SRTDRIVE: SAFETY-AWARE TRAJECTORY REFINEMENT VIA REFERENCE-RELATIVE SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-end planners output a single final plan that can still collide with surrounding agents, even when a small adjustment could avoid the collision. Retraining the planner changes its weights and acquired behavior, and learned candidate scorers are not trained to judge a replacement relative to the original plan. We propose SRTDrive, a post-hoc refinement framework that leaves the planner frozen, keeps its output unchanged as a reference trajectory, and learns whether to retain or replace it. A generator produces alternatives as deformations of the reference along and across its path, so that every alternative is directly comparable to the original plan. A selector then scores each candidate relative to the reference and is trained so that a replacement that increases collision at any horizon is never the target, however close it is to the logged trajectory; trajectory error only ranks equally safe candidates. At inference, the top-ranked candidate, possibly the reference itself, is output without a collision threshold. On Bench2Drive, SRTDrive raises the Driving Score of VAD-Base from 43.07 to 47.77 and reduces the routes with collisions from 95 to 86, whereas the same candidates selected by a TTC rule do not yield these gains. On nuScenes, applied to seven frozen planners without retraining, it lowers L2 error on five and the collision rate on three, with the largest gains on VAD. The code will be released at https://srtdriveiclr2027.github.io/.

1 INTRODUCTION

End-to-end autonomous driving (Hu et al., 2023; Jiang et al., 2023; Hu et al., 2022) increasingly plans by generating multiple candidate trajectories and selecting one among them (Liao et al., 2025; Zheng et al., 2025a; Jiang et al., 2026; Zheng et al., 2025b), often with candidate scores supervised by rule-based metrics such as collision (Li et al., 2024a; Ang et al., 2026; Dauner et al., 2024; Lu et al., 2023).

Nevertheless, the final plan still retains collision risk (Hu et al., 2023; Jiang et al., 2023; Liao et al., 2025) (Fig. 1(a)), even in cases where a small deviation from it would avoid the collision. Retraining the planner with collision signals is one remedy, but it modifies the planner's weights and the behavior it has already acquired. Candidate scorers are another, but they are not trained to judge a replacement relative to the original plan: the original is scored like any other candidate or used only as a deviation anchor (Ang et al., 2026), so whether a replacement introduces a collision that the original did not have never enters the supervision.

We propose **SRTDrive** (Fig. 1(b)), a post-hoc refinement framework that leaves the base planner untouched and trains a lightweight module on its outputs. SRTDrive keeps the planner's output unchanged as the reference trajectory in the candidate set and decides between retaining and replacing it by comparison with alternatives generated around it. A deformation-flow generator based on rectified flow (Liu et al., 2023) produces these alternatives as deformations of the reference along and across its path, so that every alternative is directly comparable to the original plan.

A set-contextual selector then scores every candidate relative to the reference. Its supervision encodes a simple priority: a replacement that makes collision worse than the original can never be the target, however close it is to the logged trajectory, and trajectory error decides only among equally

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

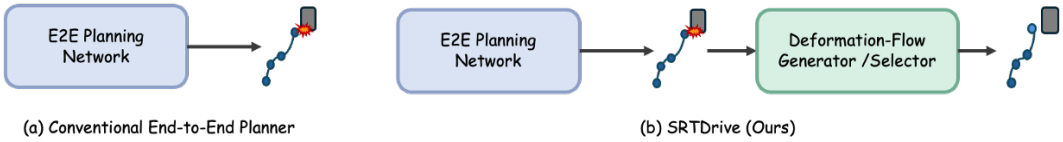


Figure 1: (a) A conventional end-to-end planner outputs a single final plan that may still collide with surrounding agents. (b) SRTDrive keeps the planner frozen and takes its output as the reference; the deformation-flow generator produces alternatives around it, and the selector either retains the reference or replaces it with an alternative, here removing the collision without retraining the planner.

safe candidates. Because the reference is scored as one candidate rather than assumed to be safe, a single selection function decides when to keep the plan and when to replace it: it replaces a risky plan with a safer alternative, replaces a safe plan only with an equally safe alternative closer to expert behavior, and never targets a less safe replacement. The top-ranked candidate is output as is without a collision threshold.

We evaluate SRTDrive in open loop on nuScenes with seven base planners and in closed loop on Bench2Drive (Jia et al., 2024). Our contributions are summarized as follows.

- A reference-relative post-hoc refinement framework that leaves the planner's weights intact, keeps the original plan as a candidate, and learns retention and replacement with a single selection function; Sec. 4.4 analyzes when it retains and when it replaces.
- A deformation-flow generator that produces alternatives as deformations of the reference, keeping them directly comparable to the original plan.
- A set-contextual selector trained with collision-prioritized lexicographic supervision, which outperforms a TTC rule applied to the same candidates in closed loop (Sec. 4.3).

2 Related Work

2.1 Generative Multimodal Trajectory Planning

Generative planners produce multiple scene-conditioned candidates via diffusion or flow matching (Liao et al., 2025; Zheng et al., 2025a; Xing et al., 2025), and ResAD (Zheng et al., 2026) learns residuals over a scene-agnostic inertial trajectory. In all of them the generative model is the primary planner, so generation and selection are coupled and no existing plan is preserved. SRTDrive instead uses a deformation-flow generator based on rectified flow (Liu et al., 2023) to produce deformations of the scene-aware base-planner output, conditioned on the reference trajectory, local BEV features, scene context, driving command, and ego state, thereby separating alternative generation from the decision to replace the plan.

2.2 Trajectory Refinement and Scoring

Refinement methods correct an initial plan with predicted future features (Zhang et al., 2026), a thresholded correction direction (He et al., 2026), or evaluator-guided search (Xu et al., 2026), modifying the original or deciding replacement by a separate threshold or post-hoc comparison. Several planners also refine their own output at inference: UniAD optimizes its plan against predicted occupancy (Hu et al., 2023), and SparseDrive re-scores its candidates with a collision-aware module (Sun et al., 2025). Scoring methods (Li et al., 2024a; 2025b;a; Ang et al., 2026) learn metric-based supervision but score each candidate independently and combine predicted metrics at inference, so the original plan, when present, is judged by the same stand-alone criterion as any alternative. The closest to our setting is the anchor-assisted reranking of CLOVER (Ang et al., 2026), which adds a penalty on the deviation from a single-mode planner's output to independently predicted scores. There the anchor limits how far a replacement may move, but the scores are not conditioned on the anchor, and whether a replacement is safer or less safe than the anchor is not supervised. SRTDrive conditions every score on the reference and supervises exactly this comparison, without combining metrics or applying a safety threshold at inference.

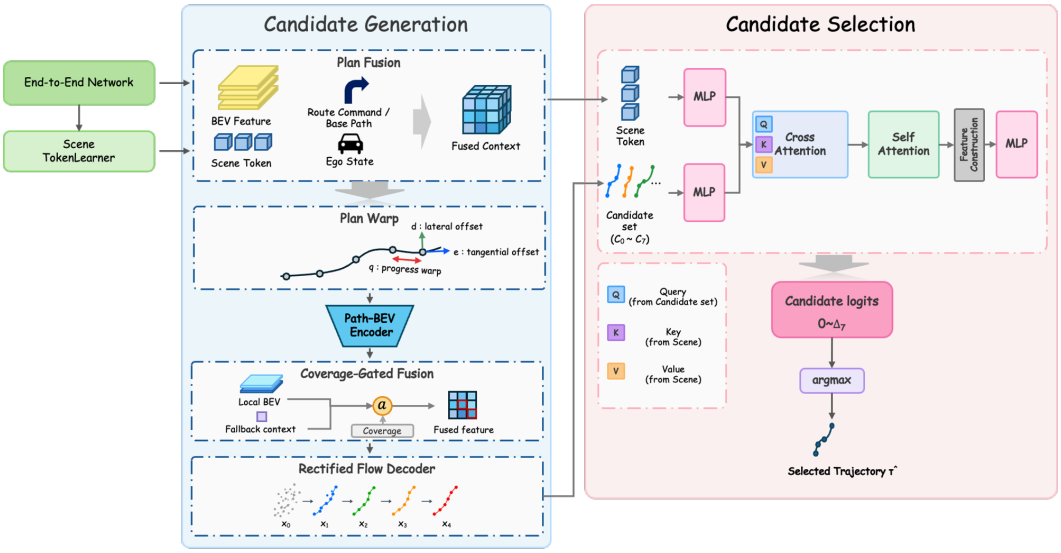


Figure 2: Overview of SRTDrive. A frozen base planner provides BEV features and a single planned trajectory, which is kept unchanged as the reference τ_0. The deformation-flow generator produces K alternatives as progress and normal–tangential deformations of the reference, conditioned on the reference, path-local BEV features, scene tokens obtained with a TokenLearner-style module (Ryoo et al., 2021), driving command, and ego state. The set-contextual selector encodes all candidates including the reference, contextualizes them through scene–candidate and candidate–candidate attention, predicts a selection logit ρ_k from each candidate's relation to the reference, and outputs the top-ranked candidate as is. Collision labels and logged trajectories are used only to construct training targets. In the figure, q, d, and e denote the progress, normal, and tangential deformations (δ^s, δ^n, $\delta^{\|}$ in Eq. (2); q is the progress state of App. A.3), and C_0–C_7 denote the candidates τ_0–τ_7.

2.3 Prioritized Objectives in Planning

Prioritizing safety over other objectives is established in rule-based planning. Rulebooks (Censi et al., 2019) order driving rules hierarchically so that violating a higher rule cannot be compensated by satisfying lower ones, and the PDM score of NAVSIM (Dauner et al., 2024) multiplies collision and drivable-area terms with a weighted average of the remaining sub-scores, so that collisions reduce the score multiplicatively rather than additively. These priorities are applied when specifying or evaluating behavior, whereas learned scorers that predict such sub-scores still combine them with tunable weights at inference (Li et al., 2024a), which permits trade-offs between collision and other terms. SRTDrive instead builds a collision-first priority into the training target of a learned selector and defines it relative to the original plan rather than in absolute terms.

3 Method

Fig. 2 shows the overall framework. Given the BEV representation $\mathbf{B}$, the reference trajectory τ^{ref}, the driving command, and the ego state from a frozen base planner, SRTDrive outputs one trajectory from a candidate set that contains the reference, and is trained to prioritize collision non-worsening relative to the reference and, at equal collision level, to prefer candidates closer to the logged trajectory. A deformation-flow generator produces the alternatives as progress and normal/tangential deformations of the reference (Sec. 3.2), and a set-contextual selector predicts the relative selection priority of every candidate, including the reference, and outputs one candidate as is (Secs. 3.3–3.5).

3.1 Frozen Planner Interface and Path-Aware Conditioning

From a pretrained planner F_ψ with frozen ψ, we obtain BEV features $\mathbf{B} \in \mathbb{R}^{H\times W\times C}$ and the reference trajectory $\tau^{\mathrm{ref}} = (\mathbf{p}_1^{\mathrm{ref}}, \ldots, \mathbf{p}_T^{\mathrm{ref}})$ in the ego frame, $(\mathbf{B}, \tau^{\mathrm{ref}}) = F_\psi(\mathbf{O}, \mathbf{e}, u)$, where $\mathbf{O}$, $\mathbf{e}$, and

u are the multi-view observations, ego state, and driving command. With K generated alternatives, the candidate set is $\mathcal{T} = \{\tau_0, \ldots, \tau_K\}$ with $\tau_0 = \tau^{\text{ref}}$, and $\mathcal{V} \subseteq \{0, \ldots, K\}$ denotes the candidates whose trajectory representation is well-formed (finite deformation, admissible per-step motion, and a consistent initial connection; Appendix A.1); the reference always belongs to $\mathcal{V}$. For planners whose scene representation is not a dense BEV map (SparseDrive and DiffusionDrive), a learnable spatial adapter aggregates their native object and map features into a spatial grid that serves as $\mathbf{B}$ (Appendix B.1).

Useful alternatives depend on what lies along the current plan, so we query BEV features around the reference path. With $\mathbf{R}_t^{\text{ref}}$ the tangent–normal frame at timestamp t and $\{\boldsymbol{\delta}_m\}_{m=1}^{M^G}$ sampling offsets (M^G=5),

$$\begin{aligned} \boldsymbol{\ell}_t^{\text{ref}} &= E_{\text{loc}}^G\left(\left[\mathbf{B}\big(\mathbf{p}_t^{\text{ref}} + \mathbf{R}_t^{\text{ref}}\boldsymbol{\delta}_m\big)\right]_{m=1}^{M^G}\right), \\ \mathbf{c}_{\text{ref}} &= E_{\text{cond}}^G\Big(E_{\text{temp}}^G\big([\mathbf{p}_t^{\text{ref}}, \Delta\mathbf{p}_t^{\text{ref}}, \boldsymbol{\ell}_t^{\text{ref}}, \text{PE}(t)]_{t=1}^T\big), \mathbf{S}, \mathbf{e}, u\Big), \end{aligned} \tag{1}$$

where $\mathbf{B}(\cdot)$ is bilinear sampling on the BEV grid, positions outside the BEV extent return zero features with a binary coverage indicator concatenated to $\boldsymbol{\ell}$, $\Delta\mathbf{p}_t^{\text{ref}}$ is the inter-waypoint displacement, $\text{PE}(t)$ a temporal positional encoding, and the superscript G marks generator-owned encoders. The global scene tokens $\mathbf{S} \in \mathbb{R}^{16\times256}$ are pooled from command-modulated $\mathbf{B}$ with learned spatial weights in the manner of TokenLearner (Ryoo et al., 2021) (Appendix A.2). The condition $\mathbf{c}_{\text{ref}}$ thus encodes where, and with what temporal pattern, the current plan travels, together with the global scene.

3.2 Alternative Generation via Deformation Flow

Alternatives generated in absolute coordinates would not be tied to the existing plan, which makes them hard to compare with it. We therefore parameterize each alternative as a deformation of the reference along progress, normal, and tangential axes. Let $\mathbf{r}_{\text{ref}}(s)$ be the interpolated reference path with arc length s, s_t^{ref} the arc length at waypoint t, and $\mathbf{t}_{\text{ref}}, \mathbf{n}_{\text{ref}}$ its tangent and normal. From $\mathbf{z}_k = [\delta_{k,t}^s, \delta_{k,t}^n, \delta_{k,t}^{\parallel}]_{t=1}^T$,

$$\mathbf{p}_{k,t} = \mathbf{r}_{\text{ref}}(\widetilde{s}_{k,t}) + \delta_{k,t}^n\,\mathbf{n}_{\text{ref}}(\widetilde{s}_{k,t}) + \delta_{k,t}^{\parallel}\,\mathbf{t}_{\text{ref}}(\widetilde{s}_{k,t}), \qquad \widetilde{s}_{k,t} = s_t^{\text{ref}} + \delta_{k,t}^s, \tag{2}$$

denoted $\tau_k = D(\tau^{\text{ref}}, \mathbf{z}_k)$. Progress deformation expresses temporal adjustments such as deceleration; normal and tangential deformations express lateral and longitudinal displacements. The progress and tangential axes coincide in direction on straight segments but play different roles: progress is monotone and accumulates along the path, encoding the speed profile, whereas the tangential offset is a small local correction (bounded by 0.5 m; Appendix A.1) that absorbs residuals of the monotone matching. Degenerate (near-)stationary references, whose path has vanishing length, are handled separately (Appendix A.3). The reference candidate τ_0 bypasses generation and decoding and preserves the input reference trajectory as is.

We learn the conditional distribution of deformations with rectified flow (Liu et al., 2023), applied unchanged in deformation space. Given a target deformation $\mathbf{z}^\star$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\xi \sim \mathcal{U}(0,1)$, and $\mathbf{z}_\xi = (1-\xi)\boldsymbol{\epsilon} + \xi\mathbf{z}^\star$,

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\left[\left\|v_\theta(\mathbf{z}_\xi, \xi; \mathbf{c}_{\text{ref}}) - (\mathbf{z}^\star - \boldsymbol{\epsilon})\right\|_2^2\right]. \tag{3}$$

At generation, K deformations are obtained by integrating $d\widehat{\mathbf{z}}_k/d\xi = v_\theta(\widehat{\mathbf{z}}_k, \xi; \mathbf{c}_{\text{ref}})$ from $\widehat{\mathbf{z}}_k(0) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and decoded as $\tau_k = D(\tau^{\text{ref}}, \widehat{\mathbf{z}}_k(1))$. The generator only constructs alternatives; the final choice is left to the selector.

Supervision targets. Targets $\mathbf{z}^\star$ are drawn from a per-scene pool with three slots: the reference itself ($\mathbf{z}^\star = \mathbf{0}$), the logged ego future, and repair trajectories. Repair trajectories are produced by offline optimization that takes the reference, GT future object boxes, and the drivable area, reduces collision and off-road costs while suppressing deformation from the reference, and must pass explicit collision, off-road, and geometric checks; this optimization does not use the logged ego future, and all GT enters SRTDrive only through training targets (Appendix A.4). For the logged trajectory,

E_{def} aligns each waypoint to a reference arc length by dynamic programming under a monotone arc-length constraint (Appendix A.3); repair trajectories provide their optimized deformation directly. The identity slot anchors the distribution at the plan itself, so the generator learns adjustments around the plan rather than a single logged mode.

3.3 Set-Contextual Trajectory Selector

Scoring each candidate in isolation ignores how it differs from the original plan and from the other candidates in the same scene. Whether a replacement is worthwhile also depends on the alternatives available: which of several equally safe candidates is best depends on what the others offer, so the selector compares candidates jointly rather than one at a time. It encodes each candidate τ_k from its positions, their first and second differences, and path-local features $\boldsymbol{\ell}_{k,t}$ extracted as in Eq. (1) with its own encoder and $M^S{=}3$ offsets, and adds a role embedding that marks the reference:

$$\mathbf{q}_k = E^S_{\text{traj}}\big(\tau_k, \boldsymbol{\ell}_k; \mathbf{e}, u\big) + \mathbf{r}_k, \qquad \mathbf{r}_k = \begin{cases} \mathbf{r}_{\text{ref}}, & k = 0, \\ \mathbf{r}_{\text{alt}}, & k > 0. \end{cases} \tag{4}$$

The tokens attend to the scene tokens $\mathbf{S}$ and then to each other through self-attention biased by a pairwise geometric descriptor $\mathbf{g}_{ij} = \Gamma(\tau_i, \tau_j) \in \mathbb{R}^8$ (same-timestamp distances, endpoint offsets, path-length and heading differences), yielding contextualized representations $\mathbf{h}_k$ over the candidates in $\mathcal{V}$ (Appendix A.6). The generator and the selector share $\mathbf{B}$ and $\mathbf{S}$ as inputs but no parameters, and the generator is frozen during selector training. The selection logit is

$$\rho_k = f_\phi([\mathbf{h}_k,\ \mathbf{h}_0,\ \mathbf{h}_k - \mathbf{h}_0,\ \mathbf{g}_{k0}])\,. \tag{5}$$

Every logit, including that of the reference, is thus computed from a candidate–reference pair: $\mathbf{h}_k - \mathbf{h}_0$ and $\mathbf{g}_{k0}$ supply semantic and geometric contrast, and for $k = 0$ the contrast vanishes, so ρ_0 acts as the baseline against which replacements are ranked. The score is therefore relative by construction rather than a stand-alone quality estimate. ρ_k is a selection logit, not a collision probability, and is never compared against a threshold.

3.4 Collision-Prioritized Lexicographic Supervision

Displacement error alone would target the candidate closest to the log and penalize safe alternatives, while a weighted sum of collision and displacement lets a small displacement gain offset a collision increase. We therefore define a lexicographic target, which we call reference-relative supervision because its first priority is defined against the original plan: non-worsening of collision relative to the reference, then minimum collision, then displacement.

With $\Delta t = 0.5\,\text{s}$ and horizons $\mathcal{H} = \{1, 2, 3\}\,\text{s}$, $c_{k,h}$ is the fraction of timesteps up to h at which the candidate ego footprint overlaps the occupancy of GT future objects (zeroed where the GT ego itself collides, following the VAD convention), and d_k is the mean over horizons of the per-horizon average L2 error to the logged trajectory. Let $\mathcal{V}_c \subseteq \mathcal{V}$ be the candidates with valid collision labels ($0 \in \mathcal{V}_c$). The worsening amount and the non-worsening set are

$$\Delta^+_k = \max_{h \in \mathcal{H}} [c_{k,h} - c_{0,h}]_+\,, \qquad \mathcal{A} = \{k \in \mathcal{V}_c : \Delta^+_k = 0\}, \tag{6}$$

so an increase at one horizon cannot be offset by a decrease at another. Since the reference cannot worsen relative to itself, $0 \in \mathcal{A}$ always holds; this follows from including the baseline in the set, not from assuming it safe. Within $\mathcal{A}$ we take the minimum-average-collision subset $\mathcal{M} = \{k \in \mathcal{A} : \bar{c}_k = \min_{j \in \mathcal{A}} \bar{c}_j\}$ with $\bar{c}_k = |\mathcal{H}|^{-1} \sum_h c_{k,h}$, and define the target from d_k only inside $\mathcal{M}$:

$$y_k = \begin{cases} \exp(-\beta d_k) \big/ \sum_{j \in \mathcal{M}} \exp(-\beta d_j), & k \in \mathcal{M}, \\ 0, & k \notin \mathcal{M}, \end{cases} \qquad \beta > 0. \tag{7}$$

If $\mathcal{M} = \{0\}$, keeping the original is the target; if a candidate reduces collision without worsening at any horizon, the reference drops out of $\mathcal{M}$ and replacement becomes the target; if all candidates in $\mathcal{V}_c$ are collision-free, $\mathcal{M} = \mathcal{A}$ and d_k orders them. Candidates outside $\mathcal{M}$, including those without collision labels, receive $y_k = 0$, and displacement never compensates for a collision increase.

Requiring non-worsening at every horizon makes the target favor the reference whenever collisions are involved: a replacement that removes a later collision but introduces an earlier one is never

preferred to the reference. The priority thus acts exactly on the sets in which a replacement could trade safety for accuracy, and leaves the others to be ordered by d_k; on nuScenes these form the vast majority (Appendix C.1), so replacement toward expert-like alternatives is the common outcome (Sec. 4.4).

The selector is trained by cross-entropy between the softmax of logits and y, plus a relative penalty ensuring that no worsening candidate $k \in \mathcal{B} = \mathcal{V}_c \setminus \mathcal{A}$ outranks the reference:

$$\mathcal{L}_{\text{select}} = -\sum_{k\in\mathcal{V}} y_k \log \frac{\exp \rho_k}{\sum_{j\in\mathcal{V}} \exp \rho_j} + \frac{\lambda_{\text{rel}}}{\max(1, |\mathcal{B}|)} \sum_{k\in\mathcal{B}} \left[\rho_k - \rho_0 + m_0 + m_1 \Delta_k^+\right]_+, \quad (8)$$

with base margin $m_0 > 0$ and worsening-proportional margin $m_1 \geq 0$. The first term learns which candidate to select within the set, so no separate keep-versus-replace classifier is needed; the second directly supervises the order between worsening replacements and keeping the original. In practice we add an auxiliary term that further penalizes candidates whose 1 s collision exceeds that of the reference; the implemented loss terms and their weights are given in Appendix A.6.

3.5 Training and Inference

Training proceeds in two stages. The generator is trained with Eq. (3) on reference trajectories and BEV features from the frozen planner (schedule in Appendix A.5). The selector is then trained with Eq. (8), in practice by fine-tuning an earlier checkpoint (Appendix A.6), on a fixed candidate cache produced by the same frozen generator and settings as inference (K=7 alternatives, four Euler steps, three noise seeds per scene), on which $c_{k,h}$ and d_k are computed offline. Neither loss updates the planner or corrects coordinates. Collision metrics, displacement errors, the supervision pool, and the sets $\mathcal{A}, \mathcal{M}$ define training targets only; they are not inputs or filters at inference. At inference,

$$\widehat{k} = \arg\max_{k\in\mathcal{V}} \rho_k, \qquad \widehat{\tau} = \tau_{\widehat{k}}, \quad (9)$$

so the planner's output is kept when $\widehat{k} = 0$ and the selected alternative is output otherwise, with no coordinate regression, averaging, or collision-score gate. Sec. 4.4 analyzes how these decisions affect collision relative to the reference.

4 Experiments

4.1 Dataset and Metric

nuScenes. We conduct the open-loop evaluation on the nuScenes (Caesar et al., 2020) validation split (150 scenes, 6,019 samples). Metrics are the L2 displacement error (m) between the planned and logged trajectories and the collision rate (%) with ground-truth future objects, each averaged over the 1, 2, and 3 s horizons. Open-loop metrics on nuScenes are known to be sensitive to ego-status shortcuts (Li et al., 2024b); we therefore complement them with closed-loop evaluation.

Bench2Drive. We conduct the closed-loop evaluation on the 220 official routes of Bench2Drive (Jia et al., 2024) in CARLA and report the benchmark's overall metrics and multi-ability scores, all computed with the official evaluation code. Driving Score (DS) weights route completion by infraction penalties, Success Rate (SR) is the fraction of routes that satisfy the success criteria, Efficiency is the ego speed relative to surrounding traffic (and can exceed 100), and Comfort is the fraction of driving segments satisfying the acceleration, jerk, and yaw criteria. Multi-ability scores measure success on route segments requiring merging, overtaking, emergency braking, giving way, and traffic-sign compliance. We additionally count the routes whose infraction log contains at least one collision; this is not an official benchmark metric but directly reflects the collision behavior SRTDrive targets.

4.2 Implementation Details

nuScenes. We apply SRTDrive to seven base planners with released checkpoints: VAD (Jiang et al., 2023), UniAD (Hu et al., 2023), GenAD (Zheng et al., 2024), DMAD (Shen et al., 2025), IR-WM (Mei et al., 2026), SparseDrive (Sun et al., 2025), and DiffusionDrive (Liao et al., 2025).

Table 1: Open-loop results on nuScenes validation. L2 (m) and collision rate (%) are averaged over 1, 2, and 3 s. "+SRT" applies SRTDrive on top of the frozen base planner. Each planner is evaluated with its own protocol, so values are comparable within a row but not across rows. The better value in each pair is in bold.

Method	L2 (m) ↓		Collision Rate (%) ↓	
	Base	+SRT	Base	+SRT
VAD (Jiang et al., 2023)	0.723	**0.677**	0.165	**0.125**
UniAD (Hu et al., 2023)	1.086	**1.079**	0.282	**0.277**
GenAD (Zheng et al., 2024)	0.542	**0.541**	0.183	**0.167**
DMAD (Shen et al., 2025)	0.619	**0.614**	**0.147**	0.161
IR-WM (Mei et al., 2026)	0.901	**0.888**	0.644	0.644
SparseDrive (Sun et al., 2025)	**0.608**	0.613	**0.103**	0.109
DiffusionDrive (Liao et al., 2025)	**0.567**	0.573	**0.079**	0.084

Each base planner is frozen, and a separate SRTDrive module is trained for it on the 700 training scenes (28,130 frames) as described in Sec. 3.5. The generator produces $K=7$ alternatives with four Euler steps. The selector is trained with AdamW (learning rate 5×10^{-5}, effective batch size 8) on 8 NVIDIA RTX A5000 GPUs. Each planner is evaluated with its own released evaluation code, and we compare the reference plan and the SRTDrive output within each planner (Appendix B.1). Further details are given in Appendix A.

Bench2Drive. Among the open-loop planners, we use VAD as the representative base planner for closed-loop evaluation and for the analyses in Secs. 4.4–4.5, since it is a standard end-to-end baseline with an official Bench2Drive implementation. We use VAD-Base with the same weights and PID controller as the official Bench2Drive baseline. SRTDrive is trained with this planner frozen on the full Bench2Drive training set, with collision labels and repair trajectories constructed as in Sec. 3.2 from the ground-truth agent annotations of the dataset. At every step SRTDrive selects one trajectory, which is passed unchanged to the controller; generating seven alternatives and selecting adds 98 ms per step on average, about 11% of the 876 ms per step taken by VAD-Base in our Bench2Drive runs on an RTX A5000 GPU. As a rule-based alternative to the learned selector, a TTC variant keeps the same candidates and selects among them with a hand-designed rule based on the time to collision with surrounding agents (Appendix B.2).

4.3 Main Results

nuScenes. Table 1 reports within-planner comparisons. Without retraining any planner, SRTDrive lowers L2 on five of the seven planners and the collision rate on three, with the clearest gain on VAD (L2 from 0.723 to 0.677, collision from 0.165 to 0.125), where it also halves the 1 s collision rate (Appendix C.2). The size of the gain depends on how much room the reference leaves for local refinement: several planners already refine their own output at inference, e.g., UniAD against predicted occupancy and SparseDrive with collision-aware re-scoring (Sec. 2.2), and on the other six planners the changes stay within 0.016 in either metric. Appendix C.3 discusses SparseDrive and DiffusionDrive, whose sparse scene features enter SRTDrive through a spatial adapter.

Bench2Drive. Table 2(a) shows that SRTDrive raises the Driving Score of VAD-Base from 43.07 to 47.77 and the Success Rate from 16.82% to 19.09%, i.e., five additional successful routes out of 220, with higher Efficiency (163.84 to 168.60). At the same time, SRTDrive completes more of each route (RC 63.64% to 70.04%) yet reduces the number of routes with at least one collision from 95 to 86. Driving farther while colliding on fewer routes suggests that the collision reduction is consistent in closed loop rather than a byproduct of more conservative, shorter runs; since Coll. is not normalized by driven distance, the reduction per distance is larger than the per-route count suggests.

Learned versus rule-based selection. To test whether this gain comes from the candidates or from the selector, we keep the same candidates and select among them with the TTC rule. This variant reaches the highest Route Completion (79.69%), but collisions occur on 172 of the 220 routes, nearly twice as many as the base planner, and its Driving Score (39.56) and Success Rate

Table 2: Closed-loop results on the 220 official Bench2Drive routes. (a) Overall driving performance: Driving Score (DS), Success Rate (SR), Efficiency, and Comfort, with Route Completion (RC) and the number of routes containing at least one collision (Coll., out of 220) as auxiliary metrics. (b) Multi-ability success rates (%); Mean is the average of the five abilities. All rows use the same VAD-Base weights and PID controller. "+SRT" passes the trajectory selected by SRTDrive to the controller; "+SRT (TTC rule)" keeps the same candidates but replaces the learned selector with a TTC-based rule. The best value in each column is in bold.

Method	Closed-Loop Metric				Auxiliary	
	DS ↑	SR (%) ↑	Efficiency ↑	Comfort (%) ↑	RC (%) ↑	Coll. ↓
VAD-Base (Jiang et al., 2023)	43.07	16.82	163.84	**50.80**	63.64	95
VAD-Base + SRT (TTC rule)	39.56	12.27	165.67	33.73	**79.69**	172
VAD-Base + SRT	**47.77**	**19.09**	**168.60**	40.26	70.04	**86**

Method	Multi-Ability Success Rate (%) ↑					
	Merging	Overtaking	Emerg. Brake	Give Way	Traffic Sign	Mean
VAD-Base (Jiang et al., 2023)	11.25	**20.00**	20.00	30.00	23.68	20.99
VAD-Base + SRT (TTC rule)	10.00	13.33	8.33	**40.00**	26.32	19.60
VAD-Base + SRT	**12.50**	11.11	**33.33**	20.00	**30.00**	**21.39**

(12.27%) fall below even those of the base planner. The same candidates selected by a simple TTC rule therefore do not yield these gains: the rule trades route progress for frequent collisions, whereas SRTDrive improves both.

Ability breakdown and comfort. Across abilities (Table 2(b)), SRTDrive yields its largest gain in Emergency Brake (20.00 to 33.33), consistent with progress deformations that express deceleration along the reference, followed by Traffic Sign (23.68 to 30.00) and Merging (11.25 to 12.50), raising the ability mean from 20.99 to 21.39, whereas the TTC rule collapses in Emergency Brake (8.33). Overtaking (20.00 to 11.11) and Give Way (30.00 to 20.00) decrease; overtaking can require a larger departure from the reference path than local deformations provide, and since each ability is scored on a subset of the 220 routes, per-ability differences can correspond to only a few routes. Comfort is lower for both variants that switch candidates at every step (40.26% for SRTDrive and 33.73% for the TTC rule, versus 50.80%), as independent per-step selection can change the commanded trajectory between consecutive steps; enforcing temporal consistency is a natural extension.

4.4 Analysis

Retention and replacement. For the VAD-based model on nuScenes, the selector retains the reference in 22.1% of the 5,119 valid samples and replaces it in the remaining 3,987. At every horizon, more than 99.3% of replacements leave the collision metric unchanged; at 3 s, 25 replacements lower it and 2 raise it (Appendix C.4). Replacement is thus the common case: as intended by the supervision, the reference is kept only when no equally safe alternative is closer to the logged trajectory. Most replacements reorder collision-free candidates by trajectory error, which accounts for the L2 gain, whereas the collision gain comes from a small number of replacements that remove a collision, with very few that introduce one.

4.5 Visualization

Fig. 3 shows representative scenes. In each, the SRTDrive output stays close to the logged trajectory, whereas the VAD reference drifts from it, toward the parked vehicles in the top scene and across the lane toward a vehicle in the middle scene. SRTDrive corrects these plans with small lateral and progress adjustments of the reference rather than a new route, consistent with the deformation parameterization of Sec. 3.2.

5 Conclusion

We presented SRTDrive, a post-hoc refinement framework that leaves an end-to-end planner untouched, keeps its output as a reference, and learns whether to retain or replace it with alternatives



Figure 3: Qualitative results on nuScenes validation with VAD as the base planner. Left: front camera. Right: BEV with ground-truth map elements and agents (gray boxes: vehicles, triangles: pedestrians), the logged future (black, dotted), the VAD reference (orange), the TTC rule of Table 2 applied to the same candidates (red, dashed), and the SRTDrive output (teal).

generated as deformations of that reference, under supervision that never prefers a replacement that makes collision worse. SRTDrive is most effective when the base planner leaves collisions that a local adjustment can remove: in closed loop on Bench2Drive, it raises the Driving Score of VAD-Base from 43.07 to 47.77 while reducing the routes with collisions from 95 to 86, whereas the same candidates chosen by a TTC rule do not yield these gains, and on nuScenes it lowers L2 error on five of seven frozen planners, with the largest gains on VAD. On planners that already refine their own output, the changes are small.

Limitations and Future Work. The collision priority is imposed on training targets rather than enforced at inference, and its labels come from recorded data. Selection is made independently at each step, which lowered comfort in closed loop, and alternatives are confined to deformations of the reference. Future work will pursue temporally consistent selection, closed-loop evaluation across more planners, and extension to planners that emit multiple references.

USE OF LARGE LANGUAGE MODELS

Large language models were used as assistants for writing the manuscript in English and for writing code. The reasoning, methods, and conclusions of the paper, as well as all code and experimental results, were reviewed and verified directly by the authors, who take full responsibility for the content.

REFERENCES

Sining Ang, Yuguang Yang, Canyu Chen, and Yan Wang. CLOVER: Closed-loop value estimation and ranking for end-to-end autonomous driving planning. *arXiv preprint arXiv:2605.15120*, 2026.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11621–11631, 2020.

Andrea Censi, Konstantin Slutsky, Tichakorn Wongpiromsarn, Dmitry Yershov, Scott Pendleton, James Fu, and Emilio Frazzoli. Liability, ethics, and culture-aware behavior specification using rulebooks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Chong He, Yuechen Luo, Fang Li, Shaoqing Xu, and Fuxi Wen. DriveVer: Lightweight trajectory evaluator as test-time verifier for autonomous driving. *arXiv preprint arXiv:2607.00399*, 2026.

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision (ECCV)*, pp. 533–549, 2022.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862, 2023.

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), Datasets and Benchmarks Track*, 2024.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-end vectorized autonomous driving via probabilistic planning. In *International Conference on Learning Representations (ICLR)*, 2026.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, Yu-Gang Jiang, and Jose M. Alvarez. Hydra-MDP: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024a.

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*, 2025b.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M. Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Rebecca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, Dragomir Anguelov, and Sergey Levine. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

Jianbiao Mei, Yu Yang, Xuemeng Yang, Licheng Wen, Jiajun Lv, Botian Shi, and Yong Liu. Vision-centric 4d occupancy forecasting and planning via implicit residual world models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. TokenLearner: Adaptive space-time tokenization for videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yinzhe Shen, Ömer Şahin Taş, Kaiwen Wang, Royden Wagner, and Christoph Stiller. Divide and merge: Motion and semantic learning in end-to-end autonomous driving. *Transactions on Machine Learning Research*, 2025.

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

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Yihong Xu, Éloi Zablocki, Yuan Yin, Elias Ramzi, Ellington Kirby, Alexandre Boulch, and Matthieu Cord. Test-time trajectory optimization for autonomous driving. *arXiv preprint arXiv:2606.07170*, 2026.

Jinqing Zhang, Zehua Fu, Zelin Xu, Wenying Dai, Qingjie Liu, and Yunhong Wang. ResWorld: Temporal residual world model for end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2026.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. GenAD: Generative end-to-end autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2024.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-based planning for autonomous driving with flexible guidance. In *International Conference on Learning Representations (ICLR)*, 2025a.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, and Dongbin Zhao. World4Drive: End-to-end autonomous driving via intention-aware physical latent world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3729–3739, 2026.

A Implementation Details

A.1 Candidate Validity

The reference candidate τ_0 is admitted after input validation (finite coordinates and a consistent ego-frame transform). An alternative τ_k, $k \geq 1$, is valid if all of the following hold: (i) the generated deformation $\widehat{\mathbf{z}}_k(1)$ and the decoded coordinates are finite; (ii) the deformation components lie within admissible ranges: the progress component satisfies $|q_{k,t}| \leq 30$ (dimensionless, in the inverse-softplus space of App. A.3), and the normal and tangential components satisfy $|\delta^n_{k,t}| \leq 4\,\mathrm{m}$ and $|\delta^{\parallel}_{k,t}| \leq 0.5\,\mathrm{m}$; (iii) the decoded arc length extends at most 15 m beyond the last reference waypoint; (iv) the per-step displacement satisfies $\|\Delta \mathbf{p}_{k,t}\|_2 \leq 10\,\mathrm{m}$; (v) the second-order position difference satisfies $\|\Delta^2 \mathbf{p}_{k,t}\|_2 \leq 6\,\mathrm{m}$; and (vi) the first waypoint lies within 4 m of the reference path. Validity concerns the trajectory representation only; it does not depend on BEV coverage. Waypoints outside the BEV extent receive zero path-local features together with a binary coverage indicator (Sec. 3.1).

A.2 Global Scene Tokens

The scene tokens $\mathbf{S} \in \mathbb{R}^{16\times 256}$ are built from the BEV features $\mathbf{B}$ as follows. The channels of $\mathbf{B}$ are modulated by the driving command u through squeeze-and-excitation-style channel gating, a 2D positional encoding is concatenated, and the resulting map is pooled into 16 tokens with learned spatial weights (a softmax over the grid per token), following TokenLearner (Ryoo et al., 2021), followed by a linear projection to 256 dimensions and a single-layer Transformer encoder with 8 heads. The same $\mathbf{S}$ is consumed by the generator condition encoder and the selector cross-attention, but the two modules own separate parameters.

A.3 Progress-Matching Encoder E_{def}

Given the reference polyline and a supervision trajectory τ^{sup}, each waypoint $\mathbf{p}^{\mathrm{sup}}_t$ is projected onto every segment of the reference polyline and onto an end extension of length 15 m beyond the last waypoint. Dynamic programming selects one matched arc length s^{sup}_t per waypoint under the constraint $s^{\mathrm{sup}}_1 \leq \cdots \leq s^{\mathrm{sup}}_T$, minimizing a weighted sum of five costs: the projection distance (weight 1.0), the direction difference between the trajectory step and the reference tangent (0.05), a segment-jump cost on the change of matched segment between consecutive waypoints (0.01), and progress-continuity (0.05) and lateral-continuity (0.02) costs between consecutive waypoints. The normal and tangential components are the residual of $\mathbf{p}^{\mathrm{sup}}_t$ with respect to $\mathbf{r}_{\mathrm{ref}}(s^{\mathrm{sup}}_t)$ in the local frame. The progress component is encoded in inverse-softplus space by comparing the matched arc-length increment $\Delta s^{\mathrm{sup}}_t = s^{\mathrm{sup}}_t - s^{\mathrm{sup}}_{t-1}$ with the reference segment length $\Delta s^{\mathrm{ref}}_t = s^{\mathrm{ref}}_t - s^{\mathrm{ref}}_{t-1}$. Writing sp for softplus,

$$
\begin{aligned}
q_t &= \mathrm{sp}^{-1}\big(\Delta s^{\mathrm{sup}}_t\big) - \mathrm{sp}^{-1}\big(\Delta s^{\mathrm{ref}}_t\big), \\
\delta^s_{k,t} &= \sum_{t' \leq t} \Big[\mathrm{sp}\big(q_{t'} + \mathrm{sp}^{-1}(\Delta s^{\mathrm{ref}}_{t'})\big) - \Delta s^{\mathrm{ref}}_{t'}\Big],
\end{aligned}
\tag{10}
$$

so that the decoder in Eq. (2) operates on arc-length offsets while the flow operates on q_t, which keeps matched increments positive by construction. Eq. (10) states the relation in exact arithmetic; the implementation additionally stabilizes sp^{-1} numerically near zero and treats (near-)stationary segments, where the increments vanish, separately. The identity slot uses $\mathbf{z}^\star = \mathbf{0}$ and repair trajectories use the deformation state returned by the optimizer directly, without re-matching.

A.4 Repair Trajectory Optimization

For each training frame, repair trajectories are obtained by optimizing the deformation state $\mathbf{z}$ with the objective

$$
J(\mathbf{z}) = \sum_{j \in \mathcal{J}} w_j\, J_j(\mathbf{z}), \qquad \mathcal{J} = \{\mathrm{ref}, \mathrm{col}, \mathrm{road}, \mathrm{bev}, \mathrm{prog}, \mathrm{sm}, \mathrm{bnd}\}, \tag{11}
$$

where each term is evaluated on $\mathbf{z}$ and the decoded trajectory $D(\tau^{\mathrm{ref}}, \mathbf{z})$: J_{ref} penalizes deviation from the reference path, J_{col} penalizes overlap between the ego footprint and the occupancy of GT

future objects over the horizon, J_{road} penalizes leaving the drivable area, J_{bev} penalizes leaving the BEV extent, J_{prog} penalizes insufficient progress, J_{sm} penalizes non-smooth motion, and J_{bnd} penalizes violations of the admissible deformation bounds. The weights are $w_{\text{ref}} = 1$, $w_{\text{col}} = 12$, $w_{\text{road}} = 8$, $w_{\text{bev}} = 12$, $w_{\text{prog}} = 4$, $w_{\text{sm}} = 0.2$, and $w_{\text{bnd}} = 2$. The bound penalty applies a hinge loss to values exceeding the limits of App. A.1 on the inverse-softplus progress state, the normal offset, and the tangential offset. The optimizer runs Adam (learning rate 0.05) for 80 iterations from eight initial states, and a result is kept only if it passes explicit checks, including zero collision at every timestep, no departure from the BEV extent, and an off-road waypoint ratio of at most 0.20. The optimizer does not access the logged ego future. Frames without a valid repair contribute only the identity and logged slots. Among valid target slots, the identity, logged, and repair slots are weighted $4:2:1$ when the reference is safe, $0.25:1:4$ when it is risky and a valid repair exists, and $4:1:0$ when it is risky without one; the weights are renormalized per frame after invalid slots are masked.

A.5 Generator Training Schedule

The generator is first trained with Eq. (3) for 8 epochs on 8 GPUs with one frame per GPU, using AdamW with a learning rate of 2×10^{-4}. In a final coverage-adaptation phase of 8 epochs (8 GPUs, batch size 1 per GPU), the same objective is used to update only the condition encoder E^G_{cond} and the coverage-aware context module, while E^G_{loc}, E^G_{temp}, and the flow decoder v_θ remain frozen from the first stage; this phase adapts the conditioning to the coverage indicator without changing the learned velocity field. The velocity network v_θ is a 3-layer Transformer with 8 heads and width 256; flow time ξ is embedded sinusoidally. At generation we use four Euler steps and $K = 7$ alternatives.

A.6 Selector Architecture and Training

E^S_{traj} in Eq. (4) takes, for each waypoint, the position $\mathbf{p}_{k,t}$, its first and second differences $\Delta\mathbf{p}_{k,t}$ and $\Delta^2\mathbf{p}_{k,t}$, the path-local feature $\boldsymbol{\ell}_{k,t}$, and $\mathrm{PE}(t)$, is conditioned on the ego state and command, and is a 2-layer temporal Transformer over the T waypoint tokens with 8 heads and width 256, followed by mean pooling. The selector samples $M^S = 3$ path-local offsets per waypoint at lateral offsets of -0.925, 0, and $+0.925$ m, i.e., the ego centerline and the two sides of the 1.85 m-wide footprint (the generator uses $M^G = 5$). Scene cross-attention uses one layer and geometry-biased self-attention across candidates uses two layers, each with 8 heads and width 256. With $\mathbf{Q} = [\mathbf{q}_0, \ldots, \mathbf{q}_K]$, one such layer computes

$$\begin{aligned} \widetilde{\mathbf{Q}} &= \mathrm{CrossAttn}(\mathbf{Q}, \mathbf{S}, \mathbf{S}), \qquad a_{ij} = \mathrm{softmax}_j\left(\frac{(W_Q\widetilde{\mathbf{q}}_i)^\top (W_K\widetilde{\mathbf{q}}_j)}{\sqrt{d}} + b(\mathbf{g}_{ij})\right), \\ \mathbf{h}_i &= E^S_{\text{out}}\left(\widetilde{\mathbf{q}}_i + \textstyle\sum_j a_{ij} W_V \widetilde{\mathbf{q}}_j\right), \end{aligned} \tag{12}$$

where only candidates in $\mathcal{V}$ participate. The geometric descriptor $\Gamma(\tau_i, \tau_j) \in \mathbb{R}^8$ concatenates, in order, the mean and minimum of same-timestamp waypoint distances, the endpoint x and y differences and their Euclidean distance, the difference in cumulative path length from the origin, and the cosine and sine of the heading difference of the last segment; $b(\cdot)$ is a two-layer MLP producing one bias per attention head. The logit head f_ϕ is an MLP.

The training cache is produced once with the frozen generator checkpoint used at inference. It covers the 28,130 frames of the 700 nuScenes training scenes; for each frame, $K = 7$ alternatives are generated with four Euler steps under three noise seeds, giving three candidate sets of eight candidates (the reference and seven alternatives). Collision labels are valid when GT future object annotations exist for all timesteps up to 3 s; candidates in frames with missing annotations form $\mathcal{V} \setminus \mathcal{V}_c$. Supervision metrics follow Sec. 3.4 with

$$c_{k,h} = \frac{1}{n_h}\sum_{t=1}^{n_h} \widetilde{b}_{k,t}, \qquad d_k = \frac{1}{3}\sum_{h\in\{1,2,3\}} \frac{1}{n_h}\sum_{t=1}^{n_h} \left\|\tau_{k,t} - \tau_t^{\text{GT}}\right\|_2, \qquad n_h = 2h, \tag{13}$$

where the ego footprint is $4.084\,\text{m} \times 1.85\,\text{m}$ and GT occupancy is rasterized at 0.5 m resolution.

The reported selector starts from a selector checkpoint trained for 12 epochs with a safety-first objective and is fine-tuned for 3 epochs with AdamW (learning rate 5×10^{-5}, effective batch size

8). Its loss is the sum of four terms: (i) a collision-priority term that favors candidates in the lowest-collision group; (ii) a reference-relative term that suppresses candidates whose collision is worse than that of the reference τ_0; (iii) the same term restricted to the 1 s horizon, weighted by 2; and (iv) an L2 ranking term that orders candidates by d_k within the lowest-collision group with temperature 0.1. Terms (i), (ii), and (iv) have unit weight and together implement the priorities of Sec. 3.4: term (ii) enforces non-worsening relative to the reference, term (i) favors the minimum-collision group, and term (iv) orders candidates by displacement within it.

B Experimental Details

B.1 Base Planners and Evaluation Protocol

Table 1 applies SRTDrive to seven base planners: VAD, UniAD, GenAD, DMAD, IR-WM, SparseDrive, and DiffusionDrive. For each planner, its released checkpoint is kept frozen and provides both the reference trajectory τ_0 and the scene features $\mathbf{B}$ on which SRTDrive is conditioned (Sec. 3.1), and a separate SRTDrive module is trained for that planner. The reference and the selected trajectory are evaluated with the planner's own released open-loop evaluation code, so the collision definition, the ego footprint, the occupancy rasterization, and the set of valid samples follow each planner's protocol and differ across rows. Table 3 lists the number of valid validation samples (out of 6,019) entering each row; for VAD, the footprint is $4.084\,\mathrm{m} \times 1.85\,\mathrm{m}$, occupancy is rasterized at 0.5 m, and 5,119 samples are valid. Results in Table 1 are therefore comparable within a row but not across rows. Our re-evaluated reference numbers can differ from those reported in the original papers, because we evaluate only the samples that are valid under each released pipeline and use the released checkpoints as they are. For SparseDrive and DiffusionDrive, whose scene representation is not a dense BEV map, a learnable spatial adapter aggregates their native object and map features into a spatial grid that serves as $\mathbf{B}$.

Table 3: Open-loop evaluation protocol per base planner. Each planner supplies its own scene features and is evaluated with its own released code.

Planner	Scene features for SRTDrive	Valid samples
VAD	BEV (own)	5,119
UniAD	BEV (own)	6,019
GenAD	BEV (own)	5,119
DMAD	BEV (own)	6,019
IR-WM	BEV (own)	4,969
SparseDrive	Object/map + adapter	5,119
DiffusionDrive	Object/map + adapter	5,119

B.2 Closed-Loop Protocol

Closed-loop evaluation uses the 220 official Bench2Drive routes in CARLA 0.9.15 with the official leaderboard evaluator and scoring code. All three configurations in Table 2 use the VAD-Base weights and the PID controller released with the benchmark. For SRTDrive, the generator and selector run at every planning step, and the selected trajectory replaces the planner output before the controller; no temporal smoothing or hysteresis is applied across steps. Each configuration is evaluated in a single run over the 220 routes.

TTC-rule variant. The TTC-rule variant uses the same frozen generator and the same $K = 7$ alternatives per step as SRTDrive, but replaces the learned selector with a hand-designed selection rule based on the time to collision between each candidate and surrounding agents; no learned component is used for selection.

Collision routes. The auxiliary metric "Coll." in Table 2 counts the routes, out of 220, whose infraction log contains at least one collision event, including collisions with vehicles, pedestrians, and static objects. Each route is counted at most once regardless of the number of collisions it contains. This count is not part of the official Bench2Drive metrics and is reported to expose the collision behavior that the Driving Score aggregates with other infractions.

Ability subsets. Multi-ability scores are computed by the official evaluator on the subsets of routes associated with each ability, so their denominators differ and are smaller than 220; a change in an ability score can therefore correspond to a small number of routes.

C Additional Results

C.1 Supervision Statistics

Table 4 reports, over the 71,790 valid candidate sets of the VAD training cache, the fraction of sets in which each regime of the collision priority occurs: $\mathcal{A} \neq \mathcal{V}_c$ (some alternative worsens collision relative to the reference), $\mathcal{M} \neq \mathcal{A}$ (some non-worsening alternative reduces collision), and $\mathcal{M} = \{0\}$ (retention is the target), together with the fraction of sets in which all candidates are collision-free and d_k alone orders them. The regimes overlap, so the columns do not sum to 100%. Collision-free sets dominate the cache, which is consistent with the small changes in collision rate among replacements in Table 6.

Table 4: Fraction of valid candidate sets (%) in the VAD training cache by supervision regime. Regimes are not mutually exclusive.

Planner	$\mathcal{A} \neq \mathcal{V}_c$	$\mathcal{M} \neq \mathcal{A}$	$\mathcal{M} = \{0\}$	All collision-free
VAD	1.195	0.517	0.132	98.178

C.2 Per-Horizon Open-Loop Results

Table 5 breaks down the VAD row of Table 1 by horizon. SRTDrive lowers both metrics at every horizon and halves the 1 s collision rate.

Table 5: Per-horizon open-loop results of the VAD-based model on nuScenes validation.

	L2 (m) ↓				Collision Rate (%) ↓			
Method	1s	2s	3s	Avg.	1s	2s	3s	Avg.
VAD	0.411	0.702	1.057	0.723	0.020	0.127	0.348	0.165
VAD + SRT	**0.385**	**0.652**	**0.994**	**0.677**	**0.010**	**0.103**	**0.264**	**0.125**

C.3 Results on Sparse-Representation Planners

On SparseDrive and DiffusionDrive, SRTDrive changes L2 and the collision rate by at most 0.006 relative to the reference (Table 1). Unlike the other planners, these two expose object- and map-level features rather than a dense BEV map, so SRTDrive conditions on the grid aggregated by the spatial adapter of App. B.1, which may carry less path-local detail than native BEV features. SparseDrive additionally re-scores its own candidates with a collision-aware module (Sec. 2.2), leaving little residual collision for local deformations to remove. Scene interfaces that expose sparse features to SRTDrive more directly are a natural extension.

C.4 Per-Horizon Retention and Replacement

Table 6 gives the full statistics behind the retention analysis of Sec. 4.4 for the VAD-based model. Statistics are computed in VAD's original evaluation pipeline on the same 5,119 valid validation samples. The reference is retained in 22.11% of samples and replaced in 77.89% (3,987 samples). Among replacements, we report the fraction in which the collision metric of the selected trajectory is lower than (↓), equal to (=), or higher than (↑) that of the reference at each horizon; ↑ counts any increase of the collision metric rather than only newly introduced collisions. At every horizon, more than 99.3% of replacements leave the collision metric unchanged, and increases occur in at most 0.05% of replacements.

Table 6: Retention and replacement statistics (%) of the VAD-based model on nuScenes validation. ↓/=/↑: collision metric of the selected trajectory lower than, equal to, or higher than that of the reference, as a fraction of replaced samples.

			1 s			2 s			3 s		
Planner	Kept	Replaced	↓	=	↑	↓	=	↑	↓	=	↑
VAD	22.11	77.89	0.025	99.975	0.000	0.125	99.850	0.025	0.627	99.323	0.050