

Supplementary Materials

A Dataset Construction

A.1 Examples of the QAs

In Fig. 8, we illustrate examples of QAs used in training TOKEN.

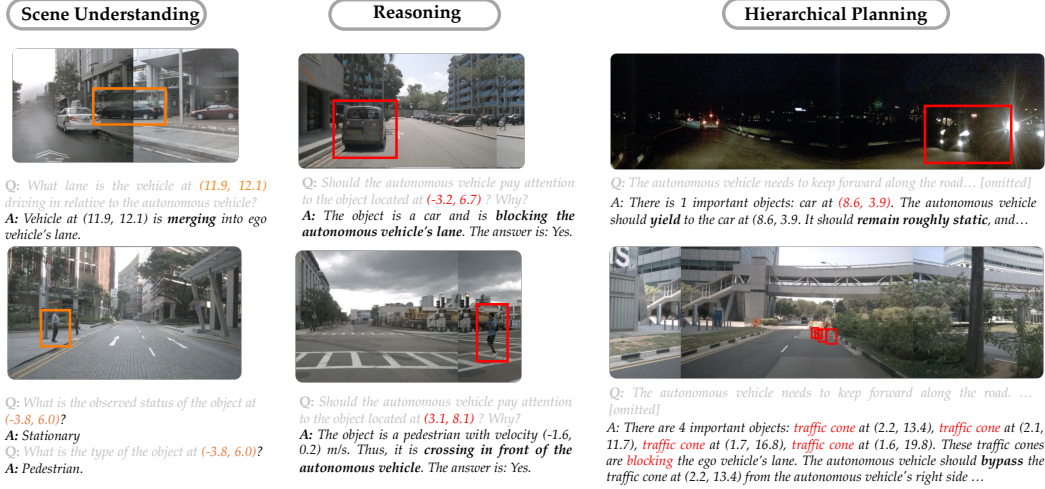


Figure 8: Examples of perception, reasoning, and planning QAs.

A.2 Road-Level Navigation Signal

Previous works on VLM/LLM for autonomous vehicle planning often prompt the model with a high-level command based on the relative position of the ground-truth ego trajectory, including "keep forward" and "turn right/left." However, these high-level commands are not only unrealistic but also simplify the planning problem by removing the need for behavior planning. Therefore, we re-labeled the NuScenes dataset to use road-level navigation signals as high-level commands, including "keep forward along the current road," "prepare to turn right/left at the next intersection," "turn right/left at the intersection," "left/right U-turn," and "left/right 3-point turn."

A.3 Interaction Mode Labeling

We use a combination of heuristics and manual labeling to annotate the interactions between the ego vehicle and the other traffic agents. We first use two types of categorical modes to describe the lane-relationship between a traffic agent and the ego vehicle (*agent-ego lane mode*) and the relative motion between a traffic participant and the ego vehicle (*homotopy*)[29]. Agent-ego lane mode at a time step t encodes the topology relationship between the ego's current lane and the traffic agent's lane, including: *LEFT*, *RIGHT*, *AHEAD*, *BEHIND*, and *NOTON*, where *NOTON* describes that the traffic agent is not on any derivable lanes in the scene (e.g., a parked vehicle in a parking lot). To compute the agent-ego lane mode for each traffic agent, we follow [29] to first identify the lane on which each agent is located and then leverage the lane topology map to annotate the agent-ego lane mode. We project the agent's center to the lane polyline and use its relative position in the local Frenet frame to determine its lane association. Homotopies describe the relative motion between a pair of agents shown in the video, including: *[S, CW, CCW]* (*static, clockwise, counterclockwise*). Combining agent-ego lane mode, homotopy, agent ground truth state information, and scene context information (e.g., ego is located near an intersection) together, we can leverage heuristics to annotate the interaction. For example, within a 3-second horizon, a static object's agent-ego lane mode changes from *AHEAD*, to *LEFT*, to *BEHIND*, while the ego vehicle performs *RIGHT-LANE-*

CHANGE, *KEEP-LANE*, then *LEFT-LANE-CHANGE*, indicating the ego vehicle overtakes that object from the ego vehicle’s left side. Finally, we use human labelers to verify and correct interaction labels in the following categories: 1) bypass blocking traffic cones to navigate around a construction zone; 2) yield to pedestrians; 3) yield to vehicles; 4) overtake traffic agents via straddling the lane dividers; 5) overtake traffic agents via lane-change.

B Evaluation Protocol

In this section, we provide a detailed description about our evaluation protocol. In Section 5.1 of the main text, we introduce the three variants of trajectory L2 error (the overall, turning, and progress errors) and the collision rate used to evaluate the predicted motion plans. As noted in [2], different evaluation protocols used to compute these metrics can lead to significant metric variations. We use the same evaluation protocol as described in [2] with one exception: we exclude samples where any future motion is missing near the end of a sequence (the frame masking strategy described in [2]). Including these partially invalid samples would significantly lower the L2 errors, as the L2 errors of these invalid frames are set to zero.

C Additional Result: On the Value of Object-Centric Tokenization

In Tab. 5, we present the full quantitative evaluation of each model’s performance in the planning task. We observe that TOKEN significantly outperforms all baselines across all planning metrics.

Method	Traj L2 (m) ↓					Heading L2 (rad) ↓					Lon. weighted traj L2 (m) ↓					Collision (%) ↓
	1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	Ave _{all}
Video-LLaMA	0.27	1.72	6.34	3.01	2.39	0.06	0.14	0.20	0.13	0.13	0.56	3.36	9.20	4.36	3.52	2.64
VILA-1.5	0.28	1.56	4.41	2.09	1.66	0.05	0.11	0.19	0.12	0.10	0.29	1.92	6.47	2.89	2.24	1.98
BEV-TOKEN	0.39	1.01	2.02	1.14	0.96	0.03	0.05	0.06	0.05	0.05	0.75	1.79	3.55	2.03	1.71	0.39
TOKEN	0.26	0.70	1.46	0.81	0.68	0.02	0.04	0.06	0.04	0.03	0.50	1.32	2.72	1.51	1.26	0.15

Table 5: **Planning performance evaluation.** TOKEN significantly outperforms baseline VLMs due to its use of driving-task pre-trained features and object-centric tokenization.

D Long-tail Events Construction

We manually inspected the NuScenes dataset and identified the following long-tail scenarios for evaluation, each representing less than 1% of the training data: 1) executing 3-point turns; 2) resuming motion after a full stop; 3) overtaking parked cars through the oncoming lane; and 4) navigating around construction sites.

Executing 3-point turns, which has one scene (scene-0778, frame 6-30) in the evaluation split and 0 scenes in the training distribution. We extracted 25 key frames from the scene in which the ego vehicle is performing the 3-point U-turn for evaluation to remove the noise from other nominal behaviors in the scene.

Resuming motion after a full-stop, which includes 14 scenes in the training split and 8 scenes in the evaluation split. We extract key frames from each scene where the ground truth (GT) motion plan captures the acceleration behavior after the full stop. This results in 70 key frames in the training split (0.28% of the total training samples) and 40 key frames in the evaluation split. The scenes and key frames in the training split are: scene-0208 (frame 25-29), scene-1023 (frame 21–25), scene-0067 (frame 24-28), scene-0159 (frame 4-8), scene-0185 (frame 26-30), scene-0262 (frame 8-12), scene-0862 (frame 18-22), scene-0025 (frame 6-10) scene-0072 (frame 24-28), scene-0157 (frame 12-16), scene-0234 (frame 4-8), scene-0423 (frame 6-10), scene-0192 (frame 14-18), and scene-0657 (frame 12-16). The scenes and key frames in the evaluation split are: scene-0921 (frame 21-25), scene-0925 (frame 19-23), scene-0968 (frame 7-11), scene-0552 (13-17), scene-0917 (frame 24-28), scene-0221 (frame 11-15), scene-1064 (frame 21-25), and scene-0331 (frame 8-12).

Overtaking parked cars through the oncoming lane, which includes 14 scenes in the training split and 5 scenes in the evaluation split. We extracted key frames from each scene where the ground truth

motion plan captures the ego vehicle steering into the adjacent lane to overtake a blocking object and then returning to its original lane. This results in 248 key frames in the training split (0.9% of the total training samples) and 102 key frames in the evaluation split. The scenes and key frames in the training split are: scene-0001 (frame 12-39), scene-0011 (frame 1-39), scene-0023 (frame 1-8), scene-0034 (frame 23-39), scene-0318 (frame 10-30), scene-0379 (frame 14-26), scene-0408 (frame 12-30), scene-0417 (frame 4-20), scene-0422 (frame 18-39), scene-0865 (frame 24-39), scene-1105 (frame 18-30), scene-1065 (frame 24-35), scene-0200 (frame 20-39), and scene-0752 (frame 10-28). The scenes and key frames in the evaluation split are: scene-0038 (frame 4-33), scene-0271 (frame 3-11), scene-0969 (frame 14-33), scene-0329 (frame 3-33), and scene-1065 (frame 24-35)

Navigating around construction sites. This common and challenging scenario requires the autonomous vehicle to actively change lanes to bypass a construction zone. Although there are two scenes (scene-0980 and scene-0535) in the training split, none exist in the evaluation split. Therefore, we moved one training scene (scene-0980, frames 16-30) to the evaluation split. We selected 15 key frames that capture the ego vehicle decelerating and steering into the adjacent lane to bypass the traffic cones.

E Detailed Qualitative Result

In this section, we provide an in-depth analysis of the qualitative results shown in Sec. 5.2.

Executing a 3-point turn. During a 3-point turn, a vehicle makes a sharp left turn, backs up, and then makes another left turn to complete the maneuver. In Fig. 4, we compare the motion plans from TOKEN and PARA-Drive. Despite receiving a "3-point turn" command, PARA-Drive predicts straight movements at $t = 2s$ and $t = 4s$, likely due to the absence of such examples in its training set. In contrast, TOKEN understands the command and generates the correct turning behavior. When approaching the curb to stop and back up, both PARA-Drive and TOKEN predict forward motions at $t = 8s$, likely due to the lack of 3-point turn examples in the dataset. At $t = 10s$, when the vehicle has enough clearance, both models predict left-turn motions, with TOKEN's prediction more closely aligning with the ground truth.

Overtaking parked cars through the oncoming lane. Fig. 6 shows an example of qualitative comparison between the motion plan from TOKEN and PARA-Drive at two constitutive time steps. At $t = 5s$, PARA-Drive predicts a motion that collides with the blocking vehicle, while TOKEN instructs the ego vehicle to decelerate to avoid the object. Interestingly, although TOKEN correctly predicts in language that the ego vehicle should decelerate and steer to the right, the motion plan only reflects the deceleration behavior. We hypothesize that this is caused by insufficient data that helps the LLM associate low-level motion with mid-level behavior. When the ego vehicle straddles the lane divider and prepares to overtake, TOKEN instructs the ego vehicle to drive back to its original lane after overtaking, while PARA-Drive predicts a forward motion that straddles the lane divider.

Navigating around construction sites. Fig. 7 shows an example of qualitative comparison between the motion plan from TOKEN and PARA-Drive at two constitutive time steps. At $t = 8s$, TOKEN instructs the ego vehicle to steer and bypass the traffic cones from the ego vehicle's right side, while PARA-Drive predicts a motion that collides with the blocking traffic cones. At $t = 12s$, TOKEN instructs the ego vehicle to steer to the right to keep forward along the current lane, while PARA-Drive predicts a motion that deviates from the lane center.

F Comparison with the SOTA LLM-based Planner - Agent-Driver

We compare TOKEN with the SOTA LLM-based planner Agent-Driver [5]. They have the following differences:

Scene representation. Agent-Driver queries text-based scene information using various tools (e.g., object detection, mapping, etc.) and uses the queried text-based information as an input prompt to instruct the LLM to plan the ego vehicle's motion. TOKEN tokenizes the scene into a few object-level tokens. This makes TOKEN more efficient in terms of information density per-token (e.g.,

TOKEN uses one token to encode an object’s semantic, geometry, and dynamic information while the same information is tokenized into around 60 text tokens in Agent-Driver).

LLM-backbone. Agent-Driver fine-tunes the entire GPT3.5 while TOKEN uses LORA (with a rank of 64) to fine-tune LLaMA2.

Chain-of-Thought Reasoning. Both Agent-Driver and TOKEN utilize chain-of-thought reasoning to align the model’s planning process. However, Agent-Driver uses a coarse reasoning process that instructs the LLM to directly generate the ego vehicle’s discretized steering and acceleration command description based on the scene description, as opposed to TOKEN’s more structured reasoning process that instructs the model to reason about the semantically meaningful effect of the objects (e.g., blocking the ego vehicle’s path) and interaction plans (e.g., overtake).

We use the Agent-Driver’s predictions from the official repository and show the overall quantitative evaluation in Tab. 3 of the main text. In Tab. 6, we show the detailed quantitative evaluation of each long-tail scenario. In addition, since Agent-Driver includes the ego-state information in the prompt, we train a variant of TOKEN (denoted as TOKEN⁺) that also uses the ego-state information as input for planning (current velocity and current acceleration) and show the quantitative evaluation of TOKEN⁺ in Tab. 6. We see that TOKEN and Agent-Driver have similar overall performance, but TOKEN significantly outperforms Agent-Driver in long-tail scenarios. Furthermore, when using ego-state information as input, TOKEN⁺ significantly outperforms Agent-Driver in all splits.

Split	Method	Traj L2 (m) ↓					Heading L2 (rad) ↓					Lon. weighted traj L2 (m) ↓					Collision (%) ↓
		1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	
val	Agent-Driver	0.23	0.68	1.50	0.80	0.66	0.79	0.85	0.90	0.85	0.80	0.43	1.26	2.75	1.48	1.22	0.13
	TOKEN	0.26	0.70	1.46	0.81	0.68	0.13	0.18	0.21	0.17	0.18	0.50	1.32	2.72	1.51	1.26	0.15
	TOKEN ⁺	0.17	0.52	1.21	0.64	0.52	0.10	0.16	0.19	0.15	0.17	0.33	1.00	2.30	1.21	1.00	0.13
3-point turn	Agent-Driver	0.38	1.31	2.93	1.54	1.26	0.20	0.74	1.23	0.72	0.63	0.64	2.26	4.78	2.56	2.11	8.67
	TOKEN	0.39	1.29	2.60	1.43	1.18	0.21	0.35	0.71	0.42	0.36	0.68	2.15	4.33	2.39	1.98	4.00
	TOKEN ⁺	0.20	0.73	1.77	0.90	0.73	0.17	0.22	0.38	0.26	0.22	0.37	1.28	2.93	1.53	1.24	1.46
Resume motion from full stop	Agent-Driver	0.14	0.84	2.51	1.16	0.91	0.17	0.47	0.42	0.35	0.32	0.25	1.49	4.48	2.07	1.62	0.00
	TOKEN	0.13	0.70	1.58	0.80	0.65	0.09	0.24	0.31	0.22	0.19	0.24	1.24	2.66	1.38	1.13	0.00
	TOKEN ⁺	0.06	0.43	1.27	0.59	0.46	0.05	0.13	0.17	0.12	0.10	0.11	0.78	2.30	1.06	0.84	0.00
Overtake	Agent-Driver	0.27	0.89	2.07	1.08	0.88	0.05	0.13	0.24	0.14	0.12	0.47	1.46	3.37	1.77	1.45	0.77
	TOKEN	0.29	0.77	1.63	0.90	0.74	0.04	0.07	0.11	0.07	0.09	0.53	1.36	2.86	1.58	1.31	0.19
	TOKEN ⁺	0.15	0.46	1.04	0.55	0.46	0.02	0.07	0.13	0.07	0.06	0.29	0.83	1.75	0.95	0.80	0.00

Table 6: **Quantitative comparison with an LLM-based planner - Agent-Driver.** TOKEN significantly outperforms Agent-Driver in long-tail scenarios. TOKEN⁺ denotes a variant of TOKEN that uses ego-state as input, similar to Agent-Driver.

G On the Value of Alignment - Qualitative Results

In Fig. 9, we show a few qualitative comparisons between TOKEN and a variant of TOKEN trained with representation alignment but without reasoning alignment. We can see that the predicted motion plans are more aligned the GT motion with reasoning process alignment.

H Additional Results

H.1 TOKEN with HD-map Information

In the main text, we use multi-view video as the sensory input. In this section, we include HD-map as an additional input to evaluate the performance of TOKEN. We utilize the CTT encoder described in [29] to fuse each traffic agent’s past state history with each lane’s ground truth center line and produce a traffic agent token as an additional token for each traffic object. We use TOKEN⁺_{map} to denote the variant of TOKEN with HD-map information and show its quantitative evaluation in Tab. 7. We can see that the additional map information and the past state history significantly improve the planning performance in both evaluation split and long-tail scenarios.

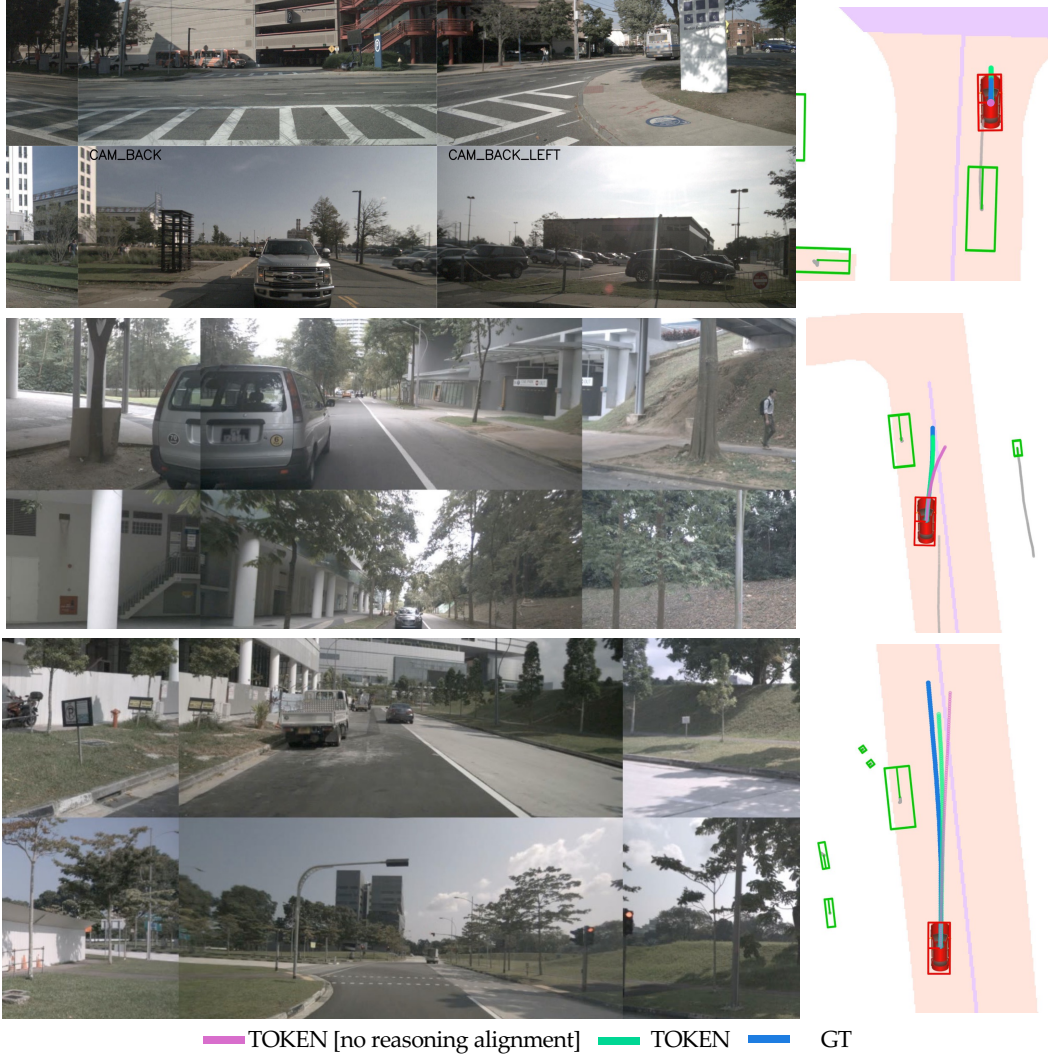


Figure 9: Qualitative comparison between TOKEN and a variant of TOKEN trained with representation alignment but without reasoning alignment. With reasoning process alignment, TOKEN is able to instruct the ego vehicle to resume motion after a full-stop (top figure) and safely overtake static obstacles (middle and bottom figures).

Split	Method	Traj L2 (m) ↓					Heading L2 (rad) ↓					Lon. weighted traj L2 (m) ↓					Collision (%) ↓
		1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	Ave _{all}
Val	TOKEN	0.26	0.71	1.47	0.81	0.68	0.02	0.04	0.06	0.04	0.03	0.50	1.32	2.73	1.52	1.27	0.15
	TOKEN ^{+map}	0.15	0.42	1.18	0.58	0.51	0.02	0.02	0.06	0.03	0.03	0.33	0.92	2.11	1.12	0.97	0.08
Long-tail	TOKEN	0.26	0.81	1.77	0.95	0.78	0.05	0.10	0.18	0.11	0.09	0.50	1.47	3.09	1.69	1.40	0.35
	TOKEN ^{+map}	0.20	0.71	1.37	0.76	0.64	0.03	0.08	0.11	0.07	0.05	0.37	1.18	2.73	1.43	1.27	0.31

Table 7: Quantitative performance of TOKEN^{+map}, a variant of TOKEN with HD-map information.

504 H.2 Ablation on the Effect of Structured Reasoning Process Alignment.

505 One of the unique features of TOKEN is that it reasons about semantically meaningful interactions
 506 with identified critical objects (e.g., bypassing the blocking traffic cones). We hypothesize that this
 507 structured reasoning process supervision enhances the model’s planning performance by encour-
 508 aging the model to understand the interactions between the ego vehicle and other traffic objects,
 509 aligning more closely with how an expert reasons in the real world. To ablate the effect of structured
 510 reasoning process alignment, we removed it from the planning QAs’ answer labels and used a sim-
 511 ilar chain-of-thought reasoning and task planning method as in Agent-Driver (i.e., instructing the

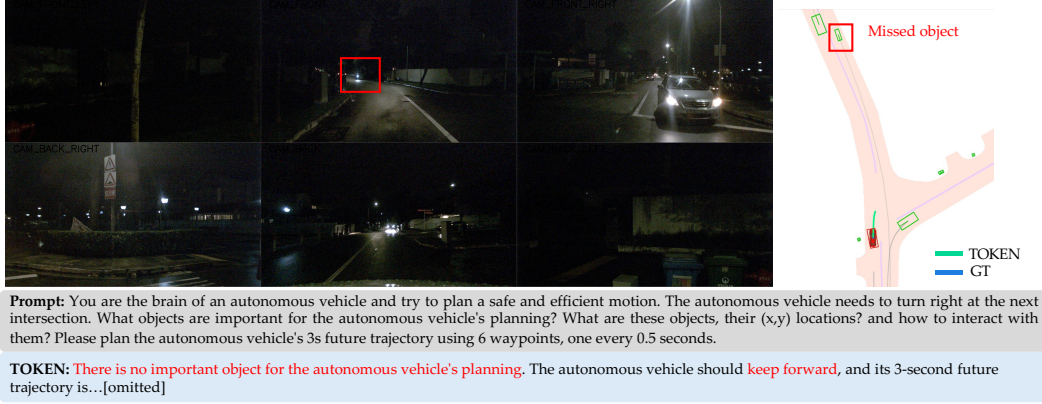


Figure 10: Failure mode example: The critical object (an oncoming motorcycle annotated by the red rectangle) is not detected by TOKEN’s scene tokenizer, resulting in a dangerous motion plan.

LLM to generate the steering and acceleration command description first, followed by the motion plan) to TOKEN (denoted as TOKEN^{-interact.}). We show the evaluation result in Tab. 8. We can see that the structured reasoning process alignment improves the planning performance in both the evaluation set and the long-tail scenarios.

Split	Method	Traj L2 (m) ↓					Heading L2 (rad) ↓					Lon. weighted traj L2 (m) ↓					Collision (%) ↓
		1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	
Val	TOKEN ^{-interact.}	0.28	0.77	1.54	0.86	0.75	0.02	0.03	0.08	0.04	0.04	0.52	1.39	2.82	1.58	1.31	0.17
	TOKEN	0.26	0.71	1.47	0.81	0.68	0.02	0.04	0.06	0.04	0.03	0.50	1.32	2.73	1.52	1.27	0.15
Long-tail	TOKEN ^{-interact.}	0.33	1.01	2.06	1.13	0.92	0.09	0.15	0.19	0.14	0.12	0.58	1.82	3.66	2.02	1.65	0.59
	TOKEN	0.26	0.81	1.77	0.95	0.78	0.05	0.10	0.18	0.11	0.09	0.50	1.47	3.09	1.69	1.40	0.35

Table 8: Quantitative performance of TOKEN^{-interact.}, a variant of TOKEN that uses a similar chain-of-thought reasoning as Agent-Driver does.

H.3 Few-Shot Learning.

To stress test TOKEN’s few shot learning ability, we further remove 50% long-tail scenes from the training split and re-train PARA-drive and TOKEN and compare their performance. In Tab. 9, we show the quantitative evaluation result in long-tail scenarios. We see that TOKEN only degrades slightly as opposed to PARA-Drive’s significant performance degradation. For example, Traj L2 Ave_{all} of PARA-Drive is degraded by 24% while TOKEN only experiences 9% degradation with 50% long-tail scenes removed. The results indicate the superior few-shot learning ability of TOKEN.

Split	Method	Traj L2 (m) ↓					Heading L2 (rad) ↓					Lon. weighted traj L2 (m) ↓					Collision (%) ↓
		1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	1s	2s	3s	Ave _{123s}	Ave _{all}	
Long-tail (full long-tail training set)	PARA-Drive	0.26	0.96	2.16	1.13	0.91	0.09	0.20	0.34	0.21	0.18	0.47	1.61	3.51	1.86	1.57	0.51
	TOKEN	0.26	0.81	1.77	0.95	0.78	0.05	0.10	0.18	0.11	0.09	0.50	1.47	3.09	1.69	1.40	0.35
Long-tail (with 50% long-tail training scenes removed)	PARA-Drive	0.51	1.17	2.30	1.32	1.12	0.09	0.17	0.29	0.18	0.16	0.76	1.96	3.93	2.22	1.86	0.89
	TOKEN	0.28	0.86	1.83	0.99	0.85	0.06	0.15	0.24	0.15	0.13	0.56	1.66	3.29	1.84	1.62	0.41

Table 9: Planning performance of TOKEN and PARA-Drive with 50% long-tail training scenes removed.

I Failure Mode

One limitation of TOKEN is using a pre-trained and frozen PARA-Drive model as the scene tokenizer, which makes the TOKEN’s performance tightly coupled with the quality of the pretrained tokenizer. In Fig. 10, we illustrate a failure mode where the critical object (the motorcycle) is not detected by the tracking querying transformer in PARA-Drive. Consequently, TOKEN assumes the road is clear to proceed and fails to generate a motion that yields to the oncoming motorcycle. Further work will focus on co-training PARA-Drive to leverage the knowledge within the LLM to improve the scene tokenizer.