SMART: Scalable Multi-agent Real-time Simulation via Next-token Prediction

Wei Wu[*]
Tsinghua University
SenseTime Research
wuwei@senseauto.com

Xiaoxin Feng[*]
SenseTime Research
fengxiaoxin@senseauto.com

Ziyan Gao[*]
SenseTime Research
gaoziyan@senseauto.com

Yuheng Kan
SenseTime Research
kanyuheng@senseauto.com

Abstract

Data-driven autonomous driving motion generation tasks are frequently impacted by the limitations of dataset size and the domain gap between datasets, which precludes their extensive application in real-world scenarios. To address this issue, we introduce SMART, a novel autonomous driving motion generation paradigm that models vectorized map and agent trajectory data into discrete sequence tokens. These tokens are then processed through a decoder-only transformer architecture to train for the next token prediction task across spatial-temporal series. This GPT-style method allows the model to learn the motion distribution in real driving scenarios. SMART achieves state-of-the-art performance across most of the metrics on the generative Sim Agents challenge, ranking 1st on the leaderboards of Waymo Open Motion Dataset (WOMD), demonstrating remarkable inference speed. Moreover, SMART represents the generative model in the autonomous driving motion domain, exhibiting zero-shot generalization capabilities: Using only the NuPlan dataset for training and WOMD for validation, SMART achieved a competitive score of 0.72 on the Sim Agents challenge. Lastly, we have collected over 1 billion motion tokens from multiple datasets, validating the model's scalability. These results suggest that SMART has initially emulated two important properties: scalability and zero-shot generalization, and preliminarily meets the needs of large-scale real-time simulation applications. We have released all the code to promote the exploration of models for motion generation in the autonomous driving field. The source code is available at https://github.com/rainmaker22/SMART.

1 Introduction

In the context of autonomous driving, leveraging vectorized maps and vehicle trajectory data facilitates various motion generation tasks, including motion planning [17, 6, 19, 18, 7], motion prediction [47, 11, 38], and Sim Agents [14]. Previous research [9, 5, 27] has predominantly employed encoder networks to represent driving scenes and decoder networks to generate multi-modal motions. These generated motions are then directly regressed to continuous trajectory distributions using Gaussian [4] or Laplace [53] mixture loss functions. While this framework demonstrates strong performance in prediction tasks that prioritize regression accuracy, it often underperforms in motion generative tasks that emphasize the safety and reasonableness of driving behavior, such as planning [3] or Sim Agents [26]. The primary reasons for this underperformance are as follows: First, the framework does not

[*]Equal contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

represent future interactions between the motions of different agents, leading to inconsistent scene-level forecasting. Second, the model generates multi-modal motion by initializing multiple intention queries in the decoder, which is typically limited by GPU memory, resulting in a fixed number of motion modalities. Consequently, it is uncertain whether the generated modalities sufficiently represent the diversity of future behaviors. Thirdly, these models struggle to generalize across different datasets, requiring new data collection for training in new urban environments or maps.

The advent of autoregressive large language models (LLMs) [12, 42] has ushered in a new era in artificial intelligence. Drawing inspiration from this, some studies in the driving motion generation domain[30, 34], have tokenized agent trajectories into discrete motion tokens and employed a Next Token Prediction (NTP) task based on cross-entropy loss for autoregression. These models continue to utilize an encoder-decoder architecture, encoding continuous vectorized map and historical trajectory data with an encoder, and decoding discrete tokens solely in the decoder module. Compared to continuous distribution regression methods, the autoregressive paradigm of NTP has the following advantages: the model adopts a step-by-step next token prediction, allowing it to model interactions between agents' motions at each time step, and the number of modalities is not limited, leading to better diversity in generative tasks.

However, existing NTP-based motion models still fail to address the aforementioned issues of generalizability and scalability, which have a critical impact on industrial applications. Generalizability means achieving satisfactory results across diverse datasets through zero-shot and few-shot learning, while scalability involves improving model performance as dataset size or model parameters increase, following scaling laws defined by [16]. This shortfall is due to two main factors: First, current model architectures lack generalizability under the constraints of limited data scale. Due to the high cost of acquiring extensive driving data, open-source datasets typically cover only a few hundred hours of driving in specific urban areas, with significant domain gaps caused by perceptual and regional differences. Second, unlike tasks involving the serialization of a single dimension, motion generation requires the serialization of both the temporal dimension of trajectories and the spatial interactions between maps and agents. To tackle these challenges, this paper introduces the SMART model: Scalable Multi-Agent Real-Time Motion Generation via Next-token Prediction. The model incorporates a tokenizer for map data and proposes an autoregressive prediction task for the next road token prediction to enhance the model's spatial comprehension. Subsequently, a GPT-style approach is adopted, tokenizing agent trajectories across the entire time series to establish a decoder-only transformer model. The decoder-only transformer allows SMART to compute the next token for the upcoming frame at the current moment during inference, eliminating the need to re-encode historical motion tokens with each inference, which significantly improves inference efficiency for real-time interactive autonomous driving simulation.

In summary, our contributions to the community include: (1) We propose a novel framework for motion generation, incorporating a tokenization scheme for both vectorized road and agent trajectories and utilizing a decoder-only transformer for training on the next token prediction task. This approach offers new insights into the design of motion generation algorithms for autonomous driving. (2) In the field of driving motion generation, we have pioneered a focus on the model's zero-shot generalizability across different datasets. Notably, the model trained solely on the NuPlan dataset performed well on the WOMD test dataset, despite the lack of overlap between the map areas of these two datasets. An empirical validation of SMART models' scalability emulates the appealing properties of large fundamental models. (3) SMART achieves state-of-the-art performance across most metrics in the generative Sim Agents challenge, ranking **1$^{\text{th}}$** on the WOMD leaderboards[2]. Furthermore, SMART's single-frame inference time is within 15ms, meeting the real-time requirements for interactive simulation in autonomous driving.

2 Related work

2.1 Properties of auto-regressive large models

Scalability and zero-shot generalization Power-law scaling laws [22, 12, 31] mathematically describe the relationship between the growth of model parameters, dataset sizes, computational resources, and the performance improvements of machine learning models, providing several distinct

[2] `https://waymo.com/open/challenges/2024/sim-agents/`

benefits. Firstly, they enable the extrapolation of a larger model's performance by scaling up model size, data size, and computational cost. Secondly, the scaling laws have demonstrated a consistent and non-saturating increase in performance, corroborating their sustained advantage in enhancing model capabilities. Zero-shot generation refers to the ability of models to generate predicted motions for time series from unseen datasets. Previous work [29, 21] on zero-shot generation typically involves training on a single time series dataset and testing on a different dataset. In this study, we utilize the NuPlan dataset for training SMART models and the WOMD validation dataset for testing. Existing methods in the autonomous driving field [37, 40] often rely on LLMs or VLMs to assist in decision-making and planning to enhance generalizability and interpretability. However, no studies have attempted to directly construct a foundational model for the driving motion field to validate scalability and zero-shot generalizability.

2.2 Tokenizer in continuous domains

Language models [42, 43] rely on Byte Pair Encoding or WordPiece algorithms for text tokenization. Visual generation models[49, 48] based on language models also necessitate the encoding of 2D images into 1D token sequences. Early endeavors VQVAE [44] have demonstrated the ability to represent images as discrete tokens, although the reconstruction quality was relatively moderate. In the driving motion domain, MotionLM[34] used a simple uniform quantization of axis-aligned deltas between consecutive waypoints of agent trajectories.

2.3 Driving motion generation

Our work builds heavily on recent advancements in driving motion generation. A comprehensive range of generative models has been applied to this problem, including continuous motion distribution regression [33, 1, 39], diffusion models [50, 20], and discrete autoregressive models [30, 34]. MotionDiffuser [20] is a diffusion-based representation method for modeling the joint distribution of future trajectories across multiple agents, leveraging a simple predictor design and PCA compression for efficient, top-performing multi-agent motion prediction. While these diffusion-based models produce multi-modal future trajectories of individual agents, they only capture the marginal distributions of possible agent movements and do not model interactions among agents' future motions. Typical distribution regression models use parametric continuous distributions such as Gaussian [36] or Laplace [53] to model the future motion distribution. A limitation of these models is the uncertainty of whether the Gaussian or Laplace mixture distribution is flexible enough to represent the distribution over future states. Additionally, to generate multi-modal future motions, these models often need to incorporate motion goal candidates [13] or learnable latent embeddings [45] as multi-modal queries in the decoder module, resulting in significant memory usage and increased inference time. MotionLM [34] treats multi-agent motion prediction in autonomous vehicles as a language modeling task, generating interactive trajectories through a simplified autoregressive process without requiring complex optimizations and latent anchor embeddings. On this basis, Trajeglish [30] targets multi-agent offline closed-loop simulation.

3 Method

In this section, we introduce SMART, an autoregressive generative model for dynamic driving scenarios. While both language and agent motions are sequential, they differ in their representation—natural language consists of words from a finite vocabulary, whereas agent motions are continuous real-valued data. This distinction necessitates the unique design outlined in Sec. 3.1 for agent motion and road vector tokenizer, including the construction of vocabulary and the tokenization of motion sequences. Sec. 3.2 provides a comprehensive description of the model's architecture. Sec. 3.3 elaborates on the training tasks designed for the proposed model to learn the distribution of the motion token within the temporal sequence and the distribution of the road token within the spatial sequence.

3.1 Tokenization

Agent Motion tokenization To apply discrete sequence modeling in continuous domains, prior works typically follow one of two approaches: either use a pre-trained tokenizer, such as VQVAE [44] or VQGAN [10], to encode continuous features into discrete tokens, or normalize continuous

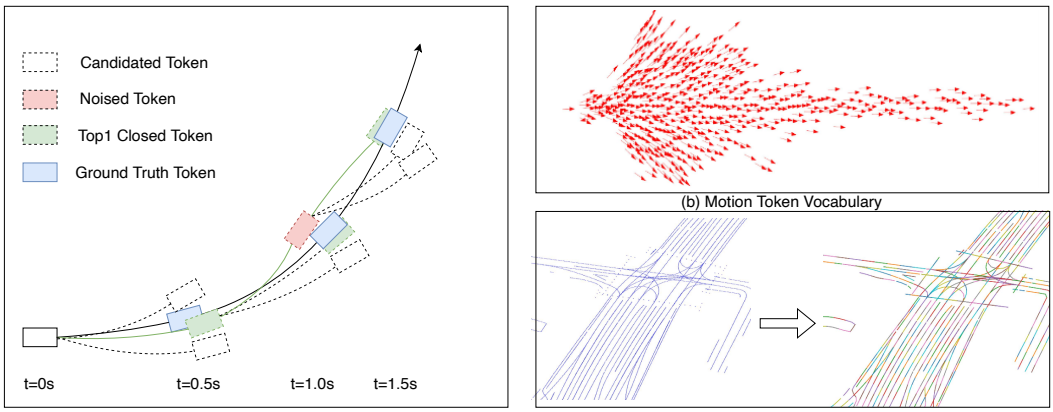


Figure 1: (a) At time t=0s, the current vehicle state is used as the reference to select the token closest to the ground truth bounding box within the token set. At time t=0.5s, the matched token from the previous step is used to select the next predicted token. At time t=1.0s, a noised token serves as the reference to determine the token for t=1.5s. This iterative process continues. (b) Motion token vocabulary with time granularity equal to 0.5s.(c) The original road vector features are represented as continuous sequences of map points. We divide the original map into multiple segments, each within 5 meters in length, and then perform matching with discrete tokens. The final map is composed of road vector tokens represented by different colored segments.

features and divide continuous values into discrete slots at equal intervals [2, 34]. For the former approach, establishing a latent vocabulary often requires a large amount of raw data to train the tokenizer; otherwise, the tokenizer itself will be biased towards the pre-training dataset. Since our work aims to enable the model to generalize effectively when trained on a small number of data samples, SMART opts to discretize explicit trajectory and map features. Specifically, similar to [30], we segment the continuous trajectories of all agents in the dataset into trajectory sets by fixed time intervals $t = 0.5s$. Then, we cluster the trajectory sets using the k-disks algorithm. As shown in Figure 1(b), the sampled trajectories serve as our final agent motion token vocabulary V_a.

As shown in Figure 1(a), the blue box represents the tokens obtained after discretizing the ground truth trajectory. At every 0.5-second interval, a search is conducted within the token vocabulary for candidate tokens, from which an appropriate(closest) token is selected to represent the current moment. Note that to prevent matching errors that may occur during the tokenization process of the agent motion sequence, we implement a rolling matching approach for the entire continuous motion sentence in a given period T. This implies that the token for the next time step is matched by referring to the position of the token currently matched, rather than relying on the actual correct position. However, due to the transformer decoder must perform sequential inference step by step, this approach inevitably leads to out-of-distribution issues due to compounding errors[32]. Especially, in the field of autonomous driving, these accumulated errors may result in collisions and off-map events[51]. To address this issue, we introduce noise into the tokenization process to enable the model to simulate distribution shifts during training. Specifically, we perturb the currently matched token by selecting one from the top-k tokens closest to the ground truth token in the vocabulary. Then, in the next time step, we match the motion token based on the perturbed vehicle state. This data augmentation method allows the model to effectively handle issues such as distribution shifts and accumulated errors, thereby enhancing robustness in generative tasks. Finally, the agent motion token is represented as $A \in \mathbb{R}^{N_A \times N_T \times F_A}$, where N_A denotes the total number of agents, and N_T represents the number of time steps, with a feature size of F_A, containing coordinates, heading, and shapes.

Road vector tokenization To enhance the model's generalization capabilities, we have applied a similar tokenization process to road vectors as we did with agent motion. Each road vector is a directed lane segment with features including start and end positions, length, turn direction, and other semantics from the dataset. To obtain fine-grained inputs for the road network, all road vectors are segmented into tokens spanning no longer than 5 meters in length. Unlike the motion sequence,

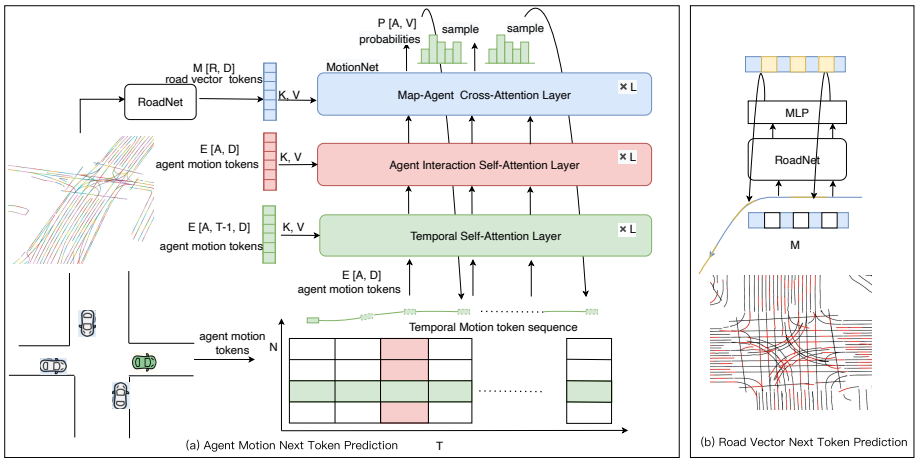


Figure 2: The architecture of SMART framework (a) We train a decoder-only transformer that predicts the motion tokens of multi-agents conditional on previous motion tokens, interactive agent motion tokens, and encoding road tokens. The model is trained to predict the next motion token. (b) Illustration for our proposed road spatial understanding training task.

the tokenization process of the road sentence does not have a time-series dependency. As shown in Figure 1(c), the tokenization of the road sentence is performed in parallel, directly tokenizing all the original road vector segments. The road vector token is represented as $R \in \mathbb{R}^{N_R \times F_R}$, where N_R denotes the total number of road vectors, and F_A represents the token features.

3.2 Model Architecture

Figure 2 illustrates the simple but expressive model architecture of SMART. The model comprises an encoder for road map encoding and a motion decoder that predicts a category distribution based on motion token embeddings.

RoadNet: road token encoder We employ multi-head self-attention (MHSA) to model the relationships among road tokens, after which the updated road token encodings will assist motion token decoding. For the i^{th} road token, we derive a query from its embedding r_i and let it attend to the neighboring tokens $r_j \in R_i$:

$$r_{i'} = \text{MHSA}\left(q(r_i),\ k(r_j, \text{RPE}_{ij}), v(r_j, \text{RPE}_{ij})\right), \quad j \in R_i \tag{1}$$

where R_i denotes the neighbor set of the road tokens. To incorporate spatial awareness for map encoding, we generate the j^{th} key/value vector from the concatenation of r_j and the relative positional embedding RPE_{ij}[8].

MotionNet: factorized agent motion decoder Prevailing methods for encoding agents prioritize capturing the temporal dynamics of an agent's movements, followed by the integration of agent-map and agent-agent interactions, as highlighted by [35]. Factorized attention effectively captures detailed agent-map interactions across temporal scales [28]. In our work, we leverage a factorized Transformer architecture with multi-head cross-attention (MHCA) to decode complex road-agent and agent-agent relationships along the time series. Akin to query-centric methodologies [52], we utilize relative positional embeddings to differentiate between agents' local coordinate frames, enabling symmetric encoding. Take the i^{th} agent at time step t as an example. Denoted as Eq.2a, given the query derived from the agent motion token's embedding e_i^t, we employ temporal attention by computing the key and value based on which are the i^{th} agent's token embeddings from time step $t-\tau$ to time step $t-1$ and the corresponding relative positional embeddings.

$$e_{i'} = \text{MHSA}\left(q(e_i^t), k(e_i^{t-\tau}, \text{RPE}_i^{t,t-\tau}), v(e_i^{t-\tau}, \text{RPE}_i^{t,t-\tau})\right), \quad 0 < \tau < t \tag{2a}$$

$$e_{i'} = \text{MHCA}\left(q(e_i^t), k(r_j, \text{RPE}_{ij}), v(r_j, \text{RPE}_{ij})\right), \quad j \in N_i \tag{2b}$$

$$e_{i'} = \text{MHSA}\left(q(e_i^t), k(e_j^t, \text{RPE}_{ij}^t), v(e_j^t, \text{RPE}_{ij}^t)\right), \quad j \in N_i \tag{2c}$$

Likewise, in Eq.2b and Eq.2c, the key and value for agent-map and agent-agent attention are derived from road token $r_j, j \in N_i$ and agents' motion token $e_j^t, j \in N_i$ in the neighborhood respectively, where the neighbor set N_i is determined by a distance threshold of 50 meters. We stack the temporal, the agent-agent, and the agent-map attention sequentially as one fusion block and repeat such blocks K times.

3.3 Spatial-temporal next token prediction

In the training stage, we train SMART to understand the temporal and spatial relationships in the traffic scene. This is achieved with two next token prediction tasks on RoadNet and MotionNet, the model is optimized with the summation of the two tasks' objectives.

Road vector next token prediction As shown in Figure 2(b), the road vector NTP task targets RoadNet to learn the spatial structure of road vector inputs. Unlike agent motions, road vectors form a graph rather than a sequence, making it challenging to apply next token prediction tasks directly. To address this issue, we extract the original topological information of roads and model the road vector tokens with sequential relationships based on their predecessor-successor connections. As depicted in Figure 2(b), in the pre-training NTP task, the subsequent road vector token is predicted using the preceding road token based on the road topology. This approach requires RoadNet to understand the connectivity and continuity among unordered road vectors. The loss function for a single tokenized road vector sequence is defined as:

$$\text{loss}(\gamma) = -\sum_{j=1}^{J}\sum_{i=1}^{V_r}(\mathrm{r}_i^{j+1} == \mathrm{r}_{i^{gt}}^{j+1}) \log(p_\gamma(\mathrm{r}_i^{j+1}|\mathrm{r}_i^{1:j})) \tag{3}$$

where $p_\gamma(\mathrm{r}_i^{j+1}|\mathrm{r}_i^{1:j})$ denotes the categorical distribution predicted by the RoadNet parameterized by γ, J represents a complete polyline that has not yet been split into road vector tokens, $\mathrm{r}^{1:j}$ Representing the road token embedding of the predecessor, and r_i^{j+1} is the next predicted road vector token. This loss function ensures that RoadNet learns to predict the correct next road vector token given the preceding tokens, thereby capturing the spatial continuity and connectivity within the road network.

Motion next token prediction Motion NTP task targets MotionNet to understand not only the temporal dependencies in agents' motions but also the spatial dependencies between agent-map and agent-agent. SMART is trained to minimize the cross entropy between the distribution of the ground truth token label and the predicted distribution. Formally, the loss function for a single tokenized motion sentence is given by:

$$loss(\theta) = -\sum_{t=1}^{T}\sum_{i=1}^{V_a}(a_i^{t+1} == a_{i^{gt}}^{t+1}) log(p_\theta(a_i^{t+1}|e_i^{1:t}, r_j)) \tag{4}$$

where $p_\theta(a_i^{t+1}|e_i^{1:t}, m_j)$ denotes the categorical distribution predicted by the model parameterized by θ, $e_{1:t}$ is the historical tokenized agent motion embeddings, $a_i^{t+1} \in A$ is the next predicted agent motion token and r_j is the tokenized nearby road vector series. Note that SMART performs autoregression via classification[41]. Opting for a categorical output distribution offers a key advantage: it imposes no restrictions on the structure of the output distribution, allowing the model to learn arbitrary distributions, including multimodal ones. This flexibility is especially valuable for a fundamental model, as agent and road tokens from diverse datasets may follow distinct output distribution patterns.

4 Experiments

To validate the generalizability and scalability of the SMART model, we conducted extensive experiments and trained models across various scales. On the official WOMD Sim Agents Challenge (WOSAC), we employed the SMART 7 Million parameters (7M) model, which was exclusively trained on the WOMD dataset. Concurrently, the SMART 7M model was also utilized for generalization experiments and ablation studies. In the scale law experiments, we integrated additional datasets

Table 1: Comparison with state-of-the-art models on WOMD 2023 Sim Agents benchmark

Method	Realism Meta metric↑	Kinematic metrics↑	Interactive metrics↑	Map-based metrics↑	minADE ↓
SMART 7M	**0.6587**	0.4190	**0.8014**	**0.8523**	1.7453
Trajeglish[30]	0.6451	0.4166	0.7845	0.8216	1.5712
MVTE[46]	0.6448	0.4202	0.7666	0.8387	1.6770
VPD-PRIOR	0.6315	**0.4261**	0.7233	0.8330	1.3400
QCNeXt[53]	0.4538	0.3109	0.5654	0.5051	**1.0830**
MultiPath[45]	0.4766	0.1792	0.6380	0.6866	2.0517

and trained on models of multiple scales. For all experiments, the testing datasets employed the split validation dataset from WOMD. Detailed hyperparameters for the SMART architecture can be found in Section A.1. In the following sections, Section 4.1 presents the results of rollouts generated by SMART on the WOSAC benchmark [26]. Evaluations of SMART's generalizability and scalability are detailed in Sections 4.2 and 4.3, respectively. Finally, an ablation analysis of our design methods is conducted in Section 4.4.

4.1 Comparison for motion generation task

Performance comparison We compare proposed SMART with existing motion generation approaches including diffusion models[15], continuous distribution regression models [46, 36], and next token autoregressive model[30]. Because the Sim Agents challenge metrics were changed twice, to compare it more broadly with the previous methodology, we test the performance of our model using both the WOMD Sim Agents 2023 and 2024 Benchmark[26]. As shown in Table 1 and Table 2, SMART achieves not only the best Realism Meta metric but also a high prediction precision. SMART's modeling approach for maps and motion enables it to learn the behavioral distribution within the data more effectively than prior work. Notably, SMART-zeroshot represents a model trained solely on the NuPlan dataset and directly inferred on the Waymo test set. As shown in Table 2, it achieves performance close to that of MVTE. For further detailed comparisons, please refer to A.2.

Table 2: Comparison with state-of-the-art models on WOMD 2024 Sim Agents benchmark

Method	Realism Meta metric↑	Kinematic metrics↑	Interactive metrics↑	Map-based metrics↑	minADE ↓
SMART 101M	**0.7614**	**0.4786**	**0.8066**	**0.8648**	**1.3728**
SMART 7M	0.7591	0.4759	0.8039	0.8632	1.4062
BehaviorGPT	0.7473	0.4333	0.7997	0.8593	1.4147
GUMP	0.7431	0.4780	0.7887	0.8359	1.6041
MVTE	0.7302	0.4503	0.7706	0.8381	1.6770
SMART-zeroshot	0.7210	0.4311	0.7806	0.8099	2.5703
VBD	0.7200	0.4169	0.7819	0.8137	1.4743
TrafficBOTv1.5	0.6988	0.4304	0.7114	0.8360	1.8825
congniBOTv1.5	0.6288	0.3293	0.7129	0.6918	-

Efficiency comparison SMART also demonstrates remarkable speed in multi-agent motion generation. Previous encoder-decoder models [34, 36] suffer from high computational costs, as the model requires multiple query embeddings in the decoder module to generate multi-modal motions. Benefiting from the advantages of the decoder-only transformer architecture, SMART only needs to compute the next token for the upcoming frame at the current moment during inference, without the need to re-encode historical motion tokens. By reusing the token embeddings computed in previous observation time horizons, the complexity of the agent motion decoder is reduced to $O(N_A N_T) + O(N_A N_R) + O(N_A^2)$. In contrast, for encoder-decoder models like [24], besides the computational load of the encoder module, additional computations of $O(N_A^2 N_M) + O(N_A N_M N_R)$ are required for generating multi-modalities of trajectories, where N_M represents the number of modalities. The average single-step inference time of SMART is influenced by the number of map

tokens and agent motion tokens, fluctuating between 5 to 20 ms, and averaging under 10 ms. Thus, it significantly meets the current needs of interactive real-time online simulation in autonomous driving.

4.2 Generalization

Zero-shot generalization on different dataset Zero-shot generation is the ability of models to generate motions for time series from different datasets. In this work, we use the training data from NuPlan dataset to train SMART models and the test data from WOMD validation dataset. As shown in Table 3, SMART* still achieves good performance in the overall metrics. Due to significant differences in the accuracy of the calibrated ground truth values for agent position and heading between different datasets, there may be a larger gap in the agent kinematic metrics, resulting in lower scores. However, SMART* demonstrated excellent generalization in the metrics of agent interaction and drivable map. It is worth mentioning that the size of the two datasets does not differ greatly, so the SMART model can have good generalization ability based on a small number of data training.

Table 3: Zero-shot generalization on different datasets. SMART denotes a model trained on WOMD only. SMART* denotes a model trained on NuPlan dataset only. SMART** denotes a model after 1 epoch of finetuning with an initial learning rate of 0.0001 on WOMD based on SMART* model.

Method	Kinematic metrics↑	Interactive metrics↑	Map-based metrics↑	minADE ↓
SMART	0.4537	0.8034	0.8514	1.5127
SMART*	0.4161	0.7853	0.7970	2.3041
SMART**	0.4310	0.8087	0.8559	1.5671

Zero-shot generalization on unseen scenarios Multiple map scenarios as shown in Figure 3 are present only in the WOMD but not in the NuPlan dataset. Without modifications to the network architecture or tuning parameters, SMART trained only on NuPlan has achieved decent results in these scenarios, substantiating the generalization ability of SMART.

4.3 Scalability

Prior research [22, 42] have established that scaling up large language models (LLMs) leads to a predictable decrease in test loss L. This trend correlates with parameter counts N, training tokens T, following a power-law:

$$log(L) = \beta log(X) + \alpha \tag{5}$$

where X can be any of N, T. The exponent α reflects the smoothness of power-law, and L denotes the reducible loss normalized by irreducible loss. The data sources for validating scaling laws are detailed in the A.3. Overall, we trained models across four sizes, ranging from 1M to 100M parameters, on a training set containing 2.2M scenarios (or 1B motion tokens under 0.5s agent motion tokenization).

Scaling laws with model parameters We investigate the test loss trend as the model size increases. We assessed the final test cross-entropy loss L on the validation set of 100,000 traffic scenarios. The results are plotted in Figure 4, where we observed a clear power-law scaling trend for Loss L as a function of model size N. The power-law scaling laws can be expressed as:

$$log(L) = -0.157log(X) + 1.52 \tag{6}$$

These results verify the strong scalability of SMART, providing valuable insights into how model performance scales with dataset size.

4.4 Ablation

In this study, we aim to verify the effectiveness of each component of SMART. Results are reported in Table 4. The initial model, denoted as $M1$, is constructed on the architecture depicted in Sec.3.2, employing solely agent tokenization. The introduction of the road vector tokenization in $M2$, which tokenized the road vector states into discrete tokens, results in marked improvements over $M1$ in

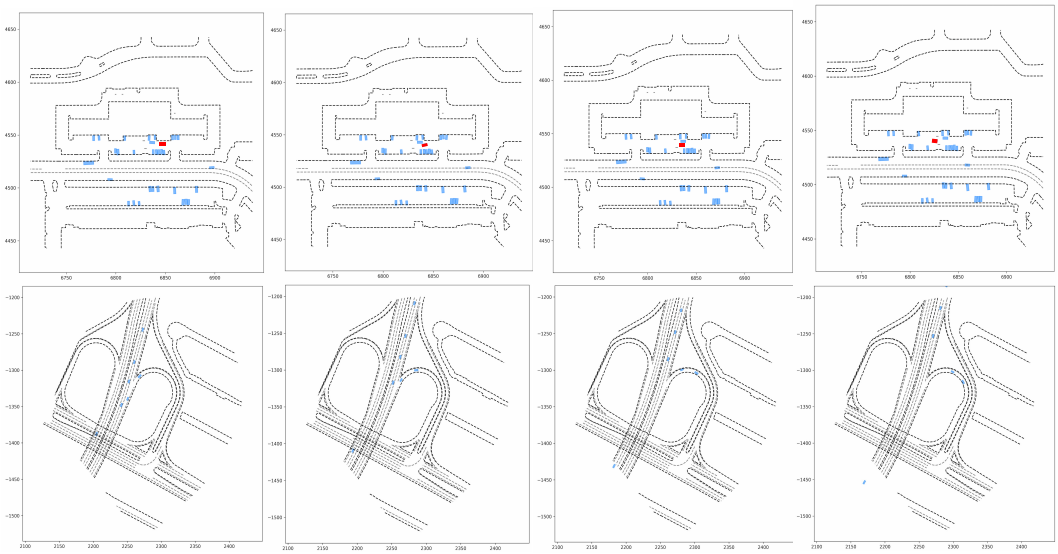
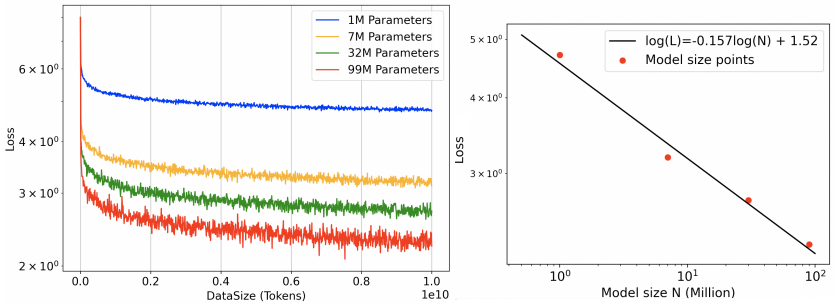


Figure 3: Qualitative results of closed-loop planning for two representative scenarios from the test set. Each scenario (every row) lasts 8 seconds and we take 4 snapshots with a 2-second interval. SMART controls all the agents in the scenario. The first row depicts a parking lot area. The red vehicle in the picture effectively completed a detour around a stationary vehicle ahead in the parking lot. The second row shows a scene of a large curvature U-turn in a ramp zone, where the traffic flow in the right lane of the ramp has completed the behavior of ramp exit under the control of SMART. It is recommended to refer to supplementary materials for more videos



(a) Loss reduction curves of models with different scales trained on 1Billion tokens

(b) Scaling laws with SMART transformer model size , with power-law fits (solid line) and equations (in legend).

Figure 4: Due to limitations in dataset size, we trained models at multiple scales ranging from 1M to 101M on a total of 1 billion tokens. (a) Training loss of different models (b) Axes are all on a logarithmic scale. The power-law scaling law can be expressed as a solid line. Exponents $\beta = -0.157$ suggest a smooth decline in test loss L when scaling up SMART models.

the generalization capability. Comparing models $M1$ and $M2$ reveals that when trained solely on the WOMD dataset, the tokenization of road vectors results in a certain reduction in overall metrics. We speculate that discretized map tokens may lose some fine-grained geometric information about roads. $M4$ incorporates noised agent motion tokenization, designed to address cumulative errors and distributional shifts during inference. This modification leads to enhancements in both the interaction metric and the map-based metric.

5 Conclusions

In this paper, we have introduced SMART, a novel paradigm for autonomous driving motion generation that leverages vectorized map and agent trajectory data, processed through a decoder-only transformer architecture in a GPT-style framework. We have observed that SMART emulates two

Table 4: Ablation study on each component of SMART. Experimental results are based on the **WOMD validation set**. "RVT" indicates road vector tokenization, "RVNTP" indicates road vector next token prediction, "NAT" indicates noised agent tokenization, "NRVT" indicates noised road vector tokenization

SMART					Train on WOMD			Train on NuPlan		
Model Number	RVT	NAT	NRVT	RVNTP	kinematics	interactive	map	kinematics	interactive	map
M_1					**0.459**	**0.827**	**0.857**	0.376	0.593	0.603
M_2	$\surd$				0.434	0.807	0.840	0.389	0.696	0.724
M_3	$\surd$	$\surd$			0.448	0.809	0.848	0.413	0.750	0.743
M_4	$\surd$	$\surd$	$\surd$		0.437	0.801	0.837	0.411	0.747	0.741
M_5	$\surd$	$\surd$		$\surd$	0.453	0.813	0.853	0.413	0.780	0.785
M_6	$\surd$	$\surd$	$\surd$	$\surd$	0.453	0.803	0.851	**0.416**	**0.785**	**0.797**

critical properties: scalability and zero-shot generalization, which are essential for advancing large models. We believe that our findings and the release of all codes will encourage further exploration and development of models for motion generation in the autonomous driving field, ultimately contributing to more reliable autonomous driving systems.

Limitations In this work, we primarily focus on the design of the learning paradigm and maintain a relatively simple design for the discrete token vocabulary. We believe that iterating SMART with an advanced tokenizer[25] or sampling technique can further improve the performance. Although we have collected training data from multiple datasets, we are still limited by the dataset size when validating the model's scalability, restricting us to models with a maximum scale of 100 million parameters. Given the focus of this work on generalization and scaling laws, a large number of hyperparameter ablation experiments remain to be verified, including the time granularity of agent motion tokens and the size of the token vocabulary. As a motion generation model, the ability of SMART to migrate to planning and prediction tasks still needs to be verified, and this is our top priority for future work.

Acknowledgement We thank the anonymous reviewers, area chairs, and program committee members for their valuable suggestions, which have greatly improved the quality of our work. We also appreciate the thoughtful discussions with Yue Gong and Shuxiang Lu. Authors Wei Wu, Xiaoxin, and Ziyan contributed equally to this work. Wei Wu led the project and provided funding support, while Xiaoxin and Ziyan focused on algorithm design, implementation, model training, and manuscript writing.

References

[1] Elmira Amirloo, Amir Rasouli, Peter Lakner, Mohsen Rohani, and Jun Luo. Latentformer: Multi-agent transformer-based interaction modeling and trajectory prediction. *arXiv preprint arXiv:2203.01880*, 2022.

[2] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

[3] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

[4] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

[5] Yongli Chen, Shen Li, Xiaolin Tang, Kai Yang, Dongpu Cao, and Xianke Lin. Interaction-aware decision making for autonomous vehicles. *IEEE Transactions on Transportation Electrification*, 2023.

[6] Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking imitation-based planner for autonomous driving. *arXiv preprint arXiv:2309.10443*, 2023.

[7] Alexander Cui, Sergio Casas, Abbas Sadat, Renjie Liao, and Raquel Urtasun. Lookout: Diverse multi-future prediction and planning for self-driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16107–16116, 2021.

[8] Alexander Cui, Sergio Casas, Kelvin Wong, Simon Suo, and Raquel Urtasun. Gorela: Go relative for viewpoint-invariant motion forecasting. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7801–7807. IEEE, 2023.

[9] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019.

[10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[11] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, 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*, pages 9710–9719, 2021.

[12] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

[13] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021.

[14] Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. *Advances in Neural Information Processing Systems*, 36, 2024.

[15] Zhiming Guo, Xing Gao, Jianlan Zhou, Xinyu Cai, and Botian Shi. Scenedm: Scene-level multi-agent trajectory generation with consistent diffusion models. *arXiv preprint arXiv:2311.15736*, 2023.

[16] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[17] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023.

[18] Zhiyu Huang, Peter Karkus, Boris Ivanovic, Yuxiao Chen, Marco Pavone, and Chen Lv. Dtpp: Differentiable joint conditional prediction and cost evaluation for tree policy planning in autonomous driving. *arXiv preprint arXiv:2310.05885*, 2023.

[19] Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. *arXiv preprint arXiv:2303.05760*, 2023.

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

[21] Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning*, pages 10280–10297. PMLR, 2022.

[22] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[24] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7144–7153, 2019.

[25] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

[26] Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, et al. The waymo open sim agents challenge. *Advances in Neural Information Processing Systems*, 36, 2024.

[27] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2980–2987. IEEE, 2023.

[28] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021.

[29] Bernardo Pérez Orozco and Stephen J Roberts. Zero-shot and few-shot time series forecasting with ordinal regression recurrent neural networks. *arXiv preprint arXiv:2003.12162*, 2020.

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

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[32] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models, 2023.

[33] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

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

[35] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. *Advances in Neural Information Processing Systems*, 35:6531–6543, 2022.

[36] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *arXiv preprint arXiv:2306.17770*, 2023.

[37] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023.

[38] Haoran Song, Wenchao Ding, Yuxuan Chen, Shaojie Shen, Michael Yu Wang, and Qifeng Chen. Pip: Planning-informed trajectory prediction for autonomous driving. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 598–614. Springer, 2020.

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

[40] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Chenxu Hu, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

[41] Luís Torgo and João Gama. Regression by classification. In *Advances in Artificial Intelligence: 13th Brazilian Symposium on Artificial Intelligence, SBIA'96 Curitiba, Brazil, October 23–25, 1996 Proceedings 13*, pages 51–60. Springer, 1996.

[42] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[43] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[44] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[45] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7814–7821. IEEE, 2022.

[46] Yu Wang, Tiebiao Zhao, and Fan Yi. Multiverse transformer: 1st place solution for waymo open sim agents challenge 2023. *arXiv preprint arXiv:2306.11868*, 2023.

[47] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.

[48] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

[49] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation, 2024.

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

[51] Jinyun Zhou, Rui Wang, Xu Liu, Yifei Jiang, Shu Jiang, Jiaming Tao, Jinghao Miao, and Shiyu Song. Exploring imitation learning for autonomous driving with feedback synthesizer and differentiable rasterization. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1450–1457, 2021.

[52] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17863–17873, 2023.

[53] Zikang Zhou, Zihao Wen, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Qcnext: A next-generation framework for joint multi-agent trajectory prediction. *arXiv preprint arXiv:2306.10508*, 2023.

A Appendix

A.1 Implementation and Simulation Inference

Architecture details Table 5 summarizes the hyperparameters of the different models used in our implementation. We train a single model to generate the future motion of all three categories (i.e., Vehicle, Pedestrian, Cyclist), with each category having its own motion token vocabulary. The input road token feature contains three types of information: the position of each road token point, the road token direction at each point, and the type of each road token. For the prediction head in each decoder layer, we use a three-layer MLP, and the model weights are not shared across different decoder layers.

Table 5: Hyperparameters of different SMART models

Modules	Hyperparameters	Values			
		SMART 1M	SMART 7M	SMART 26M	SMART 101M
RoadNet	Number of self attention layers	1	3	1	3
	Road token embeddings	64	128	256	512
	Size of road token vocabulary	1024	1024	1024	1024
	Road token attention radius	10	10	10	10
MotionNet	Number of temporal attention layers	1	6	6	6
	Number of agent-agent attention layers	2	6	6	6
	Number of map-agent attention layers	2	6	6	6
	Number of attention head	8	8	8	8
	Dimension of attention head	8	16	32	64
	Feature dimension of Agent token embeddings	64	128	256	512
	Size of motion token vocabulary	512	1024	1024	2048
SMART	Total parameters	1.0M	7.2M	26.9M	101.0M

Training details The simulation model is trained end-to-end for all three agent types using the AdamW optimizer [23]. Both the dropout rate and the weight decay rate are set to 0.1. The learning rate is decayed from 0.0002 to 0 using a cosine annealing scheduler. Training includes all vehicles within a scene. The batch size is set to 4, with a maximum GPU memory usage of 30GB.

Inference for WOSAC The test set comprises 44,920 scenes, and each scene requires running the model inference $32 \times T$ times to generate the 32 simulations for a group of agents. During model inference, each simulation step produces the classified distribution of next tokens. There are two options for next token sampling: selecting the maximum-likelihood token or sampling among the top-k motion tokens with the redistributed probability. The first approach, while accurate, tends to yield less varied generations. Conversely, opting for the top-k motion tokens encourages diversity but can compound errors, generating trajectories with unrealistic kinematic motions or even drift. To balance realism and diversity, we use top-5 sampling at every step during the simulation. Videos of rollouts can be found on our project page or supplementary materials. For each scenario, the SMART model directly controls all agents within the scene. Due to the focus of this article on the generalization and scalability of the model, we have achieved good results in specific scene generation without extensive exploration of detailed tricks.

A.2 Detailed comparison in the WOSAC leaderboard

Table 6: Per-component metric results on the test split of WOMD, representing likelihoods. Due to updates in the calculation of WOSC evaluation metrics, methods are ranked by the composite metric on the 2023 Leaderboard for a broader comparison. For latest WOSC, please refer directly to the updated 2024 Leaderboard for detailed comparisons.

Method	KINEMATIC				INTERACTIVE			MAP		minADE↓
	LINEAL SPEED↑	LINEAR ACCEL↑	ANG. SPEED↑	ANG. ACCEL↑	DIST TO OBJ.↑	COLLISION ↑	TTC ↑	DIST TO ROAD↑	OFF ROAD↑	
WAYFORMER	0.202	0.144	0.248	0.312	0.192	0.449	0.766	0.379	0.305	6.823
SBTA-ADIA	0.317	0.174	0.478	0.463	0.265	0.337	0.770	0.557	0.483	3.611
CAD	0.346	0.252	0.432	0.311	0.33	0.311	0.789	0.637	0.539	2.314
JOINT-MULTIPATH++	0.431	0.230	0.019	0.035	0.349	0.485	0.811	0.637	0.613	2.051
MTR+++	0.411	0.106	0.483	0.436	0.345	0.414	0.796	0.654	0.577	1.681
QCNeXt	**0.477**	0.242	0.325	0.198	0.375	0.324	0.756	0.609	0.360	**1.083**
MVTE	0.442	0.221	0.535	0.481	0.382	0.450	0.832	0.664	0.640	1.677
Trajeglish	0.450	0.192	**0.538**	0.485	**0.387**	0.922	**0.836**	**0.659**	0.886	1.571
SMART 7M	0.363	**0.296**	0.423	**0.564**	0.376	**0.963**	0.832	**0.659**	**0.936**	1.749

The Waymo Open Sim Agents Challenge (WOSAC) is a significant initiative aimed at advancing the development and evaluation of simulation agents for autonomous vehicles. This challenge leverages the Waymo Open Motion Dataset (WOMD) to provide high-fidelity object behaviors and shapes produced by a state-of-the-art offboard perception system. Participants are required to simulate scenarios involving up to 128 agents, focusing on closed-loop evaluation to ensure realism in agent behaviors and interactions. The evaluation framework employs various metrics, including kinematic features, interaction-based features, and map-based features, to assess the performance of simulation agents in generating realistic and diverse behaviors that match real-world driving data. WOSAC computes three metrics over nine measurements: kinematic metrics (linear speed, linear acceleration, angular speed, angular acceleration magnitude), object interaction metrics (distance to nearest object, collisions, time-to-collision), and map-based metrics (distance to road edge, road departures).

In the benchmark comparisons presented in Table 6, the SMART 7M method, developed by our team, demonstrates superior performance across multiple metrics, particularly excelling in interactive and safety-related indicators. Notably, SMART 7M achieved the highest scores in angular acceleration, distance to nearest object, collision avoidance, and off-road metrics, underscoring its effectiveness in complex driving scenarios. These results highlight the robustness of SMART 7M in ensuring safety and reliability, indicating its advanced capability in managing dynamic and potentially hazardous traffic conditions more effectively than other evaluated methods. This performance also suggests the potential of the SMART model to be applied to planning tasks.

A.3 Additional ablation studies

Comparison of the scalability and generalization of different architectures This section presents experiments comparing the architecture proposed in this paper with the MVTE model. The MVTE model[3], derived from MTR, represents continuous distribution regression models. The experimental

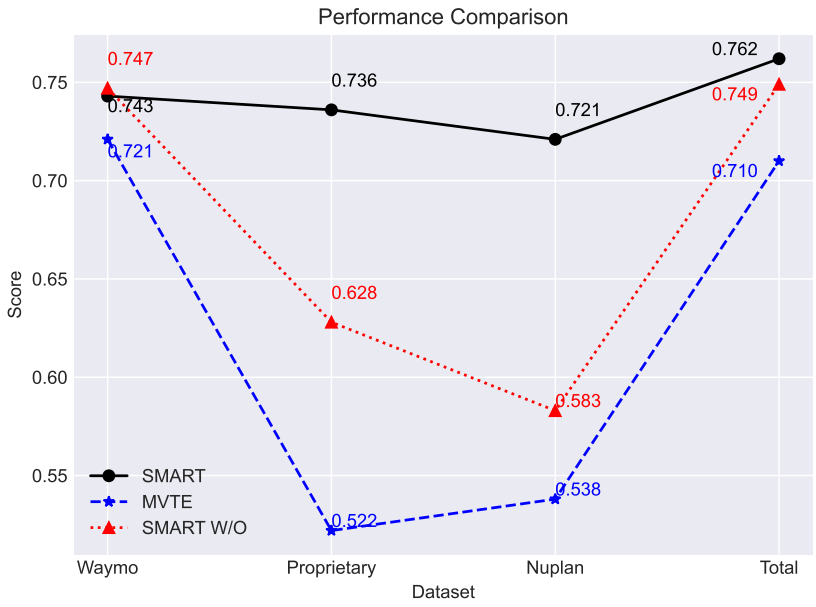


Figure 5: SMART w/o refers to SMART model without the road vector tokenization and noise tricks proposed in this paper. To ensure the fairness of the experiments, all model parameters were adjusted to the 90-100M. Models were trained on various datasets and validated only on the WOMD validation dataset

results shown in Figure 5 indicate that distribution regression-based models have poor generalization capabilities across different datasets. Models trained with incremental data from other datasets performed worse overall than models trained solely on the WOMD. An interesting phenomenon is

[3]Since MVTE does not have open-source code, we reproduced the results by relying on the MTR model

that although our private dataset contains more data than the NuPlan dataset, the performance of MVTE trained on it was inferior to that on the NuPlan dataset. This suggests that the distribution regression-based paradigm is likely to cause the model to overfit to a dataset. From the SMART w/o results, it can be seen that the model's generalization performance is poor, but the effect of incremental data can improve the performance compared to a single training dataset. Based on the above experiments, it can be inferred that discrete tokenization is a very effective way to eliminate the dataset gap. Moreover, autoregressive models based on cross-entropy classification loss are key to the scalability of trajectory generation models, which aligns with why large language models (LLMs) have significant scaling capabilities.

Comparison of different tokenizer In Trajeglish[30], a detailed comparison of various discretized tokenizers is conducted. As introduced in Sec. 3.1 of this paper, we ultimately adopted the k-disks approach for token vocabulary construction. Prior to our work, no studies had attempted to construct a vocabulary for agent and road motion tokens using latent tokenizer methods [44]. Therefore, we drew on the visual domain's VQ-VAE approach to perform latent autoencoding of motion tokens and provided a comparison of this tokenizer with the method selected in this paper.

Table 7: Comparison of different tokenizer. Experimental results are based on the SMART 7M

	Train on WOMD			Train on NuPlan		
Tokenizer	Kinematics	Interactive	Map	Kinematics	Interactive	Map
VQ-VAE	0.461	0.810	0.853	0.376	0.687	0.703
K-disks	0.453	0.803	0.851	0.416	0.785	0.797

From the results in Table 7, it is evident that VQ-VAE performs better on a single dataset compared to k-disks. Specifically, both methods achieve similar results in interactive and map-based metrics, but VQ-VAE outperforms k-disks in kinematic metrics. The k-disks approach loses fine-grained trajectory information during discretization, whereas VQ-VAE better fits the true distribution of the dataset when reconstructing trajectories. However, when comparing the two methods' performance in zero-shot generalization, k-disks significantly outperform VQ-VAE. We speculate that during the training of the VQ-VAE tokenizer to construct motion and road token vocabularies, the tokenizer may have already memorized or overfitted to the training dataset. Therefore, to achieve better generalization performance using the VQ-VAE approach, it is essential to pre-train the VQ-VAE tokenizer on a large-scale dataset.

Comparison of SMART models with different scales For language models, large and diverse datasets are relatively easy to obtain. In contrast, the autonomous driving motion domain lacks a data source of comparable size and diversity. To validate scaling laws on a larger dataset, we integrated data from Waymo, Nuplan, and our proprietary dataset. We introduced our proprietary dataset solely for validating scaling laws. For the WOSC leaderboard evaluation, we exclusively used the Waymo dataset. For generalization and other ablation experiments, we utilized both Nuplan and Waymo open-source datasets to facilitate reproducibility of the experiments by providing access to widely available datasets. Table 8 below summarizes the scenario count, duration, and total motion token count for each dataset.

The results in Table 9 highlight the performance of SMART models with different parameter scales across various metrics. As the model scale increases from SMART 1M to SMART 101M, there is a significant improvement in both the interactive metrics and the map-based metrics. This indicates that larger models are better at capturing interactions and understanding map-based context, leading to enhanced performance in these areas. However, the kinematic metrics show minimal variation.

Table 8: Data sources

Dataset	Scene Count	Single Scenario Duration	Total Motion Token Count
Nuplan	30w	10s	0.13B
Waymo	48w	9s	0.18B
Proprietary	150w	11s	0.68B
Total	228w	-	1B

Table 9: Comparison of SMART models with different scales. Training time refers to the duration required for the model to converge using the entire dataset. Inference time refers to the time taken by the model to predict the next token for a single frame.

Method	Kinematic metrics↑	Interactive metrics↑	Map-based metrics↑	Training time	Average inference time
SMART 1M	0.423	0.782	0.835	8hours	10.30ms
SMART 7M	0.436	0.809	0.852	23hours	17.21ms
SMART 26M	0.442	0.817	0.864	3days	25.94ms
SMART 101M	**0.457**	**0.819**	**0.872**	1week	46.58ms

Additionally, the training time and average inference time increase substantially with larger models, reflecting the trade-off between model performance and computational cost. Validation is conducted every 50,000 train steps. The model is considered to have converged if there is no significant loss reduction or metric improvement after five consecutive validations. The training and inference time is measured on 32 NVIDIA TESLA V100 GPUs.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the contributions of the SMART framework, emphasizing its novel autonomous driving motion generation paradigm, state-of-the-art performance, scalability, and zero-shot generalization capabilities. These claims are substantiated in the subsequent sections of the paper. Specifically, Sections 3 and 4 provide a detailed explanation of the methodology and experimental results that support these claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In section 5, we discussed the potential limitations of this study and possible directions for future iterations of the model.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper does not propose new theorems; In Section4.3, it references the scale power-law and demonstrates through experiments that the proposed model conforms to this scale law.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper provides detailed instructions for replicating the experimental results. This includes methodology 3, data preprocessing and tokenization details3.1, model architecture3.2 and parameter settings3.3, training procedures4 and A.2, and any datasets used A.3. The training code has been made open-source to ensure the reproducibility of all results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and dataset used in this paper are all in the code base. This paper introduces private datasets in the experimental verification of scale laws, which are not open-source.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In section 4, this paper briefly introduces the training and validation test design, and in A.2, all modules and training hyperparameters are introduced in detail.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: In Section 4, we outline the composition of the training and testing datasets used. In Appendix A.1, we describe the sampling method employed for scenario generation. Due to WOSAC's evaluation requirement for the model to randomly generate multiple scenarios, there may be slight deviations in the final metrics compared to those reported in the paper. However, given the large scale of the evaluation dataset, consisting of over 20,000 samples, such error is negligible. Therefore, the experimental results in this paper can be accurately reproduced using the provided open-source code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In appendix A.3, we have recorded the experimental testing environment, training time, and model inference time. All models in this paper were trained using 32 V100 GPUs. The training process requires a GPU memory of at least 25GB, while model inference typically requires only 10GB of memory.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In the introduction, we summarize this paper's contributions to autonomous driving applications, particularly highlighting the model's application in multi-agent simulation for autonomous driving. In the conclusion, we provide an outlook on the model's application in planning. We have released the largest model and codes to promote the exploration of autoregressive models for drive motion generation.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our proposed generative model can generate a large number of autonomous driving simulation scenarios or synthesis test data. Although safety is a critical aspect of autonomous driving, the current primary application is in simulation. Therefore, the model does not pose significant risks or potential for misuse at this stage.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The training and evaluation datasets used in this study are cited within this paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our study involves simulation-based autonomous driving scenarios without the direct involvement of human subjects or crowdsourcing, negating the need for IRB approval

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.