Simformer: 1st Place in the Waymo Open Scenario Generation Challenge 2025

Sen Wang, Xu Jianrong, Xiaoyong Zhang, Fangqiao Hu, Zhijun Huang,
JiaChen Luo, Kechen Zhu, Jiaxiang Zhu, Yong Zhou, Zhenwu Chen
SUTPC (SHENZHEN URBAN TRANSPORT PLANNING CENTER CO.,LTD)
{wangsen, xujianrong, zhangxiaoyong, hufangqiao, huangzhijun,
luojc, zhukechen, zhujiaxiang, zhouyong, czw}@sutpc.com

Abstract

*We present **Simformer**, the 1st place solution for the Waymo Open Scenario Generation Challenge 2025. Simformer is a unified framework for large-scale, realistic multi-agent traffic scenario generation, drawing inspiration from UniGen and SMART. Our method discretizes agent positions, attributes, and trajectories into interpretable tokens via ego-centric clustering and joint attribute binning. Leveraging a GPT-style Transformer, Simformer autoregressively generates scenario tokens conditioned on map context and the ego route. Technical innovations include efficient token vocabulary construction, a reverse-matching strategy for trajectory initialization, and a scalable autoregressive architecture. Extensive experiments on the Waymo Open Motion Dataset demonstrate that Simformer achieves state-of-the-art performance in realism, diversity, and physical plausibility, setting a new benchmark for data-driven autonomous driving simulation. The code will be made publicly available at https://github.com/XiaomuWang/SimFormer.*

1. Introduction

Autonomous vehicles (AVs) must be evaluated in a broad spectrum of realistic, complex, and safety-critical scenarios. However, collecting rare events from the real world is challenging. Recent scenario generation frameworks such as UniGen [5] unify the modeling of agent placement and behavior, but rely on continuous distributions, which complicate training and limit scalability. Alternatively, SMART [8] introduces discrete tokenization—converting continuous states into discrete tokens—and a decoder-only Transformer (GPT-style) for autoregressive next-token prediction, achieving better generalization and efficiency. Simformer combines these advances: it discretizes agent initial positions, attributes, and trajectories, learning to generate scenarios via an autoregressive Transformer. Simformer enables high-quality, efficient, and generalizable scenario generation for AV simulation and testing.

2. RELATED WORK

Scenario generation for autonomous driving has evolved from rule-based simulation to advanced data-driven generative models. Rule-based and simulator methods (e.g., CARLA [2], SUMO [4]) rely on predefined traffic rules or microscopic simulation engines to synthesize scenarios, but often lack realism and diversity, especially for rare events. With the availability of large-scale datasets, continuous regression-based models such as SimNet [1], TrafficGen [3], and UniGen [5] emerged. These approaches model agent positions, attributes, and trajectories as continuous variables, typically regressing to real-world distributions using neural networks. While effective for basic realism, such models face challenges in diversity, scalability, and efficient generalization. Recently, discrete tokenization and autoregressive methods have shown strong promise. Inspired by language modeling, these approaches (e.g., SMART [8], MotionLM [7], Trajeglish [6]) convert continuous scene elements and trajectories into discrete tokens and leverage GPT-style Transformers to predict token sequences. This enables scalable, flexible, and generalizable multi-agent scenario synthesis. Simformer (this work) combines the unified scene modeling of UniGen with the efficient discrete tokenization and next-token prediction paradigm of SMART. By clustering positions in the ego-centric frame, using joint attribute token vocabulary, and leveraging autoregressive Transformers, Simformer achieves state-of-the-art realism, efficiency, and scenario diversity.

3. Methodology

3.1. Problem Formulation

Given a high-definition map segment M, the one-second history of the ego vehicle's motion R_{ego}, and the expected number of agents to generate for each type $\mathcal{C} = \{\text{vehicle}, \text{pedestrian}, \text{cyclist}\}$, the scenario generation task

requires synthesizing a complete multi-agent traffic scenario. Specifically, the model must generate a set of non-ego agents $\mathcal{A} = \{a_1, ..., a_N\}$ and their full trajectories $\{\tau_1, ..., \tau_N\}$ over a fixed time horizon $T = 91$ timesteps (including past, present, and future). Each agent a_i is characterized by its type, initial state (position, heading, and 3D bounding box dimensions), and a trajectory τ_i composed of center positions (x, y, z) and heading angles over time. The number of generated agents per type must match the provided count. The objective is to model the conditional distribution:

$$p\left(\mathcal{A}, \{\tau_i\}_{i=1}^{N} \mid M, R_{ego}, \text{Count}(a^{\text{type}})\right) \tag{1}$$

such that each sampled scenario is realistic, physically plausible, map-compliant, and exhibits natural multi-agent interactions. In Simformer, this problem is approached by autoregressively generating discrete tokens that represent agent types, placements, attributes, and full trajectories, conditioned on the map context, ego history, target agent type counts, and previously generated agents.

3.2. Overall Pipeline

SimFormer generates realistic traffic scenarios through an end-to-end autoregressive token prediction pipeline. The process consists of the following stages:

1. **Input Preparation:** Given a high-definition map M, a one-second history of the ego agent R_{ego}, and a specification of the expected number of agents to generate per type (vehicles, pedestrians, cyclists), we vectorize the map and extract relevant polyline and context features. This input is encoded into fixed-length token sequences to serve as the conditioning context.

2. **Tokenization:** All agent-related elements—including initial positions (in the ego-centric frame), physical attributes (length, width, height), dynamic attributes (speed, heading), and trajectory fragments—are discretized into interpretable token vocabularies via clustering or binning techniques (detailed in Section 3.3).

3. **Autoregressive Agent Generation:** Guided by the agent type counts, SimFormer iteratively generates agents in a grouped and type-aware manner. For each agent, it sequentially predicts (a) an agent type token, (b) a position token, (c) tokens for size and dynamic attributes, and (d) a sequence of trajectory tokens. Each prediction is conditioned on the map, ego history, agent type specification, and all previously generated agents.

4. **Decoding:** The generated tokens are mapped back to continuous representations, recovering full initial states and 91-timestep trajectories (including past, present, and future). These are then serialized into structured outputs for evaluation and submission.

3.3. Tokenization Strategies

3.3.1 Initial Position Tokenization

To capture the spatial distribution of non-ego agents, we represent each agent's location relative to the ego vehicle's local coordinate frame as $(\Delta x, \Delta y)$. We apply clustering to these relative positions across the training data to obtain a finite set of representative position tokens:

$$\mathcal{V}_{pos} = \{(\hat{x}_k, \hat{y}_k) \mid k = 1, \ldots, K_{pos}\} \tag{2}$$

where $(\hat{x}_k, \hat{y}_k)$ denotes the center coordinates of the k-th position token in the ego-centric frame, and K_{pos} is the total number of position tokens.

During both training and scenario generation, the model predicts a probability distribution over this discrete vocabulary, indicating where agents are likely to appear relative to the ego vehicle. This tokenization strategy enables flexible and scalable modeling of complex multi-agent spatial layouts.

3.3.2 Attribute Tokenization

To efficiently represent agent properties and capture essential attribute correlations, we discretize the physical and dynamic attributes of each agent into two separate joint token vocabulary, each with matched bin resolutions for the two dimensions.

Size Token Table The agent's physical dimensions, specifically length and width, are jointly discretized. Each size token corresponds to a bin in the two-dimensional (length, width) space:

$$\mathcal{V}_{size} = \{(l_m, w_m) \mid m = 1, \ldots, M\} \tag{3}$$

where (l_m, w_m) denotes the center values of the m-th bin for both length and width, and M is the number of bins for each attribute.

Dynamic Token Table The agent's initial speed and yaw angle (heading) are also jointly discretized, forming the dynamic token vocabulary:

$$\mathcal{V}_{dynamic} = \{(v_p, \theta_p) \mid p = 1, \ldots, P\} \tag{4}$$

where (v_p, θ_p) represents the center values of the p-th bin for both speed and heading, and P is the number of bins for each attribute.

During both training and generation, the model predicts probability distributions over these two attribute token vocabulary for each agent, enabling flexible, data-driven synthesis of plausible and diverse physical and dynamic properties. This two-table strategy balances modeling power, sample efficiency, and coverage of real-world agent attribute distributions.

3.3.3 Trajectory Tokenization

To efficiently represent and generate agent motion, we discretize short segments of each agent's trajectory into a finite set of trajectory tokens. Specifically, all agent trajectories are divided into fixed-length intervals (such as 0.5 seconds, sampled at 0.1-second resolution), and k-disks clustering [8] is applied on these segments to construct a trajectory token vocabulary. Each trajectory token thus represents a prototypical motion pattern relative to its starting point.

For trajectory initialization, we employ a reverse matching strategy to ensure that the first token is physically consistent with the agent's given initial state. This is achieved by comparing the recent movement of the agent—estimated using its position, velocity, and heading—with the candidate trajectory tokens, and selecting the best match. After the initial token is determined, the remainder of the trajectory is generated autoregressively as a sequence of tokens, each conditioned on the scenario context and previously generated tokens.

3.4. Model Architecture and Token Table

Transformer Architecture SimFormer discretizes the prediction task by clustering trajectory data into four token types: Position, Size, Dynamic, and Trajectory. Each token is embedded into a 128-dimensional space using separate embedding tables. As shown in Figure 1, the model adopts a decoder-only Transformer with a dual-stream architecture. A 3-layer encoder processes HD map features, while a 6-layer decoder autoregressively predicts agent tokens based on the encoded context and ego-history.

To capture spatial and temporal structure, 64 frequency bands are used to encode input coordinates and motion cues. Type and positional embeddings are added to preserve spatial locality, temporal order, and agent-specific information. The Transformer backbone uses 8 attention heads with 16-dimensional projections per head, and applies dropout with a rate of 0.1 for regularization.

Token Table Statistics SimFormer constructs all token vocabularies from training data. Position tokens are obtained via clustering on ego-relative coordinates, while size and dynamic attributes are discretized through uniform binning. Trajectory tokens are derived from clustered motion fragments. Each token type corresponds to a specific feature representation, as summarized in Table 1.

Token Type	Token Count	Algorithm	Feature Dimension
Position	1024	k-means	ego-relative 2D position
Size	256	uniform binning	length-width attributes
Dynamic	256	uniform binning	speed and yaw
Trajectory	2048	k-disks	0.5s motion fragment

Table 1. Token vocabulary statistics and clustering parameters.

4. Experiments

4.1. Dataset

We conduct all experiments on the Waymo Open Motion Dataset (WOMD), a large-scale and diverse benchmark for autonomous driving. WOMD consists of over 487,000 real-world driving sequences with high-definition map information and rich agent annotations, including positions, headings, velocities, and physical dimensions for vehicles, pedestrians, and cyclists. All experiments follow the official Scenario Generation Challenge 2025 data splits and evaluation protocols.

4.2. Metrics

For overall evaluation, we adopt the Realism Meta Metric (RMM) proposed by Waymo in the Scenario Generation Challenge 2025. RMM is a composite metric that quantifies the realism of generated scenarios by aggregating multiple aspects, including kinematic, interactive, and map-based features, into a single scalar score. Higher RMM values indicate scenarios that more closely match the distribution and characteristics of real-world driving data.

4.3. Implementation Details

Simformer is implemented in PyTorch and trained on 8 NVIDIA A100 GPUs. We use the AdamW optimizer with an initial learning rate of 5×10^{-4}, a minimum learning rate ratio of 1×10^{-2} (relative to the initial value), weight decay of 0.1, and a cosine annealing learning rate schedule. The batch size is set to 64 per GPU. Dropout with a rate of 0.1 is applied to all Transformer layers. All token vocabularies are constructed from the training set using uniform binning and clustering. Models are trained for 48 epochs, with early stopping based on the validation Realism Meta Metric (RMM).

4.4. Results

Our method achieves state-of-the-art performance on the Waymo Open Motion Dataset under the official Scenario Generation Challenge 2025 protocol. As shown in Table 2 and Figure 2, Simformer demonstrates superior realism and diversity across all key metrics, including the Realism Meta

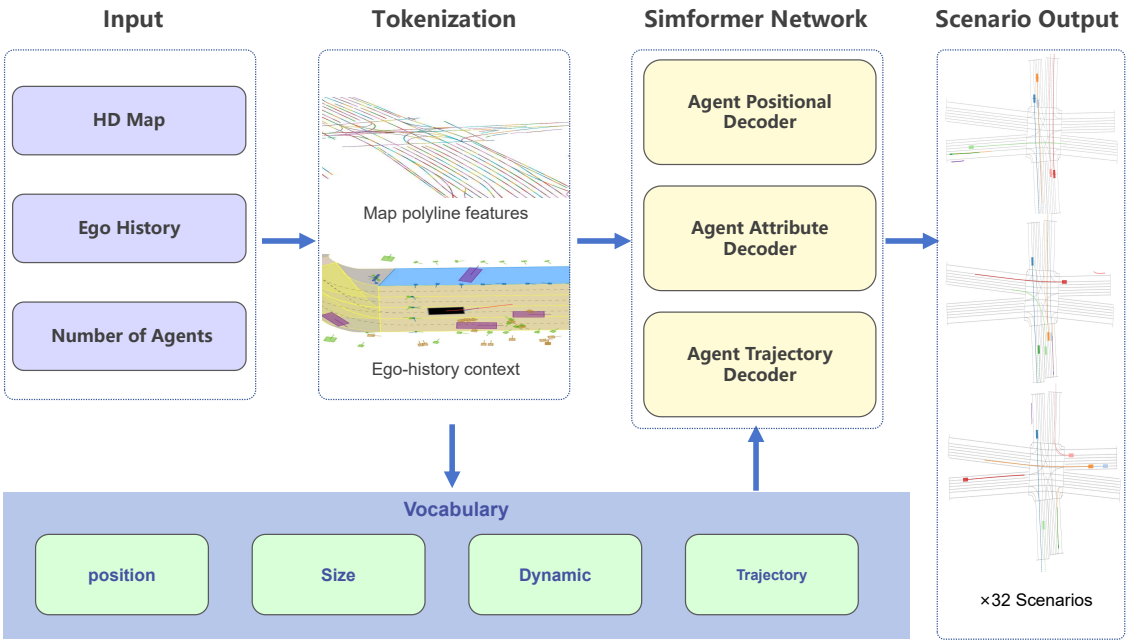

Figure 1. Model Architecture.

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 2. Comparison results with different methods in the Waymo Scenario Generation challenge.

Metric (RMM), static collision rate, and dynamic collision rate. The visualizations illustrate diverse, physically plausible multi-agent interactions, with the blue vehicle representing the ego agent in each scenario.

5. Conclusion

We presented **Simformer**, a unified and scalable framework for realistic scenario generation in autonomous driving. By leveraging discrete tokenization and autoregressive modeling, Simformer effectively captures the complexity and diversity of real-world traffic. Extensive experiments on the Waymo Open Motion Dataset demonstrate that our approach achieves state-of-the-art performance in realism, diversity, and physical plausibility.

References

[1] Luca Bergamini and et al. Simnet: Learning reactive self-driving simulations from real-world observations. In *NeurIPS*, 2021. 1

[2] Alexey Dosovitskiy and et al. Carla: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 2017. 1

[3] Liang Feng and et al. Trafficgen: Learning to generate diverse and realistic traffic scenarios. *arXiv preprint arXiv:2206.02051*, 2022. 1

[4] Pablo A. Lopez and et al. Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018. 1

[5] Reza Mahjourian and et al. Unigen: Unified modeling of initial agent states and trajectories for generating autonomous driving scenarios. *arXiv*, 2024. 1

[6] Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Learning the language of driving scenarios. *arXiv preprint arXiv:2312.04535*, 2023. 1

[7] Ari Seff and et al. Motionlm: Multi-agent motion forecasting as language modeling. In *ICCV*, 2023. 1

[8] Wei Wu and et al. Smart: Scalable multi-agent real-time simulation via next-token prediction. *arXiv*, 2024. 1, 3

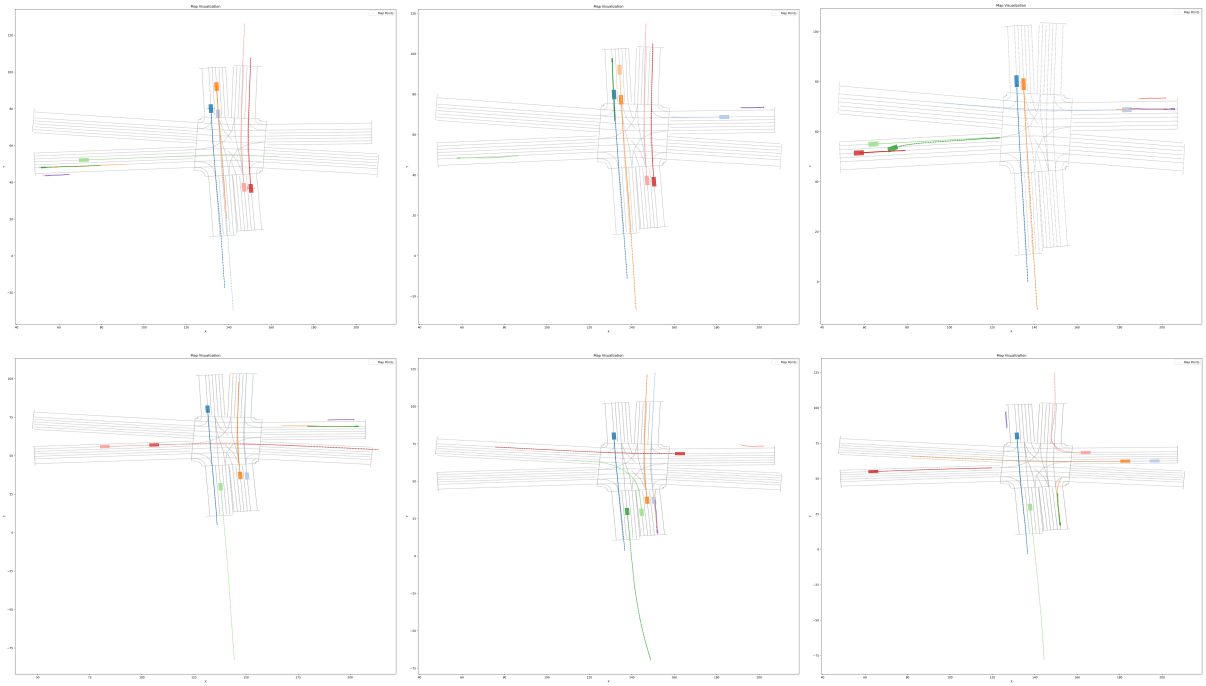

Figure 2. Visualization of six diverse scenarios generated by SimFormer. The blue vehicle denotes the ego agent. Each frame includes five vehicles, two cyclists, and one pedestrian.