S4-Driver: Scalable Self-Supervised Driving Multimodal Large Language Model with Spatio-Temporal Visual Representation

Supplementary Material

The supplementary material is organized as follows. Sec. 7 provides implementation details of our proposed methods including the concrete input prompt and target output. Sec. 8 provides the model performance for each ego-vehicle behavior separately. Sec. 9 gives additional visualization of S4-Driver motion planning in diverse scenarios. Finally, Sec. 10 conducts extra ablation studies on WOMD-Planning-ADE benchmark to justify the design of S4-Driver.

7. Implementation Details

WOMD-Planning-ADE benchmark. This benchmark contains 487k scenarios for model training and 44k for validation, which are divided from 103k sequences of $20s$ length. Each scenario contains $1s$ history and $8s$ future. We only consider the future $5s$ for the open-loop evaluation of motion planning since motion planning in real task is conducted in an iterative manner. At each time stamp, the dataset provides multi-view images captured by 8 cameras (namely *front left, front, front right, side left, side right, rear left, rear, rear right*). The dataset is also equipped with fine-grained labels for all the other agents and the roadgraph although they are not used in our self-supervised motion planning algorithm.

Ego-vehicle coordinate system. We conduct the motion planning task in the ego-vehicle coordinate system. At current timestamp, the ego-vehicle position is defined as the origin. The $x-$axis is oriented towards the current heading angle of the ego-vehicle. The $y-$axis is oriented towards the left of the ego-vehicle. The $z-$axis is oriented upwards. Although it is a 3D coordinate system, we only consider the $xy-$plane for the description of the positions, velocities, and accelerations of the vehicles in the motion planning task.

Architecture and hyperparameters. Unless otherwise specified, we build up the S4-Driver framework based on multimodal large-language model PaLI3-5B [10]. The 3D scene representation (Sec. 3.3) in default covers the range of $(-30m, 80m) \times (-30m, 30m) \times (-2m, 8m)$ in the ego-vehicle coordinate along x, y, z axes separately. The position embedding in Eq. 3 and Eq. 7 are implemented with a two-layer MLP attached to the Fourier embedding of (x, y, z). To generate the gate values, we reduce the dimension of visual features from $C = 1536$ to $C' = 96$ in Eq. 5. The volume resolution is $1m \times 1m \times 2m$. For each scene, we select $M = 6000$ sparse volumes based on the gate values. For attention bias (Sec. 3.3.3), there are independent intra-modality biases for visual tokens and text tokens separately. We leave the details of bias function $b(\cdot)$ in the extra ablation study (Sec. 10). For each attention head at each self-attention layer, we also include learnable scalar inter-modality biases for the attention values between text tokens and visual tokens. In Sec. 3.4, for temporal fusion, apart from the current frame, we also take camera images from $T = 1$ historical frame at the timestamp $-0.5s$ as model input. For multi-decoding aggregation (Sec. 3.5), the model outputs $K = 16$ trajectories in parallel through Top-p sampling (nucleus sampling) [21] ($p = 0.9$), which are aggregated as one unique planning result. For both datasets, the input images are resized to 448×448 as model inputs.

Model initialization. To ensure the alignment within the pretrained MLLM, we apply the following special rules to relieve the disturbance of injected modules to the pretrained weights in the early finetuning stage. The last FC-layer of MLP for the position embeddings in Eq. 3 and Eq. 7 are zero initialized. The learnable vacant feature $\mathbf{f}_{vac}$ for sparse volume representation is also initialized as zeros (Eq. 8). We also set the initial bias for each bin in the bias function $b(\cdot)$ (Eq. 9) as zeros (details in Sec. 10). Finally, the temporal fusion (Sec. 3.4), we maintain the channel-wise semantics of visual features by initializing the weight matrix of the FC layer as $\mathbf{W} = \left[I \in \mathbb{R}^{C \times C}; \mathbf{0}^{C \times (T-1) \cdot C}\right]$ and the bias as zero. Based on the above rules, the inserted modules would not significantly change the channel-wise semantics of the visual features from original pretrained perspective view features, which allows for a stable and efficient finetuning process.

Input prompt and target output. The input prompt is composed of the high-level behavior command and ego-vehicle historical states. We represent all historical states in the language space. The position, velocity, and acceleration at each time stamp are represented with two floating numbers (two decimal places) separately for the $x-$component and $y-$component in the ego-vehicle coordinate. The target output is composed of meta-decision (Sec. 3.2 only in S4-Driver) and future waypoints. The future waypoints are also represented in the language space with two floating numbers for each time stamp. The historical and future states are sampled at a frequency of $5Hz$ on WOMD-Planning-ADE

Input Prompt

Assume I am at the coordinate 0, 0.

high-level behavior command

The high-level behavior attention is: go straight forward.

historical ego-vehicle states

The past trajectory under vehicle coordinate is: -17.77, 0.03 and -14.19, 0.02 and -10.63, 0.02 and -7.07, 0.01 and -3.53, 0.01. The past ego velocity under vehicle coordinate is: 17.92, -0.04 and 17.81, -0.02 and 17.79, -0.03 and 17.72, -0.03 and 17.66, -0.05. The past ego acceleration under vehicle coordinate is: -0.24, 0.10 and -0.49, 0.14 and -0.02, -0.13 and -0.45, -0.06 and -0.31, -0.05.

What is my future trajectory in next 5 seconds under vehicle coordinate?

Target Output

meta-decision

The ego vehicle is going to keep speed then decelerate.

future waypoints

The future trajectory under vehicle coordinate is: 3.53, -0.02 and 7.04, -0.03 and 10.56, -0.05 and 14.07, -0.07 and 17.60, -0.09 and 21.13, -0.11 and 24.65, -0.13 and 28.17, -0.15 and 31.69, -0.18 and 35.19, -0.20 and 38.68, -0.23 and 42.15, -0.26 and 45.61, -0.29 and 49.06, -0.33 and 52.48, -0.36 and 55.90, -0.39 and 59.28, -0.42 and 62.65, -0.44 and 65.99, -0.47 and 69.31, -0.49 and 72.60, -0.51 and 75.86, -0.52 and 79.07, -0.54 and 82.25, -0.56 and 85.39, -0.58.

Figure 9. Example input prompt and target output on WOMD-Planning-ADE.

where we consider the historical states of past $1s$ and future waypoints of future $5s$. On nuScenes dataset, we consider the historical states of past $1s$ and future waypoints of future $3s$ at a frequency of $2Hz$. An example of the input and output on WOMD-Planning-ADE is visualized in Fig. 9.

Heuristics for high-level behaviors. We consider the seven behaviors for the ego-vehicle on WOMD-Planning-ADE benchmark, which are determined based on the ego-vehicle ground-truth future trajectories following heuristic rules.

1. **Stop**: The ego-vehicle movement is $< 5m$ and the maximal speed is $< 2m/s$.
2. **Do left turn**: The ego-vehicle does not stop. The final heading angle is $> 30°$. The final position is $\geq -5m$ along $x-$axis.
3. **Do left U-turn**: The ego-vehicle does not stop. The final heading angle is $> 30°$. The final displacement is $< -5m$ along $x-$axis.
4. **Do right turn**: The ego-vehicle does not stop. The final heading angle is $< -30°$.
5. **Go straight left**: The ego-vehicle does not stop. The final heading angle is in the range $[-30°, 30°]$. The final displacement is $> 5m$ along $y-$axis.
6. **Go straight right**: The ego-vehicle does not stop. The final heading angle is in the range $[-30°, 30°]$. The final displacement is $< -5m$ along $y-$axis.
7. **Go straight forward**: The ego-vehicle does not stop. The final heading angle is in the range $[-30°, 30°]$. The final displacement along $y-$axis is in the range $[-5m, 5m]$.

For the calculation of *bADE* metric in the model evaluation, we follow the above rules to determine the behavior based on the ground-truth future trajectories of $8s$ although our planning horizon is only $5s$. For the high-level behavior command in the model inputs, "stop" is excluded to avoid future information leakage. To determine the high-level behavior command input, we start from the ground-truth $8s$ future trajectories. If none of behaviors 2-7 is satisfied, we would prolong the future horizon by $2s$ until at least one of behaviors 2-7 is satisfied. If the end of the collected trajectory sequence is reached, we would just consider that scenario as "go straight forward".

Heuristics for meta-decisions. In Sec. 3.2, we design a meta-decision strategy as a preliminary prediction before the waypoints. The meta-decision of ego-vehicle includes four categories determined with following heuristic rules based on the ground-truth future information.

1. **Keep stationary**: The maximal speed is $< 2m/s$ and the final displacement is $< 1.5m$.
2. **Keep speed**: The ego-vehicle does not keep stationary. The average acceleration is in the range of $[-0.5m/s^2, 0.5m/s^2]$.
3. **Accelerate**: The ego-vehicle does not keep stationary. The average acceleration is $> 0.5m/s^2$.
4. **Decelerate**: The ego-vehicle does not keep stationary. The average acceleration is $< -0.5m/s^2$.

The model should firstly predict the meta-decision and then auto-regressively output the future waypoints. On WOMD-Planning-ADE with $5s$ future prediction horizon, we divide the $5s$ into two stages of $2.5s$, where meta-decisions are predicted independently for each stage. On nuScenes dataset with $3s$ prediction horizon, we only consider one-stage meta-decision.

8. Behavior-wise Model Performance

In Tab. 7, we report the *ADE@5s* metric of S4-Driver for each ego-vehicle behavior on WOMD-Planning-ADE benchmark separately. Results show the superiority of S4-Driver especially in complicated scenarios like turnings,

Methods	ADE@5s for each behavior							ADE@5s	bADE@5s
	stop	straight forward	straight left	straight right	left turn	right turn	left U-turn		
Vanilla PaLI	**0.048**	0.960	1.252	1.297	1.566	1.323	1.039	0.798	1.069
S4-Driver (ours)	0.063	0.843	1.077	1.177	1.252	1.158	0.925	0.693	0.928
S4-Driver* (ours)	0.065	**0.806**	**0.957**	**1.074**	**1.124**	**1.027**	**0.756**	**0.655**	**0.830**
Models that take high-quality objects, tracks, and roadgraphs as inputs instead of using raw camera images.									
MotionLM	0.048	0.832	1.293	1.275	1.239	1.172	0.990	0.697	0.978

Table 7. Behavior-wise sliced metrics on WOMD-Planning-ADE benchmark. "*" denotes methods with internal data pretraining.

Camera configuration	ADE@5s	bADE@5s
front	0.765	1.036
front, front left/right	0.751	1.012
front, front left/right, side left/right	0.746	**0.981**
all eight surrounding cameras	**0.732**	0.985

Table 8. Ablation studies on camera configurations.

where a significant performance gain is witnessed from vanilla PaLI baseline to S4-Driver. Motion planning for these difficult behaviors requires a great understanding of the roadgraph and other agents from raw camera images, which reflects the strong reasoning ability of our proposed spatio-temporal visual representation.

9. Additional Qualitative Results

In Fig. 10, we visualize more planning results on WOMD-Planning-ADE. Examples cover different behaviors, speeds, lighting conditions, and weathers. Results show the robust performance of S4-Driver in all these diverse scenarios.

10. Additional Ablation Studies

In this part, we conduct several additional ablation studies to further justify the design of our S4-Driver including the camera configuration, relative attention bias, multi-decoding aggregation, and motion tokens.

Camera configuration. We apply different configurations of camera sensors in Tab. 8. Consistent with intuitions, the *front* camera is the most important, which can solely guarantee a reasonable planning performance. Adding other cameras continuously improves the model performance since they provide more and more complete information of the surrounding environment. The three cameras in the back can notably boost the planning model since they provide cues about other agents behind ego-vehicle. This information helps to determine the future ego-vehicle velocities and accelerations.

Relative attention bias. We consider two types of relative position bias function $b(\cdot)$ in Eq. 9.

1. **Linear bias:** The bias is linearly related to the relative distance between tokens. For the sparse volume visual tokens, $b(x, y, z) = b_x(\Delta x) + b_y(\Delta y) + b_z(\Delta z)$, we take the $x-$axis as an example,

$$b_x(\Delta x) = \tau_x \cdot |\Delta x|, \tag{14}$$

where τ_x is a learnable parameter separately for each attention head at each layer, and Δx is the relative position between sparse volume tokens along $x-$axis. Similarly, for 1D text tokens, the separate bias function is

$$b_p(\Delta p) = \tau_p \cdot |\Delta p| \tag{15}$$

where τ_p is a learnable parameter separately for each attention head in each layer, and Δp is the relative position between text tokens.

2. **Bin-wise bias (our choice):** The relative positions $\Delta x, \Delta y, \Delta z$ (visual tokens) and Δp (text tokens) for each axis are divided into 32 bins independently. For each axis, 16 bins cover the range $[-8(m), 8(m)]$ linearly with an interval $1(m)$[2]. The other 16 bins symmetrically cover the range $(-128(m), -8(m))$ and $(8(m), 128(m))$ in log scale, where the relative distances are truncated to at most $128(m)$. In this case, if we take $b_x(\cdot)$ as an example, the relative bias function is written as

$$b_x(\Delta x) = m_x(\text{bin}(\Delta x)) \tag{16}$$

where $m_x(\cdot)$ maps each bin to a learnable bias value independently for each attention head at each self-attention layer.

Tab. 9 justifies the design of bin-wise bias, which brings greatly better performance. We think it is important to distinguish the two directions along each axis (*e.g.* front or back) in the motion planning task. Besides, local feature aggregation is more sensitive to close neighbors at each locality, so the bin-wise bias function has finer grains in close distance compared to the linear bias function.

Multi-decoding aggregation. We dig into the multi-decoding strategy which can bring notable performance

[2]The unit (m) is only for visual tokens ($\Delta x, \Delta y, \Delta z$) throughout this paragraph.

Figure 10. Additional qualitative results of motion planning. We show the *front left, front, front right* cameras for each case.

Attention bias	ADE@5s	bADE@5s
no bias	0.750	1.005
linear bias function	0.770	1.082
bin-wise bias function	**0.732**	**0.985**

Table 9. Ablation studies on attention bias.

gain. The motivation is that the MLLM is prone to assigning high confidence scores to simple future behaviors such as *stop*. To this end, we encourage the model to output multiple future trajectories. Their aggregation serves as the final planning result, which can counteract the model's high confidence in simple behaviors. In Tab. 10, we find that 1) The aggregation of more decoded trajectories leads to better performance. 2) Nucleus sampling can outperform beam search since it can generate more diverse outputs. 3) Weighted average is inferior to mean average since the model is prone to assigning high confidence to simple degenerated behaviors.

Motion tokens. Several prior works [46, 49] benefit from specialized trajectory tokenization modules, which converts motion trajectories into extra discrete tokens added to the vocabulary of language models. However, in

Sample number	Sample strategy	Aggregation	ADE@5s	bADE@5s
1	greedy sampling	-	0.728	0.986
4	beam search	average	0.739	0.997
4	nucleus sampling	average	0.709	0.941
4	nucleus sampling	weighted average	0.747	1.005
16	nucleus sampling	average	**0.693**	**0.928**

Table 10. Ablation studies on multi-decoding aggregation.

Trajectory representation	ADE@5s	bADE@5s
motion tokens	0.779	1.061
floating numbers	**0.750**	**1.005**

Table 11. Ablation studies on motion trajectory tokenization.

Tab. 11, the trajectory tokenization strategy similar with MotionLM [46] hurts the performance of our S4-Driver in comparison with naïve floating number waypoints representation (Fig. 9). We also witness a much slower convergence speed with this extra trajectory tokenization. Since the MLLM is already pretrained on large-scale data, additionally injected trajectory tokens may not align with the pretrained model. In contrast, the floating number representation can align the historical and future states in the language space to exploit the large-scale MLLM pretraining with lower requirement for finetuning.