Supplementary Material for LEAD: Minimizing Learner-Expert Asymmetry in End-to-End Driving

Long Nguyen[1,3] Micha Fauth[1] Bernhard Jaeger[1,3] Daniel Dauner[1,3]
Maximilian Igl[2] Andreas Geiger[1,3] Kashyap Chitta[2]
[1]University of Tübingen, Tübingen AI Center [2]NVIDIA Research [3]KE:SAI

Abstract: This supplementary document provides an extended description of the LEAD dataset, the TFv6 learner, and the experimental pipeline used in the main paper. It includes detailed definitions of the evaluation metrics for all benchmarks studied, descriptions of the sensor configurations and expert-policy components, and the occlusion- and intent-alignment procedures applied during data collection. Additional implementation details are provided for the extended route set, newly introduced scenarios, perception-label updates, sensor perturbations, and controller adjustments. The document also outlines the co-training setup for combining synthetic LEAD samples with real-world datasets such as NAVSIM and Waymo, together with the filtering process used to assemble targeted subsets of CARLA data. The supplementary videos can be found on https://ln2697.github.io/lead and offer additional qualitative demonstrations of the model's driving behavior in CARLA. For improved interpretability, we present both a top-down BEV visualization and a cinematic third-person view, as the raw input camera streams provide only limited scene coverage.

A. Benchmark Metrics

The following section discusses metrics used from every benchmark in detail.

A.1. CARLA Leaderboard 2.0

The Leaderboard uses the official CARLA metrics: Driving Score, Route Completion, and Infraction Score. Each metric is calculated independently for each route. After all routes are completed, the final metrics are derived by taking the arithmetic mean of the metrics across all routes. The overall driving score, calculated using the global values, is the primary metric for ranking methods.

Driving Score. The primary evaluation criterion is the Driving Score, denoted as:

$$\mathrm{DS}_i = \mathrm{RC}_i \cdot \mathrm{IS}_i,$$

where RC_i represents the percentage of the i-th route completed, and IS_i is a penalty factor accounting for infractions incurred during the route.

Route Completion. This metric quantifies the proportion of the route completed by the agent, expressed as a percentage.

Infraction Penalty. The penalty due to infractions, IS_i, is calculated as a product of all infractions:

$$\mathrm{IS}_i = \prod_{j=1}^{N_I} (p_j)^{\#\mathrm{infractions}_j},$$

where p_j denotes the penalty coefficient for the j-th type of infraction out of a total of N_I infraction types. $\#\mathrm{infractions}_j$ is the number of times this infraction occurred. The calculation begins with a base score of 1.0, which decreases with each infraction.

Infractions are categorized by severity, each associated with a penalty coefficient that reduces the driving score. Key infractions include:

- Collisions with pedestrians: $p_j = 0.50$.
- Collisions with vehicles: $p_j = 0.60$.
- Collisions with static objects: $p_j = 0.65$.
- Running a red light: $p_j = 0.70$.
- Ignoring a stop sign: $p_j = 0.80$.
- Failure to yield to emergency vehicles: $p_j = 0.70$.
- Failure to maintain minimum speed: Up to $p_j = 0.70$.
- Off-road driving: reduces route completion score proportionally.

When one of the following events occurs, the route stops immediately:

- Route deviation (more than 30 meters off route).
- Blocked agent (more than 180 simulation seconds without action).
- Communication timeout (more than 60 seconds).
- Route timeout (exceeding allowed simulation time).

Leaderboard 2.0 metric discussion. The Driving Score is calculated in a way that it can be advantageous not to

complete the whole route. This is the case if the infractions incurred during a segment of the route reduce the driving score more than the potential gain from continuing the route. In this case, stopping early to avoid further penalties leads to an overall higher driving score. This tradeoff only occurs for long routes. [68]

For completeness, we emphasize that our evaluation does not rely on this effect: all reported results try to follow the full routes without using early stopping.

A.2. Bench2Drive

Driving Score. The Driving Score is calculated similarly to the Leaderboard 2.0. The only difference to the original Driving Score is that the penalty "Failure to maintain minimum speed" is ignored.

Success Rate. The Success Rate measures the percentage of completed routes without any infractions (ignoring the minimum speed penalty).

Efficiency. This uses the ratio of the ego vehicle speed to the speed of the surrounding actors. Since there was no penalty in Leaderboard 1.0 for low speeds, most models used a very low speed, which makes driving and reacting to other dynamic actors much easier. The higher this efficiency metric, the faster the model drives, making the driving task harder.

Comfortness. The comfort metric considers the jerk magnitude, lateral and longitudinal accelerations, yaw acceleration, longitudinal jerk, and yaw rate.

Table 16 and Table 17 compare TFv6 against all existing baselines on Bench2Drive at the time of writing using the numbers reported in the respective publications. TFv6 outperforms all baselines by a wide margin in terms of success rate and driving score. TFv6 even outperforms the privileged method Think2Drive by 3 DS and 1 SR.

Method	**Efficiency** ↑	**Comfortness** ↑
HiP-AD [51]	203	19
BridgeDrive [35]	238	17
R2SE [32]	245	23
SimLingo [42]	**259**	**33**
TFv6 (Ours)	210	22

Table 1. **Efficiency and Comfortness on Bench2Drive.**

Table 1 reports the Efficiency and Comfortness scores, providing an additional view on driving speed and motion smoothness beyond the standard Bench2Drive metrics.

In addition to aggregated metrics, Bench2Drive provides a fine-grained analysis of driving capabilities by evaluating five key urban driving abilities: Merging, Overtaking, Emergency Brake, Give Way, and Traffic Sign handling. Each ability is measured over a curated subset of short routes designed to isolate a specific driving skill under diverse towns and weather conditions. Table 2 summarizes the performance across these abilities.

A.3. NAVSIM v1

NAVSIM v1 evaluates planners using the Predictive Driver Model Score (PDMS), which combines multiplicative penalties with a weighted average of progress, safety, and comfort metrics. All subscores lie in $[0, 1]$ and are computed after a 4-second non-reactive rollout.

- *NC* returns 1 if no collision occurs, 0.5 if ambiguous, 0 if at-fault.
- *DAC* checks whether the ego stays within the drivable area.
- *TTC* binary score for maintaining safe time-to-collision.
- *EP* normalized forward progress over the horizon.
- *C* smoothness of accelerations, jerk, and yaw motion.

The PDMS used in NAVSIM v1 is:

$$\text{PDMS} = \left(\prod_{m \in \{\text{NC,DAC}\}} m \right) \cdot \left(\frac{5\,\text{TTC} + 5\,\text{EP} + 2\,\text{C}}{5 + 5 + 2} \right)$$

A.4. NAVSIM v2

NAVSIM v2 extends the NAVSIM v1 PDMS metric to the *Extended Predictive Driver Model Score* (EPDMS). The new metric broadens the set of behaviors covered by evaluation, adds explicit lane-keeping and extended comfort terms, and introduces a mechanism to filter out penalties that are also caused by the human reference driver.

Compared to NAVSIM v1, EPDMS adds two new multiplier metrics and three weighted metrics:

- *DDC* penalizes driving against the allowed flow of traffic; strong violations reduce the score more than minor ones. The weight is multiplicative in the range $\{0, 0.5, 1\}$.
- *TLC* penalizes running red lights or otherwise violating traffic light rules. The weight is multiplicative in the range $\{0, 1\}$.
- *LK* penalizes driving far from the lane centerline for extended periods; it is disabled in intersections where map annotations are unreliable.
- *HC* evaluates how well the planned trajectory matches the recent motion history of the vehicle, discouraging abrupt changes in behavior.
- *EC* compares subsequent planned trajectories and their dynamic states; large changes in acceleration, jerk, or yaw between frames are treated as uncomfortable.

Weights are set to 2 for LK, HC, and EC. In addition, for the multiplicative metrics, a filtering step is applied so that a penalty is only activated when the planner itself causes the violation; if the human reference already violates the rule, the metric is treated as satisfied, and no penalty is applied.

Method	Merg. ↑	Overtak. ↑	Emer. Brake ↑	Give Way ↑	Traf. Sign ↑	Mean ↑
TFv5 [68]	58.75	57.77	83.33	40.00	82.11	64.39
HiP-AD [51]	50.00	84.44	83.33	40.00	72.10	65.98
SimLingo [42]	54.01	57.04	88.33	**53.33**	82.45	67.03
R2SE [32]	53.33	61.25	90.00	50.00	84.21	67.76
BridgeDrive [35]	63.50	58.89	88.34	50.00	88.95	69.93
TFv6 (Ours)	**72.50**	**97.77**	**91.66**	40.00	**89.47**	**78.28**

Table 2. **Multi ability scores on Bench2Drive evaluation protocol.**

Pseudo closed-loop aggregation. NAVSIM v2 further refines evaluation by approximating closed-loop behavior in an open-loop setting:

1. *First stage.* Starting from an initial logged scene, the planner is evaluated for 4 seconds using EPDMS, following the same non-reactive simulation procedure as in NAVSIM v1.
2. *Second stage.* Several follow-up scenes corresponding to different 4-second plans are pre-generated. Each follow-up scene starts from a perturbed ego state (e.g., lateral offset or different speed). The submitted planner is evaluated on each follow-up scene with EPDMS.
3. *Aggregation.* Follow-up scores are combined using a Gaussian kernel over the distance between the planner's actual end state in the first stage and the start state of each follow-up scene, giving higher weight to more consistent continuations. The final NAVSIM v2 score is obtained by multiplying the first-stage EPDMS by this weighted second-stage aggregate.

A.5. Waymo E2E

The Waymo End-to-End Challenge uses the *Rater Feedback Score* (RFS) as the primary ranking metric at horizons $3\,$s and $5\,$s, averaged over 11 scenario categories. As a secondary metric, the standard *Average Distance Error* (ADE) is reported between the predicted trajectory and the highest-rated rater trajectory.

For each scenario, three rater-specified trajectories are provided, each with an associated driving quality score $s \in [0, 10]$, where 10 denotes excellent driving. Given a model prediction, the evaluation first finds the closest rater trajectory using the adjusted miss-distance defined by the trust-region procedure used in the Motion Prediction and Interaction Prediction Challenges [1].

A trust region is defined around a rater-specified trajectory using lateral and longitudinal thresholds at time $T \in \{3, 5\}$ seconds:

$$\tilde{\eta}_{\text{lat}}(3) = 1.0, \qquad \tilde{\eta}_{\text{lat}}(5) = 1.8,$$

$$\tilde{\eta}_{\text{lng}}(3) = 4.0, \qquad \tilde{\eta}_{\text{lng}}(5) = 7.2.$$

Speed scaling. These thresholds are scaled by the initial speed v (m/s) of the rater trajectory using

$$\text{scale}(v) = \begin{cases} 0.5, & v < 1.4, \\ 0.5 + 0.5\dfrac{v - 1.4}{11 - 1.4}, & 1.4 \le v < 11, \\ 1, & v \ge 11. \end{cases}$$

The final trust-region thresholds are

$$\eta_{\text{lat}}(T, v) = \text{scale}(v)\,\tilde{\eta}_{\text{lat}}(T) \quad \eta_{\text{lng}}(T, v) = \text{scale}(v)\,\tilde{\eta}_{\text{lng}}(T).$$

Scoring within trust regions. If the predicted trajectory lies inside any of the trust regions of a rater-specified trajectory, it receives that rater's score.

Scoring outside trust regions. If the prediction lies outside all trust regions, it receives an exponentially decayed score based on the distance Δ to the closest rater trajectory:

$$\text{rater\_feedback\_score} = s \times 0.1^{\max\left(\max\left(\frac{\Delta_{\text{lng}}}{\eta_{\text{lng}}}, \frac{\Delta_{\text{lat}}}{\eta_{\text{lat}}}\right) - 1,\, 0\right)}.$$

A floor score of 4 is assigned to all predictions outside the trust regions.

B. Baseline Expert Overview

Our baseline expert policy consists of three sequential stages, first two decide lateral driving controls and the last stage is responsible for longitudinal control decision. Further heuristics use simulator's internal states to solve specific corner cases

Dense geometric trajectory. The expert first generates a dense reference path using an A* planner on the road graph. The resulting trajectory is sampled at $0.1\,$m resolution and provides a smooth geometric route that can also serve as a baseline motion plan. This path, however, does not account for static obstacles that are absent from the HD map (e.g., temporary roadblocks, parked vehicles).

Obstacle avoidance refinement. When static obstacles intersect or occlude the geometric path, the expert locally adjusts the trajectory to maintain progress. This includes short-range maneuvers such as lane changes or lateral deviations to overtake stationary obstacles while still respecting roadway topology.

Target-speed proposal. Given the refined trajectory, the expert proposes a target speed based on three factors: (i) the speed limit along the current road segment, (ii) the Intelligent Driver Model for maintaining safe distances to dynamic agents, and (iii) predicted bounding box intersections to prevent imminent collisions. Two PID controllers then track the route and the target speed to produce the final steering and throttle/brake commands.

C. State Alignment

Leveraging new occlusion-measurement tooling, we adjust how the expert regulates velocity so that its behavior matches what can realistically be inferred from the sensor suite. All modifications operate exclusively on the *third stage* of LEAD (i.e., the target-speed proposal), leaving the geometric route and obstacle-aware trajectory augmentation unchanged.

C.1. Visibility Alignment

We adjust the expert policy in several ways to produce sensor-grounded demonstrations:

- **Traffic-light behavior.** We refine the expert's traffic-light handling to better match camera visibility. Previously, certain traffic-light layouts (e.g., overhead or region-specific designs) sometimes caused the expert to stop unrealistically close to the stop line. We explicitly identify these cases and increase the stopping distance, while enforcing stronger deceleration when approaching the signal to produce smoother, more sensor-consistent braking behavior.
- **Weather-adaptive behavior.** The previous expert used a fixed target speed across all weather presets. We introduce a weather-aware controller that reduces speed under rain, fog, or low-light conditions, reflecting decreased visibility and traction. This produces more realistic demonstrations and avoids instructing the learner to drive too aggressively in poor weather.
- **Occlusion-aware junction entry.** Using the depth-aligned camera point cloud, we estimate an occlusion score for each camera based on the density and distance of visible structure. When approaching a junction with high occlusion, the expert slows down preemptively. This behavior teaches the learner to treat uncertain junctions as inherently hazardous, without requiring privileged map knowledge.
- **Narrow urban streets.** Dense rows of parked vehicles create environments with severely reduced visibility. The expert detects these situations by evaluating the concentration of static obstacles and lowers its target speed accordingly.
- **Speed Limits Computation.** We use CARLA's native speed-limit API and the second-highest speed among nearby dynamic actors. The minimum of these two values is used as the effective speed limit, preventing unrealistic acceleration while preserving natural traffic flow.

In practice, these reductions are small and cumulative: the maximum total decrease in target speed is only 1.75m/s. Since the affected situations are almost exclusively low-speed urban segments with a speed limit of 30 km/h, the expert still drives at least 23.7km/h at its peak. The adjustments, therefore, nudge the expert toward more realistic, visibility-aware driving rather than turning it into an overly conservative controller. The latter is also penalized explicitly by the metrics.

From a practical standpoint, these adjustments are also consistent with real driving behavior. Even if the speed limit is 30km/h, no human driver maintains that speed when entering a visually constrained situation such as a narrow urban street, a row of parked cars, or a partially occluded junction. Slowing down in these cases is a basic safety principle: reduced visibility naturally demands reduced speed. Therefore, the expert behaves more like a real, visibility-limited driver rather than an omniscient planner following nominal speed limits under all circumstances.

On the implementation side, visibility in each camera is summarized by a single scalar occlusion score. We first keep only points that are within a reasonable height and distance range and belong to relevant semantic classes. From these points, we measure their distances to the ego and take a high quantile q to estimate how far the camera can "see." Let $\mathcal{P}$ be all valid points and $\mathcal{P}_{\text{vis}}$ the filtered subset. The score is

$$s_{\text{occ}} = \frac{q}{d_{\max}} \cdot \frac{|\mathcal{P}_{\text{vis}}|}{|\mathcal{P}|},$$

which lies in $[0, 1]$ and decreases when the view becomes more occluded.

C.2. Uncertainty Alignment: Pedestrian Interactions

Previous implementations relied on bounding-box forecasting and collision prediction, which often resulted in late stops, if at all, and therefore ambiguous cues for the learner. We replace this behavior with a direct, visibility-based criterion derived from the camera point cloud:

- the expert stops when a pedestrian reaches a minimum number of visible pixels.
- a motion threshold ensures stationary pedestrians do not trigger unnecessary braking.

This yields early, unambiguous stopping decisions grounded entirely in observable visual evidence, which aligns well with the learner's perception module.

C.3. Uncertainty Alignment: Emergency Vehicle Interactions

Emergency vehicles can approach junctions at high speed and cause irreversible collisions when the ego reacts too slowly. If the ego hesitates or yields too late, the resulting collision typically ends the episode almost immediately, an effect that becomes far more pronounced on long routes. The previous expert avoided such failures using privileged full-state access and often resolved the situation by slightly slowing down until the priority vehicle passed.

To provide clearer demonstrations:

- when an emergency vehicle is detected with sufficiently strong visual evidence (enough number of pixels/radar points), the expert issues an immediate full stop;
- driving only resumes once the hazard is completely cleared.

C.4. Uncertainty Alignment: Miscellaneous

In unprotected turn scenarios, precise velocity and acceleration estimates of oncoming actors play a critical role in collision prediction. However, these quantities are difficult to infer reliably from visual observations alone, especially at long distances or under partial occlusion. To reduce the expert's reliance on such privileged motion signals, we deliberately make collision checking more conservative in these situations.

Specifically, when the ego vehicle approaches or executes an unprotected turn at a junction, we uniformly enlarge the 3D bounding boxes of potentially conflicting actors by a fixed factor of 2 along all spatial dimensions (length, width, and height). This enlargement increases the spatial margin used during collision checks, causing braking decisions to be triggered based on proximity and visibility rather than on precise motion forecasts.

D. Intent Alignment

The navigation interface maintains a queue of sparse target points. A target point is *popped* from this queue once the ego comes within a fixed radius. This radius is the *pop distance*. It simply determines when the current target point is considered "visited" and the next one becomes active.

Single target point. With only one target point as conditioning, we found that the default pop distance of 7.5m is empirically optimal. A larger radius causes premature popping and weakens directional cues, while a smaller radius makes the target point appear too late for smooth planning. At one-point conditioning, 7.5m gives the cleanest and most stable supervision.

Multiple target points. When introducing additional target points (previous, current, next), the pop distance becomes a critical factor for whether the *next* point carries meaningful information. At the default 7.5m, the current target point is popped too early: in most frames, the next point appears far ahead and therefore adds little to no contextual structure. Practically, the model behaves as if only a single target point were provided.

Scheduling pop distance. To expose the learner to usable local geometry from the next target point, we reduce the pop distance to 3m during training. This delays the popping of the current target point and increases the number of frames in which the next point lies close enough to influence local decision-making. With this adjustment, multi-point conditioning begins to provide consistent benefits instead of collapsing back to the single-point regime.

In closed-loop evaluation, we tune the pop distance separately. A value of 5m provides the best performance empirically. In addition, we further normalize the target points by constants $(x, y) = (200, 50)$ before tokenizing them with a shared linear layer.

E. Extending CARLA Dataset

After focusing on intent and visual alignment, we also focused on enhancing the dataset's quality in general. Table 3 summarizes the differences between the two datasets.

Original Driving Scenarios CARLA Leaderboard 2.0. In each Town, we collect data containing different scenarios, which we detail in the following (descriptions are taken from https://leaderboard.carla.org/scenarios/):

- **Control Loss without Previous Action**: The ego-vehicle loses control due to poor road conditions and must recover.
- **Unprotected Left Turn at Intersection with Oncoming Traffic**: The ego-vehicle performs an unprotected left turn at an intersection (can occur at both signalized and unsignalized intersections).
- **Right Turn at Intersection with Crossing Traffic**: The ego-vehicle makes a right turn at an intersection while yielding to crossing traffic (signalized and unsignalized intersections).
- **Crossing Negotiation at Unsignalized Intersection**: The ego-vehicle navigates an unsignalized intersection by negotiating with other vehicles, assuming the first vehicle entering the intersection has priority.
- **Crossing Traffic Running a Red Light at an Intersection**: While traveling straight through an intersection, the ego-vehicle encounters crossing traffic that runs a red light.
- **Crossing with Oncoming Bicycles**: The ego-vehicle must turn at an intersection while yielding to bicycles crossing the road.

- **Highway Merge from On-Ramp**: The ego-vehicle merges into moving traffic on a highway.
- **Highway Cut-In from On-Ramp**: A vehicle merges into the ego-vehicle's lane from an on-ramp, requiring the ego-vehicle to decelerate, brake, or change lanes.
- **Static Cut-In**: Another vehicle cuts into the ego lane from a queue of stationary traffic. The ego-vehicle must react appropriately.
- **Highway Exit**: To exit the highway, the ego-vehicle needs to cross a lane of moving traffic.
- **Yield to Emergency Vehicle**: An emergency vehicle approaches from behind; the ego must create space for it to pass safely.
- **Obstacle in Lane - Same Direction**: An obstacle blocks the ego lane, requiring a lane change into same-direction traffic.
- **Obstacle in Lane - Opposite Direction**: An obstacle blocks the lane, requiring the ego-vehicle to bypass it by moving into a lane with opposite-direction traffic.
- **Door Obstacle**: The ego-vehicle must avoid the door of a parked vehicle opening into its lane.
- **Slow-Moving Hazard at Lane Edge**: A slow-moving hazard (e.g., bicycle) partially obstructs the lane; the ego-vehicle must brake or carefully bypass it.
- **Vehicle Invading Lane on Bend**: On a bend, an oncoming vehicle invades the ego lane; the ego must brake or move aside.
- **Longitudinal Control after Leading Vehicle's Brake**: The leading vehicle brakes suddenly, and the ego-vehicle must execute an emergency maneuver.
- **Obstacle Avoidance without Prior Action**: The ego-vehicle encounters an unexpected obstacle and must brake or evade.
- **Pedestrian Emerging from Behind Parked Vehicle**: A pedestrian suddenly enters the lane from behind a parked vehicle; the ego must brake or evade.
- **Obstacle Avoidance with Prior Action — Pedestrian or Bicycle**: While turning, the ego-vehicle encounters an obstacle such as a pedestrian, bicycle, or stopped vehicle and must react.
- **Parking Cut-In**: A parked vehicle exits a parking space into the ego's path; the ego must slow down to allow it to merge.
- **Parking Exit**: The ego-vehicle must exit a parking space and merge into moving traffic.

Original Driving Routes CARLA Leaderboard 2.0. For data collection, we follow the structure of the long Leaderboard 2.0 routes introduced in [68], which cover Town12 and Town13 with average lengths of 8–12 km and contain roughly one hundred scenarios per route. These long routes provide the full scenario distribution of Leaderboard 2.0, but are impractical for training because they contain many kilometers of redundant driving. Therefore, we split every long route into shorter segments, each containing exactly one scenario and its surrounding context. This preserves the original scenario diversity while making route sampling more balanced and computationally efficient. For each scenario type, we then sample up to 50 short routes with replacement when constructing the training set, ensuring uniform scenario coverage independently of how often a scenario appears in the original long routes. Additional routes are collected from Towns 01–05 and 10 to cover the six classical Leaderboard 1.0 scenarios.

New Driving Routes Introduced by LEAD As the authors of Bench2Drive pointed out [20], the official routes lack geographical diversity that is required to develop robust driving policies. To mitigate this issue, we extend the route set in several ways. First, we incorporate additional routes from Town06 and Town07 by converting previously missing route definitions into the Leaderboard 2.0 XML format. Second, since Town15 provides no official routes, we create a new set of more than 800 routes spanning dense urban grids. Third, we exploit structural redundancies in existing XML scenario descriptions to generate additional high-quality routes with minimal manual effort.

Many scenario types in CARLA share identical XML structures, such as accident obstacles, door opening events, and construction-site hazards. Those differ only in their naming and geographical placement across the map. We leverage this observation to increase scenario diversity in two complementary ways. (1) For underrepresented scenario types, we duplicate existing XML route descriptions and rename the scenario class, effectively creating new routes with identical logic but different spatial distributions. This significantly increases the frequency of rare scenarios without requiring additional manual labeling. (2) During data annotation, a single manually curated XML file can be reused for multiple scenario classes that share the same underlying structure. This reduces labeling overhead and increases annotation throughput, while maintaining consistent semantics across the dataset.

New Scenario of LEAD: Stopping at Red Lights. In the default CARLA data, most red-light stops occur behind queued vehicles, meaning the ego often learns to brake by following a lead car rather than by interpreting the traffic-light state itself. To correct this bias, we extend the red phase for all lights and remove all vehicles that would otherwise stand in front of the ego. This forces the model to base its stopping behavior solely on the visual signal of the red light.

New Scenario of LEAD: Defective Traffic Lights. To simulate ambiguous or faulty signal behavior, we introduce

intersections where all directions display green simultaneously. The ego encounters continuous cross traffic and must behave as if the junction were uncontrolled, relying on gap detection rather than signal color. This encourages cautious approach behavior, waiting for safe temporal gaps, and decisive merging once a gap appears.

New Scenario of LEAD: Unprotected Left Turn with Competing Flow. We add a more challenging variant of unprotected left turns in which vehicles from multiple approaches attempt to merge into the same lane as the ego. This creates competitive interactions where several agents vie for limited temporal gaps. The ego must judge whether a gap is safe while anticipating the actions of vehicles joining from different directions. The scenario produces more realistic negotiation behavior and reduces failure cases in dense urban left-turn settings.

	Hours	Routes	Samples	Towns	Storage
TFv5	40	5600	600k	8	500GB
Ours	73	9300	1003k	11	300GB

Table 3. **Dataset overview.**

F. Further Improvement of Data and Evaluation Pipeline

Besides extending the dataset, we also enhance the quality of the perception data labels, driving data trajectory, and align the controller to fit more to the model's driving outputs.

CARLA Sensor Configuration The original TFv5 setup uses a minimal sensor suite consisting of a single front camera with a wide 110° field of view and one LiDAR. In our experiments, we extend this configuration. Depending on the experiment, we use either a 3-camera setup, each with 60 degree field of view and horizontal overlap, or a full 6-camera rig that provides 360° coverage. In addition, we employ a dual-LiDAR configuration for denser point clouds and a four-radar layout, where each radar covers 90° horizontally.

Adapting Perception Labels All BEV and perspective annotations are updated to reflect the expanded scenario set introduced by Leaderboard 2.0. We introduce dedicated BEV classes for emergency vehicles, stop signs, opened-door obstacles, construction zones, and accident sites, ensuring consistent semantics across both original and newly added scenarios. Emergency vehicles and stop signs additionally receive their own perspective-semantic categories to capture their distinct behavioral relevance. Because pedestrians and bicycles occupy only a few cells at CARLA's default BEV resolution, we enlarge their BEV footprints by a factor of four.

Marking Overhead Traffic Light. The introduction of Town12 and Town13 in CARLA features overhead traffic lights positioned unusually high above the road, making them difficult to detect when the ego is close to the stop line.

To avoid inconsistent annotations and missed signals, we mark the locations of these overhead lights explicitly and ensure that the ego halts at an appropriate distance before the stop line, where the signal is still fully visible. Bounding box for those red traffic lights also gets its own class.

Depth Label Downsampling. Depth maps provided by CARLA are high resolution and expensive to store and process. Since depth supervision was shown to be less effective than other modalities [4], we downsample depth labels by a factor of four before storing them. This preserves metric structure while reducing storage and memory footprint.

Sensor Perturbations. To simulate the compounded errors that occur in closed-loop driving, baseline TransFuser++ augments the sensor rig by duplicating it and applying small rigid perturbations (up to 1 m horizontal translation and 5° yaw rotation). We extend this strategy by allowing rotations of up to 15°, while constraining translations to remain plausible in narrow urban environments and avoiding rotations that would contradict the ego's intended driving direction. In particular, we sample a horizontal translation T and a yaw rotation R. The translation is restricted by the available lateral space $w_{\min}$ in the narrowest lane with respect to the width w_{ego} of ego vehicle:

$$T \sim \text{Uniform}([-T_{\max}, T_{\max}])$$

$$T_{\max} = \tfrac{1}{2}(w_{\min} - w_{\text{ego}}).$$

The rotation range is scaled by a single factor

$$\phi(T) = \text{sgn}(T)\left(1 - \frac{|T|}{T_{\max}}\right),$$

so large translations reduce the allowable rotation and enforce alignment with the translation direction:

$$R \sim \text{Uniform}(-15\phi(T),\ 15\phi(T))\,.$$

For perception pre-training on real-world datasets, we further introduce a third perturbed rig with up to 1.25 m

translation and 35° rotation to reflect the wider range of off-center driving and complex road geometry encountered outside simulation. During training, images from the original and perturbed rigs are sampled uniformly, while all trajectory labels remain defined in the global frame. This setup exposes the model to off-center observations and explicitly teaches it to recover when the ego drifts away from the centerline.

Weather Diversity. The original dataset used a limited set of 21 weather presets. We extend this to a pool with 25 further presets by introducing additional fog levels, sunrise variants, and generally more adverse weather combinations. Each route samples a preset with slight randomization of sun altitude, fog density, cloudiness, and rain intensity.

Controller Tuning. The lateral PID controller uses a discrete lookahead index n_{la} that specifies which future point the controller aims toward when computing the steering command. To improve sensitivity in sharp curves, we apply a curvature-dependent adjustment based solely on the local route curvature κ and a single sensitivity parameter γ:

$$n_{\text{la}} \leftarrow n_{\text{la}} + \text{clip}(\lfloor \kappa\gamma \rfloor, 0, 2)$$

Higher curvature increases the correction, while γ controls how strongly the controller reacts to it.

As for target points, we add two edge cases to the controller. When two of the three target points are in 10m proximity, we decrease the pop distance from 5m to 4m. And if the future target point is more than 50m away, we replace it with the current target point.

We tune every parameter introduced in this section on a 2km evaluation route in Town06, containing many rapid turning transitions, and select the value that maximizes the driving score.

G. Pre-processing Pipeline of LiDAR and Radar.

To provide short-term motion cues, we stack the five most recent LiDAR frames after ego-motion compensation, resulting in a temporally consistent point cloud history rather than a single instantaneous sweep. After aggregation, ground points are removed using a RANSAC-based plane fitting algorithm, and the remaining points are stored as a joint LiDAR point cloud.

Radar detections are incorporated by treating them as additional LiDAR returns and merging them into the same point cloud representation. To emphasize nearby dynamic actors during rasterization, radar detections close to the ego vehicle are duplicated five times, increasing their saliency in the resulting LiDAR pseudo-image.

G.1. Radar Detection Training

The radar detection module predicts a fixed set of $Q = 20$ radar objects per frame, each represented by a 2D position (x, y), a radial velocity v, and a validity score. Predictions are produced from learned radar queries via cross-attention over BEV features, ego-velocity tokens, and radar point tokens. Training is formulated as a set prediction problem following a DETR-style matching scheme, where predicted queries are matched one-to-one with ground-truth radar detections using Hungarian matching. The matching cost combines a normalized L1 regression term over (x, y, v) and a binary classification term over the validity label:

$$\mathcal{C} = \lambda_{\text{reg}}\, \mathcal{L}_{\text{L1}}(x, y, v) + \lambda_{\text{cls}}\, \mathcal{L}_{\text{BCE}}(\hat{l}, l). \tag{1}$$

After matching, losses are computed only on matched pairs. The regression loss is normalized by the spatial and velocity ranges and masked by the ground-truth validity label, while the classification loss is a binary cross-entropy loss on the validity logits. The final radar loss is the weighted sum of both terms, averaged over the batch. The radar detection module is trained jointly with the rest of the network and remains active during both the perception pretraining stage and the final planning training stage.

G.2. Limitations of the CARLA Radar Noise Model

CARLA's radar sensor returns range and radial velocity without modeling several artifacts commonly encountered in real-world automotive radar:

- **Multipath and clutter.** Ghost detections from multipath reflections, ground clutter, and sidelobe returns are absent.
- **Extended targets and speckle.** Real objects produce multiple fluctuating detections across frames; CARLA returns idealized point detections.
- **Doppler ambiguity.** CARLA provides exact radial velocities without the aliasing that occurs when targets exceed the maximum unambiguous velocity.
- **Weather attenuation.** Although CARLA models weather visually, rain and fog do not affect the radar sensor model.

As a result, the radar fusion gains reported in our CARLA experiments (Tables 4 and 5) should be interpreted as an upper bound on the benefit achievable with an ideal radar input. Transferring these gains to real-world deployments would require either training with realistic radar noise augmentation or fine-tuning on real radar data.

H. Detailed CARLA Ablation Experiments

The results in Tables 4 and 5 track the contribution of each component across both Bench2Drive and Longest6v2. Each row represents an additive modification applied on top of the previous configuration, isolating the effect of every individual change.

Method	DS ↑	Backbone
TFv5	83.56 ± 0.34	RegNetY-032
Align State	84.94 ± 0.50	RegNetY-032
Align Intent	89.29 ± 0.54	RegNetY-032
Radar Fusion	90.01 ± 0.42	RegNetY-032
Enhanced Dataset	94.01 ± 1.48	ResNet34
Tuned Controller	94.72 ± 0.72	ResNet34
TFv6	**95.28 ± 0.36**	**RegNetY-032**

Table 4. **Ablation summary on Bench2Drive.**

Method	DS ↑	Backbone
TFv5	22.51 ± 4.42	RegNetY-032
Align State	34.05 ± 3.52	RegNetY-032
Align Intent	42.13 ± 1.80	RegNetY-032
Radar Fusion	42.60 ± 2.56	RegNetY-032
Enhanced Dataset	50.01 ± 2.86	ResNet34
Tuned Controller	57.74 ± 2.99	ResNet34
TFv6	**62.92 ± 1.58**	**RegNetY-032**

Table 5. **Ablation summary on Longest6v2.**

I. Fully End-to-End Driving Without Heuristics

The results reported in the main paper use three lightweight post-processing heuristics inherited from the TFv5 codebase: a Kalman filter for smoothing target point locations, a stop-sign deceleration rule, and a creeping heuristic that nudges the vehicle forward when blocked. These heuristics are common in prior work [16, 42, 68] but are not part of the learned policy.

To verify that TFv6's performance is not dependent on these components, we disable all three and evaluate the resulting fully end-to-end model. As shown in Table 6, performance remains comparable, confirming that TFv6 drives effectively without any hand-crafted post-processing.

Configuration	B2D ↑	L6v2 ↑	Town13 Train↑
TFv6 (with heuristics)	95.2 ± 0.3	62 ± 1	5.2 ± 0.8
TFv6 (fully E2E)	94.4 ± 0.5	62 ± 5	10.3 ± 0.7

Table 6. **TFv6 without post-processing heuristics.** Removing the Kalman filter, stop-sign rule, and creeping heuristic has minimal impact, confirming that TFv6 operates as a fully end-to-end policy.

J. Real-World-Data Benchmarks

Waymo E2E Benchmark. The dataset consists of 4021 run segments, each 20 seconds long. Of these, 2037 segments are used for training, and 479 segments are used for validation. The remaining 1050 segments form the test set.

The training split contains roughly 400k samples from diverse US urban environments, collected at 10Hz frequency. Each sample provides synchronized multi-camera RGB images, high-precision ego motion, and expert trajectories. No BEV labels are available; the benchmark provides only trajectory supervision. The evaluation focuses on long-horizon consistency, rare-event robustness, and multi-agent interaction quality.

For training, at each epoch, we subsample 300k samples from the training split. For the baseline, we train the model supervised only on the expert trajectory for 60 epochs. For co-training with LEAD, the first 30 epochs are trained entirely on CARLA samples, while the remaining 30 epochs use a mixture of CARLA and Waymo data.

For evaluation, we select the model checkpoint of the epoch that yields the highest RFS on the validation set.

NAVSIM Benchmarks. NAVSIM offers a pseudo-simulation environment built from real-world perception logs, enabling closed-loop evaluation without real-world rollouts. Its core training source is the *NavTrain* dataset, which contains 103k challenging driving samples explicitly curated to break constant-velocity baselines. Each sample includes multi-camera RGB, ego motion, and dense BEV annotations such as road/lane semantics and 3D bounding boxes for all agents.

For training, at each epoch, we subsample 100k samples from the training split. For the baseline, we train the model supervised only on the expert trajectory for 120 epochs. For co-training with LEAD, the first 30 epochs are trained on a mix of CARLA and NAVSIM samples, while the remaining 90 epochs use only NAVSIM data to ensure the model's learned statistics are aligned with real-world data.

For evaluation, we simply take the final model.

Filtering CARLA Data. To support Sim2Real training, we extract only the most useful CARLA samples rather than training on the full simulator dataset. Every frame is evaluated by a set of about 60 simple filters. Each filter corresponds to a situation that is either rare, important for safety, or underrepresented in NavTrain or Waymo. Examples include bad weather, dense junctions, high numbers of parked obstacles, pedestrian interaction, sharp curves, roundabouts, etc.

Each filter is a lightweight check on simulator metadata (bounding boxes, distance to junction, traffic-light states, ego motion, scenario tags). When a frame satisfies a condition, it is added to the corresponding bucket exclusively. This creates many small disjoined subsets of CARLA data, each focused on a specific type of interaction.

During training, we draw samples according to a fixed mixture: buckets that capture difficult or safety-critical sit-

uations are upweighted, while common or uninformative ones are heavily downsampled. Across an entire epoch, we cap the contribution of CARLA to at most 100k frames, ensuring that synthetic data acts as a targeted supplement rather than overwhelming the real-world distribution. This strategy allows CARLA to contribute exactly those interactions that are expensive or infeasible to obtain from real logs, without distorting the overall training mix.

Co-training with Synthetic Data. To ensure comparability with the real-data-only experiments, we keep the number of gradient steps and the batch size identical to the baseline, avoiding increased compute as a confounding factor.

To enrich the NAVSIM and Waymo benchmarks with synthetic data, we re-run the LEAD expert on the same route set used for the CARLA Leaderboard, but with the ego vehicle equipped with the camera intrinsics/extrinsics of the respective real-world-data benchmark.

In training, we gradually shift the training distribution from synthetic CARLA samples to real-world NavTrain/Waymo samples. At the start of training, most samples in the batch come from CARLA, allowing the model to learn rare and safety-critical interactions with dense labels that are easy to generate in simulation but scarce in real logs. As training progresses, the ratio is slowly annealed toward real data, ensuring that the final model is aligned with real-world sensor statistics and driving behavior.

For NAVSIM, we discard all nighttime and adverse-weather episodes to remain consistent with the conditions represented in NavTrain. For Waymo, we remove segments recorded under fog, which rarely appear in the official Challenge distribution.

Method	PDMS ↑	EPDMS ↑
Baseline NAVSIM	85.1 ± 0.4	28.3 ± 0.8
No Scheduled Co-Training	86.2 ± 0.4	30.8 ± 1.0
Scheduled Co-Training	**86.4** ± 0.3	**31.4** ± 1.0

Table 7. **Co-training NAVSIM Results.**

Tables 7 compare three representative configurations: the baseline trained only on data of the respective benchmark, co-training without annealing, and our scheduled co-training strategy. Overall, scheduled co-training leads to a minor improvement over the other settings, but it consistently provides the best performance among the tested variants.

K. Training Hyperparameters

We summarize the training hyperparameters of TFv6 in Table 8.

Common hyperparameters

Hyperparameter	Value
Batch size	64
Optimizer	AdamW
Initial LR	3e−4
End LR	0
Weight decay	0.01
LR Scheduler	Cosine LR Decay
Grad Clipping	No

Dataset-specific hyperparameters

Hyperparameter	CARLA	NAVSIM	Waymo
Pre-training epochs	30	60	30
Post-training epochs	30	60	30
LR Decay Restart	Yes	Yes	No
Image Augm. Prob.	0.2	0.5	0.5
Image Resolution	384×384	256×480	288×262
Number of Cameras	3	4	3
Back Camera Used	No	Yes	No
Sensor Per. Prob.	0.5	0.75	0.75
Past States Input	0	0	6
Future States Pred.	2s	4s	5s
Future Pred. Freq.	4Hz	2Hz	2Hz
Future Heading Pred.	No	Yes	No
Backbone	RegNetY32	ResNet34	ResNet34

Table 8. **Training hyperparameters across benchmarks.**

From TFv5 to TFv6, we increase the number of detectable bounding boxes from 30 to 90.

For the CARLA Leaderboard, we allow the model to detect at most 20 objects with Radar. For reliable modeling of unprotected turns in an intersection, we increase the range of the BEV grid width from $[-32\text{m}, 32\text{m}]$ to $[-40\text{m}, 40\text{m}]$ so vehicles coming from left or right can be modeled more explicitly. LiDAR points above 4m and below -4m are discarded to save storage.

For radar encoding, we use a 4-layer transformer decoder with 8 heads. The matching and optimization cost function for radar detection is a weighted sum of object properties, regression, and object presence classification. The regression is weighted with a factor of 5, and the classification is weighted with a factor of 1.

For training with mixed-precision training in BF16, we make sure every normalization layer and softmax, as well as cross-entropy loss, are computed in 32-bit precision.

L. Further Changes from PDM-Lite to LEAD

Table 9 summarizes the small but notable adjustments from PDM-Lite to LEAD. Time–headway parameters tune the IDM's temporal safety buffer before braking, shaping how cautiously the expert approaches stop signs, red lights,

pedestrians, and bicycles. Minimum-distance parameters set the spatial clearance that the IDM maintains around other agents. The urban-junction speed cap regulates the maximum speed permitted inside urban intersections, while the overtake-speed parameter defines the target velocity used when passing static obstacles or slower vehicles.

Parameter	PDM-Lite	LEAD
idm_stop_sign_time_headway	0.1	0.5
idm_red_light_min_distance	6.0	3.0
idm_red_light_time_headway	0.1	0.5
idm_pedestrian_min_distance	4.0	4.5
idm_pedestrian_time_headway	0.1	0.2
idm_bicycle_min_distance	1.0	6.0
idm_bicycle_time_headway	4.0	6.0
max_speed_in_junction_urban	17.77	7.0
default_overtake_speed	13.88	11.11

Table 9. **Key differences between PDM-Lite and LEAD.**

M. Inference Time

Student forward pass time. Table 10 reports the forward pass time of TFv5 and TFv6 (140° FOV front camera-setup, LiDAR, and radar variant) on a single RTX 2080 Ti.

Model	Time per Frame ↓
TFv5 ResNet34	41 ms
TFv6 ResNet34 (Ours)	40 ms
TFv5 RegNetY-032	57 ms
TFv6 RegNetY-032 (Ours)	70 ms

Table 10. **Forward Pass Time Comparison.**

Expert inference time. Table 11 reports the runtime overhead introduced by LEAD relative to PDM-Lite. For data collection mode, experts are evaluated under identical conditions: the ego vehicle operates in full data-collection mode with three RGB cameras, one LiDAR, and bounding-box extraction.

While PDM-Lite can run without any sensors when used purely as a driving policy, LEAD is meaningful only in data-collection mode.

Expert	Time per Frame ↓
PDM-Lite without Data Collection [17]	**18 ms**
PDM-Lite with Data Collection	124 ms
LEAD with Data Collection (Ours)	182 ms

Table 11. **Inference time comparison in full data-collection mode.**

N. NAVSIM v1 Summary

We summarize the current NAVSIM v1 leaderboard results, ranked by PDMS, in Table 12.

Method	PDMS ↑	Year	Venue
RAP [6]	**93.80**	2025	–
iPad [10]	91.72	2025	–
DiffusionDrive [31]	88.02	2024	CVPR
LTFv6 (Ours)	86.43	2025	–
TransFuser [5]	83.88	2024	NeurIPS
LTF [5]	83.52	2024	NeurIPS
Ego Status MLP [5]	66.40	2024	NeurIPS
CV [5]	20.65	2024	NeurIPS

Table 12. **NAVSIM v1 leaderboard snapshot as of 20 Nov 2026.**

Table 13 reports NAVSIM v1 results restricted to methods evaluated with a unified ResNet-34 backbone, following the controlled comparison protocol of Table 1 in [48]. By fixing the perception backbone, the comparison isolates differences in planning, trajectory generation, and decision-making components.

In addition, we indicate whether a method explicitly optimizes for the PDMS evaluation metric.

Method	PDMS ↑	Year	PDMS Opt.
MindDrive [48]	**88.9**	2025	Yes
WoTE [26]	88.3	2025	Yes
DIVER [38]	88.3	2025	Yes
Hydra-MDP++ [23]	86.6	2024	Yes
Hydra-MDP [27]	86.5	2024	Yes
GoalFlow [57]	85.7	2025	Yes
DiffusionDrive [31]	**88.1**	2025	No
LTFv6 (Ours)	86.4	2025	No
DRAMA [61]	85.5	2024	No
TransFuser [5]	84.0	2022	No
LTF [5]	83.8	2022	No
UniAD [13]	83.4	2023	No

Table 13. **NAVSIM v1 results under a unified ResNet-34 backbone.** Values are taken from Table 1 of [48].

O. NAVSIM v2 Summary

We summarize the current performance of existing methods on NAVSIM v2 in Table 14.

Table 15 reports NAVSIM v2 results on the Navhard split, restricted to methods evaluated with a unified ResNet-34 backbone, following the controlled comparison protocol of [48]. By fixing the backbone and sensor configuration,

Method	EPDMS ↑	Year	Venue
ZTRS [29]	**48.12**	2025	–
RAP [6]	39.61	2025	–
LTFv6 (Ours)	31.91	2025	–
LTF [5]	25.12	2024	NeurIPS
Ego Status MLP [5]	14.17	2024	NeurIPS
CV [5]	11.48	2024	NeurIPS

Table 14. **NAVSIM v2 leaderboard snapshot as of 20 Nov 2026.**

the comparison isolates differences in planning robustness under perturbed and safety-critical conditions.

We focus exclusively on the final EPDMS score, which aggregates Stage I and Stage II evaluations. As for NAVSIM v1, we additionally indicate whether a method explicitly optimizes for the EPDMS metric.

Method	EPDMS ↑	Year	EPDMS Opt.
MindDrive [48]	**30.5**	2025	Yes
GuideFlow [34]	27.1	2025	Yes
GTRS-DP [30]	23.8	2025	Yes
LTFv6 (Ours)	**31.9**	2025	No
DiffusionDrive [31]	24.2	2025	No
LTF [5]	23.1	2022	No

Table 15. **NAVSIM v2 (Navhard) results under a unified ResNet-34 backbone.** Values are taken from Table 3 of [48]. We report the final EPDMS score, which aggregates Stage I and Stage II evaluations. We restrict all methods to the same backbone.

P. Bench2Drive Summary

Table 16 summarizes Bench2Drive results for methods that have been published in peer-reviewed venues or publicly available technical reports. These results represent the current state of the literature and provide a reference point for comparing established approaches under a common evaluation protocol.

Table 17 reports Bench2Drive results for methods that have not yet undergone peer review, including preprints, technical reports, and leaderboard submissions without an accompanying publication. We include these results for completeness and transparency, but distinguish them from peer-reviewed methods.

Method	DS ↑	SR ↑	Year	Venue
TCP [56]	40.70	15.00	2022	NeurIPS
VAD [22]	42.35	15.00	2023	ICCV
SparseDrive [50]	44.54	16.71	2025	ICRA
GenAD [67]	44.81	15.90	2024	ECCV
Dual-AEB [66]	45.23	10.00	2025	ICRA
UniAD [13]	45.81	16.36	2023	CVPR
MomAD [45]	47.91	18.11	2025	CVPR
BridgeAD [64]	50.06	22.73	2025	CVPR
SeerDrive [65]	58.32	30.17	2025	NeurIPS
WoTE [25]	61.71	31.36	2025	ICCV
DriveDPO [43]	62.02	30.62	2025	NeurIPS
ThinkTwice [19]	62.44	31.23	2022	CVPR
DriveTransformer-L. [21]	63.46	35.01	2025	ICLR
DriveAdapter [18]	64.22	33.08	2023	ICCV
OAIAD [60]	68.73	48.86	2025	Machines
Raw2Drive [59]	71.36	50.24	2025	NeurIPS
VL [2]	73.29	65.44	2025	ICRA
ETA [12]	74.33	48.33	2025	ICCV
DiffusionDrive-temp [31]	77.68	52.72	2025	CVPR
ORION [7]	77.74	54.62	2025	ICCV
PGS [15]	78.08	48.64	2025	NeurIPS
GaussianFusion [36]	79.1	54.4	2025	NeurIPS
SimLingo [42]	85.07	67.27	2025	CVPR
HiP-AD [51]	86.77	69.09	2025	ICCV
TFv6 (Ours)	**95.28**	**86.81**	2025	-
Think2Drive [24]	*91.85*	*85.41*	2024	ECCV
PDM-Lite [44]	*97.02*	*92.27*	2024	ECCV
LEAD (Ours)	*96.78*	*96.59*	2025	-

Table 16. **Reported Bench2Drive results for peer-reviewed methods.**

Method	DS ↑	SR ↑	Year	Venue
AD-MLP [62]	18.05	0.00	2023	-
ReAL-AD [39]	41.17	11.36	2025	-
FUMP [33]	45.67	16.36	2025	-
FocalAD [49]	45.77	17.30	2025	-
CogAD [54]	48.30	24.00	2025	-
E³AD [41]	50.07	20.12	2025	-
X-Driver [37]	51.7	18.1	2025	-
DiFSD [47]	52.02	21.00	2024	-
SpaRC-AD [55]	55.6	30.0	2025	-
VeteranAD [63]	64.22	33.85	2025	-
iPad [10]	65.02	35.91	2025	-
StuckSolver [3]	65.23	36.32	2025	-
GEMINUS [52]	65.39	37.73	2025	-
VDRive [11]	66.25	50.51	2025	-
RAP-ResNet [6]	66.42	37.27	2025	-
CAPS [40]	66.76	52.87	2025	-
SNG [14]	67.17	35.90	2025	-
DiffAD [53]	67.92	38.64	2025	-
DIVER [46]	68.90	36.75	2025	-
Hydra-NeXt [28]	73.86	50.00	2025	-
DriveMoE [58]	74.22	48.64	2025	-
TFv5 [68]	84.21	67.27	2024	-
R2SE [32]	86.28	69.54	2025	-
BridgeDrive [35]	86.87	72.27	2025	-
TFv6 (Ours)	**95.28**	**86.81**	2025	-
Think2Drive [24]	*91.85*	*85.41*	2024	ECCV
PDM-Lite [44]	*97.02*	*92.27*	2024	ECCV
LEAD (Ours)	*96.78*	*96.59*	2025	-

Table 17. **Reported Bench2Drive results for non-peer-reviewed methods.**

Q. Fail2Drive

Fail2Drive [9] is a recent CARLA closed-loop benchmark that measures the generalization of driving policies under controlled distribution shift. It consists of 200 short routes on Town13, paired into 100 *In-Distribution/Generalization* couples. Both routes of a pair share the same location, spawn points, and traffic configuration, and differ only by a targeted structural change, such as a novel obstacle asset, a non-standard agent behavior, or a visual perturbation. This paired design isolates the causal factor of failure, so that performance differences directly quantify sensitivity to the induced shift.

The benchmark introduces 17 new scenario classes grouped into four categories: *Behavior* (high-level behaviors rarely demonstrated in standard CARLA data), *Visual-lat* and *Visual-lon* (appearance, geometry, or layout shifts affecting lateral or longitudinal control), and *Robustness* (irrelevant environmental changes that should *not* trigger a reaction). Models are scored with Driving Score (DS), Success Rate (SR), and the harmonic mean HM $= 2\,\mathrm{DS}\cdot\mathrm{SR}\,/\,(\mathrm{DS}+\mathrm{SR})$. Route Completion is omitted because routes are short and almost always finished [9].

TFv6 results: We evaluate TFv6 on Fail2Drive, averaging over three independently trained seeds. Overall, TFv6 reaches 91.7 HM on the In-Distribution split and 74.8 HM on the Generalization split, a relative drop of -18.4%. Driving Score falls from 90.2 to 79.5 (-11.9%) and Success Rate from 93.3 to 70.7 (-24.2%). Table 18 reports the per-category breakdown and Table 19 the per-scenario HM.

Per-category analysis: Consistent with the cross-model findings in [9], we observe a pronounced asymmetry across categories. *Robustness* scenarios, which only introduce irrelevant environmental changes, degrade by merely -2.7%, indicating that TFv6 does not over-react to spurious distractors. *Visual-lon* and *Visual-lat* shifts produce moderate drops of -14.5% and -26.2%. In contrast, *Behavior* scenarios drop by -38.2%: these require the policy to execute high-level behaviors rarely demonstrated in the training distribution, such as stopping for a fully blocked road or following pedestrians walking on the lane.

Per-scenario analysis.: The largest failures occur in *Wall* (-87.7%) and *PedestriansOnRoad* (-45.8%). Among Visual-lat scenarios, *CustomObstacles* accounts for nearly all of the degradation (-62.8%). This reproduces the observation of [9] that avoidance behaviors are triggered by familiar CARLA-specific obstacle templates rather than the spatial presence of obstacles in the drivable lane.

At the same time, TFv6 remains essentially unaffected by texture-level perturbations (*ObscuredStop*: $+0.1\%$) and by obstacles placed outside the ego trajectory (*PassableObstacles*: -0.6%, *PedestrianCrowd*: $+0.7\%$). We consider behavior-level generalization an important direction for future work.

Category	ID HM ↑	Gen HM ↑	Rel.
Robustness	95.5	92.9	-2.7%
Visual-lon	92.3	79.0	-14.5%
Visual-lat	87.0	64.1	-26.2%
Behavior	92.1	56.9	-38.2%
Overall	**91.7**	**74.8**	$\mathbf{-18.4\%}$

Table 18. **TFv6 on Fail2Drive, per-category summary.** Harmonic Mean of Driving Score and Success Rate, averaged over three training seeds. ID = In-Distribution.

Category	Scenario	ID ↑	Gen ↑	Rel.
Behavior	Wall	98.4	12.1	-87.7%
Behavior	FullyBlocked	98.3	83.9	-14.7%
Behavior	PedestriansOnRoad	93.2	50.5	-45.8%
Behavior	ConstructionPedestrian	78.5	64.0	-18.5%
Visual-lat	BadParking	96.5	89.2	-7.6%
Visual-lat	ConstructionPermutations	80.5	69.2	-14.0%
Visual-lat	CustomObstacles	80.6	30.0	-62.8%
Visual-lon	ObscuredStop	99.7	99.8	$+0.1\%$
Visual-lon	HardBrake	90.8	82.0	-9.7%
Visual-lon	RightOfWay	98.9	83.6	-15.5%
Visual-lon	Animals	86.6	60.7	-30.0%
Visual-lon	PedestrianOtherBlocker	91.2	86.6	-5.0%
Robustness	RightConstruction	99.0	97.6	-1.4%
Robustness	OppositeConstruction	99.2	94.4	-4.9%
Robustness	ImageOnObject	93.7	87.1	-7.0%
Robustness	PassableObstacles	98.0	97.4	-0.6%
Robustness	PedestrianCrowd	87.2	87.8	$+0.7\%$

Table 19. **TFv6 on Fail2Drive, per-scenario HM.** Harmonic Mean of Driving Score and Success Rate on the In-Distribution (ID) and Generalization (Gen) splits, averaged over three training seeds.

Finally, we compare TFv6 against the published numbers of [9] in Table 20. Among all learned methods, TFv6 achieves the highest HM on both splits (91.7 In-Distribution, 74.8 Generalization), outperforming the previous best sensor-based method TransFuser++ [16] by 7.3 HM on the generalization split and the privileged learned planner PlanT 2.0 [8] by 10.0 HM. Only the privileged rule-based expert PDMLite-F2D [9] scores higher. Nevertheless, the absolute gap of -18.4% HM confirms the central finding of Fail2Drive: even strong closed-loop policies that saturate Bench2Drive and Longest6 v2 still fail to generalize to genuinely unseen scenarios.

Method	RGB	LiDAR	Privileged	Learned	Bench2Drive	Fail2Drive					
						In-Distribution			Generalization		
					DS ↑	DS ↑	SR(%) ↑	HM ↑	DS ↑	SR(%) ↑	HM ↑
TCP [56]	✓	✗	✗	✓	59.9	24.7	39.1	30.3	24.5 (-0.8%)	31.4 (-19.7%)	27.5 (-9.1%)
UniAD [13]	✓	✗	✗	✓	45.8	47.5	36.3	41.2	44.0 (-7.4%)	27.6 (-24.0%)	33.9 (-17.6%)
Orion [7]	✓	✗	✗	✓	77.8	53.0	52.0	52.5	51.2 (-3.4%)	46.0 (-11.5%)	48.5 (-7.7%)
HiP-AD [51]	✓	✗	✗	✓	86.8	74.1	70.7	72.4	67.1 (-9.4%)	56.7 (-19.8%)	61.5 (-15.1%)
SimLingo [42]	✓	✗	✗	✓	85.1	82.6	79.3	80.9	71.7 (-13.2%)	55.0 (-30.6%)	62.2 (-23.1%)
TF++ [16]	✓	✓	✗	✓	84.2	83.3	78.5	80.8	75.4 (-9.5%)	61.1 (-22.2%)	67.5 (-16.5%)
TFv6 (Ours)	✓	✓	✗	✓	**95.0**	**90.2**	**93.3**	**91.7**	**79.5** (-11.9%)	**70.7** (-24.2%)	**74.8** (-18.4%)
PlanT 2.0 [8]	-	-	✓	✓	92.4	87.8	85.0	86.4	73.3 (-16.5%)	58.0 (-31.8%)	64.8 (-25.0%)
PDMLite-F2D [9]	-	-	✓	✗	97.0	95.6	97.0	96.3	94.0 (-1.7%)	95.3 (-1.8%)	94.6 (-1.7%)

Table 20. **Results on Fail2Drive.** In-Distribution evaluates on known CARLA scenarios; Generalization measures robustness under distribution shift. We include reported scores on Bench2Drive for comparison. Baseline numbers are taken from [9]; TFv6 is evaluated by us and averaged over three training seeds.

R. Qualitative Results

We provide more qualitative results of the effect of intent and visual alignment in Figures 1, 2, 3, 4, 5.

Figure 1. **Top.** Limited intent conditioning results in ambiguous behavior and makes TFv5 miss the exit. **Bottom.** TFv6 resolves these ambiguities through improved intent conditioning, enabling reliable lane-change execution and safer interaction with surrounding traffic.

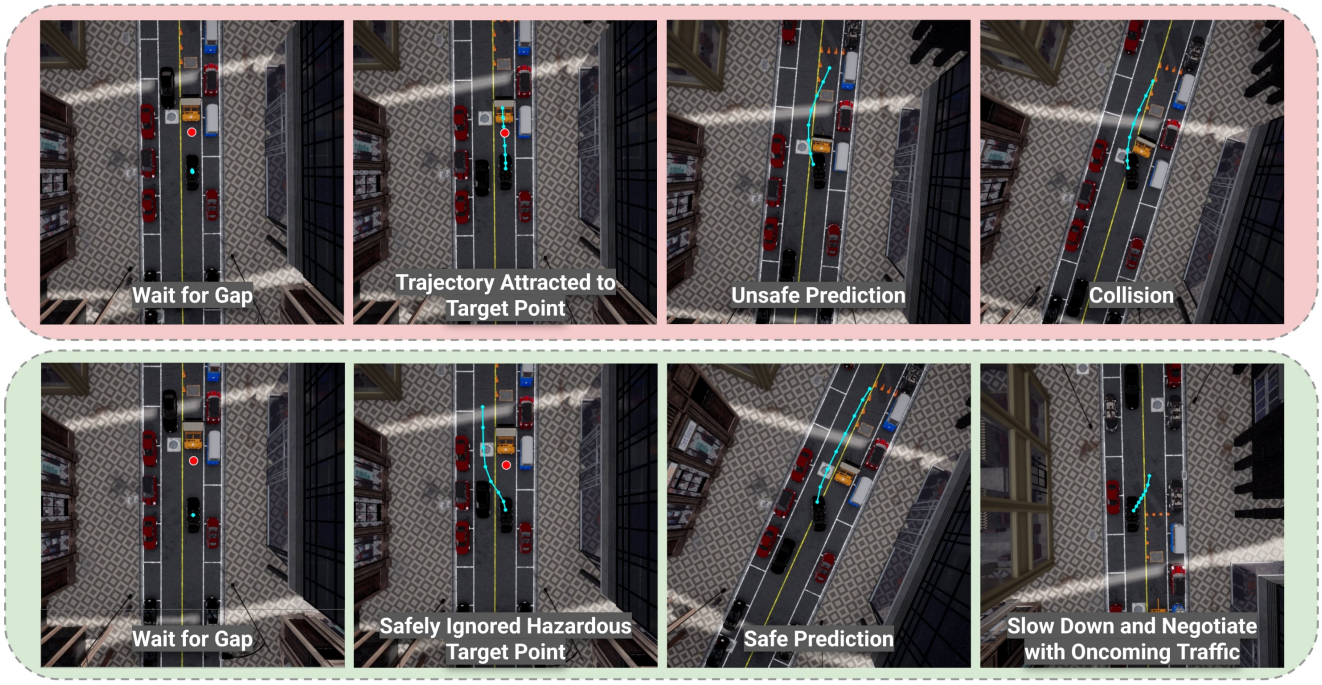

Figure 2. **Top.** Driving model abuses the target point bias and produces an unsafe trajectory, leading to a collision. **Bottom.** TFv6 relies less on target point bias. With the expert's recovery demonstrations coming from visual alignment (d), the driving model also learns to negotiate after a sub-optimal gap estimation.

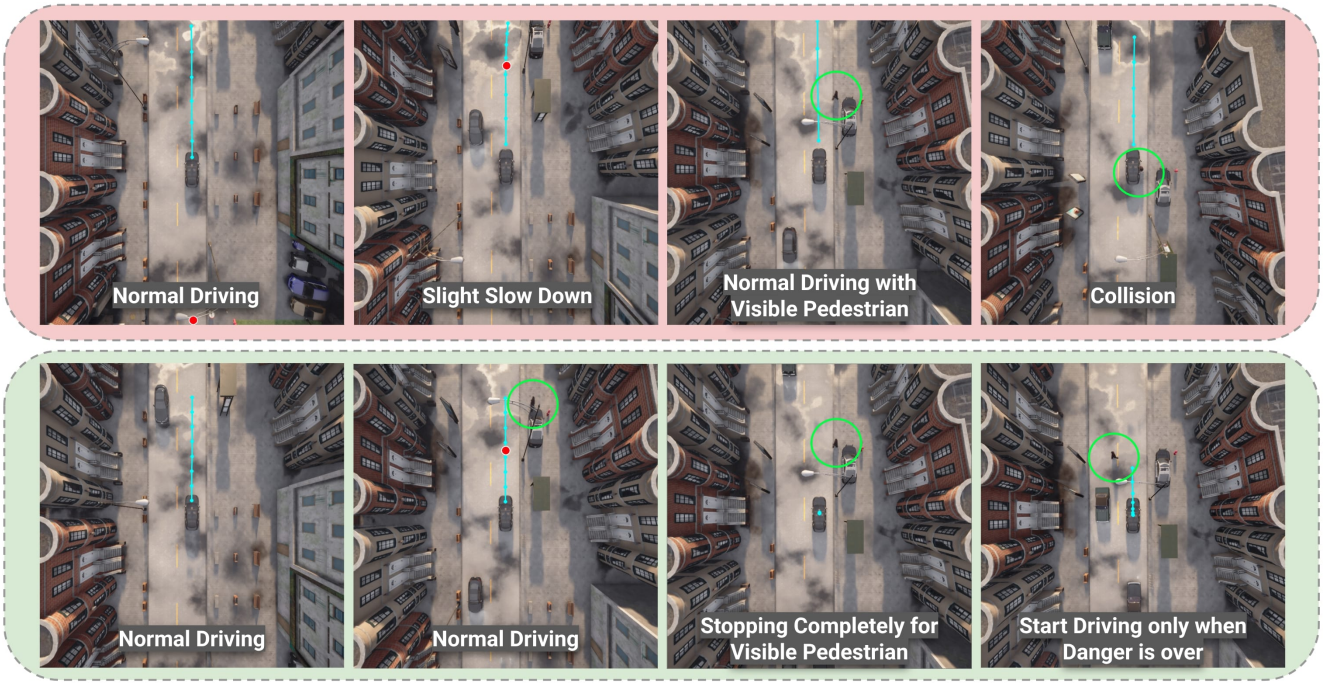

Figure 3. **Top.** PDM-Lite brakes only when its bounding-box collision predictor triggers, providing a weak and delayed learning signal for pedestrian interactions in TFv5. **Bottom.** LEAD stops immediately on sight of visible pedestrians and resumes only once the hazard is fully cleared. Trained on these causal demonstrations, TFv6 handles the scene safely.

Figure 4. Effect of the current target point on the planning output before (**left**) and after (**right**) intent alignment. In the pre-alignment model, the predicted trajectory follows the target point almost verbatim. After intent alignment, the learner interprets target points more robustly and produces a consistent, lane-aligned plan. While this reduces the tendency to overreact to small variations in target-point placement, it also removes the "recovering" behavior that occasionally helped the model correct itself on long-route evaluations.

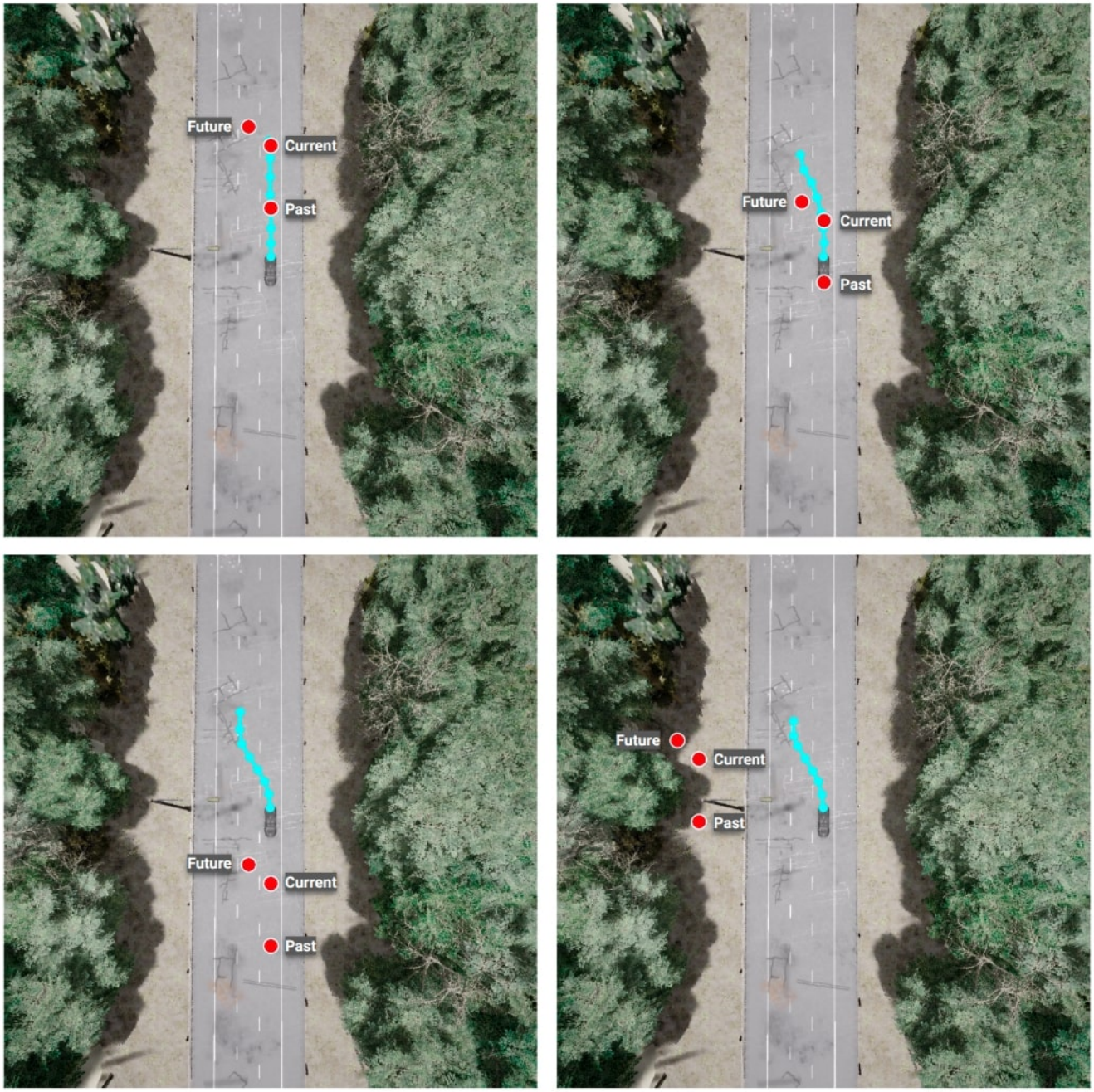

Figure 5. Preliminary investigation shows that with 3 target points, the planning does not always depend on the exact locations of the target points, but also on the local structure of the points.

References

[1] Waymo open dataset: An autonomous driving dataset. https://www.waymo.com/open, 2019. 3

[2] Fazel Arasteh, Mohammed Elmahgiubi, Behzad Khamidehi, Hamidreza Mirkhani, Weize Zhang, and Cao Tongtong. Validity learning on failures: Mitigating the distribution shift in autonomous vehicle planning. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2025. 12

[3] Zhipeng Bao and Qianwen Li. Large language model-assisted autonomous vehicle recovery from immobilization. *arXiv.org*, 2510.26023, 2025. 13

[4] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2023. 7

[5] 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. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 11, 12

[6] 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.org*, 2510.04333, 2025. 11, 12, 13

[7] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2025. 12, 15

[8] Simon Gerstenecker, Andreas Geiger, and Katrin Renz. Plant 2.0: Exposing biases and structural flaws in closed-loop driving. *arXiv.org*, 2025. 14, 15

[9] Simon Gerstenecker, Andreas Geiger, and Katrin Renz. Fail2drive: Benchmarking closed-loop driving generalization. *arXiv.org*, 2026. 14, 15

[10] Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. ipad: Iterative proposal-centric end-to-end autonomous driving. *arXiv.org*, 2505.15111, 2025. 11, 13

[11] Ziang Guo and Zufeng Zhang. Vdrive: Leveraging reinforced vla and diffusion policy for end-to-end autonomous driving. *arXiv.org*, 2510.15446, 2025. 13

[12] Shadi Hamdan, Chonghao Sima, Zetong Yang, Hongyang Li, and Fatma Güney. Eta: Efficiency through thinking ahead, a dual approach to self-driving with large model. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2025. 12

[13] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, et al. Planning-oriented autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 11, 12, 15

[14] Zhihua Hua, Junli Wang, Pengfei Li, Qihao Jin, Bo Zhang, Kehua Sheng, Yilun Chen, Zhongxue Gan, , and Wenchao Ding. Unveiling the surprising efficacy of navigation understanding in end-to-end autonomous driving. URL: https://zhihua-hua.github.io/NavigationDrive-web/static/NavigationDrive.pdf, 2025. 13

[15] Yi Huang, Zhan Qu, Lihui Jiang, Bingbing Liu, and Hongbo Zhang. Prioritizing perception-guided self-supervision: A new paradigm for causal modeling in end-to-end autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 12

[16] Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden biases of end-to-end driving models. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 9, 14, 15

[17] Bernhard Jaeger, Daniel Dauner, Jens Beißwenger, Simon Gerstenecker, Kashyap Chitta, and Andreas Geiger. Carl: Learning scalable planning policies with simple rewards. *Proc. Conf. on Robot Learning (CoRL)*, 2025. 11

[18] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 12

[19] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 12

[20] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 6

[21] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *Proc. of the International Conf. on Learning Representations (ICLR)*. OpenReview.net, 2025. 12

[22] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 12

[23] 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, 2025. 11

[24] Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in CARLA-V2). In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 12, 13

[25] 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. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2025. 12

[26] 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, 2025. 11

[27] 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. 11

[28] Zhenxin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Zuxuan Wu, and José M. Álvarez. Hydra-next: Robust closed-loop driving with open-loop training. *arXiv.org*, 2503.12030, 2025. 13

[29] Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Jingde Chen, Nadine Chang, Maying Shen, Jingyu Song, Zuxuan Wu, Shiyi Lan, and Jose M. Alvarez. Ztrs: Zero-imitation end-to-end autonomous driving with trajectory scoring, 2025. 12

[30] 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, 2025. 12

[31] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 11, 12

[32] 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. *arXiv.org*, 2506.09800, 2025. 2, 3, 13

[33] Lin Liu, Caiyan Jia, Ziying Song, Hongyu Pan, Bencheng Liao, Wenchao Sun, Yongchang Zhang, Lei Yang, and Yandan Luo. Fully unified motion planning for end-to-end autonomous driving. *arXiv.org*, 2504.12667v2, 2025. 13

[34] Lin Liu, Caiyan Jia, Guanyi Yu, Ziying Song, JunQiao Li, Feiyang Jia, Peiliang Wu, Xiaoshuai Hao, and Yandan Luo. Guideflow: Constraint-guided flow matching for planning in end-to-end autonomous driving, 2025. 12

[35] Shu Liu, Wenlin Chen, Weihao Li, Zheng Wang, Lijin Yang, Jianing Huang, Yipin Zhang, Zhongzhan Huang, Ze Cheng, and Hao Yang. Bridgedrive: Diffusion bridge policy for closed-loop trajectory planning in autonomous driving. *arXiv.org*, 2509.23589, 2025. 2, 3, 13

[36] Shuai Liu, Quanmin Liang, Zefeng Li, Boyang Li, and Kai Huang. Gaussianfusion: Gaussian-based multi-sensor fusion for end-to-end autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 12

[37] Wei Liu, Jiyuan Zhang, Binxiong Zheng, Yufeng Hu, Yingzhan Lin, and Zengfeng Zeng. X-driver: Explainable autonomous driving with vision-language models. *arXiv.org*, 2505.05098, 2025. 13

[38] Meixiu Long, Duolin Sun, Dan Yang, Junjie Wang, Yecheng Luo, Yue Shen, Jian Wang, Hualei Zhou, Chunxiao Guo, Peng Wei, Jiahai Wang, and Jinjie Gu. Diver: A multi-stage approach for reasoning-intensive information retrieval, 2025. 11

[39] Yuhang Lu, Jiadong Tu, Yuexin Ma, and Xinge Zhu. Real-ad: Towards human-like reasoning in end-to-end autonomous driving. *arXiv.org*, 2507.12499, 2025. 13

[40] Hamidreza Mirkhani, Behzad Khamidehi, Ehsan Ahmadi, Fazel Arasteh, Mohammed Elmahgiubi, Weize Zhang, Umar Rajguru, and Kasra Rezaee. CAPS: context-aware priority sampling for enhanced imitation learning in autonomous driving. *arXiv.org*, 2503.01650, 2025. 13

[41] Ling Niu, Xiaoji Zheng, Han Wang, Chen Zheng, Ziyuan Yang, Bokui Chen, and Jiangtao Gong. Embodied cognition augmented end2end autonomous driving. *arXiv.org*, 2511.01334, 2025. 13

[42] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3, 9, 12, 15

[43] Shuyao Shang, Yuntao Chen, Yuqi Wang, Yingyan Li, and Zhaoxiang Zhang. Drivedpo: Policy learning via safety dpo for end-to-end autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 12

[44] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 12, 13

[45] Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 12

[46] 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.org*, 2507.04049, 2025. 13

[47] Haisheng Su, Wei Wu, and Junchi Yan. Difsd: Ego-centric fully sparse paradigm with uncertainty denoising and iterative refinement for efficient end-to-end autonomous driving. *arXiv.org*, 2409.09777, 2024. 13

[48] Bin Sun, Yaoguang Cao, Yan Wang, Rui Wang, Jiachen Shang, Xiejie Feng, Jiayi Lu, Jia Shi, Shichun Yang, Xiaoyu Yan, and Ziying Song. Minddrive: An all-in-one framework bridging world models and vision-language model for end-to-end autonomous driving, 2025. 11, 12

[49] Bin Sun, Boao Zhang, Jiayi Lu, Xinjie Feng, Jiachen Shang, Rui Cao, Mengchao Zheng, Chuanye Wang, Shichun Yang, Yaoguang Cao, and Ziying Song. Focalad: Local motion planning for end-to-end autonomous driving. *arXiv.org*, 2506.11419, 2025. 13

[50] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2025. 12

[51] Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. Hip-ad: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2025. 2, 3, 12, 15

[52] Chi Wan, Yixin Cui, Jiatong Du, Shuo Yang, Yulong Bai, and Yanjun Huang. Geminus: Dual-aware global and scene-adaptive mixture-of-experts for end-to-end autonomous driving. *arXiv.org*, 2507.14456, 2025. 13

[53] Tao Wang, Cong Zhang, Xingguang Qu, Kun Li, Weiwei Liu, and Chang Huang. Diffad: A unified diffusion modeling approach for autonomous driving. *arXiv.org*, 2503.12170, 2025. 13

[54] Zhennan Wang, Jianing Teng, Canqun Xiang, Kangliang Chen, Xing Pan, Lu Deng, and Weihao Gu. Cogad: Cognitive-hierarchy guided end-to-end autonomous driving. *arXiv.org*, 2505.21581, 2025. 13

[55] Philipp Wolters, Johannes Gilg, Torben Teepe, and Gerhard Rigoll. Sparc-ad: A baseline for radar-camera fusion in end-to-end autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV) Workshops*, 2025. 13

[56] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 12, 15

[57] 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. *arXiv preprint arXiv:2503.05689*, 2025. 11

[58] Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv.org*, 2505.16278, 2025. 13

[59] Zhenjie Yang, Xiaosong Jia, Qifeng Li, Xue Yang, Maoqing Yao, and Junchi Yan. Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 12

[60] Jialun Yin, Kun Zhao, Xiaohan Ma, Siping Yan, Haoran Li, Junru Yang, and Yin Chen. Occlusion-aware interactive end-to-end autonomous driving for right-of-way conflicts. *Machines*, 2025. 12

[61] Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, and Marcelo H. Ang Jr. Drama: An efficient end-to-end motion planner for autonomous driving with mamba, 2024. 11

[62] Jiang-Tian Zhai, Ze Feng, Jihao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv.org*, 2023. 13

[63] Bozhou Zhang, Jingyu Li, Nan Song, and Li Zhang. Perception in plan: Coupled perception and planning for end-to-end autonomous driving. *arXiv.org*, 2508.11488, 2025. 13

[64] Bozhou Zhang, Nan Song, Xin Jin, and Li Zhang. Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 12

[65] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 12

[66] Wei Zhang, Pengfei Li, Junli Wang, Bingchuan Sun, Qihao Jin, Guangjun Bao, Shibo Rui, Yang Yu, Wenchao Ding, Peng Li, and Yilun Chen. Dual-aeb: Synergizing rule-based and multimodal large language models for effective emergency braking. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2025. 12

[67] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 12

[68] Julian Zimmerlin, Jens Beißwenger, Bernhard Jaeger, Andreas Geiger, and Kashyap Chitta. Hidden biases of end-to-end driving datasets. *arXiv.org*, 2412.09602, 2024. 2, 3, 6, 9, 13