End-to-End Driving with Online Trajectory Evaluation via BEV World Model

Supplementary Material

1. More Ablation Studies

Robust against reward weights As discussed in Sec. 3.3 of the main paper, the reward model predicts a set of rewards and uses reward weights to combine them to generate the final reward. To analyze the impact of reward weights, we perform an ablation study, as shown in Table 1. The results demonstrate that our model is robust to variations in these hyper-parameters. Among them, w_1, the weight of the imitation reward, has the most significant influence. Increasing w_1 results in a performance drop of 1.3 PMDS. For the other reward weights, the results remain relatively stable. This experiment highlights that the combination of multiple rewards enables our framework to produce a stable final reward, effectively reducing the impact of poorly performing individual reward on the overall evaluation.

2. More Implementation Details

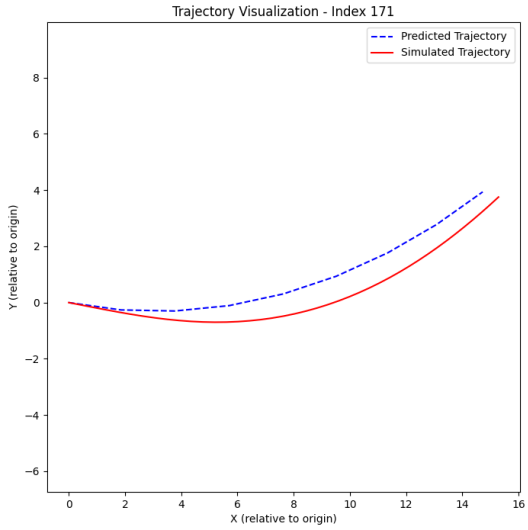
A key advantage of training with a simulator is its ability to generate more realistic ego vehicle trajectory simulations compared to traditional open-loop settings. Open-loop methods often assume that the predicted trajectory perfectly matches the actual trajectory, which neglects critical factors such as the ego vehicle's initial speed, acceleration, and performance limitations. In contrast, simulations account for these dynamics, producing trajectories that better reflect real-world conditions, as shown in Figure 1. These simulated trajectories allow for a more accurate evaluation of collision risks in complex environments.

Specifically, after the model predicts a trajectory, the simulator utilizes a Linear Quadratic Regulator (LQR)[**?**] tracker in combination with a bicycle model[**?**] to simulate the ego vehicle's actual path, enhancing the realism and reliability of the results.

LQR Tracker The LQR tracker is an optimal control algorithm designed to minimize a quadratic cost function, balancing state errors and control efforts. Importantly, the trajectory generated by the LQR tracker is influenced by the vehicle's current speed and acceleration. In the context of vehicle trajectory tracking, the LQR computes the optimal steering and acceleration inputs needed for the ego vehicle to closely follow the desired trajectory. The control input $\mathbf{u}(t)$ is determined by:

$$\mathbf{u}(t) = -\mathbf{K}\left(\mathbf{x}(t) - \mathbf{x}_{\text{ref}}(t)\right)$$

where $\mathbf{x}(t)$ is the current state vector, $\mathbf{x}_{\text{ref}}(t)$ is the reference state vector from the predicted trajectory, and $\mathbf{K}$ is the feedback gain matrix obtained by solving the Riccati equation.



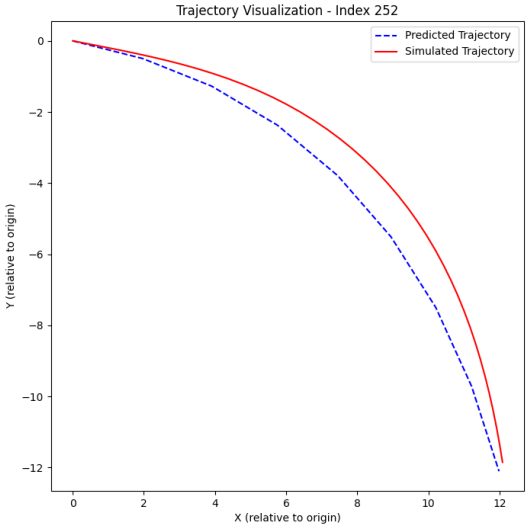


Figure 1. **Comparison of model-predicted trajectories and the actual trajectories executed by the LQR tracker and bicycle model.** The predicted trajectories cannot be executed as expected in practice.

By continuously updating the feedback gains, the LQR adjusts the control inputs in real time, ensuring stability and responsiveness in the tracking performance.

w_1	w_2	w_3	w_4	NC ↑	DAC ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑
0.1	*0.5*	*0.5*	*1*	98.0	94.7	79.9	93.4	100.0	85.6
1	0.5	0.5	1	97.9	93.4	78.8	92.9	100.0	84.3
0.01	0.5	0.5	1	97.7	95.4	79.8	92.3	100.0	85.4
0.1	**5**	0.5	1	98.2	94.7	79.6	93.8	100.0	85.7
0.1	**0.05**	0.5	1	98.0	94.8	79.9	93.3	100.0	85.6
0.1	0.5	**5**	1	97.8	95.7	79.1	93.2	99.8	85.7
0.1	0.5	**0.05**	1	98.0	94.1	79.6	93.4	100.0	85.2
0.1	0.5	0.5	**0.1**	97.9	94.6	79.5	93.1	100.0	85.3
0.1	0.5	0.5	**10**	97.7	95.2	80.8	92.3	100.0	85.7

Table 1. **Ablation study on different reward weights.** The weights are defined in Eq. 4. Weight in each row that differ from the *default setting* (first row) are highlighted in **bold**.

Bicycle Model The bicycle model is a simplified representation of a vehicle's dynamics that captures the essential lateral and longitudinal motions using two virtual wheels—one at the front and one at the rear. By incorporating the bicycle model, the resulting trajectory adheres to the vehicle's dynamic constraints. The model is governed by the following equations:

$$\dot{x} = v\cos(\theta + \beta)$$
$$\dot{y} = v\sin(\theta + \beta)$$
$$\dot{\theta} = \frac{v}{L}\sin(\beta)$$
$$\beta = \arctan\left(\frac{L_r}{L}\tan(\delta)\right)$$

Here, x and y are the vehicle's position coordinates, θ is the heading angle, v is the speed, L is the wheelbase, L_r is the distance from the rear axle to the center of gravity, δ is the steering angle, and β is the slip angle. By considering critical parameters such as vehicle mass, center of gravity, tire forces, and wheelbase, the bicycle model maintains a high level of accuracy in simulating the vehicle's behavior under various driving conditions.

3. Additional Visual Analysis

Here, we provide the failure case analysis as shown in Figure 2 and additional visualizations of the planning trajectories as shown in Figures 3, 4, and 5.

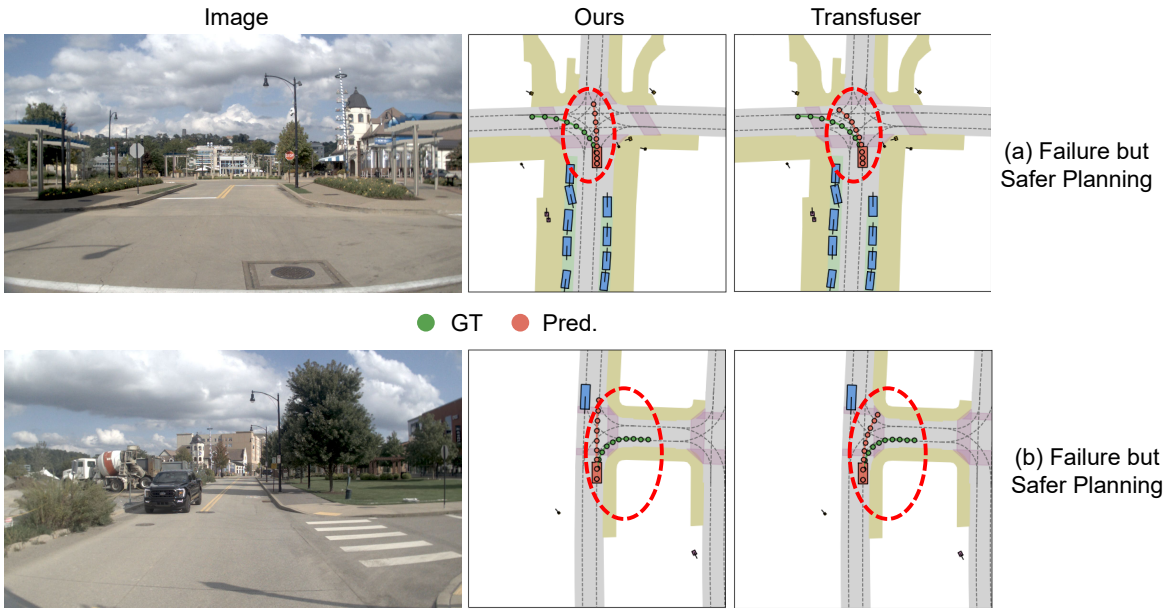


Figure 2. **Failure Case Analysis.** In the illustrated example, both our method and Transfuser [**?**] fail to predict the correct future trajectory. However, with the aid of the trajectory evaluation module, our approach generates a safe trajectory that remains within the road boundaries. In contrast, the trajectory predicted by Transfuser leads directly to a collision with the curb.

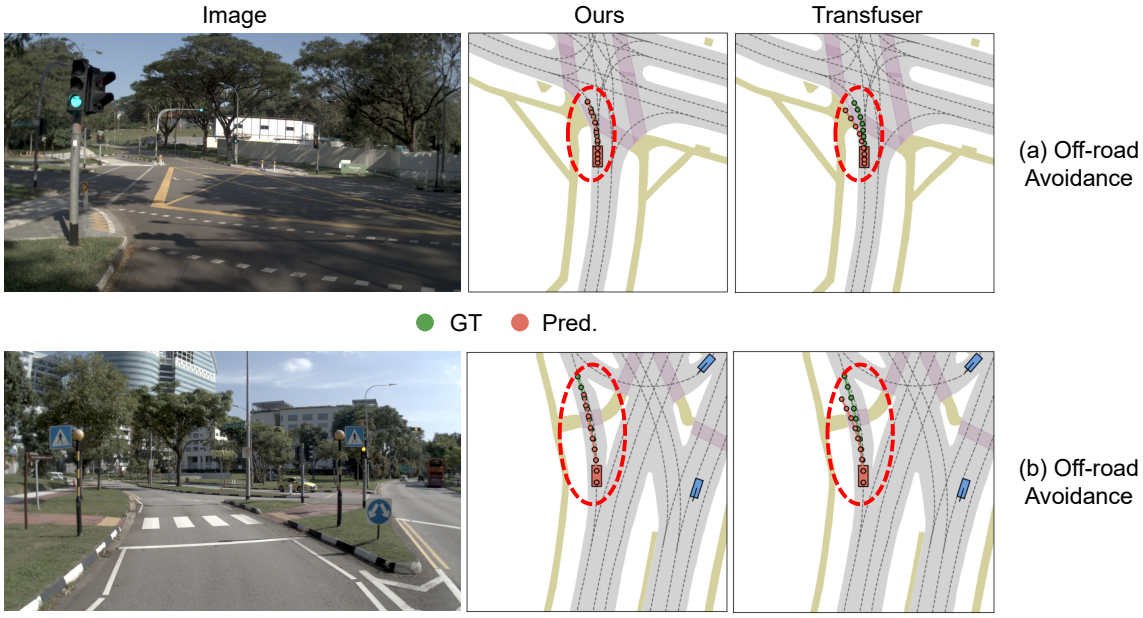


Figure 3. **Off-road avoidance.** The trajectory evaluation module ensures that the trajectory remains on the road and avoids deviations that could lead to off-road driving.

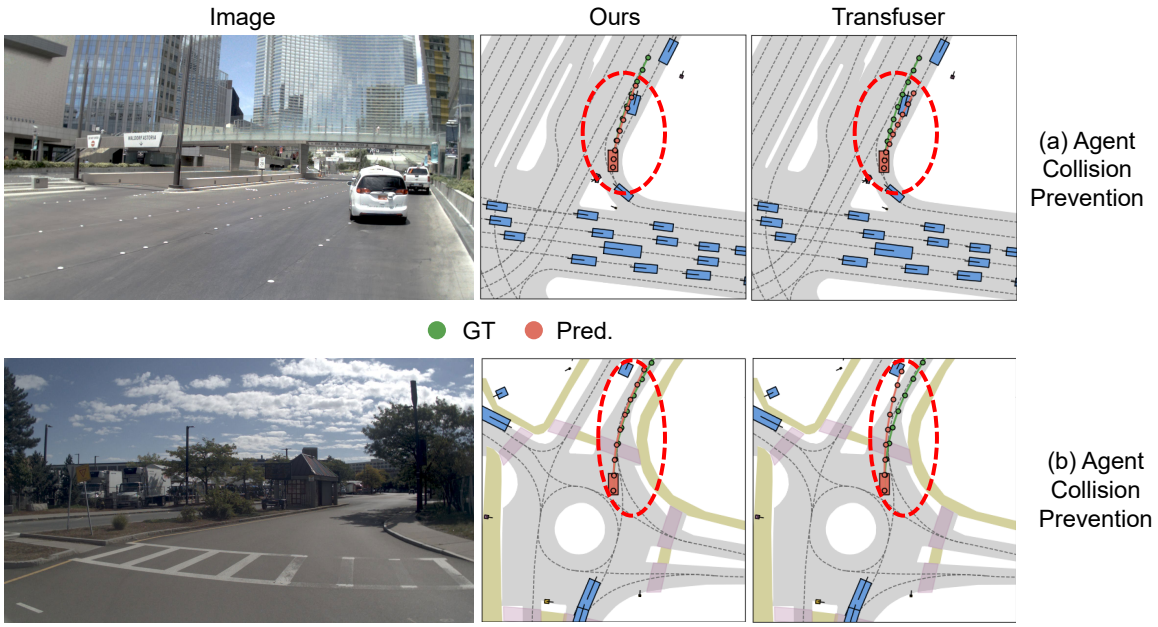


Figure 4. **Agent collision prevention.** The trajectory evaluation module ensures a safe distance is maintained from other agents, preventing the vehicle from coming too close and reducing the risk of potential collisions.

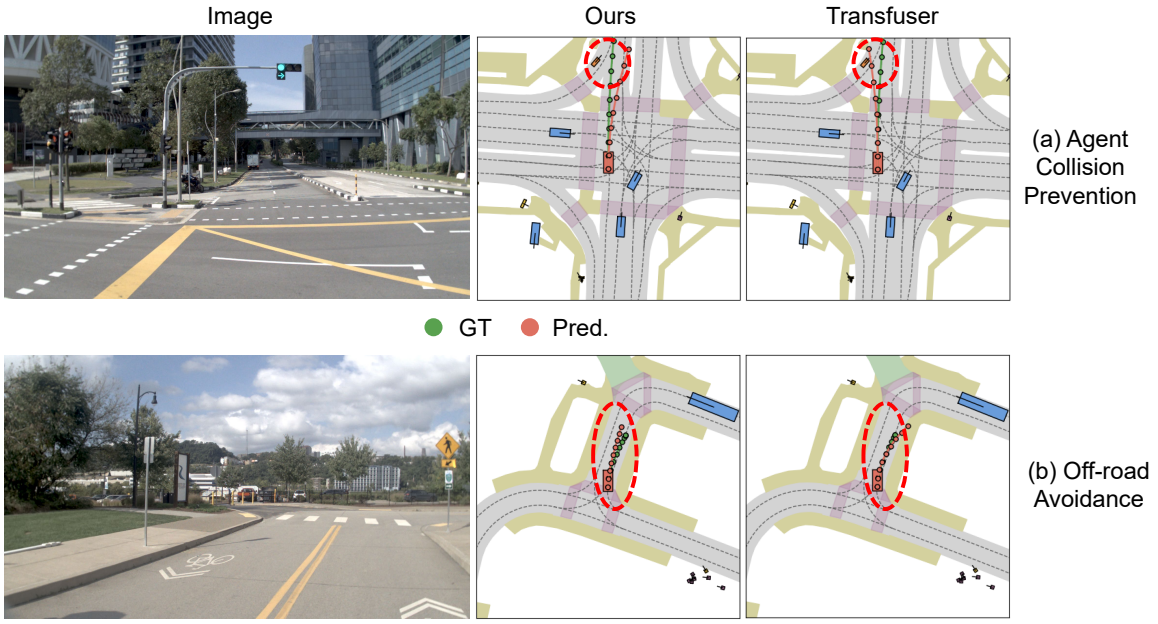


Figure 5. **Ensuring a safer distance.** In (a), Transfuser does not account for potential collisions with agents, whereas our approach maintains a safer distance. In (b), Transfuser's trajectory shifts closer to the road edge, while our method avoids proximity to the edge for safer navigation.