CANONICAL OUTPUT CONDITIONING: DECOUPLING NAVIGATION INTERFACES FROM DIFFUSION PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

A diffusion prior and the interface through which external conditions enter it are usually learned together: condition encoders, control branches, or classifier-free channels bind the trained model to the formats seen during training. Training-free guidance avoids retraining, yet still requires a compatibility function that relates each condition to the generated variable, typically a measurement operator, a differentiable loss, or a separately trained predictor. When condition and output live in different coordinates, constructing that function becomes its own learned component. We argue that this gap is fixed by a design choice made before either training or guidance: the choice of what the model generates. **Canonical-output conditioning** selects a representation whose coordinates both support the downstream task and admit direct comparison with externally converted conditions. The prior is then trained without conditions; heterogeneous formats reduce to explicit geometric adapters that feed a shared quadratic compatibility interface. We validate the design in route-guided navigation, where conditions are naturally heterogeneous. A LiDAR-conditioned diffusion prior over orientation fields, trained with no route inputs, is guided at inference by waypoint, corridor, and map route cues through **Orientation Mixing**, a confidence-weighted clean-estimate update. On six SemanticKITTI sequences, the single frozen prior serves all four interfaces and outperforms route-conditioned predictors trained on a fixed interface (mean ADE 0.29,m vs. 0.38,m), supporting representation choice as a practical lever for decoupling condition interfaces from diffusion priors.

1 INTRODUCTION

Conditioning a diffusion model involves two commitments that are easy to conflate: learning a generative prior, and learning how a condition enters that prior. Most systems make both at once. Condition encoders and control branches train integration modules for each condition format, as in ControlNet (Zhang et al., 2023) and T2I-Adapter (Mou et al., 2023). Classifier-free guidance amortizes the conditional score into training itself (Ho & Salimans, 2022). The prior may be reusable, but the condition interface is not. A new format triggers another round of paired data and training, even when the underlying prior already represents everything the task needs.

Training-free guidance addresses the second commitment at sampling time. Classifier guidance steers with an external score (Dhariwal & Nichol, 2021). RePaint (Lugmayr et al., 2022), DDNM (Wang et al., 2023), and DPS (Chung et al., 2023) reuse unconditional priors for inverse problems. Diffuser imposes test-time objectives on a behavior prior (Janner et al., 2022). Universal Guidance (Bansal et al., 2023), LGD (Song et al., 2023), and TFG (Ye et al., 2024) generalize and benchmark these designs. All of them still require a compatibility function that relates the condition to the generated variable, typically a known measurement operator, a differentiable task loss, or a separately trained predictor. When the condition and the output live in different coordinates, constructing this function becomes its own learned or hand-tuned component, and the interface problem reappears one level down.

Our observation is that whether this gap exists is decided before training or guidance, by a choice that is usually made implicitly: **what should the model generate?** If the generated representation is chosen so that external conditions can be expressed directly in its coordinates, with the same units, support conventions, and vector semantics, then heterogeneous conditions reduce to explicit

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.

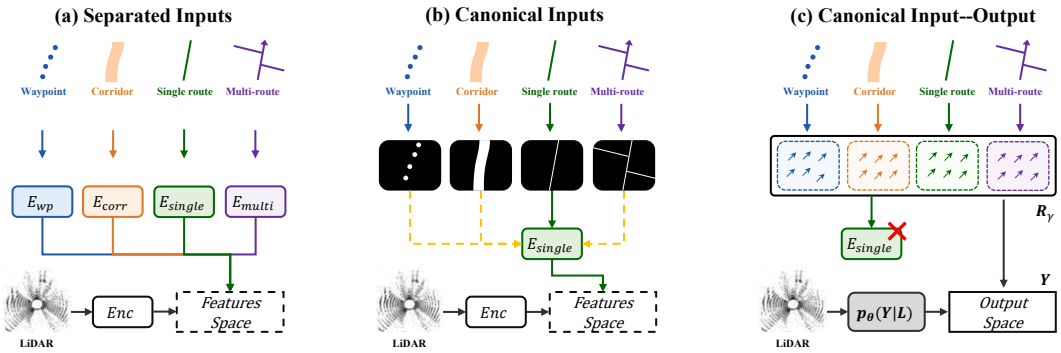


Figure 1: Three conditioning architectures. (a) Interface-specific route encoders learn integration with perception. (b) Canonicalizing inputs still leaves route integration inside training. (c) Canonical-output conditioning expresses routes in the generated field coordinates and applies them only during inference; a planner subsequently reads out a trajectory.

conversions and no compatibility function needs to be learned. We call this design canonical-output conditioning. The representation plays two roles: it supports a downstream task readout, and it serves as the common coordinate system for converted conditions. Training estimates a prior over this representation without conditions. An adapter specifies each condition's semantics, support, and reliability. A shared sampler applies the resulting quadratic compatibility energy. We characterize this interface theoretically: two adapters condition identically exactly when they preserve a coefficient pair (W, b).

A design principle needs a demanding testbed. Route-guided navigation provides such a setting because its conditioning signals are inherently heterogeneous: sparse waypoints, corridors, map polylines, and local route graphs differ in density, spatial support, and semantics (Ort et al., 2018; 2020; Xu et al., 2022; Paz et al., 2022). Existing systems typically bind these interfaces at training time (Huang et al., 2026). The domain also offers a natural canonical space. An orientation field (Huang et al., 2026), a dense bird's-eye-view map of local travel directions, is not itself an executable trajectory, yet supports both route-tangent comparison and trajectory extraction. We instantiate the design as **GOF** (Guiding Orientation Fields). A LiDAR-conditioned diffusion model learns a route-agnostic prior over orientation fields, and converted route cues enter sampling through **Orientation Mixing**, a confidence-weighted interpolation on the clean estimate that realizes the quadratic update with a schedule fixed by the diffusion signal-to-noise ratio. Unlike guidance through a task-space loss on sampled trajectories (Janner et al., 2022; Chung et al., 2023; Zhong et al., 2023; Jiang et al., 2023), the cue acts on directional structure over the whole spatial support before any single trajectory is selected. Our contributions are threefold. (1) **Canonical-output conditioning.** We formulate a conditioning design in which the generated representation is the common coordinate system for external conditions. Heterogeneous formats are converted by explicit, training-free adapters and applied only at inference, decoupling condition interfaces from the learned prior. (2) **A formal interface contract.** We characterize quadratic output-space interfaces through the coefficient pair (W, b). Adapters that preserve (W, b) induce the same conditioning update, multiple cues compose by adding coefficients, and a fiber criterion determines when conditioning can and cannot be deferred to the task readout. (3) **GOF for route-guided navigation.** We instantiate the design with a route-agnostic, LiDAR-conditioned orientation-field prior, adapters for waypoint, corridor, single-route, and multi-route inputs, and Orientation Mixing as the inference-time update. On six SemanticKITTI sequences, the single frozen prior serves all four interfaces, responds consistently to intent changes, outperforms route-conditioned predictors and gradient-guided diffusion, and tolerates moderate route-map misalignment.

2 Related Work

Condition integration in diffusion models. Conditioning is commonly introduced either by learning dedicated integration modules (e.g., ControlNet (Zhang et al., 2023) and T2I-Adapter (Mou et al., 2023)), which ties a model to a particular condition format, or by guidance at sampling time

(classifier(-free) guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2022) and training-free variants for inverse problems such as RePaint, DDNM, and DPS (Lugmayr et al., 2022; Wang et al., 2023; Chung et al., 2023), as well as broader guidance frameworks (Bansal et al., 2023; Song et al., 2023; Ye et al., 2024)). In all cases a compatibility function must relate the condition to the generated variable; if they live in different coordinates this becomes an extra learned or tuned component. We instead choose generated coordinates where converted conditions and prior outputs share a simple quadratic interface, keeping adapters reusable without per-condition losses or predictors.

Guided diffusion for planning and driving. Diffuser denoises trajectories under test-time goals and rewards (Janner et al., 2022), and Decision Diffuser studies conditional generation for sequential decision-making (Ajay et al., 2022). In driving, CTG enforces traffic rules and goal constraints on a trajectory diffusion model via guidance (Zhong et al., 2023), MotionDiffuser applies constrained sampling to multi-agent motion prediction (Jiang et al., 2023), and Diffusion Planner applies flexible, training-free guidance to closed-loop driving (Zheng et al., 2025), while DiffusionDrive (Liao et al., 2025) and GoalFlow (Xing et al., 2025) learn goal- and anchor-conditioned generation end-to-end. In all of these, a route or goal cue enters either through a trained condition channel or through a scalar cost on sampled trajectories. GOF instead guides a *field*: route preferences act on directional structure over the whole spatial support before any single trajectory is selected, and the same field admits multiple readouts. GPD, which diffuses polynomial coefficients and evaluates costs through their trajectory parameterization (Srikanth et al., 2025), shows that sampling outside the executable output is not new by itself; our claim concerns choosing that intermediate space so that heterogeneous conditions need no learned encoder.

Route-guided navigation with coarse maps. MapLite and its successors navigate from topometric maps and onboard perception without detailed prior maps (Ort et al., 2018; 2020). TridentNetV2 encodes graphical global plans for trajectory generation (Paz et al., 2022), and topometric-map trajectory prediction fuses rasterized routes with LiDAR (Xu et al., 2022). OrField introduces orientation-field outputs with field-based planners, consuming smoothed routes as trained input channels (Huang et al., 2026); we adopt its field representation and readout, and do not claim them as contributions. All of these systems bind the route interface at training time. Our question is orthogonal to the completeness of any navigation stack: how route interfaces enter a learned prior, and whether they can be changed without retraining it.

3 Canonical outputs and the conditioning

Let o be a perception observation, r_k a condition in format k, $z \in \mathcal{Z} \subseteq \mathbb{R}^d$ a generated representation, and $P : \mathcal{Z} \to \mathcal{T}$ a measurable task readout. Training estimates $p_\theta(z \mid o)$ without r_k as an input. An adapter $A_k(r_k) = c_k$ defines a compatibility energy $E(z; c_k)$. Inference factors as

$$z = \mathcal{S}_\theta(o, c_k; \xi), \qquad \tau = P(z), \qquad \pi_{c_k}(dz \mid o) \propto e^{-E(z;c_k)} p_\theta(dz \mid o). \tag{1}$$

Randomness ξ includes initialization and sampling noise; recurrent history, when used, enters as a condition below. Decoupling means that changing k changes the adapter, not the trained prior or its learned input channels.

3.1 What must be canonical?

Canonical outputs place prior predictions and converted conditions in shared coordinates. Adapters must therefore agree on units, vector conventions, and missing support, rather than only tensor shape. For the weighted quadratic interface, the adapter outputs $c_k = A_k(r_k) = (g_k, W_k)$, where $g_k \in \mathbb{R}^d$ is the external guidance expressed in the same coordinates as z, and $W_k \succeq 0$ specifies its reliability and support. Flattening the field gives

$$E(z; c_k) = \tfrac{1}{2}(z - g_k)^\top W_k (z - g_k) = \tfrac{1}{2} z^\top W_k z - b_k^\top z + \kappa_k, \tag{2}$$

where $b_k = W_k g_k$ and $\kappa_k = \frac{1}{2} g_k^\top W_k g_k$. Below we omit the format index k, writing $c = (g, W)$, $b = Wg$, and $\kappa_c = \frac{1}{2} g^\top W g$. GOF uses scalar weights for each two-dimensional cell block. The pair (W, b), not the raw route encoding, is the information this compatibility function passes to conditioning. For a clean prediction d and a fixed coefficient $m \geq 0$, the quadratic condition enters a clean-estimate update as

$$M_{m,c}(d) = d - m \nabla_z E(d; c) = (I - mW)d + mb. \tag{3}$$

To compare another converted condition $c' = (g', W')$, let $b' = W'g'$. At the same prediction d and coefficient m,

$$M_{m,c}(d) - M_{m,c'}(d) = m\big[(b - b') - (W - W')d\big]. \tag{4}$$

Preserving (W, b) therefore preserves the condition update, while changes in these coefficients make its local sensitivity explicit. The prior need not learn the original interface format to apply this update. Orientation Mixing in Equation (6) instantiates it for each field cell. This statement concerns one update; Appendix A proves the equivalence and specifies when it extends to the complete sampling procedure.

Two consequences matter in practice. Where $W = 0$, the cue is absent, regardless of the arbitrary value stored in g. A zero target with $W > 0$ instead requests a zero output and is not missing information. Further, two encodings of the same intended route need not produce the same (W, b). Canonicalization makes these differences explicit in the adapter while keeping the learned prior unchanged.

3.2 Composing conditions across sources and time

Shared output coordinates also let several sources constrain one generated state. For a collection $\mathcal{C} = (c_1, \ldots, c_J)$, composition retains one prior and adds the converted preferences:

$$E_{\mathcal{C}}(z) = \sum_j E(z; c_j), \qquad W_{\mathcal{C}} = \sum_j W_j, \qquad b_{\mathcal{C}} = \sum_j W_j g_j.$$

Equation (3) then applies with $(W_{\mathcal{C}}, b_{\mathcal{C}})$. Each source retains its support and confidence; equivalent adapter coefficients preserve the joint update.

This construction can organize information across time as well as formats. Let n index observation frames, separately from diffusion level t. Current perception o_n anchors the prior $p_\theta(z_n \mid o_n)$; interfaces $r_{n,j}$ express intended future continuation; the previous generated state h_{n-1} supplies a history cue. Their roles combine as

$$\mathcal{C}_n = \big(A_1(r_{n,1}), \ldots, A_J(r_{n,J}), A_{\text{hist}}(h_{n-1})\big), \qquad z_n = \mathcal{S}_\theta(o_n, \mathcal{C}_n; \xi_n), \qquad h_n = z_n.$$

Thus the current output reconciles present evidence, past prediction, and future-directed intent, and carries the resulting state to the next frame. Present perception enters the learned prior, rather than another cue energy; future intent is a preference, not an observed future state. This recurrent conditioning does not require a learned state-transition model. Appendix A.2 gives the composition identities and the shared-output interpretation of source-separated sectors.

3.3 Conditioning before task readout

A field readout visits only selected spatial support; field compatibility can additionally address other locations and alternative starts. Conditioning only on the extracted trajectory cannot directly distinguish fields that yield that trajectory but differ elsewhere. Conditioning in the field can therefore express route preferences before a particular trajectory is selected, while allowing a shared prior to serve different route adapters and task readouts. Appendix A gives the precise condition under which an energy can instead be expressed entirely through the task output. The navigation study evaluates the task utility of this design.

4 GOF: Decoupled route guidance for navigation

4.1 Directional outputs, adapters, and prior training

Figure 2 shows the concrete task and architecture. The representation is a dense BEV field $Y(i, j) \in \mathbb{R}^2$, with direction $Y/\|Y\|$ where nonzero and activation magnitude $\|Y\|$. The field encodes spatial directional structure rather than a single executed ego trajectory. We train a LiDAR-conditioned noise-prediction U-Net on these fields, learning continuation structure from directional targets without route inputs. A task readout subsequently constructs and selects candidate streamlines. For a route polyline γ, the adapter assigns each grid cell the unit tangent $R_\gamma(i, j)$ at its nearest route point and the distance $D_\gamma(i, j)$ to that point, with reliability $\omega(i, j) = \exp[-D_\gamma(i, j)/\ell]$,

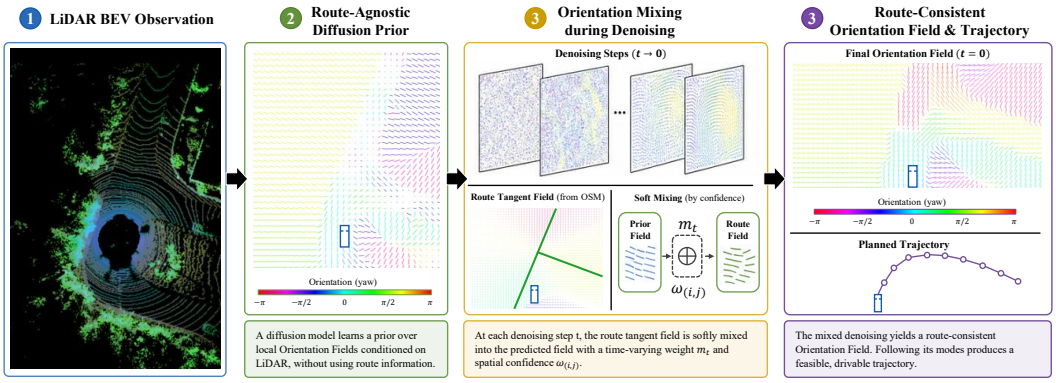


Figure 2: GOF system overview. The prior uses LiDAR only; a route adapter and Orientation Mixing act during inference; a field-based planner converts the resulting field into a trajectory.

where the spatial bandwidth ℓ is tied to the BEV horizon. Waypoints define directions toward a target, while corridors and connected multi-route crops provide richer directional support. These cues share coordinates with Y, but not necessarily the same information content: combining multiple routes introduces additional alternatives rather than simply re-encoding the same cue. Multi-route crops orient connected roads toward an assigned local target sink. Appendix B.2 details these conversions, while Appendix B.3 describes graph cleaning, pose alignment, offline route assignment, and training configuration.

4.2 Orientation Mixing

Under variance-preserving noising $X_t = a_t X_0 + \sigma_t \epsilon$, with $a_t = \sqrt{\bar{\alpha}_t}$, $\sigma_t = \sqrt{1-\bar{\alpha}_t}$, and $\epsilon \sim \mathcal{N}(0, I)$, predict the clean tensor

$$d = D_t(x_t) = \{x_t - \sigma_t \epsilon_\theta(x_t, t, o)\}/a_t. \tag{5}$$

For each cell, write its local cue as $c = (g, \omega I_2)$. Orientation Mixing combines this prediction with the converted guidance in the same coordinates:

$$E(z; c) = \tfrac{1}{2}\omega \left\| z - g \right\|^2, \qquad \mu = (1-w)d + wg, \qquad w = m_t \omega. \tag{6}$$

The cue acts directly on the predicted clean field. With $0 \le w \le 1$, the interpolation is the unique minimizer of

$$Q(z) = \tfrac{1}{2}(1-w) \left\| z - d \right\|^2 + \tfrac{1}{2} w \left\| z - g \right\|^2. \tag{7}$$

It also equals $d - m_t \nabla_z E(d; c)$, and satisfies $E(\mu; c) = (1-w)^2 E(d; c)$. Thus spatial confidence controls the local tradeoff between prior prediction and route guidance, with an explicit decrease in the cue energy. The update implements this tradeoff without a learned route encoder.

The SNR schedule fixes temporal influence:

$$m_t = (1 + \mathrm{SNR}_t)^{-1} = \frac{\sigma_t^2}{\sigma_t^2 + a_t^2}. \tag{8}$$

Once this schedule is selected, diffusion SNR and spatial confidence determine the mixing weight without an independently tuned energy step. The schedule and spatial bandwidth remain design choices. For fixed x_t with $a_t, \sigma_t > 0$, let $u_0 = \epsilon_\theta(x_t, t, o)$ initialize an independently variable noise prediction u. A gradient step of size η_t on the same energy $E((x_t - \sigma_t u)/a_t; c)$ maps back to the clean edit $d - (\sigma_t^2 \eta_t / a_t^2) \nabla_z E(d; c)$. The choice

$$m_t = \frac{\sigma_t^2 \eta_t}{a_t^2}, \qquad \eta_t = \frac{a_t^2 m_t}{\sigma_t^2} = \frac{a_t^2}{\sigma_t^2 + a_t^2}$$

recovers the selected SNR interpolation exactly. The network supplies the initial prediction. Appendix A.3 gives the coordinate conversion. Implementation details are given in Appendix B.

Table 1: Route-interface study on seq08, without the temporal sector. Cells: ADE↓/FDE↓/HitRate↑ at 20 m (m/m/fraction). For learned-route baselines, smoothed single-route is matched and other inputs are mismatched. GOF uses four spatial sectors with either repeated or distinct cues.

Route interface	Deterministic OF	Conditional Training	GOF
Waypoint	0.51/0.90/0.93	0.55/1.07/0.93	0.51/0.91/0.90
Corridor	2.52/4.98/0.67	3.37/6.77/0.62	0.41/0.74/0.94
Single-route	0.46/0.87/0.89	0.53/1.05/0.92	0.39/0.73/0.93
Multi-route	0.45/0.83/0.90	**0.42/0.79/0.94**	0.30/0.52/0.95
Smoothed single-route	**0.36/0.67/0.94**	0.45/0.85/**0.94**	–
All 4 sectors	–	–	**0.29/0.48/0.96**

4.3 Re-noising and multiple sources

Updating the clean field changes the reference for its noise. At a nonzero noise level, write the current state as $x_t = a_t d_t + \sigma_t \widehat{\epsilon}_t$, where d_t is the clean prediction and $\widehat{\epsilon}_t$ its associated residual. After replacing d_t by a mixed target μ, the old noisy state has residual

$$\frac{x_t - a_t \mu}{\sigma_t} = \widehat{\epsilon}_t + \frac{a_t}{\sigma_t}(d_t - \mu).$$

The old residual is therefore stale relative to the edited field: retaining the state leaves an offset from the clean edit in what is treated as noise. Re-noising rebuilds the noisy state before the next denoising evaluation. We use the same full-refresh operation either in place or while moving to a lower noise level:

$$X_s^+ = (a_s - a_t) d_t + a_t \mu_s + \sigma_s Z, \qquad Z \sim \mathcal{N}(0, I), \quad 0 < s \le t. \tag{9}$$

Here μ_s is formed from d_t using the mixing weights at the destination level s, and each refresh uses an independent Z. For *in-place refresh*, $s = t$, the transport term vanishes and $X_t^+ = a_t \mu_t + \sigma_t Z$. For *cross-level refresh*, $s < t$, the clean prediction is transported as the noise level changes. Both replace the retained residual with fresh noise at the destination level. Their combinations are studied in Table 5. Every in-place update recomputes its clean prediction and mixed target. Appendix A.3 shows the fixed-reference variance identity.

GOF implements the temporal organization of Section 3.2 using channel pairs called sectors. During training, the same current clean field is duplicated across sectors with independent noise, while LiDAR provides the shared perception input. At inference, spatial sectors receive route cues, while a temporal sector receives the previous generated field. Each cue acts through its own quadratic interface, and the joint denoiser reconciles them with the current LiDAR observation. The resulting field becomes the next frame's history. This is soft learned agreement rather than an exact consensus constraint. History is supplied without explicit ego-motion warping; its integration is learned reconciliation rather than an identified geometric registration or state-transition model. Appendix A.2 gives the duplicated-output interpretation and denoising rationale.

5 Experiments

We evaluate whether a route-free field prior can support heterogeneous navigation interfaces, whether its outputs remain useful for trajectory readout, and how inference-time guidance affects navigation performance. **Protocol:** We train on SemanticKITTI sequences 00, 02, 05, and 07 and evaluate 08, 13, 15, 16, 18, and 19, using a 400×400 BEV with resolution 0.16 m. ADE/FDE are meters at a 20 m radial horizon; HitRate is the percentage of paths staying inside the logged future-path corridor throughout that horizon. Evaluation is offline/open-loop: route matching assigns intended continuations, and future trajectories are not network inputs. We compare route-conditioned predictors with a route-free prior guided through output-side adapters. System, component, and common-readout results are averaged over the six test sequences; interface, schedule, budget, and map-misalignment navigation comparisons use sequence 08. The paired field-response study additionally covers eight scenes across the six test sequences. Appendix B.3 describes the training setup, baseline inputs, route assignment, and trajectory extraction.

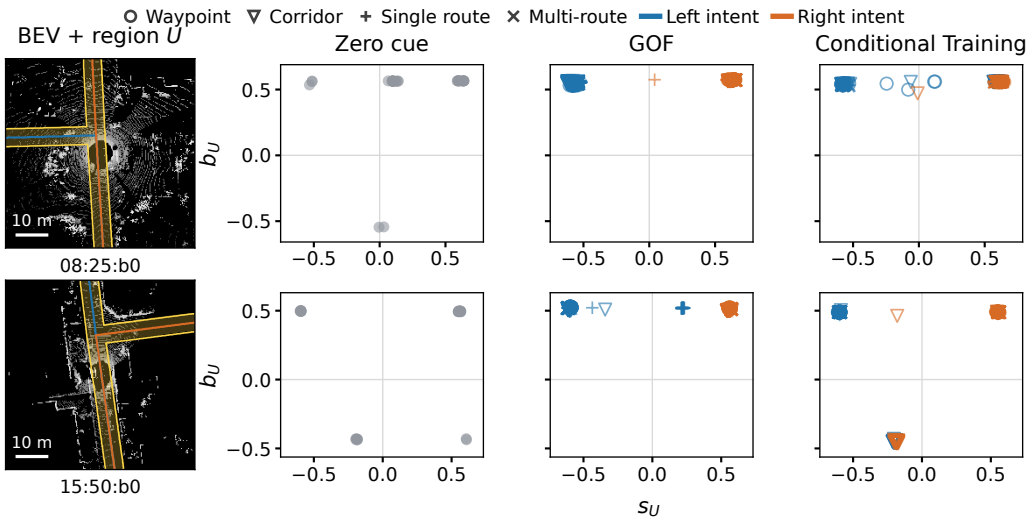


Figure 3: Union-2D responses (32 seeds per cue). Each row shows the LiDAR BEV with evaluation region U shaded yellow, prior-only with zero cue, GOF, and Conditional Training. Shapes denote interfaces; blue/orange denotes left/right intent and the corresponding BEV routes.

5.1 Changing the cue interface without changing the prior

Table 1 changes the route interface while keeping each trained predictor fixed. Deterministic OF and Conditional OSM are trained to consume smoothed single routes, whereas GOF converts waypoint, corridor, single-route, or multi-route inputs into directional cues at inference. For each single-interface setting, the same converted cue is duplicated across all four spatial sectors, while the joint setting assigns the four cues to separate sectors. This design isolates the effect of combining more route information from simply increasing the number of duplicated energies; the temporal sector is removed to focus on spatial guidance. GOF remains effective across all interfaces and improves as the route representation provides richer information: waypoints offer coarse directional guidance, while corridors and single routes provide denser cues, and multi-route guidance preserves multiple locally relevant branches near intersections. Combining all four cues achieves the best performance. Importantly, all these settings use the same perception prior without a learned route adapter. In contrast, although the learned-route baselines also produce fields and can be evaluated across different route interfaces, their condition integration remains learned and their performance degrades when the test-time interface differs from the training representation. These results demonstrate that canonical-output conditioning enables reuse of the same learned prior across route interfaces with different representations and spatial support, while the observed performance differences also reflect the amount and structure of information provided by each interface.

Intent response in the generated output. We further compare GOF with Orientation Mixing and the learned-route diffusion baseline Conditional Training. For eight geometry-selected branching scenes, we pair left/right intent under four interfaces and 32 sampling seeds, holding the observation fixed. Each method uses $N = 10$ denoiser evaluations, without history. Both views below are evaluated at U, the union of 3 m tubes around the two visible routes. Let x be the vectorized generated field on U, and t_L, t_R the reference vectors constructed from shortest paths on the visible-route graph to its left/right exits, respectively. All three use pixelwise direction normalization and scaling by $1/\sqrt{|U|}$. We define

$$s_U = \frac{x^\top (t_R - t_L)}{\|t_R - t_L\|}, \qquad b_U = \frac{x^\top (t_R + t_L)}{\|t_R + t_L\|}. \tag{10}$$

The horizontal coordinate s_U measures relative route preference (right vs. left), and b_U measures agreement with their shared directional pattern. For the same scene, interface, and sampling seed, we define the paired cue response $\Delta s_U = s_U(\text{right}) - s_U(\text{left})$. The same references and pixels are used across methods and interfaces; details are in Appendix C.1.

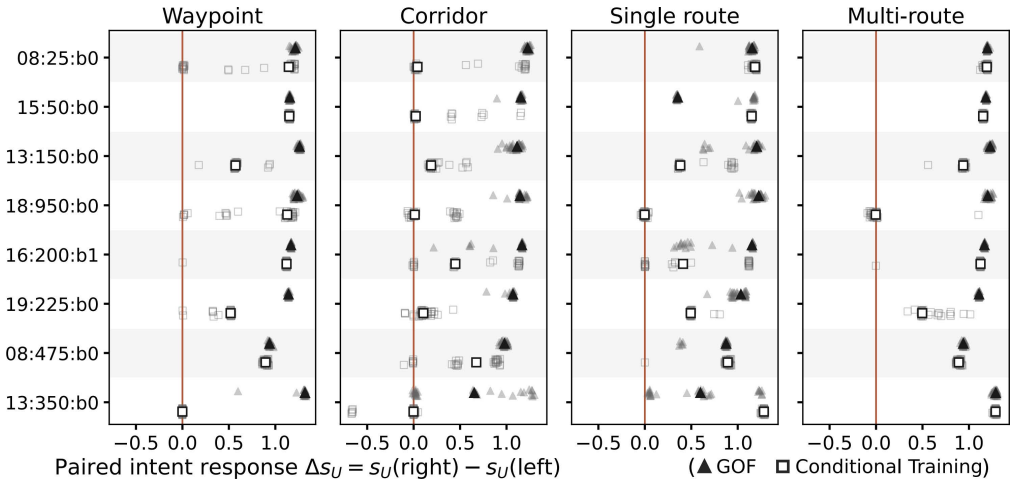


Figure 4: Compact Union-1D overview of all eight scenes and four interfaces. Faint points show each Δs_U among 32 seeds, while bold markers show the median. Triangles denote GOF and open squares Conditional Training. Positive shifts indicate response toward the changed intent. Scene labels are sequence:frame:branch.

Figure 3 shows that both models express different intents through multiple interfaces: canonical-input transfer is useful, while the output-side design adds route-free prior training and a direct condition edit. Across all scenes and interfaces, the paired response Δs_U is positive in 1022/1024 pairs for GOF and 911/1024 for Conditional Training (Figure 4); GOF also has a larger median Δs_U in most scenes, and multi-route yields a more concentrated response distribution. These results are evaluated with the output orientation field, not with the final trajectory.

5.2 Task utility of the intermediate representation

Table 2 compares complete navigation systems. TP (Xu et al., 2022) predicts trajectories directly, whereas deterministic OF (Huang et al., 2026) and Conditional Training learn route-conditioned field prediction. The latter receives either the training-time smoothed route (matched) or the same route as a raw polyline (mismatched). DPS (Chung et al., 2023) provides a gradient-guided diffusion comparison, sharing the same checkpoint as our GOF. Straight and direct-waypoint policies provide non-learned references. Among them, GOF achieves the lowest mean ADE/FDE and highest mean HitRate among these systems. This shows that keeping route integration outside training is competitive for the concrete task.

Table 2: Six-sequence comparison of complete systems at 20 m, with their respective route inputs and default readouts. Cells: ADE/FDE/HitRate (m/m/%).

Method	ADE/FDE/HitRate
Straight	1.20/2.87/72
Direct waypoint	1.76/2.72/61
TP (Xu et al., 2022)	.46/.94/91
Det. OF (Huang et al., 2026)	.38/.65/93
Cond. Training (matched)	.43/.85/94
Cond. Training (mismatched)	.48/.98/93
DPS (Chung et al., 2023)	.81/1.70/89
GOF	**.29/.53/96**

5.3 How inference-time guidance uses route information

Table 3 adds the guidance components cumulatively. Orientation Mixing produces the largest change: the LiDAR-only prior learns local continuation geometry, while the inference-time cue selects the assigned route. Re-noising refreshes the noisy state after that clean edit and further improves the mean metrics. Temporal guidance gives the best average performance. Appendix Table 8 includes the sequence-specific comparisons. Component configurations are con-

Table 3: Six-sequence component study. The full system additionally uses temporal guidance.

Configuration	ADE	FDE	HitRate (%)
LiDAR-only prior	4.88	9.79	46
+ Mixing	.44	.83	91
+ Re-noising	.31	.56	95
Full system	**.29**	**.53**	**96**

trolled to have equal number of sampling steps, and the Orientation Mixing is a pointwise interpolation with little arithmetic cost relative to denoising.

Spatial composition and temporal conditioning. The common output interface supports composition across sources and time (Section 3.2). In Table 1, combining four different spatial cues outperforms repeating any one cue across the same four sectors, testing heterogeneous cue content rather than additional sectors. Current LiDAR anchors the prior, route cues express future intent, and past predictions can condition the current output without a separately trained navigation-condition encoder. The spatial study excludes history; a separate temporal study (Table 4) holds the prior and spatial guidance fixed while comparing absent, own, five-frame-delayed, and direction-reversed (Conflicting) history. We measure temporal directional change J_{dir} as the mean angle between consecutive fields on valid overlapping scene support after ego-motion alignment. Lower values indicate greater directional stability. Own lowers J_{dir}, while Conflicting increases it,. These results support modest and content-dependent stabilization for temporal conditioning. Appendix C.3 shows the detailed defination of the metrics.

Table 4: Temporal conditioning with fixed spatial guidance: three seeds (142–144) on the same 1,800 frames from six sequences. Pooled means; ADE/FDE: m; J_{dir}: degrees. Valid-direction coverage exceeds 99.99% in every condition.

History	ADE↓	FDE↓	$J_{\mathrm{dir}}\downarrow$
None	0.317	0.600	4.625
Own	0.293	0.548	4.511
Delayed	0.292	0.546	4.533
Conflicting	0.304	0.563	4.851

Re-noising allocation. Table 5 varies the combination of refresh placements. The budget $N = T \times K$ allocates T noise-level transitions and $K - 1$ in-place updates at each nonzero destination level. With $K = 1$, refresh occurs only across levels; $K > 1$ combines both placements. At a fixed budget, their navigation results are comparable. Together with the component ablation, this supports refreshing after the clean edit as the important operation, with flexibility in where the refresh occurs. The guidance-schedule comparison in Appendix C.4 supports the selected SNR schedule.

Table 5: Seq08 re-noising allocations at budget $N = T \times K$. Entries: $T \times K$: ADE/FDE/HitRate (m/m/%); –: unreported.

N	Cross-level only ($K = 1$)	Cross-level + in-place ($K > 1$)	
5	5×1: 1.08/2.09/85	–	–
10	10×1: .31/.54/95	5×2: .29/.49/96	–
15	15×1: .30/.50/96	5×3: .29/.49/96	–
20	20×1: .29/.48/96	5×4: .29/.48/96	10×2: .29/.49/96

Imperfect cues and qualitative behavior. The map-perturbation study on SemanticKITTI seq 08 supports tolerance within the tested range of route rotations up to 10° and $+x$ translations up to 8 m (Appendix C.5). Appendix C.2 additionally illustrates generated fields and their trajectory readouts at a branching road.

6 Discussion and conclusion

Canonical-output conditioning decouples learning a continuation prior from learning a navigation interface. We choose generated coordinates that support both task readout and direct condition comparison, and apply converted conditions only at inference. The quadratic interface specifies what an adapter must preserve, and the clean-estimate update makes its effect explicit; GOF demonstrates how Orientation Mixing realizes this separation across navigation interfaces. Canonical input coordinates permit some interface transfer; the additional output-side commitment removes route integration from prior training and exposes its action on the generated fields. The paired interface study supports this design with more consistent intent responses. The prior learns directional structure from LiDAR and field supervision, while adapters supply route semantics and reliability at inference. This permits training the continuation prior independently of route-interface annotations, while retaining task knowledge in the representation and planner. The demonstrated task is selecting road-network continuations, not lane rules, actor intent, or safe closed-loop control. Cross-domain performance, field-reuse utility, and closed-loop behavior remain untested extensions. Retaining a field also permits new readouts without another generation call while perception and route remain valid. Field generation remains the dominant cost (Table 10). The resulting separation makes navigation interfaces reusable without retraining the perception prior.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, manuscript drafting, and part of coding assistance. We have not used generative AI tools for original research ideas. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 843–852, 2023. URL https://arxiv.org/abs/2302.07121.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.14687.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2006.11239.

Yuming Huang, Wei Gao, Zhiyuan Zhang, Maani Ghaffari, Dezhen Song, Cheng-Zhong Xu, and Hui Kong. Learning orientation field for OSM-guided autonomous navigation. *Journal of Field Robotics*, 43(2):717–738, 2026. doi: 10.1002/rob.70060. URL https://onlinelibrary.wiley.com/doi/10.1002/rob.70060.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915, 2022. URL https://proceedings.mlr.press/v162/janner22a.html.

Chiyu Max Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, and Dragomir Anguelov. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9644–9653, 2023. URL https://arxiv.org/abs/2306.03083.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047, 2025.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. *arXiv preprint arXiv:2201.09865*, 2022. URL https://arxiv.org/abs/2201.09865.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. URL https://arxiv.org/abs/2302.08453.

T. Ort, L. Paull, and D. Rus. Autonomous vehicle navigation in rural environments without detailed prior maps. In *IEEE International Conference on Robotics and Automation*, pp. 2040–2047, 2018.

T. Ort, K. Murthy, R. Banerjee, S. K. Gottipati, D. Bhatt, I. Gilitschenski, L. Paull, and D. Rus. MapLite: Autonomous intersection navigation without a detailed prior map. *IEEE Robotics and Automation Letters*, 5(2):556–563, 2020.

D. Paz, H. Xiang, A. Liang, and H. I. Christensen. TridentNetV2: Lightweight graphical global plan representations for dynamic trajectory generation. In *IEEE International Conference on Robotics and Automation*, pp. 9265–9271, 2022.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32483–32498, 2023. URL https://proceedings.mlr.press/v202/song23k.html.

Ajit Srikanth, Parth Mahajan, Kallol Saha, Vishal Mandadi, Pranjal Paul, Pawan Wadhwani, Brojeshwar Bhowmick, Arun Singh, and Madhava Krishna. GPD: Guided polynomial diffusion for motion planning. *arXiv preprint arXiv:2501.18229*, 2025. URL https://arxiv.org/abs/2501.18229.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2212.00490.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1602–1611, 2025.

J. Xu, L. Xiao, D. Zhao, Y. Nie, and B. Dai. Trajectory prediction for autonomous driving with topometric map. In *IEEE International Conference on Robotics and Automation*, pp. 8403–8408, 2022.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified training-free guidance for diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2409.15761.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhang_Adding_Conditional_Control_to_Text-to-Image_Diffusion_Models_ICCV_2023_paper.html.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Li, Xianyuan Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. In *International conference on learning representations*, volume 2025, pp. 37207–37227, 2025.

Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *IEEE International Conference on Robotics and Automation*, pp. 3560–3566, 2023. URL https://arxiv.org/abs/2210.17366.

A CONDITIONING IDENTITIES AND MULTI-SOURCE CONSTRUCTION

We give the identities supporting the output-side interface, the source-separated realization, and its local sampling updates.

A.1 INTERFACE EQUIVALENCE AND TASK READOUT

Proposition 1 (Quadratic interface equivalence)**.** *Two quadratic interfaces with the same* (W, b) *have energies differing only by a constant and induce the same target in Equation* (1) *for every prior for which it is normalizable. They also give the same clean edit* $d - m_t(Wd - b)$.

Expand $E(z; c) = z^\top W z/2 - b^\top z + \kappa_c$. If $W = W'$ and $b = b'$, the difference is constant, so normalization cancels it for every prior. Gradients and clean edits coincide as well.

For a common sampler, equality of edits propagates by induction through every subsequent operation when the schedule, observation, random draws, and initial history are equal and no other condition-dependent path exists. This assertion includes a state-dependent denoiser; it does not assume that denoising preserves the edit or that samples match the ideal target. For adaptive schedules or stopping rules, these must themselves agree under equivalent interfaces.

When conditioning can be deferred to the task output. For a task output τ, its fiber $P^{-1}(\tau)$ consists of all representations with that same readout. If $E(z; c) = \bar{E}(P(z); c)$, it is constant on each fiber immediately. Conversely, constancy defines $\bar{E}(\tau; c)$ by choosing any $z \in P^{-1}(\tau)$; the value is independent of the choice. This proves set-theoretic factorization on the image of P. This is only a statement of expressivity. A measurable factorization requires additional regularity; measurability is not inferred merely from constancy on fibers.

A.2 COMPOSING MULTIPLE SOURCES THROUGH CANONICAL OUTPUTS

For the aggregate energy $E_{\mathcal{C}}$ in Section 3.2, define

$$\pi_{\mathcal{C}}(dz \mid o) \propto e^{-E_{\mathcal{C}}(z)} p_\theta(dz \mid o). \tag{11}$$

This is a joint preference; a likelihood interpretation additionally requires an observation model. Addition is independent of source order, whereas sequential edits with intermediate predictions are a different operation.

For scalar cell weights, define $\omega_{\mathcal{C}} = \sum_j \omega_j$. Where $\omega_{\mathcal{C}} > 0$, the equivalent cue is $g_{\mathcal{C}} = \sum_j \omega_j g_j/\omega_{\mathcal{C}}$. Both its direction and its total weight matter. Averaging cues while discarding the total weight changes the conditioning strength. A convex local edit requires $m_t\omega_{\mathcal{C}} \le 1$, not merely $m_t\omega_j \le 1$ for each source. Correlated or repeated cues therefore need calibrated weights; copying a cue with unchanged weight counts its preference again.

Separate source slots, one underlying output. GOF retains cues in separate sectors rather than summing them before denoising. To relate this construction to Equation (11), let $\boldsymbol{z} = (z_1, \dots, z_J) \in \mathcal{Z}^J$ and define the duplication map $\iota_J(z) = (z, \dots, z)$. For any canonical prior $p_\theta(dz \mid o)$, its ideal duplicated law is $p_\theta^{\Delta} = (\iota_J)_{\#} p_\theta$. This law is supported on equal-output tuples; it is not a product of independent sector priors. Give source j its own energy $E(z_j; c_j)$ and define

$$\Pi_{\mathcal{C}}(d\boldsymbol{z} \mid o) \propto \exp\left[-\sum_j E(z_j; c_j)\right] p_\theta^{\Delta}(d\boldsymbol{z} \mid o).$$

The pushforward notation avoids assigning an ordinary joint density to this lower-dimensional support.

Proposition 2 (Source composition under duplicated outputs)**.** *For finite positive normalizers,* $\Pi_{\mathcal{C}} = (\iota_J)_{\#}\pi_{\mathcal{C}}$. *Consequently, any measurable readout* $\overline{P}$ *satisfying* $\overline{P}(\iota_J(z)) = P(z)$ *gives* $\overline{P}_{\#}\Pi_{\mathcal{C}} = P_{\#}\pi_{\mathcal{C}}$.

Proof. On the duplicated support, $z_j = z$ for every j, so the sector energy pulls back to $\sum_j E(z; c_j) = E_{\mathcal{C}}(z)$. Integration against p_θ^{Δ} is integration after ι_J against p_θ; the normalizers and all measurable-event probabilities therefore coincide. Applying the readout gives the second identity. □

The exact equivalence requires the duplicated support; separate sector energies with an arbitrary coupled prior do not suffice.

Why duplicated-target denoising learns agreement. Consider the ideal training law $z_j = z$, with finite second moment, and independent Gaussian corruption $x_j = a_t z + \sigma_t \epsilon_j$, $\epsilon_j \sim \mathcal{N}(0, I)$, at a common level with $a_t, \sigma_t > 0$. Writing $\bar{x} = J^{-1} \sum_j x_j$, the identity

$$\sum_j \|x_j - a_t z\|^2 = \sum_j \|x_j - \bar{x}\|^2 + J \|\bar{x} - a_t z\|^2$$

shows that $\bar{x}$ contains all information in the corrupted sectors about z, with effective noise variance σ_t^2/J. At a fixed level, squared-error noise prediction and clean prediction have the same optimum through the affine conversion in Equation (5). Every optimal clean-prediction block is therefore

$$D^*_{t,j}(\boldsymbol{x}, o) = \mathbb{E}[z \mid \boldsymbol{x}, o] = \mathbb{E}[z \mid \bar{x}, o], \qquad j = 1, \ldots, J. \tag{12}$$

It is the clean predictions that agree; the corresponding noise predictions need not agree because their corrupted inputs differ.

This is a training rationale for learned agreement, not an exact-consensus guarantee for the finite network or a proof that Mixing and re-noising sample $\Pi_{\mathcal{C}}$. The denoising identity also does not identify which conflicting source is correct. Temporal history is one such cue, conditioned on its realized value rather than assumed to be an independent observation. A fixed trained sector layout supports different cue assignments within that layout; increasing its channel count is an architectural change.

A.3 Orientation Mixing and re-noising identities

For fixed d, g, w, Equation (7) has Hessian I, including at $w = 0, 1$. With $E(z; c) = \omega \|z - g\|^2 / 2$, the explicit step $d - m_t \nabla_z E(d; c)$ gives Equation (6). The stationary point of Q is $(1 - w)d + wg$; the Hessian makes it the unique minimizer. Moreover, $\mu - g = (1 - w)(d - g)$, so $E(\mu; c) = (1 - w)^2 E(d; c)$. These statements concern the local edit, not the outcome of a later denoiser evaluation.

Holding x_t fixed and using the independent prediction variable u initialized at u_0 from the main text, the chain rule gives

$$\nabla_u E\left(\frac{x_t - \sigma_t u}{a_t}; c\right)\Big|_{u=u_0} = -\frac{\sigma_t}{a_t} \nabla_z E(d; c).$$

A gradient step of size η_t therefore produces

$$u^+ = u_0 + \eta_t \frac{\sigma_t}{a_t} \nabla_z E(d; c),$$

$$\widetilde{d} = \frac{x_t - \sigma_t u^+}{a_t} = d - \frac{\sigma_t^2 \eta_t}{a_t^2} \nabla_z E(d; c).$$

For the quadratic interface, this equals Equation (6) when $\eta_t = a_t^2 m_t / \sigma_t^2$. The network depends on x_t, but x_t is fixed in this optimization: the derivative concerns the affine output-to-clean map, not $x_t \mapsto D_t(x_t)$. Here η_t is an output-coordinate gradient step size. Under Equation (8), its equivalent form $\eta_t = a_t^2/(\sigma_t^2 + a_t^2)$ remains finite at $\sigma_t = 0$, where $m_t = 0$ and the clean estimate is unchanged.

Fixed-reference full refresh. The implemented in-place update in Equation (9) uses $X' = a_t \mu + \sigma_t Z$. For fixed μ and $Z \sim \mathcal{N}(0, I)$,

$$X' \sim \mathcal{N}(a_t \mu, \sigma_t^2 I), \qquad \mathbb{E}[X'] = a_t \mu, \qquad \mathrm{Cov}(X') = \sigma_t^2 I.$$

This is a conditional Gaussian identity. Recomputing μ from a state-dependent denoiser does not turn it into an invariant-posterior guarantee for the complete sampling process.

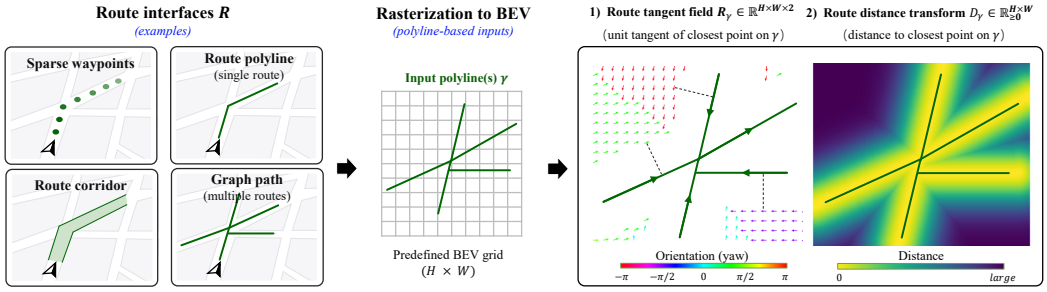


Figure 5: Route formats, polyline rasterization, nearest-route tangent field, and distance transform. Converted cues are consumed outside prior training.

B Implementation and Experimental Protocol

B.1 Sampling and Recurrent State

The cross-level update in Equation (9) has the transported clean reference

$$X_s = a_s \left[d_t + \frac{a_t}{a_s}(\mu_s - d_t) \right] + \sigma_s Z, \qquad s < t. \tag{13}$$

Here μ_s is the mixed, padded target in model units, using the destination-level weights.

The sampling procedure is:

1. Draw the initial noisy tensor and initialize this method's own history.
2. At each scheduled noise level, predict d_t and clip its components to $[-1, 1]$. If the next level is zero, return the clean output without new fusion.
3. Otherwise construct the next-level mixed target and perform the cross-level refresh in Equation (13).
4. At that next level, recompute the clean prediction and mixed target for each configured in-place refresh; then proceed to the next noise-level transition.
5. Read out the final field and retain it as this method's next history.

We label schedules by the budget $N = T \times K$, where T counts scheduled transitions and $K - 1$ counts additional updates at a nonzero destination level. In implementation, the zero-noise return omits those additional updates.

B.2 Route Adapters

On the BEV lattice, let p_{ij} be the cell center. For one or more directed polylines γ, set

$$q_{ij} \in \arg\min_{q \in \gamma} \|p_{ij} - q\|, \qquad D_\gamma(i,j) = \|p_{ij} - q_{ij}\|.$$

The cue $R_\gamma(i,j)$ is the unit tangent at the selected nearest route element. A single route gives one continuation; multiple locally connected polylines express neighboring roads directed toward the same target sink. For a target waypoint q,

$$D_\gamma(i,j) = \|q - p_{ij}\|, \qquad R_\gamma(i,j) = \frac{q - p_{ij}}{\|q - p_{ij}\|}, \quad p_{ij} \neq q.$$

For a corridor, directions follow the free-space label-construction procedure of Huang et al. (2026); distance is measured to its centerline.

The bandwidth in the main-text distance weighting is the BEV half-extent: $\ell = 32\,\mathrm{m}$ for the $64\,\mathrm{m}$ grid. When spatial or history cue vectors are combined before normalization, their resultant direction and magnitude determine the cue and reliability.

B.3 TRAINING AND EVALUATION

Dataset and observations. The split and BEV grid are specified in Section 5. LiDAR channels contain occupancy count, maximum height, and mean intensity. All compared diffusion models share this preprocessing and the backbone described below.

Map construction and assigned intent. Navigation routes are derived from OpenStreetMap (OSM). We filter OSM to road-level elements, then manually remove, add, or correct polylines that prevent correspondence with the traversed network. The sequence graph is transformed using released poses. Dynamic time warping matches the logged sequence path to the graph. In each local crop, incoming and outgoing segments are identified; other connected segments are directed toward the outgoing local target sink. Online rasterization supplies the adapters in Appendix B.2. The future path is used for evaluation and offline route assignment, without being supplied to a model as a future-trajectory tensor.

Architecture and optimization. The denoiser follows a latent-diffusion U-Net architecture (Rombach et al., 2022) with four resolution scales, base width 320, timestep embeddings, attention, and multi-scale LiDAR fusion. Table 6 specifies the optimization and forward-noise settings (Ho et al., 2020). The base noise-prediction objective is

$$\mathbb{E}_{Y,t,\epsilon} \left\| \epsilon - \epsilon_\theta(a_t Y + \sigma_t \epsilon, t, L) \right\|^2 .$$

Directional targets come from local free-space traversals, rather than OSM route inputs. This distinguishes target supervision from a learned route encoder.

Table 6: Shared optimization and forward-noise settings.

Setting	Value
Optimizer	AdamW
Learning rate	8×10^{-5}, constant
Weight decay	.01
Batch size	32
Training duration	400 epochs
Forward diffusion	1000 steps, linear beta schedule

Route-conditioned baselines. TP predicts trajectories from LiDAR and a rasterized route corridor (Xu et al., 2022). Deterministic OF uses a smoothed, Bezier-parameterized single route (Huang et al., 2026). Conditional Training trains $p_\theta(Y \mid L, R)$ with the same backbone as GOF and route input channels. Its matched test input uses the training-time Bezier-smoothed representation; its mismatched input uses a raw polyline from the same OSM route. Thus the mismatch changes preprocessing rather than the map provider. The non-learned straight policy extends the current heading; the direct-waypoint policy aims directly at the assigned waypoint.

Metrics and readout. At the main-text radial horizon, ADE averages Euclidean discrepancies between corresponding planned and logged path points; FDE compares their endpoints. HitRate is reported as a fraction in Table 1 and as a percentage in the other result tables.

The greedy readout initializes ten candidate streamlines with predefined first-step directions. Subsequent steps use local field direction and magnitude to minimize accumulated grid energy. Candidates stop at the radial horizon, outside valid BEV, or on invalid support. Scores reward aligned progress and penalize heading disagreement beyond a threshold; the best candidate is selected and smoothed. The diffusion models use this readout. The deterministic baseline's default is Field-Bezier (Huang et al., 2026). The controlled comparison applies both Field-Bezier and greedy readouts to both generators.

C SUPPORTING EXPERIMENTS

C.1 PAIRED INTERFACE RESPONSES IN THE ROUTE UNION

Pairing. The field-response study uses eight branching scenes selected and reviewed for visible route geometry before generation. Their frozen order is 08:25:b0, 15:50:b0, 13:150:b0, 18:950:b0, 16:200:b1, 19:225:b0, 08:475:b0, and 13:350:b0 (sequence:frame:branch index). Each scene fixes the LiDAR observation and two visible route continuations. Waypoint, corridor, raw single-route, and multi-route interfaces express each continuation, giving eight guided arm plus one method-specific zero-cue arm. We generate all guidance at seeds 142–173, with no temporal sector: 2304 fields per method, 4608 total.

Sampling budget. GOF uses five noise-level transitions and four in-place refresh updates, for $N = 10$ denoiser evaluations. Each guided arm repeats one cue across its four spatial sectors and averages their final fields. Its zero-cue reference sets the external drive to zero while retaining padding, corrections, and the same evaluation budget. Conditional Training uses the same number of denoiser evaluations without output-side Mixing or in-place refresh. Its LiDAR plus tangent/distance input adapter, and the first two output channels as its field.

Geometry-defined coordinates. Let Γ_L, Γ_R be the frozen visible polylines, and

$$U = \{p : \min(d(p, \Gamma_L), d(p, \Gamma_R)) \le 3\,\mathrm{m}\}, \qquad x = \frac{\mathrm{vec}(\widehat{Y}|_U)}{\sqrt{|U|}}.$$

The union counts overlap once and adds no circular probe. Each vector of $\widehat{Y}$ is the unit direction of the generated field; vectors below magnitude 0.05 contribute zero with the same fixed denominator. Validity requires finite values throughout U and at least 95% of its pixels at or above this magnitude threshold. Pixels outside U do not enter either coordinate or validity. We finally prepare 4608 fields.

We construct t_L, t_R by directing the visible-route graph toward its left or right exit along shortest paths, then extending the nearest-edge direction to every pixel in U. The reference fields use the same direction convention and normalization as x, giving $\|t_L\| = \|t_R\| = 1$. The sum and difference axes in Equation (10) are therefore orthogonal and are nondegenerate in all selected scenes. All methods and interfaces share these geometry-defined axes.

Display and interpretation. Figure 6 shows the remaining six scenes in the same preselected order and with the same display conventions as the main text. All seeds are retained, including GOF's two nonpositive pairs under corridor guidance in scene 13:350:b0. Conditional Training also responds strongly in many settings, with a larger single-route shift than GOF in scene 15:50:b0. These projections describe generated-field response to changing cues in the generated space rather than the trajectory space.

C.2 TASK READOUT, SYSTEM COMPARISONS, AND RUNTIME

Tables 7 and 8 give the per-sequence system and component results corresponding to Tables 2 and 3 in the main text. Table 9 additionally crosses field generators and trajectory extractors. ADE/FDE are in meters and HitRate is a percentage.

These metrics assess agreement with the assigned route, rather than the prior's probabilities over alternative feasible continuations.

Both generators benefit in mean error from the greedy readout. GOF has lower mean ADE/FDE within each readout group, while its mean HitRate ties deterministic OF for the greedy readout at printed precision. The sequence-level results show where the advantage depends on the scene and readout.

Qualitative field and trajectory comparison. In Figure 7, the route requests a right turn. The deterministic field is locally smooth but lacks scene-level consistency; the route-conditioned diffusion readouts favor the straight branch. GOF's single-route field bends toward the requested turn, while multi-route guidance incorporates more of the local branching structure.

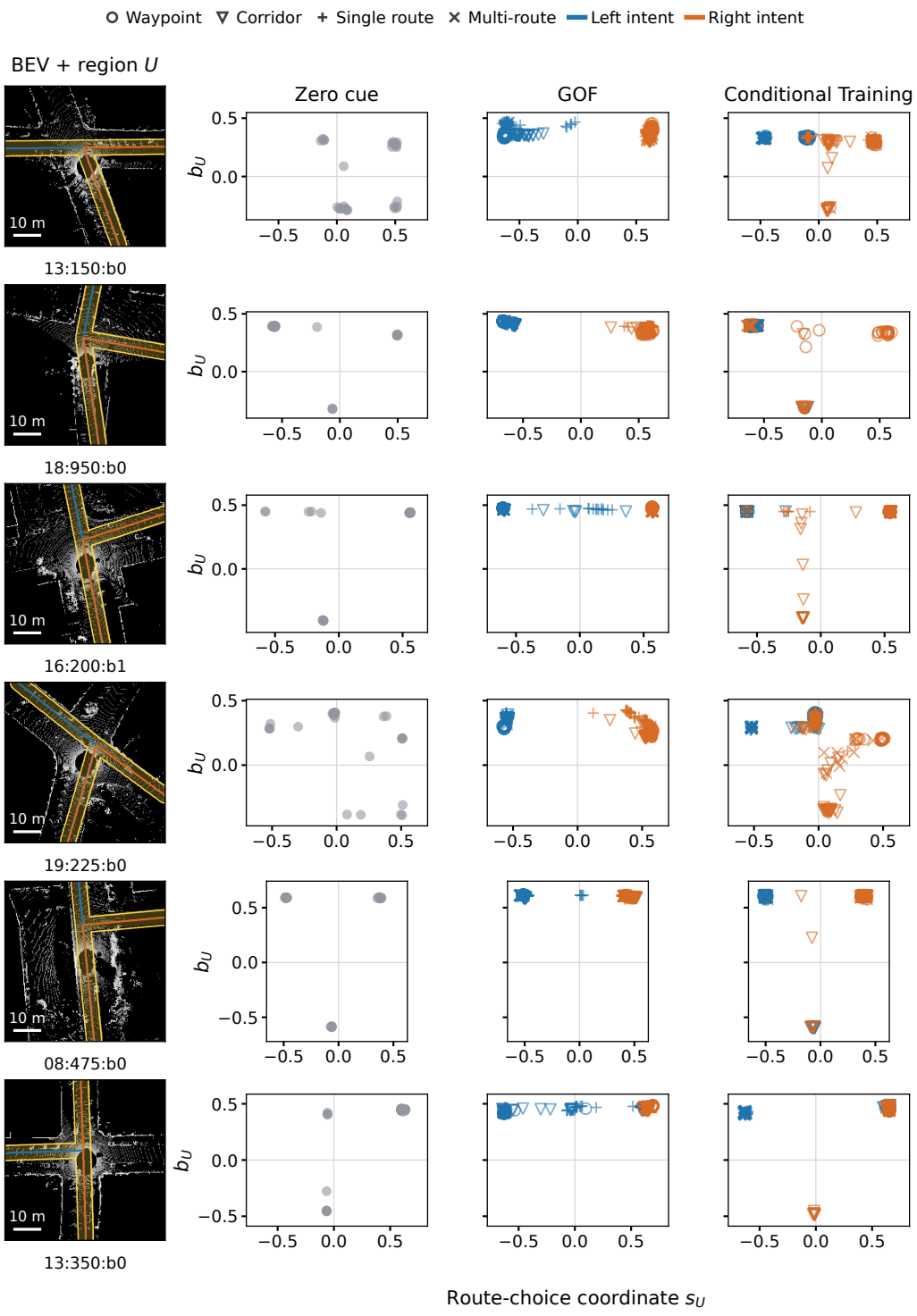


Figure 6: Remaining six preselected scenes, in frozen order, with the same Union-2D coordinates, BEV/region view, separate method panels, interface shapes and intent colors as Figure 3. Every cue setting contains all 32 seeds; no scenes or seeds are filtered by method advantage.

Table 7: Six-sequence navigation results with TP, deterministic OF, and DPS. Cells: ADE/FDE/HitRate (%). M/MM: matched/mismatched Conditional Training. This default-system comparison does not fix trajectory extraction.

Seq	TP	Det. OF	Cond. M	Cond. MM	DPS	GOF
08	.50/1.01/88	.41/.73/91	.45/.85/94	.53/1.05/92	1.03/2.16/86	**.29/.49/96**
13	.66/1.37/87	.44/.75/91	.63/1.30/92	.82/1.69/90	.96/2.06/87	**.39/.67/94**
15	.49/.93/90	.39/.57/93	.49/.98/94	.50/1.00/94	.98/2.02/87	**.31/.55/95**
16	.43/.89/90	.39/.70/94	.38/.79/93	.45/.94/92	.76/1.61/90	**.27/.49/97**
18	.24/.53/98	.29/.51/98	.23/.49/99	.22/.48/99	.41/.87/96	**.20/.41/100**
19	.44/.91/90	.38/.65/92	.36/.67/93	.38/.71/93	.71/1.51/89	**.30/.56/94**
Mean	.46/.94/91	.38/.65/93	.43/.85/94	.48/.98/93	.81/1.70/89	**.29/.53/96**

Table 8: Component results by sequence. Columns are cumulative additions; Re-noising uses the combined in-place/cross-level configuration. Cells: ADE/FDE/HitRate (%).

Seq	Prior only	+ Mixing	+ Re-noising	+ Temporal
08	6.28/12.53/32	.48/.88/89	.30/.52/95	**.29/.49/96**
13	4.30/8.61/52	.54/.93/90	.40/.72/93	**.39/.67/94**
15	5.24/10.48/42	.54/1.02/87	.34/.62/93	**.31/.55/95**
16	6.05/12.13/32	.37/.69/94	.28/.51/96	**.27/.49/97**
18	3.68/7.43/60	.24/.50/97	**.19/.39/100**	.20/.41/**100**
19	3.75/7.58/57	.47/.93/90	.32/.61/93	**.30/.56/94**
Mean	4.88/9.79/46	.44/.83/91	.31/.56/95	**.29/.53/96**

Table 9: Deterministic OF and GOF under both readouts. FB: Field-Bezier; G: greedy. Every readout is reported at 20 ms extraction time, excluding field generation. Cells: ADE/FDE/HitRate (%).

Seq	Det. OF + FB	GOF + FB	Det. OF + G	GOF + G
08	.41/.73/91	.40/.67/92	.36/.67/94	.29/.49/96
13	.44/.75/91	.46/.70/91	.34/.66/95	.39/.67/94
15	.39/.57/93	.38/.60/92	.28/.52/98	.31/.55/95
16	.39/.70/94	.36/.55/96	.32/.64/97	.27/.49/97
18	.29/.51/98	.26/.48/99	.23/.46/100	.20/.41/100
19	.38/.65/92	.39/.63/92	.33/.61/94	.30/.56/94
Mean	.38/.65/93	.37/.61/94	.31/.59/96	.29/.53/96

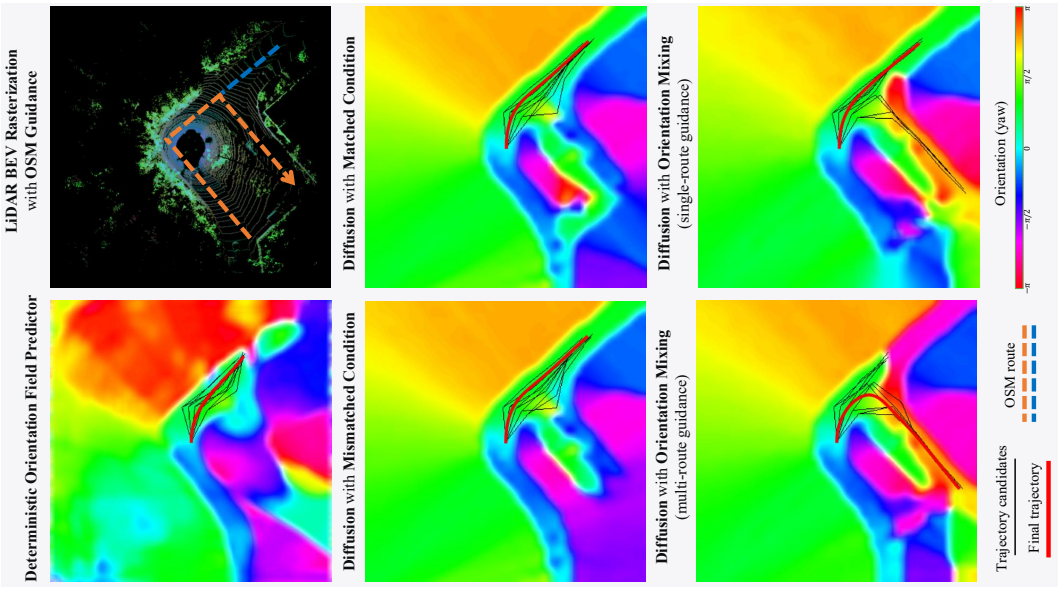


Figure 7: Qualitative comparison at a branching road. Left to right: LiDAR/navigation route, deterministic field, matched and mismatched route-conditioned diffusion, and GOF with single- and multi-route cues. Black paths are candidates; red paths are selected readouts.

Runtime. Table 10 gives the runtime breakdown on an Intel Xeon 2.20 GHz CPU and one NVIDIA RTX A5000, for the $T = 5, K = 2$ sampling configuration. Denoising dominates end-to-end latency.

Table 10: Per-frame runtime, excluding offline graph construction and cleaning. Times are approximate.

Stage	Time (ms)
LiDAR/route rasterization	10
Field sampling	550
Trajectory extraction	20
Total	580

Mixing and re-noising add pointwise arithmetic and noise generation to the scheduled denoiser evaluations.

C.3 Temporal conditioning: component evidence and controlled study

The last two columns of Table 8 isolate history after Mixing and combined re-noising. History improves ADE/FDE on five sequences but slightly worsens both on seq18.

Let z_n be the current field, $\widetilde{z}_{n-1}$ the preceding field after ego-motion alignment, and P_n their overlapping LiDAR scene support with valid vectors in both fields. The mean angular change in degrees is

$$J_{\text{dir}} = \frac{180}{\pi} \underset{n:|P_n|>0}{\text{mean}} \left[\frac{1}{|P_n|} \sum_{p \in P_n} \arccos \frac{z_n(p)^\top \widetilde{z}_{n-1}(p)}{\|z_n(p)\| \, \|\widetilde{z}_{n-1}(p)\|} \right].$$

The controlled study (4) fixes LiDAR, checkpoint, spatial guidance, and readout, with paired randomness and independent recurrent states. Own reduces mean J_{dir} by 2.44%, improving all six sequences; Conflicting increases it on all six. This supports content-dependent directional stabilization.

C.4 GUIDANCE SCHEDULES

Table 11 examines when to apply guidance. On seq08, early-only and decaying guidance outperform late-only or constant guidance; the SNR schedule has the lowest reported ADE/FDE under both compared budgets. Together with Table 5, this supports the selected schedule and refresh allocation on seq08.

Table 11: Seq08 guidance schedules. Cells: ADE/FDE/HitRate (%). Budgets follow the $T \times K$ convention in Appendix B.

Schedule	5×2	5×3
Constant	.38/.73/91	.41/.83/90
Linear decay	.33/.57/95	.31/.54/95
Cosine decay	.30/.50/96	.30/.50/96
Early only	.31/.52/95	.31/.52/96
Late only	.39/.75/91	.42/.84/90
SNR schedule	.29/.49/96	.29/.49/96

C.5 MAP-MISALIGNMENT ROBUSTNESS

We perturb route geometry on sequence 08 while holding LiDAR, checkpoint, sampling settings, planner, evaluation frames, and reference paths fixed within each method. Field methods use the same planner on direction-normalized fields; TP (Xu et al., 2022) outputs trajectories directly. R2/R5/R10 rotate each frame-local route crop by $2^\circ/5^\circ/10^\circ$. T2/T4/T8 translate the whole graph by 2/4/8 m along $+x$. Table 12 reports ADE/FDE in meters, HitRate in percent, and the relative change triple

$$100\,(\text{perturbed} - \text{same-method clean})/\text{same-method clean}.$$

HitRate change is relative percent, not percentage points. Relative changes use unrounded metrics. Absolute metrics and relative changes are ranked separately within each stress row, using the same lower-error/higher-HitRate rule.

Table 12: Seq08 map perturbations. Cells show ADE/FDE/HitRate (m/m/%), with clean-relative changes (%) below in brackets.

Stress	GOF	OriField	TP	DPS
Clean	**.286/.485/95.947** [0/0/0]	.359/.663/94.424 [0/0/0]	.504/1.015/88.357 [0/0/0]	1.035/2.155/85.949 [0/0/0]
R2	**.287/.489/95.922** [+.64/ + .99/ − .03]	.370/.692/94.178 [+2.97/ + 4.37/ − .26]	.505/1.019/88.160 [+.27/ + .38/ − .22]	1.031/2.148/86.023 [**−.31/ − .34/ + .09**]
R5	**.288/.491/95.947** [+.85/ + 1.36/0]	.406/.781/93.957 [+13.01/ + 17.77/ − .49]	.522/1.061/87.521 [+3.56/ + 4.56/ − .95]	1.036/2.158/85.974 [**+.11/ + .11/ + .03**]
R10	**.299/.520/95.898** [+4.80/ + 7.31/**−.05**]	.421/.829/92.778 [+17.33/ + 24.94/ − 1.74]	.604/1.284/84.451 [+19.84/ + 26.58/ − 4.42]	1.039/2.170/85.777 [**+.44/ + .70**/ − .20]
T2	**.285/.480/95.947** [−.37/ − .87/0]	.353/.651/94.547 [**−1.62/ − 1.78/ + .13**]	.515/1.033/87.423 [+2.21/ + 1.82/ − 1.06]	1.039/2.167/85.925 [+.40/ + .53/ − .03]
T4	**.285/.485/95.972** [**−.23/ + .18/ + .03**]	.367/.682/94.105 [+2.18/ + 2.80/ − .34]	.542/1.089/86.563 [+7.53/ + 7.31/ − 2.03]	1.045/2.184/85.753 [+1.03/ + 1.34/ − .23]
T8	**.295/.510/95.775** [+3.42/ + 5.17/**−.18**]	.394/.735/93.908 [+9.81/ + 10.82/ − .55]	.798/1.671/82.167 [+58.32/ + 64.65/ − 7.01]	1.061/2.231/85.384 [**+2.57/ + 3.49**/ − .66]

DPS changes little under these perturbations but already has higher clean ADE/FDE, so relative stability should be read alongside absolute task accuracy.

CODE AVAILABILITY

We will open-source the implementation and experiment configurations. The descriptions above give the main implementation logic; the released code will specify the exact preprocessing, sampling, and evaluation procedures for reproducibility.