# Driving on Registers

## Supplementary Material

## A. Description of PDMS and EPDMS metrics

The *Predictive Driver Model Score (PDMS)*, main metric of NAVSIM-v1, and the *Extended Predictive Driver Model Score (EPDMS)*, main metric of NAVSIM-v2, follow a similar definition, based on a group $G$ of sub metrics. $G$ can be divided into two kinds of metrics: a group $G_p$ of penalties, and a group $G_b$ of behavioral metrics. $G_p$ contains the metrics measuring the compliance to driving rules, such as drivable area compliance or collision occurrence. $G_b$ contains the metrics measuring more driver-related metrics, such as comfort or progress towards the goal. The PDMS and EPDMS metrics are defined as:

$$\text{Score}(\tau_i) = \prod_{c \in G_p} \mathcal{G}_c(\tau_i)^{\kappa_c^p} \times \frac{1}{Z} \sum_{c \in G_b} \kappa_c^b \, \mathcal{G}_c(\tau_i) \quad (5)$$

for $\tau_i$ a candidate trajectory, $Z = \sum_{c \in G_b} \kappa_c^b$ and $\kappa$ as defined in Tab. 8. $G_p$ metrics are multiplicative, meaning one failure to comply to a targeted driving rule, i.e., $\mathcal{G}_c(\tau_i) = 0$ results in a score equal to zero. On the contrary, $G_b$ metrics are additive and allow compromise between the metrics, e.g., comfort and progress.

| Abbr. | Sub-score $c$ | Grp. | PDMS $\kappa_c$ | EPDMS $\kappa_c$ |
|---|---|---|---|---|
| NC | No-at-fault Collisions | $G_p$ | 1 | 1 |
| DAC | Drivable Area Compliance | $G_p$ | 1 | 1 |
| DDC | Driving direction Compliance | $G_p$ | 0 | 1 |
| TLC | Traffic-line compliance | $G_p$ | - | 1 |
| TTC | Time to Collision | $G_b$ | 5 | 5 |
| EP | Ego Progress | $G_b$ | 5 | 5 |
| Comf. | Ego Comfort | $G_b$ | 2 | - |
| LK | Lane Keeping | $G_b$ | - | 2 |
| HC | History Comfort | $G_b$ | - | 2 |
| EC | Extended Comfort | $G_b$ | - | 2 |

Table 8. **Weights of the sub-scores** in the PDMS and EPDMS.

## B. Training loss weights

We recall here the equation of the score loss defined in the main paper:

$$\mathcal{L}_{\text{score}} = \sum_c \lambda_c \sum_i \text{BCE}\left(\mathcal{G}_{\theta_c}(\tau_i), \mathcal{G}_c(\tau_i)\right) \quad (6)$$

which corresponds an individual loss for each sub-score of the PDMS. During training, the predicted sub-scores and their associated weight $\lambda_c$ are set as defined in Tab. 9, i.e., all $\lambda_c$ are set to 1.

| Sub-score | NC | DAC | DDC | TTC | EP | Comf. |
|---|---|---|---|---|---|---|
| $\lambda_c$ | 1 | 1 | 1 | 1 | 1 | 1 |

Table 9. **Training loss weights of the sub-scores.**

Similarly, as defined in the main paper, the final loss is:

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \lambda_s \mathcal{L}_{\text{score}}. \quad (7)$$

where, again, $\lambda_s$ is set to 1.

## C. Inference weights of the sub-scores

| Abbr. | Sub-score $c$ | Grp. | PDMS $\kappa_c$ | EPDMS $\kappa_c$ |
|---|---|---|---|---|
| NC | No-at-fault Collisions | $G_p$ | 1 | 10 |
| DAC | Drivable Area Compliance | $G_p$ | 1 | 13 |
| DDC | Driving direction Compliance | $G_p$ | 0 | 6 |
| TTC | Time to Collision | $G_a$ | 5 | 14 |
| EP | Ego Progress | $G_a$ | 5 | 15 |
| Comf. | Ego Comfort | $G_a$ | 2 | 2 |

Table 10. **Inference weights of the sub-scores.**

The subscore prediction design in DrivoR enables flexible weighting strategy. In Tab. 10, we provide the subscore weights for both PDMS (NAVSIM-v1) and EPDMS (NAVSIM-v2). For simplicity, the model trained for NAVSIM-v1 keeps the same weights as in the standard PDMS metrics [6]. For NAVSIM-v2, we adjust the weights with the exception of the Comfort. The adjustment is validated with NAVSIM-v2 `warmup-two-stage`, see Figure 6 in the original paper.

## D. Implementation details

We conduct our experiments on a node of 8 × A100 GPUs. We run each training on 4 × A100 GPUs with a base learning rate of 0.0002 and batch size of 16, using AdamW optimizer and a cosine annealing learning rate scheduler. The training lasts around 1.5 hours per epoch for the competition split (`navtrain` + `navval`) and around 1 hour per epoch for `navtrain` only. As described in the main paper, we train for 25 epochs for the NAVSIM-v1 (`navtrain` + `navval`) and 10 epochs for NAVSIM-v2 `navtrain`.

## E. Efficiency analysis

Tab. 11 shows the full results of the throughput analysis. We benchmarked each model on a single batch on

| **Method** | Img. size | Cams | Encoder | Parameters ↓ | GFLOPs ↓ | Peak Memory ↓ | Throughput (ms) ↓ | NAVSIM-v2 EPDMS ↑ |
|---|---|---|---|---|---|---|---|---|
| RAP-Dino† | (448, 768) | 4 | ViT-H | 888M<br>857M‖31M | 4760<br>4622‖138 | 4.2GB<br>- | 690ms<br>653ms‖37ms | 39.6<br>- |
| *GTRS uses extra trajectories from a diffusion-based model, introducing extra overheads.* | | | | | | | | |
| +GTRS-DP | (512, 2048) | 4 | +V2-99<br>- | +116M<br>110M‖6M | +1249<br>770‖479 | +1.15GB<br>- | +389ms<br>60ms‖329ms | -<br>- |
| GTRS-D<br>- | (512, 2048)<br>- | 1<br>- | V2-99<br>- | 81M<br>69.5M‖11.5M | 404<br>345‖59 | 1.0GB<br>- | 96ms<br>22ms‖74ms | 45.0<br>- |
| GTRS-A<br>- | (512, 2048)<br>- | 1<br>- | V2-99<br>- | 171M<br>69.5M‖101.5M | 439<br>345‖94 | 1.42GB<br>- | 243ms<br>22ms‖221ms | 45.4<br>- |
| GTRS-D‡ | (512, 2048)<br>- | 1<br>- | ViT-L<br>- | 321M<br>313M‖8M | 1730<br>1610‖120 | 1.6GB<br>- | 400ms<br>352ms‖48ms | 47.0<br>- |
| GTRS-D‡ | (512, 2048) | 1 | ViT-S<br>- | 32M<br>24M‖8M | 234<br>117‖117 | 0.38GB<br>- | 303ms<br>46ms‖257ms | 45.9<br>- |
| ZTRS | (512, 2048) | 2 | V2-99<br>- | 81M<br>69.5M‖11.5M | 840<br>690‖150 | 1.0GB<br>- | 193ms<br>119ms‖74ms | 48.1<br>- |
| DrivoR | (672, 1148) | 4 | ViT-S<br>- | **41M**<br>24M‖17M | **351**<br>350‖1 | **0.5GB**<br>- | **110ms**<br>107ms‖3ms | **48.3**<br>- |

†: RAP is trained on a dataset that is 10× larger than navtrain (the default training set). ‡: Reproduced.

Table 11. **Efficiency.** We compare the number of parameters, GFLOPs, peak memory consumption, and throughput w.r.t. the NAVSIM-v2 EPDMS performance. *GTRS-A* refers to *GTRS-Aug* and *GTRS-D* for *GTRS-Dense*, we decompose the parameters, GFLOPs and throughout in the image backbone and the rest of the network respectively: *image backbone*‖*rest*.

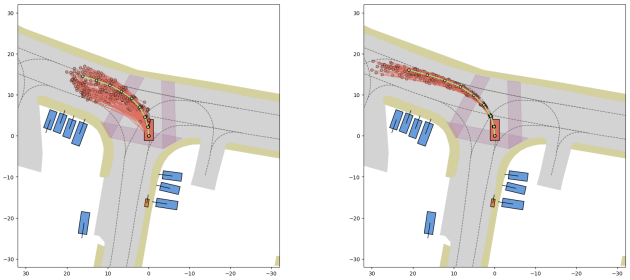
a single A100 GPU. FLOPs counts were conducted using the FVCore library. Peak memory was counted using Pytorch's `max_memory_allocated` function. (Note: FVCore currently excludes Scaled Dot Product Attention and thus FLOPs counts do not reflect the real count for each model, though relative performance should remain consistent.) Throughput was counted on a single forward pass after three warm-up iterations, with the final number averaged over 10 iterations. DrivoR has much improved throughput and lower memory consumption, demonstrating the best tradeoff between performance and efficiency.

## F. Single- vs multi- token per trajectory

Previous works such as [8] have used a single token per trajectory pose, decoding each token into an $(x, y, \theta)$ tuple. We ablate the choice of decoding a single token to a full trajectory vs decoding a set of tokens. Tab. 12 shows that mapping a single token to a trajectory leads to a large jump in performance. Fig. 7 visualizes the same scene, with trajectories decoded either from multi- or single-token. The trajectories mapped from a single token are much smoother and contain less noise, showing that representing trajectories with a single token simplifies learning.

| Trajectory representation | Multi-token | Single-token |
|---|---|---|
| PDMS | 83.9 | 90.0 |

Table 12. **Single- vs multi-token trajectory.** Quantitative comparison on `navval`.



(a) Multi-token (b) Single-token

Figure 7. **Single- vs multi- token trajectory.** Qualitative comparison: single-token trajectories are much smoother and less noisy.

## G. Expanded NAVSIM-v1 results

We provide in Tab. 13 a full comparison to state-of-the-art methods, including methods with LiDAR inputs and with post-processing such test-time training, Best-of-N scoring

| Method | | Mod. | NC | DAC | TTC | Comf. | EP | PDMS |
|---|---|---|---|---|---|---|---|---|
| PDM-Closed | [5] PMLR'23 | - | 94.6 | 99.8 | 89.9 | 86.9 | 99.9 | 89.1 |
| Human driver | [6] NeurIPS'24 | - | 100 | 100 | 100 | 99.9 | 87.5 | 94.8 |
| *Test-time training.* | | | | | | | | |
| Centaur | [23] arXiv'25 | C | 99.5 | 98.9 | 98.0 | 100 | 85.9 | 92.6 |
| *Best-of-N scores.* | | | | | | | | |
| AutoVLA | [36] NeurIPS'25 | C | 99.1 | 97.1 | 97.1 | 100.0 | 87.6 | 92.1 |
| DriveVLA-W0 | [14] arXiv'25 | C | 99.3 | 97.4 | 97.0 | 99.9 | 88.3 | 93.0 |
| EvaDrive | [11] arXiv'25 | C | 98.9 | 98.5 | 96.2 | 100 | 93.9 | 94.9 |
| TransDiffuser | [10] arXiv'25 | C+L | 99.4 | 96.5 | 97.8 | 99.4 | 94.1 | 94.9 |
| *Ensemble methods.* | | | | | | | | |
| Hydra-MDP-C (V2-99) | [17] arXiv'25 | C+L | 98.7 | 98.2 | 95.0 | 100 | 86.5 | 91.0 |
| *Methods using 10× more training data.* | | | | | | | | |
| RAP-DINO† | [7] arXiv'25 | C | 99.1 | 98.9 | 96.7 | 100 | 90.3 | 93.8 |
| *Multi-modal methods.* | | | | | | | | |
| TransFuser | [4] TPAMI'22 | C+L | 97.7 | 92.8 | 92.8 | 100 | 79.2 | 84.0 |
| DistillDrive | [31] arXiv'25 | C+L | 98.1 | 94.6 | 93.6 | 100 | 81.0 | 86.2 |
| TrajHF (EM) | [12] arXiv'25 | C+L | 96.6 | 96.6 | 92.1 | 100 | 84.5 | 87.6 |
| DiffusionDrive | [20] CVPR'25 | C+L | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1 |
| WOTE | [15] ICCV'25 | C+L | 98.5 | 96.8 | 94.9 | 99.9 | 81.9 | 88.3 |
| Hydra-MDP (V2-99) | [17] arXiv'24 | C+L | 98.0 | 97.8 | 93.9 | 100 | 86.5 | 90.3 |
| GoalFlow (V2-99) | [29] CVPR'25 | C+L | 98.4 | 98.3 | 94.6 | 100 | 85.0 | 90.3 |
| ResAD (V2-99) | [35] arXiv'25 | C+L | 98.9 | 97.8 | 94.9 | 100 | 87.0 | 90.6 |
| SeerDrive (V2-99) | [33] NeurIPS'25 | C+L | 98.8 | 98.6 | 95.8 | 100 | 84.2 | 90.7 |
| *Camera-only methods.* | | | | | | | | |
| Ego-stat. MLP | [6] NeurIPS'24 | C | 93.0 | 77.3 | 83.6 | 100 | 62.8 | 65.6 |
| UniVLA | [26] arXiv'25 | C | 96.9 | 91.1 | 91.7 | 96.7 | 76.8 | 81.7 |
| DrivingGPT | [3] ICCV'24 | C | 98.9 | 90.7 | 94.9 | 95.6 | 79.7 | 82.4 |
| UniAD | [9] CVPR'23 | C | 97.8 | 91.9 | 92.9 | 100 | 78.8 | 83.4 |
| LTF | [4] TPAMI'22 | C | 97.4 | 92.8 | 92.4 | 100 | 79.0 | 83.8 |
| PARA-Drive | [27] CVPR'24 | C | 97.9 | 92.4 | 93.0 | 99.8 | 79.3 | 84.0 |
| DriveX-S | [22] ICCV'25 | C | 97.5 | 94.0 | 93.0 | 100 | 79.7 | 84.5 |
| World4Drive | [34] ICCV'25 | C | 97.4 | 94.3 | 92.8 | 100 | 79.9 | 85.1 |
| DRAMA | [32] ISRR'24 | C | 98.0 | 93.1 | 94.8 | 100 | 80.1 | 85.5 |
| VAD-v2 | [2] arXiv'24 | C | 98.1 | 94.8 | 94.3 | 100 | 80.6 | 86.2 |
| PRIX | [28] RA-L'26 | C | 98.1 | 96.3 | 94.1 | 100 | 82.3 | 87.8 |
| DiffusionDrive | [20] CVPR'25 | C | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1 |
| DIVER | [24] arXiv'25 | C | 98.5 | 96.5 | 94.9 | 100 | 82.6 | 88.3 |
| AutoVLA | [36] NeurIPS'25 | C | 98.4 | 95.6 | 98.0 | 99.9 | 81.9 | 89.1 |
| DriveVLA-W0 | [14] arXiv'25 | C | 98.7 | 99.1 | 95.3 | 99.3 | 83.3 | 90.2 |
| ReCogDrive | [16] arXiv'25 | C | 97.9 | 97.3 | 94.9 | 100 | 87.3 | 90.8 |
| Hydra-MDP++ | [13] arXiv'25 | C | 98.6 | 98.6 | 95.1 | 100 | 85.7 | 91.0 |
| R2SE | [21] TPAMI'26 | C | 99.0 | 97.9 | 96.4 | 100 | 86.8 | 91.6 |
| iPad | [8] arXiv'25 | C | 98.6 | 98.3 | 94.9 | 100 | 88.0 | 91.7 |
| DriveSuprim | [30] arXiv'25 | C | 98.6 | 98.6 | 95.5 | 100 | 91.3 | 93.5 |
| DrivoR (train) | | C | 98.9 | 98.3 | 96.2 | 100 | 89.1 | 93.1 |
| DrivoR (trainval) | | C | 99.0 | 98.9 | 96.7 | 100 | 90.0 | 93.7 |
| DrivoR (+65k SimScale data) | | C | 99.1 | 99.0 | 96.9 | 100 | 90.3 | 94.0 |
| DrivoR (+134k SimScale data) | | C | 99.1 | 99.2 | 96.9 | 100 | 91.6 | **94.6** |

†: RAP [7] is trained on a dataset that is 10× larger than navtrain (the default training set).

Table 13. **NAVSIM-v1 scores.** Full comparison to existing methods, possibly with different modalities (Mod.), on the NAVSIM-v1 benchmark on test set (`navtest`). 'C' refers to camera, and 'L' to LiDAR.

and ensembling. We note that the Best-of-N practice aims to score all predicted trajectories with ground truth and select the best trajectory based on the scores from the evaluation, which is not aligned with the evaluation protocol defined in NAVSIM [1, 6].

We note that the goal of DrivoR is to provide a simple and efficient baseline model with registers for end-to-end driving. Without bells and whistles, DrivoR still achieves competitive performance compared to methods with post-processing techniques or additional sensor modality.

## H. NAVSIM-v2 results before benchmark fix

We provide in Tab. 14 the full NAVSIM-v2 state-of-the-art comparison, including the results of methods before the benchmark bug fix (Issue #151 in NAVSIM official GitHub), which is related to the failure of filtering out human driver errors.

From the table, we see that DrivoR ranks among the best-performing methods, with a much lighter model design (see more detail in App. E) and without extra training data. Also, DrivoR ranks first among state-of-the-art methods after the bug fix. This indicates that the errors made by DrivoR are mostly due to human errors, which are not penalized after the fix according to the NAVSIM-v2 benchmark [1].

## I. Limitations, Visualizations, and Failure Case Analysis

Fig. 8, Fig. 9, and Fig. 10 show DrivoR navigating diverse scenarios, and show the camera focus between the scoring and prediction pathways.

Fig. 9 shows a surprising case where the scoring attention is mainly focused on the back camera, despite the visible traffic light. We hypothesize that adding traffic light compliance as a score component, as in the EPDMS, could force the scoring modules attention to focus more on the traffic light as it crosses the intersection.

In order to further explore the capabilities of the learned representations, we conduct a simple linear probing experiment on top of the sensor registers. We label images in the `navtrain` dataset according to the presence of a vehicle or pedestrian within a critical distance (roughly 10 meters). We then produce a second per-scene label depending on the presence of a Red Light in the scene. We train a linear layer for a single epoch on top of sensor registers or pooled image features using the models from Table 4 (Ablations of the perception). Tab. 15 shows the results of this probing experiment. We can see that the registers remain comparable to pooled image features in terms of information retrieval. However Red Light detection remains low, highlighting a failure of the model. An interesting note is that Red Light presence is defined only by visibility in the front camera. Positive classifications using features of non-frontal cameras represents information leakage or the learning of some spurious correlation in the data, an avenue for further analysis.

Fig. 10 shows a failure case, highlighting the challenge of predicting viable trajectories without historical (past) frames. DrivoR operates only on the current timestep, which makes examples like Fig. 10 challenging, due to the

| Method | | Stage 1 | | | | | | | | | Stage 2 | | | | | | | | | EPDMS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | NC | DAC | DDC | TLC | EP | TTC | LK | HC | EC | NC | DAC | DDC | TLC | EP | TTC | LK | HC | EC | |
| *Results before official metric bug fixing.* | | | | | | | | | | | | | | | | | | | | |
| PDM-Closed | [5] | 94.4 | 98.8 | 100 | 99.5 | 100 | 93.5 | 99.3 | 97.7 | 36.0 | 88.1 | 90.6 | 96.3 | 98.5 | 100 | 83.1 | 73.7 | 91.5 | 25.4 | 51.3 |
| Const. Vel. | [6] | 88.8 | 42.8 | 70.6 | 99.3 | 77.5 | 87.3 | 78.6 | 97.1 | 60.4 | 83.2 | 59.1 | 76.5 | 98.0 | 71.3 | 81.1 | 47.9 | 97.1 | 61.9 | 10.9 |
| Ego Hist. MLP | [6] | 93.2 | 55.7 | 86.6 | 99.3 | 81.2 | 92.2 | 83.5 | 97.5 | 77.7 | 77.2 | 51.9 | 74.4 | 98.2 | 77.1 | 75.0 | 40.8 | 97.8 | 79.8 | 12.7 |
| LTF | [4] | 96.2 | 79.6 | 99.1 | 99.6 | 84.1 | 95.1 | 94.2 | 97.6 | 79.1 | 77.8 | 70.2 | 84.3 | 98.1 | 85.1 | 75.7 | 45.4 | 95.8 | 76.0 | 23.1 |
| RAP-DINO (ViT-H) † | [7] | 97.1 | 94.4 | 98.8 | 99.8 | 83.9 | 96.9 | 94.7 | 96.4 | 66.2 | 83.2 | 83.9 | 87.4 | 98.0 | 86.9 | 80.4 | 52.3 | 95.2 | 52.4 | 36.9 |
| GTRS-D (V2-99) | [19] | 98.7 | 91.4 | 95.8 | 89.2 | 99.4 | 94.4 | 99.3 | 98.8 | 72.8 | 69.5 | 98.7 | 90.1 | 95.1 | 54.6 | 96.9 | 94.1 | 40.4 | 49.7 | 41.7 |
| GTRS-A (V2-99) | [19] | 98.9 | 87.9 | 95.1 | 88.8 | 99.2 | 89.6 | 99.6 | 98.8 | 76.1 | 80.3 | 99.1 | 86.0 | 94.7 | 53.5 | 97.6 | 97.1 | 54.2 | 56.1 | 42.1 |
| GTRS-DrivoR (ViT-S)* | | 98.0 | 95.8 | 99.7 | 99.3 | 72.9 | 98.2 | 95.6 | 96.9 | 51.6 | 91.6 | 86.7 | 90.2 | 98.8 | 73.2 | 88.9 | 51.9 | 94.9 | 46.4 | 42.3 |
| GTRS-D EVA-ViT-L | [19] | 97.6 | 95.8 | 99.8 | 99.0 | 77.2 | 97.8 | 95.3 | 97.3 | 46.7 | 91.9 | 91.3 | 92.7 | 99.0 | 72.7 | 90.4 | 53.8 | 94.1 | 41.6 | 43.4 |
| GTRS-A (ViT-L) | [19] | 98.7 | 98.0 | 99.1 | 99.8 | 75.9 | 98.7 | 94.7 | 97.6 | 49.8 | 89.5 | 89.6 | 92.9 | 98.5 | 78.9 | 86.4 | 55.3 | 96.5 | 52.7 | 44.7 |
| DriveSuprim (EVA-ViT-L) | [30] | 98.7 | 98.0 | 99.1 | 99.8 | 75.9 | 98.7 | 94.7 | 97.6 | 49.8 | 89.5 | 89.6 | 92.9 | 98.5 | 78.9 | 86.4 | 55.3 | 96.5 | 52.7 | 44.7 |
| GTRS-D (ViT-L) | [19] | 98.9 | 98.2 | 99.8 | 99.6 | 73.9 | 98.9 | 95.3 | 97.3 | 40.0 | 91.5 | 90.8 | 94.7 | 98.5 | 70.8 | 90.1 | 55.4 | 97.2 | 54.2 | 45.3 |
| DrivoR (ViT-S) | | 98.8 | 95.1 | 98.9 | 100 | 72.6 | 98.7 | 94.0 | 97.6 | 73.3 | 90.2 | 88.4 | 91.9 | 98.6 | 69.8 | 88.0 | 50.1 | 98.5 | 76.2 | 45.3 |
| ZTRS (V2-99) | [18] | 98.9 | 97.6 | 100.0 | 100.0 | 66.7 | 98.9 | 96.2 | 96.7 | 44.0 | 91.1 | 90.4 | 95.8 | 99.0 | 63.6 | 89.8 | 60.4 | 97.6 | 66.1 | 45.5 |
| *Results after official metric bug fixing.* | | | | | | | | | | | | | | | | | | | | |
| RAP-DINO (ViT-H)† | [7] | 97.1 | 94.4 | 98.8 | 99.8 | 83.9 | 96.9 | 94.7 | 96.4 | 66.2 | 83.2 | 83.9 | 87.4 | 98.0 | 86.9 | 80.4 | 52.3 | 95.2 | 52.4 | 39.6 |
| GTRS-D (V2-99) | [19] | 98.9 | 96.2 | 99.4 | 99.3 | 72.9 | 98.9 | 95.1 | 96.9 | 39.1 | 91.2 | 89.4 | 94.4 | 98.8 | 69.5 | 90.0 | 54.3 | 94.0 | 48.7 | 45.0 |
| GTRS-A (V2-99) | [19] | 98.9 | 95.1 | 99.1 | 99.6 | 76.2 | 99.1 | 94.9 | 97.6 | 54.2 | 88.1 | 88.8 | 89.3 | 98.9 | 98.9 | 85.9 | 53.7 | 96.8 | 56.9 | 45.4 |
| GTRS-DrivoR (ViT-S)* | | 98.0 | 95.8 | 99.7 | 99.3 | 72.9 | 98.2 | 95.6 | 96.9 | 51.6 | 91.6 | 86.7 | 90.2 | 98.8 | 73.2 | 88.9 | 51.9 | 94.9 | 46.4 | 45.8 |
| ZTRS (V2-99) | [18] | 98.9 | 97.6 | 100 | 100 | 66.7 | 98.9 | 96.2 | 96.7 | 44.0 | 91.1 | 90.4 | 95.8 | 99.0 | 63.6 | 89.8 | 60.4 | 97.6 | 66.1 | 48.1 |
| DrivoR (ViT-S) | | 98.8 | 95.1 | 98.9 | 100 | 72.6 | 98.7 | 94.0 | 97.6 | 73.3 | 90.2 | 88.4 | 91.9 | 98.6 | 70.0 | 88.0 | 50.1 | 98.5 | 76.2 | 48.3 |
| DrivoR (+65k SimScale data, ViT-S) | | 98.9 | 97.3 | 99.2 | 99.6 | 77.7 | 99.1 | 95.3 | 97.6 | 68.4 | 92.3 | 92.2 | 97.0 | 99.0 | 72.1 | 90.3 | 56.3 | 97.1 | 38.8 | 52.3 |
| SimScale (+185k SimScale data, V2-99) | [25] | 99.6 | 99.16 | 99.9 | 100 | 69.6 | 99.6 | 95.8 | 95.6 | 28.4 | 94.5 | 94.2 | 95.8 | 99.2 | 75.8 | 92.8 | 60.1 | 96.1 | 43.2 | 53.2 |
| DrivoR (+134k SimScale data, ViT-S) | | 99.1 | 98.2 | 99.3 | 99.8 | 75.4 | 98.7 | 94.9 | 97.6 | 70.2 | 92.3 | 91.6 | 97.3 | 99.1 | 75.7 | 90.6 | 56.1 | 98.4 | 44.7 | **54.6** |

†: RAP [7] is trained on a dataset that is 10× larger than navtrain (the default training set).

*: same ViT-S backbone with DrivoR registers, the prediction and scoring heads remain the same as in GTRS.

Table 14. **NAVSIM-v2 `navhard-two-stage`**. Full comparison to other methods on the NAVSIM-v2 benchmark test set using the EPDMS. *GTRS-A* refers to *GTRS-Aug* and *GTRS-D* for *GTRS-Dense*.

| Method | Camera | Vehicle | Ped. | Red Light |
|---|---|---|---|---|
| Registers | Front | 96.5 | 98.6 | 83.3 |
| Registers | Back | 86.6 | 98.8 | 68.9 |
| Registers | Left | 97.2 | 98.1 | 80.1 |
| Registers | Right | 97.0 | 94.9 | 76.0 |
| Pooling | Front | 98.0 | 98.7 | 73.0 |
| Pooling | Back | 96.2 | 99.0 | 77.0 |
| Pooling | Left | 97.0 | 98.1 | 76.8 |
| Pooling | Right | 97.9 | 96.1 | 69.3 |

Table 15. Linear probing of Sensor Register and Pooled Image Features. Metric is classification accuracy on presence of critical objects within 10 meters, or on scene-level Red Light presence.

ambiguous nature of the vehicle's position in the scene. The front camera does not contain easily interpretable objects, and we see the model focusing entirely on the right camera. The resulting trajectories undercut the turn, and could result in the model driving in the wrong way. Future work could include the use of historical frames for planning, potentially alleviating these ambiguous failure cases.

Fig. 11 compares the predicted and selected trajectories of the two models on the same scene from the NAVSIM-v1 validation set. We can see that the trajectories of our default trained NAVSIM-v1 agent are more aggressive, traveling faster and with less spread than the trajectories from our agent tuned to have more passive behavior. The "passive" agent is however better able to navigate the out-of-distribution (OOD) scenes in NAVSIM-v2.

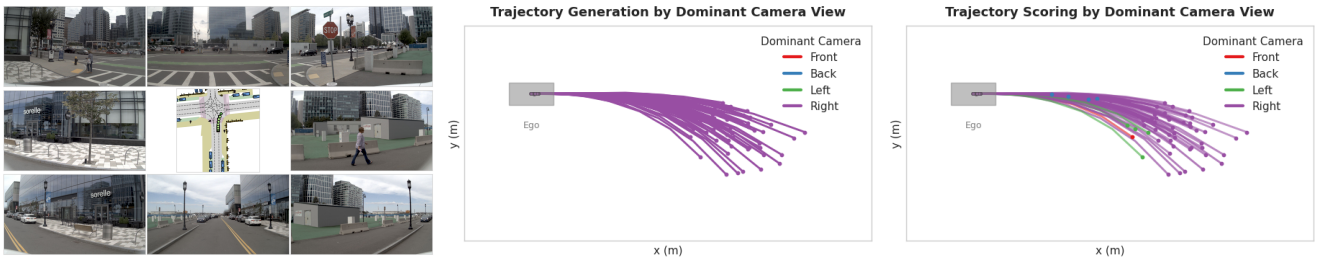



Figure 8. **Visualization of generated trajectories (right turn) in a NAVSIM-v1 val set scene.** Right turn scenario with right camera as dominant camera used in trajectory scoring and generation.

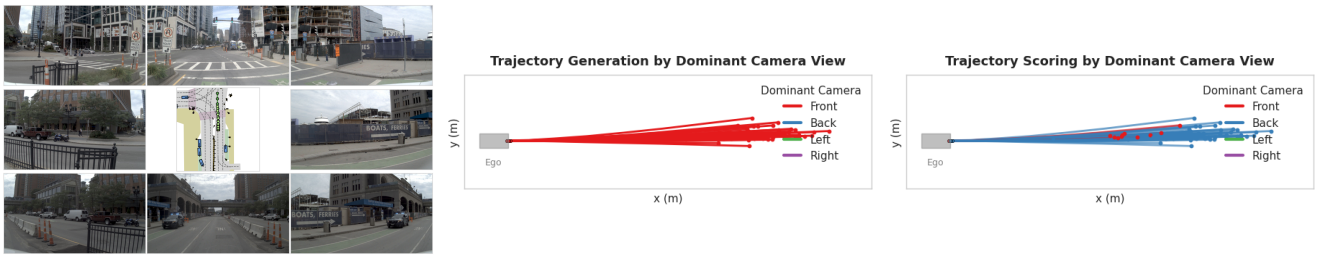



Figure 9. **Visualization of generated trajectories (intersection crossing) in a NAVSIM-v1 val set scene.** The ego-vehicle crosses a signalized intersection, but the scoring attention focuses on the rear camera. We hypothesize that the inclusion of the Traffic Light Compliance (TLC) scoring could help in such a case. As a matter of fact, our TLC score on NAVSIM-v2 `navhard-two-stage` are high (Tab. 14).

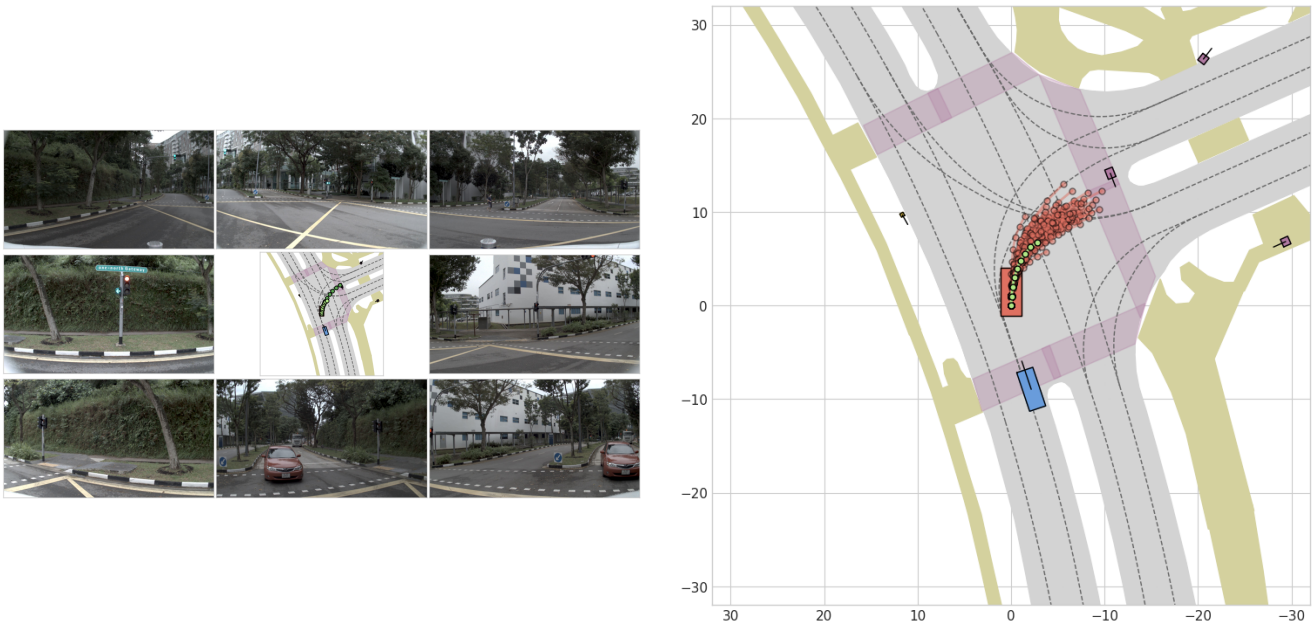



Figure 10. **Visualization of generated trajectories (failure case) in a NAVSIM-v1 val set scene.** Failure case highlighting difficulty of navigation without historical (past) frames. The front camera image is very ambiguous, resulting in full focus on the right camera and trajectories which could result in wrong-way driving.

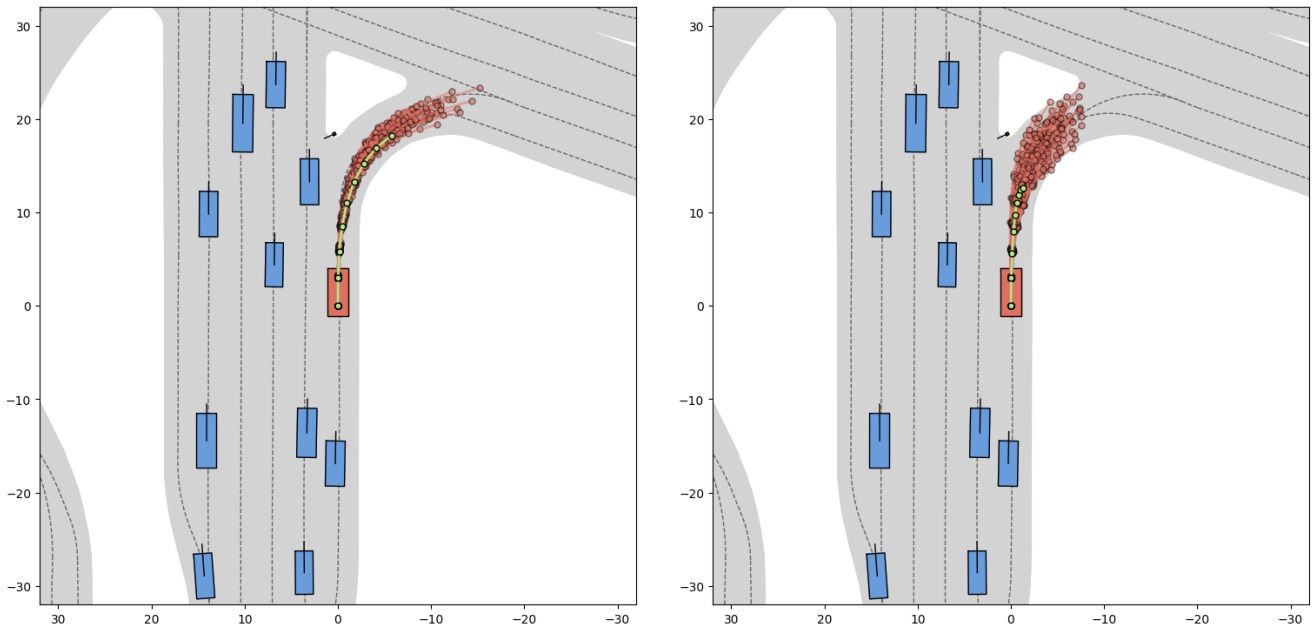



Figure 11. **Visualization of generated trajectories (agent behavior re-weighting) in a NAVSIM-v1 val set scene.**. Left: trajectories are generated from an agent using default PDMS weights on score components. Right: the agent uses our tuned weights. Note the much shorter and less aggressive trajectories of the agent in the second (right) setting, which are better suited for navigating the out-of-distribution (OOD) scenes in the NAVSIM-v2 evaluation.

## References


[1] Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *CoRL*, 2025. 3

[2] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024. 3

[3] Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In *ICCV*, 2025. 3

[4] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *TPAMI*, 2022. 3, 4

[5] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *CoRL*, 2023. 3, 4

[6] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *NeurIPS*, 2024. 1, 3, 4

[7] Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. *arXiv preprint arXiv:2510.04333*, 2025. 3, 4

[8] Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. ipad: Iterative proposal-centric end-to-end autonomous driving. *arXiv preprint arXiv:2505.15111*, 2025. 2, 3

[9] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *CVPR*, 2023. 3

[10] Xuefeng Jiang, Yuan Ma, Pengxiang Li, Leimeng Xu, Xin Wen, Kun Zhan, Zhongpu Xia, Peng Jia, XianPeng Lang, and Sheng Sun. Transdiffuser: End-to-end trajectory generation with decorrelated multi-modal representation for autonomous driving. *arXiv preprint arXiv:2505.09315*, 2025. 3

[11] Siwen Jiao, Kangan Qian, Hao Ye, Yang Zhong, Ziang Luo, Sicong Jiang, Zilin Huang, Yangyi Fang, Jinyu Miao, Zheng Fu, et al. Evadrive: Evolutionary adversarial policy optimization for end-to-end autonomous driving. *arXiv preprint arXiv:2508.09158*, 2025. 3

[12] Derun Li, Changye Li, Yue Wang, Jianwei Ren, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Peng Jia, Xianpeng Lang, Ningyi Xu, and Hang Zhao. Learning personalized driving styles via reinforcement learning from human feedback, 2025. 3

[13] Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv e-prints*, pages arXiv–2503, 2025. 3

[14] Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. *arXiv preprint arXiv:2510.12796*, 2025. 3

[15] Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. *ICCV*, 2025. 3

[16] Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025. 3

[17] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024. 3

[18] Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Jingde Chen, Nadine Chang, Maying Shen, Jingyu Song, Zuxuan Wu, Shiyi Lan, et al. Ztrs: Zero-imitation end-to-end autonomous driving with trajectory scoring. *arXiv preprint arXiv:2510.24108*, 2025. 4

[19] Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025. 4

[20] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *CVPR*, pages 12037–12047, 2025. 3

[21] Haochen Liu, Tianyu Li, Haohan Yang, Li Chen, Caojun Wang, Ke Guo, Haochen Tian, Hongchen Li, Hongyang Li, and Chen Lv. Reinforced refinement with self-aware expansion for end-to-end autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. 3

[22] Chen Shi, Shaoshuai Shi, Kehua Sheng, Bo Zhang, and Li Jiang. Drivex: Omni scene modeling for learning generalizable world knowledge in autonomous driving. In *ICCV*, 2025. 3

[23] Chonghao Sima, Kashyap Chitta, Zhiding Yu, Shiyi Lan, Ping Luo, Andreas Geiger, Hongyang Li, and Jose M Alvarez. Centaur: Robust end-to-end autonomous driving with test-time training. *arXiv preprint arXiv:2503.11650*, 2025. 3

[24] Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Breaking imitation bottlenecks: Reinforced diffusion powers diverse trajectory generation. *arXiv preprint arXiv:2507.04049*, 2025. 3

[25] Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, Hangjun Ye, Tieniu Tan, Long Chen, and Hongyang Li. Simscale: Learning to drive via real-world

simulation at scale. *arXiv preprint arXiv:2511.23369*, 2025. 4

[26] Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025. 3

[27] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*, 2024. 3

[28] Maciej K Wozniak, Lianhang Liu, Yixi Cai, and Patric Jensfelt. Prix: Learning to plan from raw pixels for end-to-end autonomous driving. *arXiv preprint arXiv:2507.17596*, 2025. 3

[29] Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *CVPR*, 2025. 3

[30] Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M Alvarez, and Zuxuan Wu. Drivesuprim: Towards precise trajectory selection for end-to-end planning. *arXiv preprint arXiv:2506.06659*, 2025. 3, 4

[31] Rui Yu, Xianghang Zhang, Runkai Zhao, Huaicheng Yan, and Meng Wang. Distilldrive: End-to-end multi-mode autonomous driving distillation by isomorphic hetero-source planning model. *arXiv preprint arXiv:2508.05402*, 2025. 3

[32] Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. In *ISRR*, 2024. 3

[33] Bozhou Zhang, Nan Song, Jingyu Li, Xiatian Zhu, Jiankang Deng, and Li Zhang. Future-aware end-to-end driving: Bidirectional modeling of trajectory planning and scene evolution. *NeurIPS*, 2025. 3

[34] Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, et al. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. In *ICCV*, 2025. 3

[35] Zhiyu Zheng, Shaoyu Chen, Haoran Yin, Xinbang Zhang, Jialv Zou, Xinggang Wang, Qian Zhang, and Lefei Zhang. Resad: Normalized residual trajectory modeling for end-to-end autonomous driving. *arXiv preprint arXiv:2510.08562*, 2025. 3

[36] Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *NeurIPS*, 2025. 3