RoaD: Rollouts as Demonstrations for Closed-Loop Supervised Fine-Tuning of Autonomous Driving Policies

Supplementary Material

A. Leaderboard Snapshot

In Fig. 5 we show a snapshot of the nuPlan WOSAC leaderboard, with our *SMART-tiny-CLSFT-RoaD* entry highlighted (red box). We briefly discuss the other high-performing methods to clarify how they are either orthogonal to our approach or specialized to the WOSAC task, and thus do not directly transfer to E2E driving.

SMART-tiny-DecompGAIL [10], *SMART-tiny-RLFTSim* [1], and *SMART-R1* [29] use reinforcement learning (RL) to optimize policies for the WOSAC task of matching the data distribution. *SMART-tiny-DecompGAIL* does so by using GAIL with PPO, while *SMART-tiny-RLFTSim* and *SMART-R1* directly optimize the WOSAC metric using RL (with small differences in implementation). However, these methods do not translate well to E2E driving, where we typically lack a well-defined reward function and RL tends to be too data-inefficient given the high cost of high-fidelity simulation.

By contrast, *TrajTok* [42] proposes an improved tokenizer for traffic models. This contribution is orthogonal to our approach. However, it is only applicable to tokenizing short actions (e.g., one-step actions), and hence does not translate to E2E driving, where policies typically predict multiple seconds into the future.

Finally, *unimotion* [19] proposes an alternative to the CatK rollout approach, whereby it finds the closest action to the ground truth not only among the top-K actions, but among all actions. As a result, it does not require additional recovery actions (since it already tracks the ground-truth trajectory as closely as possible), which yields a setup more similar to our RoaD approach, where rollouts are taken directly as demonstrations. However, this exhaustive search makes the method unsuitable for E2E driving: it requires predicting and evaluating *all* possible actions of the policy, which is only feasible for small action spaces. This excludes multi-token trajectory predictions, whose action space grows exponentially with the horizon length, and flow-matching policies, whose action space is continuous. In their work, focussing on traffic models, they use either a fixed set of up to 2024 actions or a set of up to 16 actions predicted from action queries.

B. Experimental details: end-to-end driving

B.1. Simulation

For data generation, we run the simulator at 30 Hz to match the frequency of the ground-truth logs and reuse the same dataloader as for pre-training. For evaluation, we run the simulator at 10 Hz, which matches the model's training frequency.

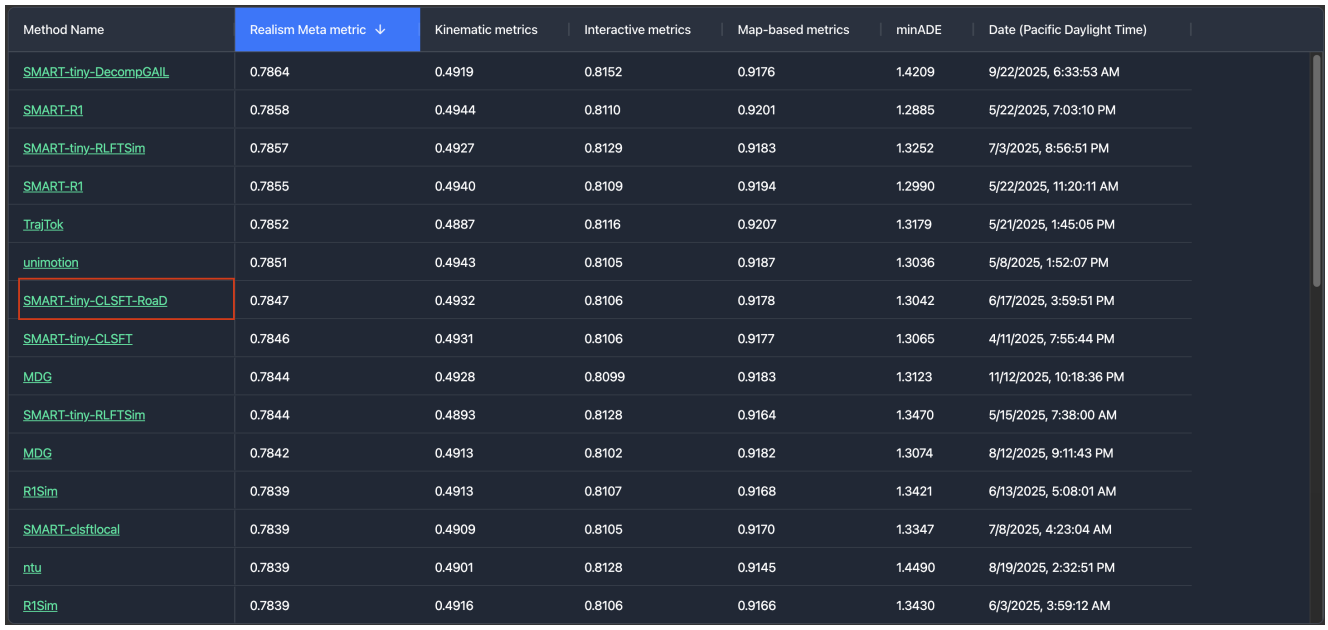
At each step, we render two camera views (front-facing wide-angle and telephoto). The policy predicts 6.4s trajectories, which are tracked by a downstream controller. As in the main text, the controller models a 200 ms control delay and ego-motion noise, and uses a dynamically extended bicycle model for the ego-vehicle dynamics.

Scenes are reconstructed using 3D Gaussian Splatting (3D-GS). Reconstruction quality degrades with distance from the recorded trajectory. This is negligible for rollout generation, where expert-guided rollouts remain close to the log, but can reduce visual fidelity during evaluation if the ego vehicle deviates too far. To avoid such artifacts, we discard rollouts whose ego trajectory deviates by more than 4 m from the recorded trajectory.

All other traffic participants, including vehicles and pedestrians, replay their logged trajectories and do not react to the ego vehicle. Consequently, they cannot avoid rear-end collisions if the ego drives more slowly than in the recording. Following prior work [4, 5], we therefore count only "at-fault" collisions for the ego: rear-end collisions caused by following vehicles are ignored, while lateral collisions are still included.

B.2. RoaD fine-tuning

For RoaD rollout generation, we sample $K{=}64$ candidate trajectories from the policy at a temperature of 0.8. To select the executed trajectory, we compute the ground-truth distance d^{g} as the average distance between the four corners of the ego vehicle along the predicted and ground-truth trajectories over the first 20 time steps (2s). The same distance metric is used to decide whether to trigger the recovery mode, with a threshold of $\delta_{\text{rec}}{=}3$ m. When recovery is triggered, we linearly interpolate between the predicted and ground-truth trajectories over $N_{\text{rec}}{=}30$ time steps and follow the ground-truth trajectory thereafter. Recovery is disabled in the last 4s of the episode because the controller requires at least 4s of input trajectory.



Method Name	Realism Meta metric ↓	Kinematic metrics	Interactive metrics	Map-based metrics	minADE	Date (Pacific Daylight Time)
SMART-tiny-DecompGAIL	0.7864	0.4919	0.8152	0.9176	1.4209	9/22/2025, 6:33:53 AM
SMART-R1	0.7858	0.4944	0.8110	0.9201	1.2885	5/22/2025, 7:03:10 PM
SMART-tiny-RLFTSim	0.7857	0.4927	0.8129	0.9183	1.3252	7/3/2025, 8:56:51 PM
SMART-R1	0.7855	0.4940	0.8109	0.9194	1.2990	5/22/2025, 11:20:11 AM
TrajTok	0.7852	0.4887	0.8116	0.9207	1.3179	5/21/2025, 1:45:05 PM
unimotion	0.7851	0.4943	0.8105	0.9187	1.3036	5/8/2025, 1:52:07 PM
SMART-tiny-CLSFT-RoaD	0.7847	0.4932	0.8106	0.9178	1.3042	6/17/2025, 3:59:51 PM
SMART-tiny-CLSFT	0.7846	0.4931	0.8106	0.9177	1.3065	4/11/2025, 7:55:44 PM
MDG	0.7844	0.4928	0.8099	0.9183	1.3123	11/12/2025, 10:18:36 PM
SMART-tiny-RLFTSim	0.7844	0.4893	0.8128	0.9164	1.3470	5/15/2025, 7:38:00 AM
MDG	0.7842	0.4913	0.8102	0.9182	1.3074	8/12/2025, 9:11:43 PM
R1Sim	0.7839	0.4913	0.8107	0.9168	1.3421	6/13/2025, 5:08:01 AM
SMART-clsftlocal	0.7839	0.4909	0.8105	0.9170	1.3347	7/8/2025, 4:23:04 AM
ntu	0.7839	0.4901	0.8128	0.9145	1.4490	8/19/2025, 2:32:51 PM
R1Sim	0.7839	0.4916	0.8106	0.9166	1.3430	6/3/2025, 3:59:12 AM

Figure 5. Snapshot of the WOSAC leaderboard with our SMART-tiny-CLSFT-RoaD entry highlighted (red box).