liu. A knowledge-driven method for autonomous driving simulation scene generation. In *2023 8th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, volume 8, pages 495–498. IEEE, 2023. 1

[17] J Lu, K Wong, C Zhang, et al. Scenecontrol: Diffusion for controllable traffic scene generation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16908–16914. IEEE, 2024. 3

[18] R Mahjourian, J Kim, Y Chai, M Tan, B Sapp, and D Anguelov. Occupancy flow fields for motion forecasting in autonomous driving. *IEEE Robotics and Automation Letters*, 7(2):5639–5646, 2022. 3

[19] Reza Mahjourian, Rongbing Mu, Valerii Likhosherstov, Paul Mougin, Xiukun Huang, Joao Messias, and Shimon Whiteson. Unigen: Unified modeling of initial agent states and trajectories for generating autonomous driving scenarios. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. arXiv:2405.15677. 2

[20] F Mütsch, H Gremmelmaier, N Becker, et al. From model-based to data-driven simulation: Challenges and trends in autonomous driving. *arXiv preprint arXiv:2305.13960*, 2023. 2

[21] Li Penghui, Dong Qianru, Yuan Henan, et al. Generation method of high coverage cut-in scene library for automatic driving simulation test. *China Journal of Highway and Transport*, 37(07):237–249, 2024. 1

[22] T Phan-Minh, E C Grigore, F A Boulton, O Beijbom, and E M Wolff. Covernet: Multimodal behavior prediction using trajectorysets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083, 2020. 3

[23] Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Learning the language of driving scenarios. *arXiv preprint arXiv:2312.04535*, 2023. 3

[24] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023. 3

[25] S Svorobej, J Byrne, P Liston, et al. Towards automated data-driven model creation for cloud computing simulation. In *Eighth EAI International Conference on Simulation Tools and Techniques*, pages 248–255. Association for Computing Machinery, 2015. 2

[26] Shuhan Tan, Kelvin Wong, Shenlong Wang, et al. Scenegen: Learning to generate realistic traffic scenes. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 892–901, 2021. 1

[27] Li Wei-nan, Wang Yu, Li Linrun, et al. Overview of simulation test scenario generation methods for autonomous vehicles. *Automotive Engineer*, (07):1–10, 2024. 1

[28] Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. Smart: Scalable multi-agent real-time simulation via next-token prediction. In *NeurIPS 2024: Conference on Neural Information Processing Systems*, 2024. arXiv:2405.15677. 2

[29] H Zhang, S Feng, C Liu, et al. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The world wide web conference*, pages 3620–3624, 2019. 2

[30] J Zhang, C Xu, and B Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15459–15469, 2024. 3