POLITECNICO DI TORINO
Repository ISTITUZIONALE

TRACE - TRaffic Autoregressive sCene gEneration

Original
TRACE - TRaffic Autoregressive sCene gEneration / Perrone, G., Rapelli, M., Casetti, C.. - (In corso di stampa). (2026 IEEE 104th Vehicular Technology Conference (VTC2026-Fall) Boston, MA (USA) 6-9 September 2026).

Availability:
This version is available at: 11583/3016324 since: 2026-10-05T10:25:40Z

Publisher:
IEEE

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©9999 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

17 November 2026

TRACE — TRaffic Autoregressive sCene gEneration

Giuseppe Perrone, Marco Rapelli, Claudio E. Casetti
Department of Control and Computer Engineering
Politecnico di Torino
Turin, Italy
{giuseppe.perrone, claudio.casetti, marco.rapelli}@polito.it

***Abstract*—Validating Connected and Autonomous Vehicle (CAV) algorithms requires diverse, realistic traffic scenarios that traditional simulators cannot provide. For this reason, in this work, we present TRACE (TRaffic Autoregressive sCene gEneration), a text-conditioned framework for multi-agent traffic scene generation that addresses this limitation through two contributions. First, we propose an automated annotation pipeline that processes driving scenarios from the Waymo Open Motion Dataset (WOMD) using the Large Language Model Meta AI (LLaMA) 3.2 3B, a compact open-source language model, to generate natural language descriptions without reliance on proprietary APIs. The pipeline, which is fully reproducible on consumer hardware, produces a large-scale, text-annotated dataset. Secondly, we introduce an autoregressive transformer that populates map scenes with heterogeneous traffic participants conditioned on natural language textual prompts. Evaluated on the full Waymo validation set (40,075 scenes), TRACE achieves a 99.6% cardinality exact-match rate, 89.4% entity-type accuracy, and a maximum per-sector spatial deviation of 4 percentage points. Failure modes are traced to distributional bias in the training data, with targeted augmentation identified as a direction for future work.**

***Index Terms*—Generative AI, Large Language Models, Autoregressive Transformer, Connected and Autonomous Vehicles**

I. Introduction

According to statistics, 94% of road accidents are due to human error, including unrecognized hazards, poor decision-making, performance impairment, and other personal factors [1]. In this context, autonomous driving can be a potentially transformative technology: self-driven vehicles promise to improve road safety, reduce urban congestion through more efficient driving behavior, and lower fuel consumption by limiting unnecessary acceleration and braking maneuvers [2]. Recent advances in artificial intelligence and deep learning have accelerated progress towards autonomous driving, enabling vehicles to process large volumes of sensor data and improve their performance in perception, navigation, and control. Autonomous vehicles already demonstrate promising results in reducing the incidence of dangerous situations and accident rates compared to human-driven benchmarks [3].

Nevertheless, before deployment, autonomous driving algorithms must undergo extensive testing to verify their safety and reliability. Simulation environments such as SUMO (Simulation of Urban MObility) [4] and CARLA (CAR Learning to Act) [5] offer a controlled and cost-effective alternative to real-world testing. However, these platforms present notable limitations: the set of available traffic scenarios is insufficient to capture the complexity of real-world driving conditions, and the behavior of simulated vehicles is governed by hand-crafted rules that fail to reproduce the variability and nuance of human driving. Generative models can address this gap by producing novel traffic scenarios that go beyond the fixed repertoire of traditional simulators, thereby expanding the testing space for autonomous driving algorithms.

Several data-driven approaches have been proposed for this purpose, spanning autoregressive transformers, diffusion models, and variational autoencoders. Despite their diversity, these methods provide no mechanism for a user to specify the content of a scenario through natural language, limiting their accessibility to non-technical users.

A complementary line of work integrates Large Language Models (LLMs) to enable textual conditioning. However, existing methods rely on closed-source proprietary APIs such as GPT-4 or Gemini, which limits reproducibility and prevents customization of annotation style and semantic granularity.

To address both limitations described above, in this paper we present TRACE (TRaffic Autoregressive sCene gEneration), a text-conditioned autoregressive transformer framework for realistic multi-agent traffic scene generation. In particular, this work makes the following contributions:

1) we propose an automated pipeline that extracts a structured representation of the traffic scenario surrounding an ego vehicle and leverages an open-source LLM to generate a natural language description of the scene. Starting from the Waymo Open Motion Dataset (WOMD) [15], this enables the construction of a large-scale, text-annotated traffic dataset;
2) we introduce TRACE, an autoregressive transformer that takes a road map and a textual description as inputs, and populates the environment with realistic traffic participants accordingly.

The remainder of this paper is organized as follows: Section II reviews related work, Section III describes the proposed data generation pipeline, Section IV presents the architecture of TRACE and Section V reports the experimental evaluation. Eventually, Section VI concludes the paper.

II. Related Work

Validating autonomous driving systems requires exposure to diverse and realistic traffic scenarios that go beyond what physical testing can economically provide. Simulation environments such as SUMO and CARLA offer a controlled alternative, but their rule-based traffic models limit the diversity and realism of generated scenarios. A growing number of works has therefore explored data-driven generative approaches for traffic scenario synthesis.

a) Map-Conditioned Generation: Autoregressive models such as TrafficGen [6] and InfGen [7] generate traffic scenarios based on HD maps and short driving logs, sequentially placing road entities via transformer-based next-token prediction. Diffusion-based approaches such as Scenario Diffusion [8] and PathDiffuser [9] offer richer spatial modeling by operating over continuous state spaces using map layout and agent-specific tokens as inputs. Variational frameworks such as STRIVE [10] leverage conditional Variational Auto-Encoders (VAEs) to perform adversarial latent optimization, generating safety-critical scenarios. However, despite their diversity, all these approaches offer no mechanism to guide their generation processes toward a desired, specific configuration.

b) Language-Conditioned Generation: A significant number of works integrate large language models to enable the generation of scenarios based on textual input. CTG++ [11] uses GPT-4 to translate user instructions into differentiable loss functions that guide a diffusion process, while LCTGen [12] employs GPT-4 via in-context learning to map textual descriptions onto structured vectors passed to a transformer generator. ChatScene [13] and ScenicNL [14] follow similar paradigms, relying on GPT-4 and Gemini respectively to produce executable Scenic scripts.

c) Positioning of this work: We propose a fully-automated pipeline for constructing large-scale, text-annotated traffic datasets without reliance on proprietary APIs or manual labeling effort. Unlike existing language-conditioned methods, that depend on closed-source models such as GPT-4 or Gemini to generate scene descriptions, our approach leverages an open-source LLM to verbalize structured scene summaries into natural language, making the annotation process fully reproducible and accessible on consumer hardware. Building on this dataset, we further introduce TRACE, an autoregressive transformer that generates traffic snapshots taking textual prompts and HD maps as inputs.

III. Dataset Construction

We derive our dataset from the Waymo Open Motion Dataset (WOMD) [15], one of the largest publicly available benchmarks for motion forecasting in urban driving. WOMD provides over 400,000 nine-second driving scenarios recorded across six U.S. cities under diverse traffic, weather, and lighting conditions by a Waymo self-driving vehicle, hereinafter referred to as the *ego vehicle* (or simply *ego*). Each scenario includes the trajectories of all surrounding road-users as well as high-definition map information, comprising road geometry and semantic road elements such as lane boundaries, crosswalks, and traffic signals.

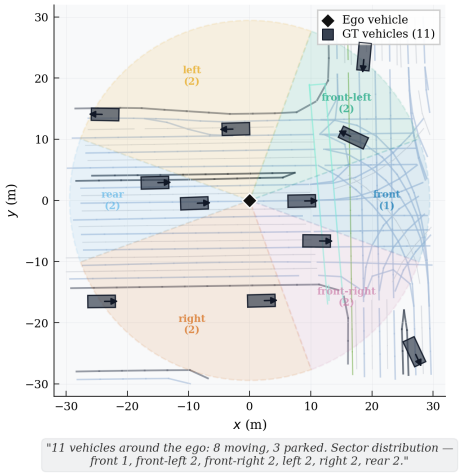


Fig. 1. Example of automatic labeling

We use the official training dataset and retain only scenarios containing at least two non-ego agents, i.e., other active participants in the scenario, such as other vehicles, pedestrians, or cyclists.

To generate descriptions automatically while minimizing the risk of incorrect or unjustified outputs, we adopt a two-phase approach. In the first phase, we transform the raw scene data into a structured, ego-centric JSON summary. The positions of all road entities are projected into the ego vehicle's reference frame via a rigid-body rotation that aligns the x-axis with the heading of the ego. We considered only elements within a perception radius of $r = 30\,\text{m}$ from the center of the ego vehicle. This covers the operationally relevant neighborhood for urban driving while excluding distant, low-relevance road elements.

The space surrounding the ego vehicle is then partitioned into six semantic sectors defined by angular thresholds in the ego reference frame (see Fig. 1). For each sector, we record the number of traffic participants present, their distance to the ego, and their type (e.g., vehicle, pedestrian, or cyclist). This yields a representation that is compact, order-invariant, and easily interpretable by a language model. To reduce input complexity, we only consider agents within the ego perception radius and we impose a maximum of 32 road users per scene.

In the second phase, the structured JSON is verbalized by LLaMA 3.2 3B (Large Language Model Meta AI), an open-source model with 3 billion parameters. The use of this model is motivated by three considerations: (i) *reproducibility*, as any researcher can obtain the model with a single command; (ii) *cost*, since no per-call charges are applied; and (iii) *throughput*, as the entire WOMD training set can be processed in reasonable time on commercial hardware.

To assess the quality of the generated descriptions, we manually examined a random subset of samples drawn from the WOMD validation set. The minimum sample size was de-

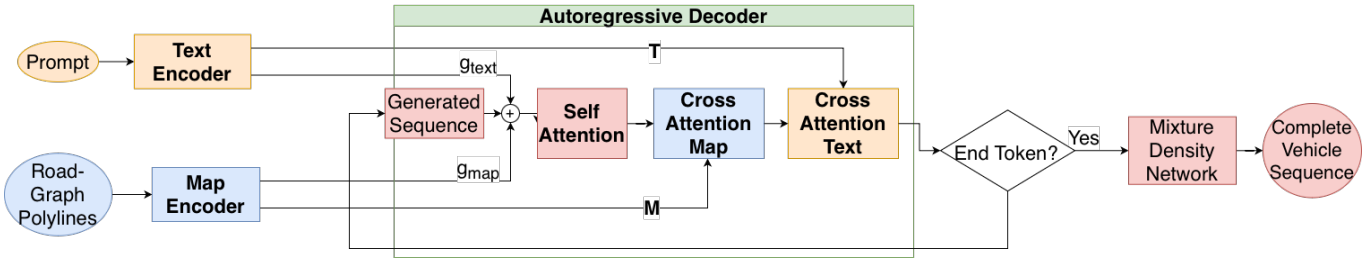


Fig. 2. Overview of the proposed model. A text encoder maps the input text prompt to a scene embedding, which conditions the Transformer decoder via cross-attention. The map encoder processes road-graph polylines into lane tokens, also attended by the decoder. At each autoregressive step, the decoder predicts a stop token probability; if generation continues, a Mixture Density Network outputs the next agent's position, heading, velocity, and entity type. The process repeats until the end token is predicted or the maximum agent count is reached.

termined using the well-known Cochran formula [16], defined as:

$$n_0 = \frac{Z^2 p(1-p)}{e^2} \qquad \text{and} \qquad n = \frac{n_0}{1 + \frac{n_0 - 1}{N}} \tag{1}$$

where Z is the Z-score corresponding to the desired confidence level, p is the estimated population proportion, e is the tolerated margin of error, and N is the total size of the finite population. Applying these equations to our population of $N = 44{,}097$ samples, and imposing a 95% confidence level with a 5% margin of error, we obtained a minimum sample size of $n = 381$ elements. This value represents the minimum number of inspected items required for the analysis to be statistically representative of the entire dataset. The evaluation criterion was regarding *adherence*: whether, or not, the generated text accurately reflects the agent configuration encoded in the structured JSON. The manual validation confirmed that, by constraining the LLM to the structured data, the descriptions remained faithful to the actual scenario in all the 381 inspected cases. An example of generated prompt is shown in the text inside the gray box of Fig. 1.

IV. TRACE Architecture

In this section we introduce TRACE (TRaffic Autoregressive sCene gEneration), a text-based autoregressive transformer that generates snapshots of realistic multi-agent traffic scenarios from a natural language prompt description and an HD map. It is important to underline that the proposed scene generation module is designed to be coupled with reactive trajectory generators, e.g. CtRL-Sim [18], to transform a static traffic snapshot into a fully interactive closed-loop environment. In such a setting, the agents placed by TRACE, can react dynamically to the maneuvers of the autonomous vehicle (AV) under test, enabling realistic closed-loop evaluation and scenario stress-testing.

Given a local HD map section, represented as a set of vectorized road elements, and a textual prompt description of a driving scenario (e.g., "Sparse urban intersection. 2 entities: 1 vehicle 1 slow, 1 cyclist."), the TRACE model populates the scene by placing the intended traffic participants, each with an initial state consistent with both the input map geometry and the provided prompt description.

The TRACE architecture, shown in Fig. 2, comprises three main components: (i) a text encoder, which maps the textual prompt into a compact embedding space, translating the natural language input into a dense numerical representation (i.e., $\mathbf{T}$ and $\mathbf{g}_{\text{text}}$) that can be processed by the decoder; (ii) a map encoder, which similarly processes all lane segments and map elements, producing a map numerical representation (i.e., $\mathbf{M}$ and $\mathbf{g}_{\text{map}}$); and (iii) an autoregressive transformer decoder, which, fed with both encoder outputs ($\mathbf{T}$, $\mathbf{g}_{\text{text}}$, $\mathbf{M}$ and $\mathbf{g}_{\text{map}}$) generates a sequence of traffic participants $\mathbf{P} = \{p_1, \ldots, p_S\}$, one at a time, ensuring each placed entity is coherent with the map layout and the textual specification.

A. Text Encoder

To encode the natural language prompt input, we leverage CLIP (Contrastive Language–Image Pre-training) [17], a large-scale vision-language model pre-trained by OpenAI on hundreds of millions of image–text pairs. CLIP learns a joint space where semantically related images and texts are mapped to similar representations. In our framework, we use CLIP to extract features from text, turning it into numbers that guide the decoder during generation. This design choice allows us to leverage powerful pre-trained language representations while keeping the training focused on the generation task. CLIP produces both a sequence of $L = 77$ token embeddings $\mathbf{T} = \{t_1, \ldots, t_L\}$, capturing fine-grained semantic details, and a global text embedding $\mathbf{g}_{\text{text}}$, providing a compact summary of the overall prompt.

B. Map Encoder

The map encoder takes as input the road layout, i.e., the lane segments, road boundaries, and traffic control elements, as a set of geometric primitives. Each element is described by its spatial extent, orientation, and semantic type. The map encoder model pairs relationships between all map elements, producing a contextual representation of the road geometry. Conversely to what is usually implemented in standard transformer architectures, where the order of inputs affects the final result, here the input is an unordered set. For this reason, we do not use positional encoding, i.e., a technique typically used in sequence models to inject information about the position of an item within a sequence.

As done with CLIP, the map encoder produces both a sequence of up to $K = 736$ map embeddings $\mathbf{M} = \{m_1, \ldots, m_K\}$, whose actual size depends on the number of road elements present in the local map section, and a global map embedding $\mathbf{g}_{\text{map}}$, which provides a compact representation of the overall road layout.

C. Autoregressive Decoder

The autoregressive decoder generates traffic agents $p_i \in \mathbf{P}$ one at a time, where each new entity is a function of three different sources of information: the map token embeddings $\mathbf{M}$, the text token embeddings $\mathbf{T}$, and the sequence of agents generated so far, $\mathbf{P'} = \{p_1, \ldots, p_{i-1}\}$.

The decoder is built as a stack of different layers (see Fig. 2 with TRACE architecture). Each layer contains a self-attention mechanism over the previously generated agents, followed by two cross-attention mechanisms: the first one allows the decoder to query $\mathbf{M}$, extracting relevant geometric information about the road layout; the second one allows the decoder to query $\mathbf{T}$, extracting information from the scene textual description. This way, every newly generated agent p_n can independently adhere to both the road geometry and the textual context.

In addition, the global embeddings $\mathbf{g}_{\text{map}}$ and $\mathbf{g}_{\text{text}}$ are summed to every element processed by the decoder, ensuring that a compact summary of both the road layout and the scene textual description is always available at each generation step during the process.

At each generation step i, the decoder predicts the next traffic agent as a vector comprising four attributes: its position in the map, its speed, its heading angle and its type (i.e., whether the agent is a vehicle, a pedestrian, or a cyclist).

For the motion attributes (position, speed, and heading), the model does not predict a single fixed value. Instead, it outputs the parameters of a Mixture Density Network (MDN), i.e., a weighted combination of multiple probability distributions. Concretely, position and speed are modeled as Gaussian mixtures, while heading is modeled as a von Mises mixture, i.e., the circular analogue of a Gaussian, better suited for angular quantities since it naturally handles the periodicity of angles without discontinuities. This allows the model to represent uncertainty and to capture situations where multiple plausible placements exist simultaneously, e.g., a vehicle that could plausibly occupy either of two adjacent lanes, according to the given map and textual inputs.

The decoder learns to terminate the generation process autonomously during training. At each decoding step, it decides whether to emit the next traffic agent p_{i+1} or a special end token p_S, which signals that the scene is complete and halts the generation.

D. Training

The model is trained end-to-end on the text-annotated dataset described in Section III. The map encoder and decoder are trained from scratch, while the CLIP text encoder is kept frozen throughout, as its parameters have already been optimized. The training objective is a weighted sum of per-component losses:

$$\mathcal{L} = \lambda_{\text{p}}\,\mathcal{L}_{\text{p}} + \lambda_{\text{s}}\,\mathcal{L}_{\text{s}} + \lambda_{\text{h}}\,\mathcal{L}_{\text{h}} + \lambda_{\text{t}}\,l_{\text{t}} + \lambda_{\text{stop}}\,l_{\text{stop}} \tag{2}$$

In equation 2, the loss terms $\mathcal{L}_{\text{p}}$, $\mathcal{L}_{\text{s}}$, and $\mathcal{L}_{\text{h}}$ are defined as the Negative Log-Likelihood (NLL) of the ground-truth values under their respective predicted MDN distributions, i.e., a measure of how well the predicted distribution fits the observed data.

The discrete terms l_{t} and l_{stop} are cross-entropy losses supervising entity-type classification and the stopping criterion, respectively. For type classification, we adopt Focal Loss, a variant of cross-entropy that down-weights the contribution of easily classified examples, focusing training on hard or rare categories. This is particularly beneficial given the class imbalance inherent in the dataset, where common entity types (e.g., cars) vastly outnumber rarer ones (e.g., motorcycles, pedestrians, or cyclists).

Preliminary experiments revealed that, in the presence of a detailed HD map given in input, the model tended to rely almost exclusively on the map representation, disregarding the text conditioning. To counteract this, with a probability $P_{\text{drop}}^{\text{map}}$, the entire map representation is zeroed out during training. This process, referred to as *map dropout*, forces the decoder to rely solely on the text conditioning to determine scene composition. Symmetrically, *text dropout* randomly zeros out the text conditioning with a probability $P_{\text{drop}}^{\text{text}}$ during training, preventing the model from ignoring the map representation. At inference time, both the map and the text are always provided.

V. Results

Before discussing the quantitative and qualitative evaluation of our proposed method, we briefly outline the implementation details and training configurations, shown in Table I.

We evaluate the proposed model on the full Waymo Open Motion Dataset (WOMD) validation set. For each scenario,

TABLE I
TRAINING AND MODEL HYPERPARAMETERS.

Group	Parameter	Value
Architecture	Transformer layers	6
	Attention heads	8
	Hidden dimension	512
	MDN components	5
Optimization	Optimizer	AdamW
	Learning rate η	10^{-4}
	Weight decay	10^{-4}
	Batch size	64
	Epochs	50
Loss weights λ	Position λ_{p}	1.0
	Speed λ_{s}	0.3
	Heading λ_{h}	0.8
	Type λ_{t}	0.5
	Stop λ_{stop}	1.5
Regularization	Map dropout $P_{\text{drop}}^{\text{map}}$	0.30
	Text dropout $P_{\text{drop}}^{\text{text}}$	0.15
	Focal Loss γ	2.0

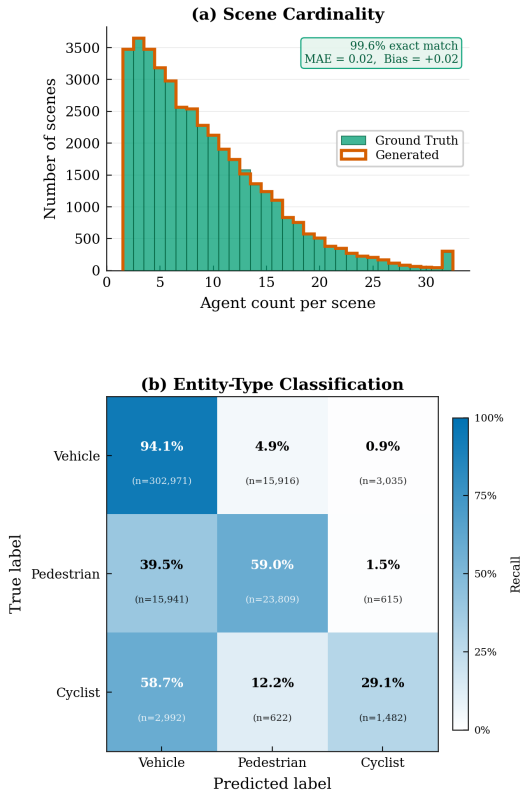


Fig. 3. Scene-level fidelity on the full Waymo Open Dataset validation set (40,075 scenes)

we extract the initial traffic snapshot (at timestep $t = 0$) as the ground-truth observation, retaining all agents within a $r = 30$ m radius of the ego vehicle, as described in Section III. This yields a validation set of 40,075 static traffic configurations.

We quantitatively assess the model along three complementary directions: *i)* its ability to reproduce the statistical distribution of real traffic snapshots; *ii)* its adherence to ground-truth agent placement, and *iii)* its compliance with prompt-specified scene configurations. Quantitative evaluation focuses on agent count, spatial distribution, and entity type, as these attributes are directly verifiable against ground-truth observations and are the most operationally relevant for initializing downstream trajectory generators.

Fig. 3a shows the distribution of agent counts per scene, with the model achieving a cardinality exact-match rate of 99.6%, with a mean absolute error of 0.02. This near-perfect cardinality adherence demonstrates that the decoder has learned to manage the numerical count specified in the text prompt, terminating generation at precisely the indicated number of agents. Since prompts are derived from real scenes, faithful count adherence implies that the marginal distribution of scene cardinality is automatically reproduced.

The overall entity-type accuracy reaches 89.4%, as shown in the confusion matrix in Fig. 3b. The dominant failure mode is the misclassifications of minority-class entities, i.e., pedestrians and cyclists, as vehicles. Vehicle recall stands at 94.1%, while pedestrian and cyclist recalls are 59.0% and 29.1%, respectively, with 39.5% of pedestrians and 58.7% of cyclists incorrectly predicted as vehicles. This is an expected consequence of training on a heavily vehicle-biased dataset, where vehicles account for 87.6% of all annotated road-users, whereas pedestrians and cyclists together represent fewer than one-eighth of the total population.

Fig. 4 shows the ego-centric spatial distribution of road users, aggregated over the full validation set of 40,075 samples. The heatmap captures where the model places agents in the space surrounding the autonomous vehicle. The spatial distribution is reproduced with high fidelity across five of the six sectors: the front sector (F) agrees exactly at 25%, the left and right flanks (L and R, respectively) differ by at most 1 percentage point, and the front-left (F-L) and front-right (F-R) quadrants show a symmetric shift of 2 percentage points each. The largest discrepancy is in the rear sector (Re: 26% ground truth vs. 22% generated, -4 percentage points), suggesting the model slightly under-populates the area directly behind the ego.

Fig. 4 also reveals a mild proximity bias: the generated agents are more concentrated in the 10–20 m ring (38% vs. 34%) at the expense of the 20–30 m area (31% vs. 36%), consistent with the 1.26 m reduction in mean ego-to-agent distance (17.77 m vs. 19.03 m). The mild proximity bias is consistent with a known limitation of finite-component MDNs: the fixed number of Gaussian components must cover all the distinct *modes* of the output distribution (i.e., the different clusters of plausible values the predicted feature can take). The likelihood objective tends to concentrate them around the most frequently observed values, leaving the tail slightly under-represented at inference time. Replacing the MDN with a continuous generative model, such as a normalizing flow or a diffusion-based model, represents a natural direction for future works.

Fig. 5 provides a qualitative illustration of how the model responds to different prompt specifications on the same underlying scene. In Fig. 5a, a front-heavy prompt (*"Dense urban intersection. 5 entities: 5 vehicles; 5 moving. Sectors: front 5."*) concentrates all generated agents in the front sector, faithfully reflecting the specified layout. In Fig. 5b, a spatially distributed prompt (*"Busy urban intersection. 6 entities: 6 vehicles; 6 moving. Sectors: front 1, front-left 1, front-right 1, left 1, right 1, rear 1."*) yields a symmetric, one-per-sector placement that spans all six angular regions. Taken together, these examples demonstrate that the model successfully associates prompt-level sector specifications with concrete spatial arrangements, redistributing agents across the scene in response to changes in the conditioning text.

To evaluate text-to-sector compliance, we condition the model on structured prompts specifying a target number of agents per angular sector and measure what fraction of generated agents are placed in the correct sector. Fig. 6a reports single-sector results (similarly to the example in Fig. 5a). The model achieves perfect compliance (100%) for axial

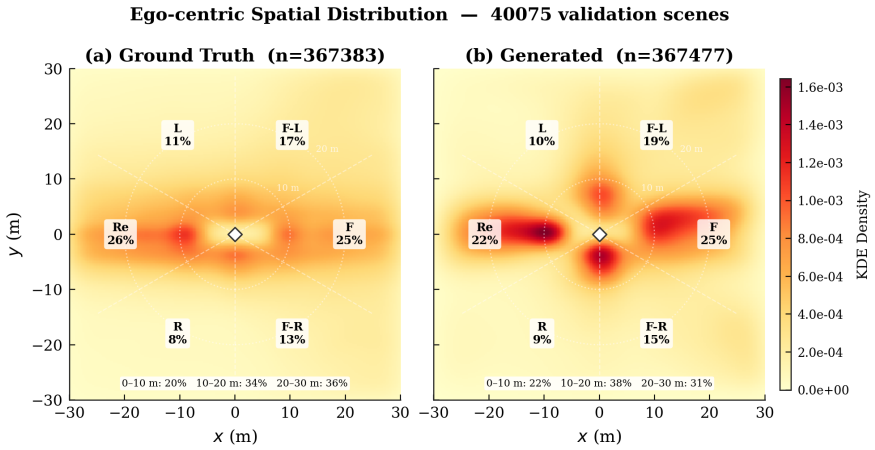
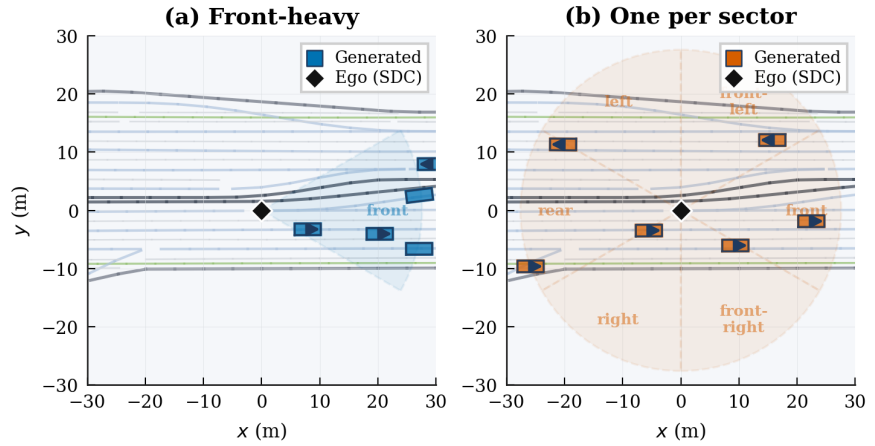


Fig. 4. Ego-centric spatial distribution of generated agents compared to ground truth, estimated via kernel density estimation (KDE, bandwidth = 2 m) over the validation dataset



"Dense urban intersection. 5 entities: 5 vehicles: 5 moving. Sectors: front 5."

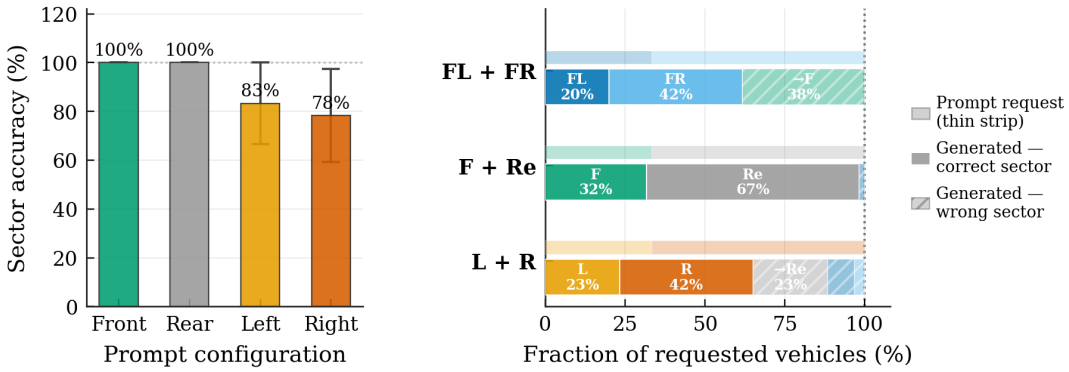
"Busy urban intersection. 6 entities: 6 vehicles: 6 moving. Sectors: front 1, front-left 1, front-right 1, left 1, right 1, rear 1."

Fig. 5. Qualitative comparison of text-conditioned generation

directions: all generated agents are placed in the front and rear sectors when so specified. Lateral directions are harder: left-sector accuracy reaches 83.3% (±16.7%) and right-sector accuracy 78.3% (±19.1%), with higher variance reflecting the structural under-representation of purely lateral traffic in the WOMD dataset.

Fig. 6b shows the actual allocation for multi-sector prompts (similarly to the example in Fig. 5b). The front+rear (F + Re) configuration is nearly perfectly resolved (98.3% ± 7.3%), matching the 32%–67% front-to-rear ratio specified in the prompt. The front-left+front-right (FL + FR) configuration achieves 61.7% compliance (±32.1%): 20% land in FL and 42% in FR, while the remaining 38% overflow into the adjacent front sector. The left+right (L + R) configuration reaches 65.0% (±24.7%), with 23% in L and 42% in R, while 23% drift into the rear sector(Re).

The origin of this failure mode is traceable to the dataset distribution. In the Waymo Open Motion Dataset (WOMD), scenarios featuring road entities solely in two distinct sectors are exceedingly rare; instead, traffic is typically distributed across multiple diverse sectors: an analysis of 1,000 randomly-sampled validation scenes reveals that *front-left* + *front-right* as the exclusive sector occupancy never occurs (0 of 1,000 scenes), and *left* + *right* exclusively occurs in only 5 scenes (0.5%). The model has therefore never encountered a training example in which traffic is confined to two lateral sectors alone, and adapts imperfectly by defaulting to the nearest over-represented mode. These findings highlight a fundamental limitation of distribution-matching training: prompt patterns that are structurally valid but statistically absent from the data cannot be reliably handled without targeted data augmentation or out-of-distribution conditioning strategies.



(a) Single-Sector Prompts **(b) Actual vs. Requested Allocation**

Fig. 6. Text-to-Sector Compliance. (a) Fill-rate for single-sector prompts (b) Vehicle allocation for mixed-sector prompts. Each row shows the actual distribution (solid bars) compared to the requested one (thin strip above). Hatched segments mark vehicles placed in the wrong sector.

VI. Conclusion

Autonomous driving has the potential to significantly impact road safety, urban congestion, and fuel efficiency. However, prior to deployment, autonomous driving algorithms require comprehensive validation. Current simulation tools exhibit important shortcomings, as they do not accurately capture the variability and subtlety of human driving behavior. Although generative models can help bridge this gap, many existing approaches lack a mechanism for users to define scenario content through natural language, thereby restricting accessibility for non-technical users.

In this paper we presented TRACE, a fully open-source framework for text-conditioned multi-agent traffic scene generation. An automated annotation pipeline based on LLaMA 3.2 3B produces natural language descriptions of Waymo driving scenarios without reliance on proprietary APIs. An autoregressive transformer decoder, conditioned on both textual prompts and HD maps via dual cross-attention, then populates scenes with heterogeneous traffic participants modeled via MDNs.

Evaluated on the full Waymo validation set (40,075 scenes), TRACE achieves a cardinality exact-match rate of 99.6%, a maximum per-sector spatial deviation of 4 percentage points, and 89.4% entity-type accuracy. Sector compliance degrades for lateral configurations statistically absent from training data, pointing to targeted data augmentation as the primary avenue for improvement. The static snapshots produced by TRACE are designed to initialize reactive trajectory generators for closed-loop autonomous vehicle evaluation; extending the framework to jointly generate initial trajectories remains a natural direction for future work.

References

[1] National Highway Traffic Safety Administration (NHTSA), "Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey," *Traffic Safety Facts Crash Stats Report No. DOT HS 812 115*, Washington, DC, Feb. 2015.

[2] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Res. Part A: Policy Pract.*, vol. 77, pp. 167–181, 2015.

[3] K. D. Kusano *et al.*, "Comparison of Waymo rider-only crash data to human benchmarks at 7.1 million miles," *Traffic Inj. Prev.*, vol. 25, no. S1, pp. S66–S77, 2024.

[4] L. P. Alvarez *et al.*, "Microscopic traffic simulation using SUMO," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2575–2580.

[5] A. Dosovitskiy *et al.*, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn. (CoRL)*, 2017, pp. 1–16.

[6] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou, "TrafficGen: Learning to generate diverse and realistic traffic scenarios," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 3567–3575.

[7] Z. Peng, Y. Liu, and B. Zhou, "InfGen: Scenario generation as next token group prediction," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2026.

[8] E. Pronovost *et al.*, "Scenario Diffusion: Controllable driving scenario generation with diffusion," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[9] "Path-Diffuser: A two-stage diffusion model for data-driven traffic simulation," *arXiv preprint arXiv:2509.24995*, 2025.

[10] D. Rempe *et al.*, "Generating useful accident-prone driving scenarios via a learned traffic prior (STRIVE)," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 17305–17315.

[11] Z. Zhong *et al.*, "Language-guided traffic simulation via scene-level diffusion," in *Proc. Conf. Robot Learn. (CoRL)*, 2023, pp. 144–177.

[12] S. Tan, B. Ivanovic, X. Weng, M. Pavone, and P. Kraehenbuehl, "Language conditioned traffic generation," *arXiv preprint arXiv:2307.07947*, 2023.

[13] H. Huang *et al.*, "Chat-scene: Bridging 3d scene and large language models with object identifiers," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, 2024, pp. 113991–114017.

[14] K. Elmaaroufi *et al.*, "ScenicNL: Generating probabilistic scenario programs from natural language," *arXiv preprint arXiv:2405.03709*, 2024.

[15] S. Ettinger *et al.*, "Large scale interactive motion forecasting for autonomous driving: The Waymo Open Motion Dataset," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 9710–9719.

[16] W. G. Cochran, *Sampling Techniques*, 3rd ed. New York, NY, USA: Wiley, 1977.

[17] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8748–8763.

[18] L. Rowe *et al.*, "Ctrl-sim: Reactive and controllable driving agents with offline reinforcement learning," *arXiv preprint arXiv:2403.19918*, 2024.