Long-term Traffic Simulation via Structured Autoregressive Modeling

Lingyu Xiao, Zexin Feng, and Xintao Yan✉

The University of Hong Kong, Pokfulam, Hong Kong
{lingyu.xiao,zexinfeng}@connect.hku.hk xintaoy@hku.hk
https://sephirex-x.github.io/rosettasim/

Abstract. Interactive traffic simulation is a vital world model for autonomous driving. A central challenge in long-horizon simulation is modeling sustained multi-agent interactions, which is further exacerbated by dynamic token cardinality as agents continuously enter and exit the scene. In this work, we propose that the solution lies in the synergy between the architectural inductive biases and statistical priors of large-scale sequence models, e.g., Large Language Models (LLMs). Our probing experiments reveal that the transferability of attention mechanisms and the distributional consistency between motion tokens and natural language enable small-scale, heavily frozen LLMs to rapidly adapt to traffic modeling. Building on this insight, we introduce RosettaSim, a unified framework that projects scene topology, agent states, and spawning intents into a structured autoregressive stream with variable length, achieving both strong short-term accuracy and stable long-horizon simulation fidelity. Furthermore, evaluating extended rollouts presents yet another hurdle, as one-to-one agent correspondence inevitably fades over time. To address this, we introduce Retrieval-based Traffic Evaluation (RTE), which retrieves semantically similar real-world scenarios as context-aware reference anchors. Experiments on the Waymo Open Sim Agent Challenge (WOSAC) demonstrate that RosettaSim achieves state-of-the-art performance in both short- and long-term simulation. Furthermore, RTE exhibits a stronger correlation with standard metrics ($r = 0.83$) than existing approaches ($r = 0.74$), indicating improved alignment with long-horizon simulation fidelity.

Keywords: Traffic simulation · Autonomous driving · Large language models

1 Introduction

Traffic simulation underpins the scalable and safe development of autonomous driving systems [9, 10, 21, 44]. While prior works [28, 29, 38, 40, 43, 49, 52] excel under short-term setting (typically < 8s), they struggle in long-term, trip-level simulation where agents continuously enter and exit the scene. To address this,

✉ Corresponding author.

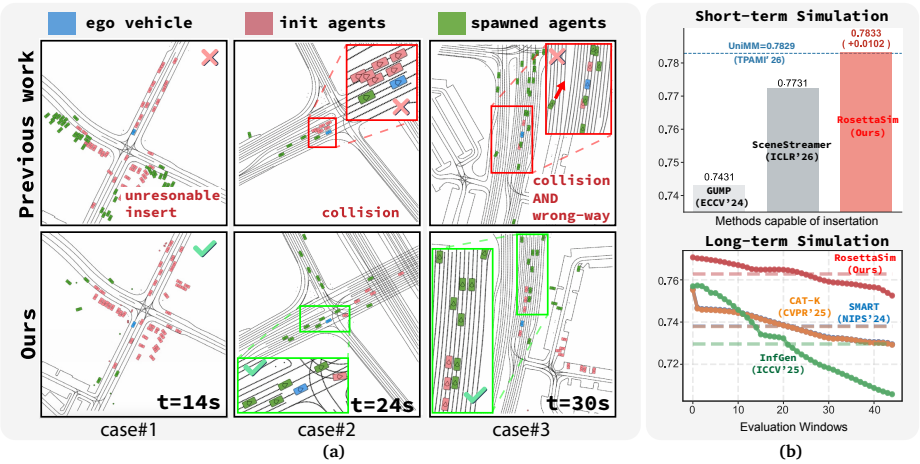

Fig. 1: Performance. (a) Visualization of long-term simulation compared with [46]. Mere demo videos can be found on the webpage. (b) Quantitative performance comparison on WOMD under close-loop short/long-term simulation.

recent works [34, 46] have reframed long-term traffic simulation as a joint modeling problem that integrates scene generation [3, 8, 30, 35] (agent injection) and motion generation within a unified rollout process. Despite this progress, two fundamental challenges remain unresolved: (i) a persistent performance trade-off when jointly modeling agent population dynamics and interaction-aware motion, and (ii) the lack of reliable evaluation protocols for long-term simulation.

The first challenge arises from the inherently different learning objectives of scene and motion generation. Scene generation focuses on spatial distributions (predicting *where* and *when* a new agent appears), whereas motion generation emphasizes temporal consistency (predicting *how* agents move). Existing approaches [27, 46] that jointly optimize these objectives often struggle to balance the two tasks. As shown on Fig. 1(b) upper, the most recent work SceneStreamer [27] falls large behind against specialized short-term models (e.g., UniMM [19]). This imbalance suggests that current modeling paradigms cannot adequately represent both agent population dynamics and long-term motion dependencies simultaneously. In parallel, large-scale sequence models, e.g., Large Language Models (LLMs), have demonstrated strong capabilities in modeling long-range dependencies across both homogeneous sequences [1, 13, 36, 45] (e.g., text) and heterogeneous modalities [17, 20, 22] (e.g., vision, robotics control). However, their application to traffic simulation remains limited. Current approach typically uses language models as a conditioning tool for controllable simulation [33] or scene generation [16]. Other efforts redesign components inspired by language models [27,38,52] or simply draw analogies to next-token prediction [28, 40, 46]. None of these works provide justification for whether LLMs' modeling capabilities for structural sequence can be applied to traffic simulation, let alone long-horizon simulation. Therefore, an intriguing question arises: Can large-scale sequence models effectively model traffic dynamics without relying on linguistic supervision, and if so, why?

We posit that the potential of LLMs lies not in linguistic semantics, but in their ability to model structured, compositional sequences autoregressively. Specifically, as shown in Fig. 2, we observe that the frequency distribution of discretized traffic motion tokens closely mirrors the statistical properties of natural language. Motivated by [2], we hypothesize that sequence models trained under similar regimes provide a highly optimized initialization manifold for traffic dynamics. To empirically validate this, we conducted probing experiments on motion generation using fully and partially frozen LLMs. The rapid adaptation of even a small-scale, heavily frozen LLM confirms that these models effectively process traffic dynamics as structured sequential data with long-range dependencies, akin to learning a "foreign language". Furthermore, we observe that LLM exhibits highly similar attention locality [1] when processing language and motion data separately, further substantiating our hypothesis. Building on these insights, we propose a unified paradigm that seamlessly projects geometric topology, existing agent states, and generation intents into a structured autoregressive token sequence with variable length. We name our method RosettaSim, acting like a "Rosetta Stone" between language and motion. By modeling the sequence structurally, RosettaSim can inherit both the modeling capability under flexible token length and the structural prior from pretraining, thus significantly mitigating the performance trade-off between scene and motion generation. As shown in Fig. 1(a), RosettaSim demonstrates improved traffic flow realism and multi-agent interaction consistency compared to the prior method [46].

The second bottleneck lies in evaluation. Standard metrics (e.g., WOSAC [24]) rely on strict one-to-one agent matching, making them inapplicable for dynamic long-term rollouts. Existing workarounds [34, 46] compare rollout histograms against entire validation sets; however, because the aggregated distribution of the entire validation set is not context-aware, global matching yields misleadingly high scores for trivial policies. For example, it is unreasonable to ask a high-speed scenario to match the overall distribution, which is dominated by low-speed scenarios. To resolve this, we propose the Retrieval-based Traffic Evaluation (RTE) framework. RTE mitigates statistical bias by leveraging a pretrained VAE [18,30] latent space to retrieve the top-K most semantically similar real-world scenarios. These serve as grounded reference anchors, ensuring a rigorous, context-aware, and fair evaluation.

Though extensive close-loop experiments on Waymo Open Motion Dataset (WOMD) [7], we demonstrate that our method achieves state-of-the-art performance in both short-term benchmarking [24] and long-term simulation compared with its counterparts (see Fig 1 (b)).

In summary, our contributions are threefold:

- We introduce a unified paradigm for long-horizon traffic simulation that formulates multi-agent scene evolution as a conditional sequence generation prob-

[1] Attention locality is defined as the sum of attention weights along the main diagonal and its immediately adjacent diagonals, measuring the proportion of attention a token allocates to itself and its direct neighbors.

lem with dynamically varying token cardinality, enabling consistent alignment between agent population dynamics and interaction-aware motion generation.

- We propose Retrieval-based Traffic Evaluation (RTE), a context-aware evaluation framework that retrieves semantically similar real-world scenarios as reference anchors for metric computation, providing a more reliable assessment of long-horizon traffic simulation.

- We present extensive empirical analysis demonstrating why and how structural priors from large-scale sequence models (LLMs) can be adapted for interactive traffic simulation, highlighting their effectiveness in modeling multi-agent dynamics beyond language domains.

2 Related Works

Closed-loop Traffic Simulation. While early works adapted motion forecasting models [24, 32], recent state-of-the-art formulates simulation as next-token prediction [28,40,49,51], often enhanced by reinforcement learning [14,26] or advanced tokenization [50]. However, these approaches primarily target short-term generation and fail to address the complex population dynamics in long-term simulations.

Language Models for Non-Language Tasks. Beyond NLP, LLMs are increasingly adapted for diverse domains via task-to-text formatting [6, 23, 33] or instruction tuning [17, 20]. Crucially, fully frozen LLMs have proven effective as structural priors for continuous tasks like 3D scene understanding [22, 25]. Yet, exploiting these pretrained priors to orchestrate autoregressive long-term traffic dynamics remains a critical open gap that our work addresses.

3 Preliminary

We formulate the long-term traffic simulation problem as a conditional sequence generation task [28, 31]. Let s_t^i denote the state token (e.g., position, heading, velocity) of the i-th agent at timestep t. The global traffic scene is represented as $\mathcal{S}_t = \{s_t^1, s_t^2, \ldots, s_t^{N_t}\}$, where the active agent count N_t varies dynamically over an extended simulation horizon T [34, 46]. Unlike standard short-term simulations that assume a fixed set of agents, long-term simulations must continuously manage agent entries and exits. Therefore, we decompose the standard transition probability $p(\mathcal{S}_{t+1} \mid \mathcal{S}_t, \mathcal{C})$ [28, 40] into two tractable processes: *Parallel Motion Update* and *Autoregressive Agent Generation*.

3.1 Formulation

We introduce an intermediate state $\tilde{\mathcal{S}}_{t+1}$ representing the motion-updated scene of existing agents before any topological changes. The next-scene generation is factorized as:

$$p(\mathcal{S}_{t+1} \mid \mathcal{S}_t, \mathcal{C}) = \underbrace{p_{\text{agent}}(\mathcal{S}_{t+1} \mid \tilde{\mathcal{S}}_{t+1}, \mathcal{C})}_{\text{Autoregressive Generation}} \cdot \underbrace{p_{\text{motion}}(\tilde{\mathcal{S}}_{t+1} \mid \mathcal{S}_t, \mathcal{C})}_{\text{Parallel Motion Update}}. \tag{1}$$

For *Parallel Motion Update*, given the current scene $\mathcal{S}_t$, the future states of all N_t existing agents are predicted simultaneously. To ensure simulation efficiency, we approximate this joint update by assuming conditional independence among agents given the rich historical scene context $\mathcal{S}_t$ and map $\mathcal{C}$:

$$p_{\text{motion}}(\tilde{\mathcal{S}}_{t+1} \mid \mathcal{S}_t, \mathcal{C}) = \prod_{i=1}^{N_t} p(\tilde{s}^i_{t+1} \mid \mathcal{S}_t, \mathcal{C}). \tag{2}$$

For *Autoregressive Agent Generation*, conditioned on the motion-updated intermediate scene $\tilde{\mathcal{S}}_{t+1}$, the model autoregressively determines the final new scene $\mathcal{S}_{t+1}$. This process dynamically handles the spawning of new agents, ensuring consistent traffic density and logical boundary interactions.

3.2 Metrics for Traffic Simulation

Standard short-term evaluations rely on WOSAC metrics [24] (kinematic, interactive, and map-based realism) computed via strict one-to-one agent correspondence with ground truth logs. For long-term settings where agents dynamically enter and exit, previous works [34, 46] augment these with statistics like number of enter/exit, distance of enter/exit (#Enter, #Exit, D_{enter}, D_{exit}) and compute aggregated distribution divergences against the entire validation set.

4 RosettaSim

4.1 Motivation

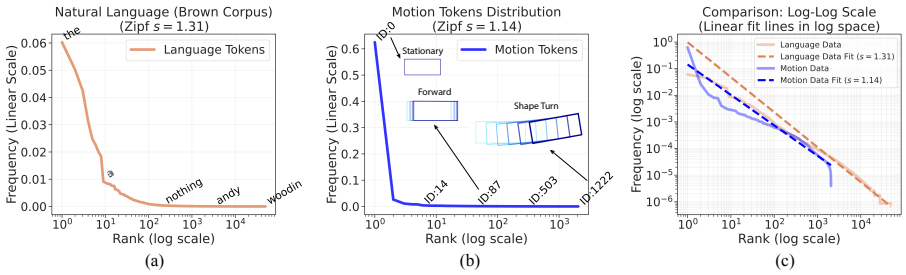

Fig. 2: The token frequency distribution of traffic motion tokens on WOMD [24]. The distribution follows Zipf's law [53], similar to natural language. (a) The frequency proportion of the language tokens [12].(b) The frequency proportion of motion tokens. (c) The comparison between motion tokens and language tokens in log-log space.

Recent advancements in traffic simulators [28, 38, 40, 44, 49] have increasingly embraced the formulation of large sequence models, adopting next-token-prediction paradigms and tokenizer-based designs [50]. However, extending standard short-term traffic simulation to long-horizon simulation presents a significant challenge. It requires a unified autoregressive framework capable of simultaneously handling continuous motion generation and discrete agent population

dynamics (i.e., agent entering and exiting) [27, 46]. Optimizing such a complex, unified sequence model from scratch is notoriously difficult, often suffering from training instability and severe compounding errors over long horizons.

To address this optimization bottleneck, we step back to investigate the fundamental statistical properties of discretized traffic dynamics. Inspired by [2], we empirically discover a striking alignment: the distribution of traffic motion tokens strictly follows Zipf's law [53] (Fig. 2(c)), a hallmark statistical property of natural language. As shown in Fig. 2(a) and (b), the frequency of motion tokens in WOMD [24] exhibits nearly identical distribution characteristics to language tokens, where a few specific actions (e.g., cruising, static) dominate the occurrences, while complex maneuvers (e.g., sharp turns) appear infrequently.

This critical observation suggests that pretrained LLMs may inherently encode similar token distributions during their massive text pretraining. Rather than treating traffic simulation as a completely isolated domain requiring from-scratch optimization, LLMs offer a highly optimized structural prior and initialization space. By conceptualizing traffic dynamics as a structured "foreign language", we can directly leverage these pretrained priors to stabilize the unified generation of motions and population dynamics.

4.2 Architecture

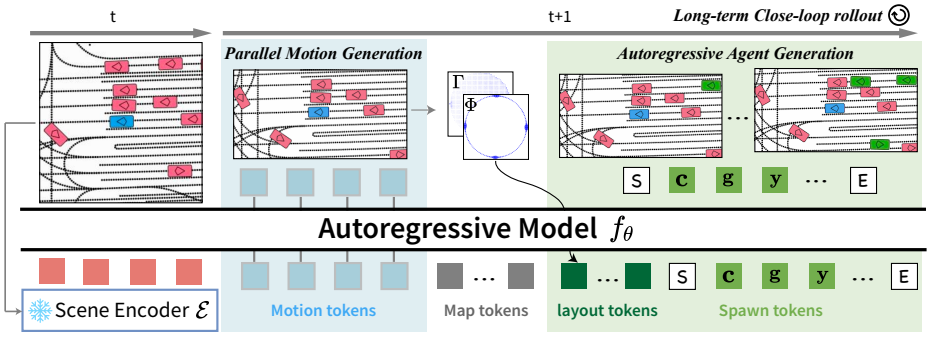

Fig. 3: Architecture of RosettaSim. We formulate the long-term traffic simulation as a structured sequence generation problem.

Parallel Motion Generation The primary goal is to effectively leverage the capabilities of pretrained large sequence model f_θ, e.g., LLMs. The overall architecture is illustrated in Fig. 3. We treat the pretrained transformer layers on [40] as our scene encoder $\mathcal{E}$ that encodes the agent-centered scene context into agent tokens: $\mathbf{a}_t^{1:N_t} = \mathcal{E}(\mathcal{S}_t, \mathcal{C})$. The dimension is aligned with LLM by MLP. Then, we use a set of learnable motion tokens $\mathbf{mo}^{1:N_t}$ to retrieve the motion context from agent tokens via self-attention in the language model f_θ:

$$\mathbf{mo}_t^{1:N_t} = f_\theta(\mathbf{mo}^{1:N_t}, \mathbf{a}_t^{1:N_t}). \tag{3}$$

To be noted that, before feeding all tokens into LLM, we will add agent-wise sinusoidal position embeddings to both agent tokens and motion tokens to encode

the agent identity:

$$\mathbf{mo}^i = \mathbf{mo}^i + \text{SinPE}(i), \quad \mathbf{a}_t^i = \mathbf{a}_t^i + \text{SinPE}(i), \tag{4}$$

where $\text{SinPE}(\cdot)$ is the sinusoidal position embedding function. The updated motion tokens $\mathbf{mo}_t^{1:N_t}$ are then projected to motion logits for each agent,

which are used to update motion in parallel:

$$p_{\text{motion}}(\cdot \mid \mathcal{S}_t, \mathcal{C}) = \text{Softmax}(\texttt{mlp}_{motion}(\mathbf{mo}_t^{1:N_t})) \tag{5}$$

For those newly spawned agents, we use a learnable embedding for $\mathcal{E}$ to get $\mathbf{a}$, instead of using a separate head to predict its initiated speed.

Autoregressive Agent Generation The process of agents entering the scene is analogous to sequence generation: each newly spawned agent depends on the existing traffic context, while the total number of agents remains variable but must remain semantically coherent. Importantly, newly generated agents should not disrupt the motion trajectories of agents already present in the scene. From this perspective, the current scene and motion tokens, $\{\mathbf{a}^{1:N_t}, \mathbf{mo}^{1:N_t}\}$ can be interpreted as a contextual prompt that conditions how traffic evolves.

To improve the accuracy and stability of this generation process, after obtaining the intermediate state $\tilde{\mathcal{S}}_{t+1}$, we introduce two additional topological-aware tokens as contextual input, map tokens $\mathbf{m}^{1:M_t}$ and layout tokens $\mathbf{l}^{1:N_t}$. The map tokens are extracted from a pretrained map encoder within scene encoder $\mathcal{E}$ and encode fine-grained geometric and topological relationships of the road network: $\mathbf{m}^{1:M_t} = \mathcal{E}(\mathcal{C})$. The layout tokens are consisted to two parts, grid tokens $\mathbf{g}$ and yaw tokens $\mathbf{y}$. Specifically, we will decompose the intermediate state $\tilde{\mathcal{S}}_{t+1}$ into position and orientation under ego-centered coordination. Following [46], we tokenize the position into a set of discretized grid index as well as the orientation into a set of discretized yaw index. Therefore, the layout tokens can be mathematically interpreted as:

$$\mathbf{l}^i = \mathbf{g}^i + \mathbf{y}^i + \text{SinPE}(i), \quad \mathbf{g}^i = \texttt{emb}_g(\Gamma(\tilde{s}_{t+1}^i)), \quad \mathbf{y}^i = \texttt{emb}_y(\Phi(\tilde{s}_{t+1}^i)), \tag{6}$$

where $\Gamma(\cdot)$ and $\Phi(\cdot)$ are tokenizers that map the grid index and yaw index from the intermediate state $\tilde{s}_{t+1}^i$, respectively. $\texttt{emb}_g(\cdot)$ and $\texttt{emb}_y(\cdot)$ are the embedding layers for grid and yaw tokens. Similar to motion tokens, we also add agent-wise sinusoidal position embeddings to layout tokens to encode agent identity.

As shown in Fig. 3, we use three tokens to represent the newly spawned agent: type token $\mathbf{c}$, grid token $\mathbf{g}$, and yaw token $\mathbf{y}$. During inference, we will first sample the agent type by agent type head: $c^j \sim \mathcal{P}(\texttt{mlp}_{type}(f_\theta(\texttt{<SOS>} + \text{SinPE}(j))))$. Here, `<SOS>` is the special token that stands for the start of autoregressive generation and $\mathcal{P}(\cdot)$ denotes sampling from a categorical distribution parameterized by the normalized logits. Then, conditioned on the sampled type token $\mathbf{c}^j$, f_θ will generate the grid token $\mathbf{g}^j$ and yaw token $\mathbf{y}^j$ autoregressively:

$$g^j \sim \mathcal{P}(\texttt{mlp}_{grid}(f_\theta(\mathbf{c}^j + \text{SinPE}(j)))), \quad y^j \sim \mathcal{P}(\texttt{mlp}_{yaw}(f_\theta(\mathbf{g}^j + \mathbf{c}^j + \text{SinPE}(j)))). \tag{7}$$

j is the index of the newly spawned agent. If the model decides to stop spawning new agents, it will sample a special token <EOS> from $\texttt{mlp}_{type}$. $\mathbf{g}^j$ and $\mathbf{y}^j$ are embedded from the same embedding layers as layout tokens. Therefore, the final number of agents are $N_{t+1} = N_t + J$ and the final generated sequence is $\mathbf{s} = [\texttt{<SOS>}, \mathbf{c}^1, \mathbf{g}^1, \mathbf{y}^1, \ldots, \mathbf{c}^J, \mathbf{g}^J, \mathbf{y}^J, \texttt{<EOS>}] \in \mathbb{R}^{(3J+2)\times D}$. D is the dimension of f_θ.

4.3 Training

Unlike previous works that either use complex weighted losses [46] or multiple stages [27], our training paradigm is fully one-stage with simple cross-entropy losses. Specifically, we optimize the following loss function:

$$\begin{aligned} \mathcal{L} &= \text{CausalCrossEntropy}(f_\theta(\texttt{seq}), \texttt{label}), \\ \texttt{seq} &= [\mathbf{a}^{1:N_t}, \mathbf{mo}^{1:N_t}, \mathbf{m}^{1:M_t}, \mathbf{l}^{1:N_t}, \mathbf{s}], \\ \texttt{label} &= [\texttt{pad}^{1:N_t}, \hat{mo}^{1:N_t}, \texttt{pad}^{1:M_t+N_t}, \hat{s}]. \end{aligned} \tag{8}$$

$\hat{mo}^{1:N_t}$ and $\hat{s}$ are the ground-truth motion tokens and spawn tokens, respectively. pad is the padding label that will be ignored during loss computation. Detailed tokenization and training hyperparameters will be described in the Appendix.

5 Retrieval-based Long-term Traffic Evaluation

5.1 Motivation

Evaluating long-term simulation is fundamentally challenging because dynamic agent entering/exiting invalidates strict one-to-one matching. Prior works [34,46] circumvent this by slicing rollouts into short windows and computing divergences against the entire validation set. However, this global matching suffers from severe statistical bias. For instance, forcing a high-speed scenario to match an overall validation distribution dominated by low-speed logs [41, 42] yields penalizing and misleading evaluations, particularly on Kinematic metrics (empirically confirmed in Fig. 8 and Sec. 6.2).

5.2 Approach

To address the aforementioned problems, one intuitive idea is to find the most similar scenario for the validation set for each simulated scenario and compute the distribution between them instead. Inspired by the LPIPS metric [48] in image generation, we propose to measure the similarity between two driving scenarios via a pretrained traffic-specific model's latent space. Works like [4, 5] has demonstrated that the latent space from a well-trained model can reflect the semantic similarity between two driving scenarios. Therefore, given a segment of long-term policy rollout, we can retrieve the top-K most similar scenarios from the validation set as reference and compute the statistical metric between them.

The whole pipeline is shown in Fig. 4. Specifically, given a simulated rollout $\mathcal{S}_{0:T}$, we will first divide it into several short-term segments $\{\mathcal{S}_{t:t+\tau}\}$, where τ is the length of each segment. Then, we will encode each segment into a latent representation via a scene VAE [18] $\mathcal{E}_\phi$ that was pretrained for scenario generation. We directly use the official checkpoint from [30] without any fine-tuning to ensure reproducibility. The latent representation is formulated as:

$$\mathbf{z} = \frac{1}{\tau}\sum_{t=1}^{\tau}\mathcal{E}_\phi(\mathcal{S}_t), \quad \mathbf{z} = [\mathbf{z}_{\text{object}}, \mathbf{z}_{\text{map}}]. \tag{9}$$

Where $\mathbf{z}_{\text{object}} := (\mu_o, \sigma_o)$ and $\mathbf{z}_{\text{map}} := (\mu_m, \sigma_m)$ are the object-level and map-level latent representation, respectively. We will also encode all the log segments in the validation set into the same latent space and build a retrieval database $\hat{\mathbf{z}}$. During evaluation, for each simulated segment, we will retrieve the top-K similar

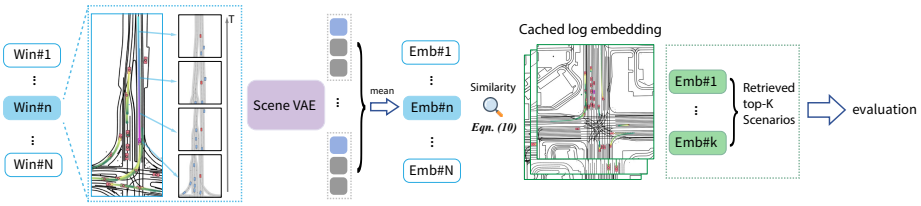

Fig. 4: Overview of our proposed retrieval-based evaluation (RTE) framework for long-term traffic simulation.

segments from the validation set based on the Wasserstein distance [37] between their latent representations:

$$\text{similarity}(\mathbf{z}, \hat{\mathbf{z}}) = W_2(\mathbf{z}_{\text{object}}, \hat{\mathbf{z}}_{\text{object}}) + \lambda \cdot W_2(\mathbf{z}_{\text{map}}, \hat{\mathbf{z}}_{\text{map}}). \tag{10}$$

Finally, we will compute the evaluation metrics between the simulated segment and the retrieved references.

5.3 Formulation of the Metrics

To rigorously evaluate long-term simulation, we propose the Realism Meta Metric F1 (RMM-F1), formulated as the harmonic mean of two critical dimensions: 1) *Behavior Realism*: Aggregates WOSAC-like metrics between generated segments and retrieved real-world references to capture fine-grained kinematic, interactive, and map-based fidelity. 2) *Traffic Flow Realism*: Evaluates macro-level population dynamics using agent entry/exit counts against the global log distribution. We use the harmonic mean ($\beta = 1$) to equally weight micro-level interactions and macro-level flows, analogous to the standard F1-score. This strictly penalizes models underperforming in either dimension, preventing deceptive compensation. Full formulations are detailed in the Appendix.

6 Results

6.1 Short-term Traffic Simulation

In this section, we manage to answer the following questions: 1) *How does our method perform on short-term traffic simulation compared with other methods?* 2) *How efficiently can large-scale sequence models adapt to continuous traffic dynamics utilizing their structural priors?*

Table 1: Comparison of different methods on WOSAC 2025 *test split*. '*' indicates results from WOSAC 2024. **Bold** and underlined denote the best and second-best performance. The method in gray (CAT-K) utilizes closed-loop tuning and is listed for reference. '-' indicates the metric is not applicable.

Method	LLM Util.	Agent Insertion	Composite Score ↑	Kinematic metrics ↑	Interactive metrics ↑	Map-based metrics ↑	min ADE ↓
ProSim* [33]	Llama2-7B [36]	✗	0.7180	0.4010	0.7780	0.8854	-
LLM2AD [38]	Arch. Inspired	✗	0.7779	0.4846	0.8048	0.9109	**1.2827**
SMART [40]	-	✗	0.7814	0.4854	0.8089	0.9153	1.3931
SimFormer [39]	-	✗	0.7820	0.4920	0.8060	**0.9167**	1.3221
UniMM [19]	-	✗	0.7829	**0.4914**	0.8089	0.9161	1.2949
CAT-K [49]	-	✗	0.7846	0.4931	0.8106	0.9177	1.3065
GUMP* [16]	GPT-2 Medium [1]	✓	0.7431	0.4780	0.7887	0.7404	1.6041
SceneStreamer [27]	Arch. Inspired	✓	0.7731	0.4492	0.8084	0.9127	1.4252
RosettaSim (Ours)	Qwen2.5-0.5B [45]	✓	**0.7833**	0.4905	**0.8105**	**0.9167**	1.3417

Comparison with SOTAs We evaluate RosettaSim on the standard 8s benchmark (Tab. 1). To align with this short-term setting, we disable the autoregressive agent spawning module during inference, requiring no additional retraining. RosettaSim achieves highly competitive results, outperforming all existing methods except those explicitly optimized with expensive closed-loop reinforcement learning or finetuning. Crucially, it significantly surpasses both LLM-inspired architectures and prior works utilizing pretrained LLMs. Furthermore, among the subset of versatile simulators capable of dynamic agent insertion, RosettaSim establishes a new state-of-the-art by a large margin. Lastly, while the recent long-term simulator InfGen [46] is absent from the WOSAC 2025 leaderboard, its reported metrics inherently trail behind SMART [40] even on a much smaller validation split. Since RosettaSim strictly outperforms SMART, it deductively establishes superiority over InfGen in this setting as well.

Ablation over tunable layers To investigate whether the performance gains stem from the pretrained linguistic knowledge or merely from increased model capacity, we conduct an ablation study by varying the number of tunable layers within the LLM backbone, while keeping the motion head and adapter layers trainable. As shown in Tab. 2, even when the LLM backbone is fully frozen ('None'), our method outperforms the baseline [40], indicating that the pretrained internal representations of the LLM already contain highly transferable structure-awareness. Furthermore, fine-tuning only the last transformer layer

('Last 1') achieves a Composite score of 0.7648, which is highly comparable to the fully fine-tuned result ('All', 0.7658). Interestingly, increasing the tunable scope to the last five layers ('Last 5') does not yield further improvements compared to 'Last 1'. This saturation suggests that the pretrained knowledge is robust and requires only minimal alignment to adapt to the traffic simulation.

To explicitly separate the contribution of pretrained knowledge from the underlying transformer architecture, we further compare our model against a randomly initialized Qwen-0.5B (Fig. 5). We observe that while architectural inductive biases alone account for 63% of the gain over the baseline, the pretrained statistical prior still contributes the remaining 37%, validating the distinct importance of LLM pretraining.

Table 2: Ablation study on the number of tunable layers within the LLM backbone. Only motion head and adapter layers are tunable. The experiment is conducted on 2% of the WOSAC val split.

Tunable Layers	Composited ↑	Kinematic↑	Interactive ↑	Map-based↑
All	0.7658	0.4854	0.8095	0.8697
Last 5	0.7647	0.4854	0.8083	0.8682
Last 1	0.7648	0.4876	0.8084	0.8671
None	0.7638	0.4843	0.8074	0.8674
baseline [40]	0.7631	0.4829	0.8065	0.8675

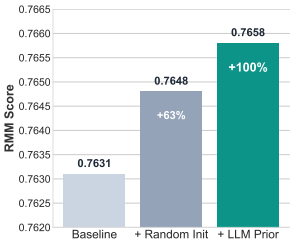

Fig. 5: Ablation study on LLM's prior. All parameters are tunable.

Can all LLMs be adapted to traffic simulation? To investigate the impact of foundational architectures, we adapt four LLMs under a fixed single-epoch budget to evaluate their rapid adaptation capability. As shown in Tab. 3, modern LLMs significantly outperform older architecture like GPT-2, proving that modern LLMs are adaptable. Notably, Qwen2.5-0.5B achieves the best performance, surpassing larger models like Llama-3.2. We hypothesize that Qwen's extensive multilingual pre-training fosters a more generalized sequence modeling capability. By treating motion tokens as a "foreign language", models exposed to diverse linguistic syntaxes during pre-training can transfer their generalized attention mechanisms more effectively than monolingual dominant models.

Table 3: Ablation study on adaptation abilities over different LMs. Evaluated on 2% of WOSAC *val split*, all the transformer layers on LMs are frozen.

LLMs	WOSAC Metric				Model Card	
	Composited↑	Kinematic ↑	Interactive↑	Map-based↑	Pretrain Tokens	Supported Languages
GPT-2-Small [1] (2019)	0.5425	0.3155	0.5590	0.6510	-	-
TinyLlama-1.1B-Chat-v1.0 [47] (2023)	0.7397	0.4684	0.7693	0.8566	3T	1
Llama-3.2-1B-Instruct [13] (2024)	0.7594	0.4817	0.8001	0.8658	9T	8
Qwen2.5-0.5B-Instruct [45] (2025)	0.7620	0.4835	0.8051	0.8656	18T	29

Why LLMs can be adapted to traffic simulation? To understand why LLMs can be adapted to traffic simulation, inspired by previous works [2], we manage to explain this in two aspects: 1) the data distribution; 2) the attention mechanism. As shown in Fig. 7, during the training, we resample the motion

tokens and enforce that the distribution follows a zipfian distribution with a different coefficient. We find that when the distribution is not extremely skewed (e.g., a uniform distribution, zipfian coefficient is 0), the performance will drop significantly, which proves our assumption. In addition, to fully understand why some LLMs are better than others, we visualize the locality of attention during inference under different sources of data across all layers. As shown in Fig. 6, the attention locality under text data [15] and motion data on Qwen2.5-0.5B are similar across different layers, where Llama and TinyLlama are not. These results give us more direct evidence that how LLMs understand language can be metaphorically transferred to how they understand motion tokens.

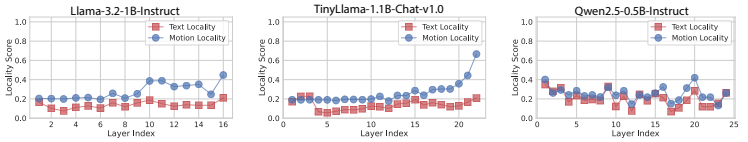

Fig. 6: Comparison of the attention mechanism in different LLMs during inference with different sources of data. A higher value means the model is focusing more on adjacent tokens.

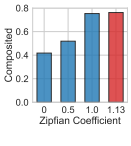

Fig. 7: Zipfian corruption.

Does scaling LLMs help traffic simulation? Finally, we want to examine whether scaling up model size benefits traffic simulation, given the intuition that larger models possess richer knowledge. We examined scaling behavior within the same model family (e.g., Qwen2.5 series). As shown in Tab. 4, increasing model size yields no improvement under the frozen setting; in fact, performance drops at 3B. However, a compelling phenomenon emerges when finetuning the last layer: despite the initial performance variance in the frozen setting, the final performance converges across different model sizes (Tab. 5 and Tab. 6).

This consistency suggests that the gains from language pretraining primarily stem from general syntactic capabilities, such as modeling sequential dependencies and statistical distributions, rather than deep, specific semantic knowledge. The "grammar" of traffic dynamics is relatively fundamental and can be fully captured by smaller architectures (e.g., 1B-1.5B). Consequently, the advanced reasoning capabilities and specific world knowledge inherent in larger models (e.g., >3B) provide diminishing returns or are even redundant for motion generation. This indicates that traffic simulation relies more on the structural similarity to language than on the complexity of the model's internal knowledge base.

Table 4: Comparison of different model sizes. LLMs are frozen.

Models	Composited
Qwen2.5-0.5B	0.7620
Qwen2.5-1.5B	0.7620
Qwen2.5-3B	0.7589

Table 5: Comparison of different model sizes. Only tune the last layer.

Models	Composited
Qwen2.5-0.5B	0.7637
Qwen2.5-1.5B	0.7644
Qwen2.5-3B	0.7641

Table 6: Comparison of different model genres. Only tune the last layer.

Models	Composited
Qwen2.5-0.5B	0.7637
TinyLlama-1.1B	0.7646
Llama-3.2-1B	0.7650

6.2 Long-term Traffic Simulation

In this section, we want to answer the following questions: 1) *How well does RTE reflect the quality of long-term traffic simulation?* 2) *How does our method perform on long-term traffic simulation?*

Rollout setup Following the practice in [46], we set the rollout length to $T = 31.1s$; there is no early stopping during rollout. If one ego-vehicle driving segment does not contain any map within a range of 64×64 throughout the window length τ, we will exclude this from evaluation.

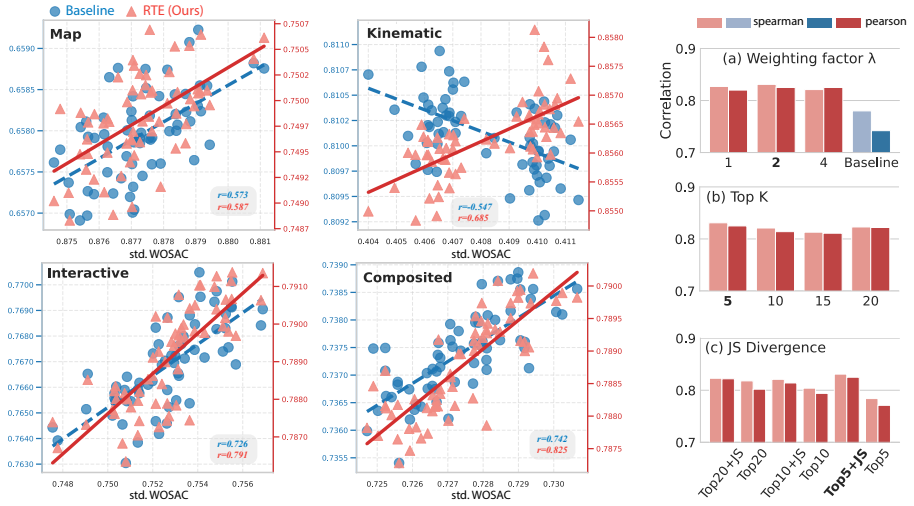

Fig. 8: Correlations. Left: Pearson correlations comparison between RTE and log-based metrics against standard metrics. Right: Ablation of RTE's parameters.

How well does RTE reflect the quality of long-term traffic simulation? Ideally, metrics designed for long-term simulation should exhibit high correlation with standard short-term WOSAC metrics when evaluated on a single short-term window. To validate this, we generate 63 diverse rollout variants by applying stochastic decoding (top-48 sampling) to two representative base models [40, 49]. Evaluating these variants on 10% of the validation set over a standard 8s duration, we compute the Pearson correlation (r, with $p \approx 0$) between the standard WOSAC metrics and both our proposed RTE-based metrics and prior log-based metrics.

As shown in Fig. 8 (left), RTE-based metrics (red) demonstrate significantly higher correlations than log-based metrics (blue). This confirms that RTE accurately reflects simulation quality even without strict one-to-one matching. Notably, log-based Kinematic metrics exhibit a *negative* correlation with standard WOSAC metrics. This empirically validates our core motivation (Sec. 5): blindly forcing specific local scenarios (e.g., low-speed traffic) to match a global kinematic distribution inevitably penalizes realistic behaviors.

Furthermore, Fig. 8 (right) presents an ablation on RTE hyperparameters (K and λ), demonstrating robust stability across various settings. We also find that applying Jensen-Shannon (JS) divergence to Bernoulli-distributed metrics (e.g., collision and offroad likelihood) further improves overall correlation. This indicates that without a rigid one-to-one assumption, JS divergence effectively smooths distribution discrepancies, yielding a more reliable quality assessment.

Evaluation results Tab. 7 presents the quantitative comparison against InfGen [46], CAT-K [49], and SMART [40]. RosettaSim establishes a new state-of-the-art on the comprehensive RMM-F1 metric (0.7624). Notably, it substantially outperforms the autoregressive baseline, InfGen, in Traffic Flow Realism (0.7846 vs. 0.7637). This validates that our unified architecture effectively manages dynamic population density and maintains global flow consistency over extended horizons. Regarding Behavior Realism, RosettaSim exhibits outstanding performance in the Kinematic metric, significantly outperforming all baselines. Rather than overfitting to specific interactive or turning behaviors, RosettaSim achieves a significantly better holistic balance, ensuring robust performance across both behavioral realism and global traffic flow. Methods lacking long-term optimization often show better behavioral realism because they do not spawn new agents over time. This lower agent density results in fewer collision opportunities, inflating their performance scores.

Furthermore, we report the metrics under D_{enter} and D_{exit} strictly for completeness. We exclude these metrics from the computation of RMM-F1 score as they can easily exploited by heuristic agent removal. Exhaustive breakdowns under each metrics are deferred to the Supp.

Table 7: Performance under long-term traffic simulation.

Method	RMM-F1	Traffic Flow Realism					Behavior Realism			
		Overall	#Enter	#Exit	D_{enter}	D_{exit}	Overall	Kinematic	Interactive	Map-based
InfGen [46]	0.7290	0.7637	0.7436	0.7838	0.2734	0.1766	0.6973	0.7312	0.7445	0.6172
CAT-K [49]	0.7381	0.7191	0.6862	0.7520	0.2294	0.3167	0.7581	0.8582	0.8256	0.6140
SMART [40]	0.7383	0.7175	0.6862	0.7489	0.2291	0.3175	0.7602	0.8619	0.8280	0.6150
RosettaSim (Ours)	0.7624	0.7846	0.7870	0.7821	0.2336	0.3095	0.7414	0.9176	0.7663	0.6089

7 Conclusion

In this work, we propose RosettaSim, a unified framework that seamlessly handles both behavior modeling and traffic flow generation. Our method achieves state-of-the-art performance on both short-term and long-horizon traffic simulation benchmarks. Moreover, we introduce RTE, a novel evaluation protocol that better captures the fidelity of long-horizon traffic simulation.

Limitations and future works We rely on heuristics for agent removal for its simplicity, though a separate head or special tokens could easily be integrated later. Future work would introduce properties like initial speeds for spawned agents to improve long-term evaluation and refine the binary collision indicator to account for collision frequency, severity, and pattern [11].

Acknowledgement This work was supported by The University of Hong Kong (HKU) Faculty Interdisciplinary Fund and the HKU Urban Systems Institute (HKU-USI) Fellowship Grant.

References

1. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
2. Chan, S., Santoro, A., Lampinen, A., Wang, J., Singh, A., Richemond, P., McClelland, J., Hill, F.: Data distributional properties drive emergent in-context learning in transformers. Advances in neural information processing systems **35**, 18878–18891 (2022)
3. Chitta, K., Dauner, D., Geiger, A.: Sledge: Synthesizing driving environments with generative models and rule-based traffic. In: European Conference on Computer Vision. pp. 57–74. Springer (2024)
4. Ding, W., Cao, Y., Zhao, D., Xiao, C., Pavone, M.: Realgen: Retrieval augmented generation for controllable traffic scenarios. In: European Conference on Computer Vision. pp. 93–110. Springer (2024)
5. Ding, W., Veer, S., Chen, Y., Cao, Y., Xiao, C., Pavone, M.: Realdrive: Retrieval-augmented driving with diffusion models. arXiv preprint arXiv:2505.24808 (2025)
6. Dinh, T., Zeng, Y., Zhang, R., Lin, Z., Gira, M., Rajput, S., Sohn, J.y., Papailiopoulos, D., Lee, K.: Lift: Language-interfaced fine-tuning for non-language machine learning tasks. Advances in Neural Information Processing Systems **35**, 11763–11784 (2022)
7. Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., Chai, Y., Sapp, B., Qi, C.R., Zhou, Y., et al.: Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9710–9719 (2021)
8. Feng, L., Li, Q., Peng, Z., Tan, S., Zhou, B.: Trafficgen: Learning to generate diverse and realistic traffic scenarios. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 3567–3575 (2023)
9. Feng, S., Sun, H., Yan, X., Zhu, H., Zou, Z., Shen, S., Liu, H.X.: Dense reinforcement learning for safety validation of autonomous vehicles. Nature **615**(7953), 620–627 (2023)
10. Feng, S., Zhu, H., Sun, H., Yan, X., He, L., Yang, J., Su, G., Li, B., Li, S., Wang, L., et al.: Breaking through safety performance stagnation in autonomous vehicles with dense learning. Nature Communications (2026)
11. Feng, Z., Xiao, L., Yan, X.: Beyond binary metrics: Unveiling the safety illusion in autonomous driving simulation. In: CVPR Workshop on Simulation for Autonomous Driving (2026)
12. Francis, W.N.: Brown corpus maunal: manual of information to accompany a standard corpus of present-day edited American English for use with digital computers. Dept. of Linguistics, Brown University, Providence, R.I, revised and amplified 1979. edn. (1979)
13. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
14. Guo, K., Liu, H., Wu, X., Lv, C.: Decompgail: Learning realistic traffic behaviors with decomposed multi-agent generative adversarial imitation learning. arXiv preprint arXiv:2510.06913 (2025)

15. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding. In: International Conference on Learning Representations
16. Hu, Y., Chai, S., Yang, Z., Qian, J., Li, K., Shao, W., Zhang, H., Xu, W., Liu, Q.: Solving motion planning tasks with a scalable generative model. In: European Conference on Computer Vision. pp. 386–404. Springer (2024)
17. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., et al.: Openvla: An open-source vision-language-action model. In: Conference on Robot Learning. pp. 2679–2713. PMLR (2025)
18. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
19. Lin, L., Lin, X., Xu, K., Lu, H., Huang, L., Xiong, R., Wang, Y.: Unimm: A unified mixture model framework for multi-agent simulation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2026)
20. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
21. Liu, H., Cao, Z., Yan, X., Feng, S., Lu, Q.: Autonomous vehicles: A critical review (2004-2024) and a vision for the future. Authorea Preprints (2025)
22. Lu, K., Grover, A., Abbeel, P., Mordatch, I.: Frozen pretrained transformers as universal computation engines. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 7628–7636 (2022)
23. Mirchandani, S., Xia, F., Florence, P., Ichter, B., Driess, D., Arenas, M.G., Rao, K., Sadigh, D., Zeng, A.: Large language models as general pattern machines. In: Conference on Robot Learning. pp. 2498–2518. PMLR (2023)
24. Montali, N., Lambert, J., Mougin, P., Kuefler, A., Rhinehart, N., Li, M., Gulino, C., Emrich, T., Yang, Z., Whiteson, S., et al.: The waymo open sim agents challenge. Advances in Neural Information Processing Systems **36**, 59151–59171 (2023)
25. Pang, Z., Xie, Z., Man, Y., Wang, Y.X.: Frozen transformers in language models are effective visual encoder layers. In: The Twelfth International Conference on Learning Representations
26. Pei, M., Shi, S., Shen, S.: Advancing multi-agent traffic simulation via r1-style reinforcement fine-tuning. arXiv preprint arXiv:2509.23993 (2025)
27. Peng, Z., Liu, Y., Zhou, B.: Scenestreamer: Continuous scenario generation as next token group prediction. In: The Fourteenth International Conference on Learning Representations (2026)
28. Philion, J., Peng, X.B., Fidler, S.: Trajeglish: Traffic modeling as next-token prediction. In: The Twelfth International Conference on Learning Representations
29. Rowe, L., Girgis, R., Gosselin, A., Carrez, B., Golemo, F., Heide, F., Paull, L., Pal, C.: Ctrl-sim: Reactive and controllable driving agents with offline reinforcement learning. In: Conference on Robot Learning. pp. 3600–3621. PMLR (2025)
30. Rowe, L., Girgis, R., Gosselin, A., Paull, L., Pal, C., Heide, F.: Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17207–17218 (2025)
31. Seff, A., Cera, B., Chen, D., Ng, M., Zhou, A., Nayakanti, N., Refaat, K.S., Al-Rfou, R., Sapp, B.: Motionlm: Multi-agent motion forecasting as language modeling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8579–8590 (2023)

32. Shi, S., Jiang, L., Dai, D., Schiele, B.: Motion transformer with global intention localization and local movement refinement. Advances in Neural Information Processing Systems (2022)
33. Tan, S., Ivanovic, B., Chen, Y., Li, B., Weng, X., Cao, Y., Krähenbühl, P., Pavone, M.: Promptable closed-loop traffic simulation. In: 8th Annual Conference on Robot Learning (CoRL) (2024)
34. Tan, S., Lambert, J., Jeon, H., Kulshrestha, S., Bai, Y., Luo, J., Anguelov, D., Tan, M., Jiang, C.M.: Scenediffuser++: City-scale traffic simulation via a generative world model. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 1570–1580 (June 2025)
35. Tan, S., Wong, K., Wang, S., Manivasagam, S., Ren, M., Urtasun, R.: Scenegen: Learning to generate realistic traffic scenes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2021)
36. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
37. Villani, C.: The wasserstein distances. In: Optimal transport: old and new, pp. 93–111. Springer (2009)
38. Wang, M., Wang, J., Ye, T., Yu, K.: Do llm modules generalize? a study on motion generation for autonomous driving. In: Conference on Robot Learning. pp. 4657–4683. PMLR (2025)
39. Wang, S., Xu, J., Zhang, X., Hu, F., Huang, Z., Luo, J., Zhu, K., Zhu, J., Zhou, Y., Chen, Z.: Improving tokenization of agents and maps with transformers for multi-agent simulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Workshop on Autonomous Driving (WAD) (2025)
40. Wu, W., Feng, X., Gao, Z., KAN, Y.: Smart: Scalable multi-agent real-time motion generation via next-token prediction. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 114048–114071. Curran Associates, Inc. (2024), https://proceedings.neurips.cc/paper_files/paper/2024/file/cef5c8dec67597b854f0162ad76d92d2-Paper-Conference.pdf
41. Xiao, L., Liu, J.J., Yang, S., Li, X., Ye, X., Yang, W., Wang, J.: Learning multiple probabilistic decisions from latent world model in autonomous driving. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 1279–1285. IEEE (2025)
42. Xiao, L., Liu, J.J., Ye, X., Yang, W., Wang, J.: Easychauffeur: A baseline advancing simplicity and efficiency on waymax. arXiv preprint arXiv:2408.16375 (2024)
43. Yan, X., Feng, S., Sun, H., Liu, H.X.: Distributionally consistent simulation of naturalistic driving environment for autonomous vehicle testing. IEEE Transactions on Intelligent Transportation Systems **26**(7), 9187–9200 (2025). https://doi.org/10.1109/TITS.2025.3571966
44. Yan, X., Zou, Z., Feng, S., Zhu, H., Sun, H., Liu, H.X.: Learning naturalistic driving environment with statistical realism. Nature communications **14**(1), 2037 (2023)
45. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
46. Yang, X., Tan, S., Krähenbühl, P.: Long-term traffic simulation with interleaved autoregressive motion and scenario generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 25305–25314 (2025)
47. Zhang, P., Zeng, G., Wang, T., Lu, W.: Tinyllama: An open-source small language model (2024)

48. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
49. Zhang, Z., Karkus, P., Igl, M., Ding, W., Chen, Y., Ivanovic, B., Pavone, M.: Closed-loop supervised fine-tuning of tokenized traffic models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
50. Zhang, Z., Jia, X., Chen, G., Li, Q., Wu, Z., Jiang, Y.G., Yan, J.: Trajtok: What makes for a good trajectory tokenizer in behavior generation? In: The Fourteenth International Conference on Learning Representations
51. Zhao, J., Zhuang, J., Zhou, Q., Ban, T., Xu, Z., Zhou, H., Wang, J., Wang, G., Li, Z., Li, B.: Kigras: Kinematic-driven generative model for realistic agent simulation. IEEE Robotics and Automation Letters **10**(2), 1082–1089 (2024)
52. Zhou, Z., Hu, H., Chen, X., Wang, J., Guan, N., Wu, K., Li, Y.H., Huang, Y.K., Xue, C.J.: Behaviorgpt: Smart agent simulation for autonomous driving with next-patch prediction. Advances in Neural Information Processing Systems **37**, 79597–79617 (2024)
53. Zipf, G.K.: Human behavior and the principle of least effort: an introduction to human ecology. Hafner Pub. Co, New York, facsim. of 1949 edition. edn. (1965)