WoRD: Diffusion-Based Posterior Inference for Online Goal Recognition

Almog Anschel
CLAIR Lab
Taub Faculty of Computer Science
Technion – Israel Institute of Technology
Haifa 3200003, Israel
anschelalmog@campus.technion.ac.il

Sarah Keren
CLAIR Lab
Taub Faculty of Computer Science
Technion – Israel Institute of Technology
Haifa 3200003, Israel
sarahk@technion.ac.il

Abstract

Goal recognition is the task of inferring an agent's intended goal from its observed behavior. Most existing work focuses on high-level symbolic recognition, abstracting away the continuous motion through which agents reach their goal. Methods that do reason over continuous trajectories rely on an explicit dynamics model that is hard to acquire and rarely transfers across settings. Neither approach explicitly learns how agents move toward each candidate goal, limiting accuracy on ambiguous trajectory prefixes where such knowledge is most needed.

We present WoRD (Waypoint Recognition with Diffusion), which uses a learned trajectory prior to infer the agent's goal from partial observations. At each step, WoRD runs one warm-started denoising chain per candidate goal, steers each chain toward the observed trajectory through a closed-form log-likelihood gradient, and reweights the resulting samples by Bayes' rule into a calibrated posterior over goals. We evaluate WoRD on simulated robot settings, pedestrian trajectories, and vehicle trajectories using four metrics targeting distinct failure modes. Across all three settings, the same inference procedure preserves calibrated multimodal beliefs through ambiguous prefixes where filter- and classifier-based observers collapse to a single mode or grow overconfident.

1 Introduction

Goal recognition is the problem of inferring an agent's intended goal from observations of its behavior. In the online setting, an observer must maintain a belief over candidate goals as observations arrive, before the trajectory becomes unambiguous [18, 27]. The problem is hardest early: paths toward adjacent goals overlap until the trajectory commits. An observer that resolves this ambiguity prematurely makes confident-and-wrong predictions, committing to a single hypothesis before evidence warrants it. Figure 1 shows a concrete instance: a pedestrian midway through a real intersection where the past trajectory and current heading remain consistent with multiple candidate exits, so a faithful belief must remain multimodal.

Ambiguity of this kind is a structural feature of goal recognition. Keren et al. [14] characterize this ambiguity via *non-distinctive paths*, prefixes of goal paths before which an observer cannot recognize the agent's goal. A principled observer must maintain multimodal beliefs through this prefix; most existing methods instead collapse the ambiguity before it ends. Kalman-based filters such as OWR [8] maintain one Gaussian per candidate goal. When two goals are equally likely, the filter locks onto whichever hypothesis aligns marginally better with early noisy measurements, and the unimodal posterior cannot recover. Particle filters represent richer posteriors, but depend on a well-specified dynamics model. For agents whose motion is goal-directed and obstacle-conditioned,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

such a model is achievable in principle, but any practical approximation is prone to error and limited in scope [16]. Discriminative classifiers produce point predictions that do not separate accuracy from calibration. Trained with cross-entropy, they are optimized to place all probability on the labeled goal, pushing probability onto the favored candidate even when observations support multiple goals [3, 22].

We exploit diffusion models [10, 26], which learn a data distribution by reversing a gradual noise process: starting from random noise, a learned score function denoises samples step by step into structured outputs. These models are well suited to trajectory data, where the distribution of plausible paths is branching and obstacle-shaped, since a score network can denoise Gaussian noise into coherent, goal-directed paths that no single parametric form captures.

We present *Waypoint Recognition with Diffusion (*WORD*)*, which uses this denoising framework for inference rather than generation. For each candidate goal in the waypoint set, WORD runs an independent denoising chain over trajectory space. At each diffusion step, a score-guidance term derived from the observer's measurement model steers the chain toward trajectories consistent with the observations so far. Because the chains share no state, each one keeps its own hypothesis, and the posterior remains multimodal through ambiguous prefixes. Warm-starting from the previous estimate, instead of pure noise, allows the update to track incoming observations in real time. The per-chain trajectory likelihoods are then combined by Bayesian reweighting into a single calibrated posterior over the waypoint set.

We evaluate WORD on complex simulation environments and on two real-world traffic datasets: rounD [15] (vehicles at roundabouts) and inD [2] (pedestrians at intersections). We compare against five baselines, including a Reverse Diffuser [11] baseline that applies the same score network with post-hoc guidance, isolating the contribution of online inference.



Figure 1: **Online goal recognition at a real intersection (illustrative example).** A pedestrian (blue dot, with an arrow indicating heading), assumed to be goal-directed, is partway along an observed trajectory (solid black) toward one of four candidate exits (colored stars); the unobserved future is dashed. At this point, the past path and the current heading remain consistent with two of the exits (green w^1, red w^2) so a faithful belief over the agent's goal must keep both alive until later observations rule one out.

Contributions.

1. **Goal recognition as diffusion inference.** We formulate online goal recognition as posterior inference via independent per-goal denoising chains, combined by Bayesian reweighting over reconstructed trajectories, providing a data-driven approach to goal recognition in complex environments where hand-specified motion models are infeasible. Unlike planning-based diffusion methods that condition generation on a known goal [11, 12], WORD inverts the conditioning: it infers goals from partial observations.
2. **Online goal recognition algorithm.** We derive a warm-start mechanism that reuses the previous trajectory estimate as a starting point for each new observation, reducing per-update computation enough for online use at the sensor sampling rate.
3. **Analytic observation guidance.** By grounding the score-guidance term in the physical geometry of trajectory observations, we derive a closed-form update that requires no learned classifier and no additional training.

2 Related Work

Goal Recognition: Prior taxonomies [17, 18, 27, 29] group goal recognition methods into three families, which map to the failure modes of section 1: planning-based methods relying on hand-crafted domain models, probabilistic filters, and discriminative classifiers.

Planning-based methods cast goal recognition as inverse planning: for each candidate goal, an observer generates an optimal plan and scores it against observed actions [20, 24]. These methods require a hand-crafted symbolic domain model and assume the agent acts optimally, limiting applicability to continuous or stochastic environments [13]. By contrast, WoRD replaces the symbolic domain model with a learned score field trained on observed trajectories, making no assumption about whether the agent plans optimally.

Probabilistic approaches treat goal recognition as online Bayesian inference, with g a candidate goal and $\mathbf{Z}_{1:k}$ the trajectory observed so far. They maintain a posterior $p(g \mid \mathbf{Z}_{1:k})$ that updates with each observation [21]. Particle filters provide a scalable approximation [16], but represent beliefs as discrete weighted samples: weight collapse under sharp-peaked likelihoods causes most particles to lose support at the moment the trajectory commits, and downstream gradient-based planning requires lossy smoothing to recover a continuous belief. WoRD avoids these discretizations: the denoising chain represents beliefs as a continuous score field, so observation-likelihood gradients are available analytically at every step without smoothing.

Guo et al. [8] formulate waypoint recognition as a recursive filtering problem, running parallel Kalman filters for each candidate waypoint and weighting them by observation likelihood. This Kalman framework constrains each per-waypoint belief to a unimodal Gaussian, preventing the system from representing ambiguous trajectories where a single partial observation is consistent with multiple divergent paths. WoRD replaces OWR's per-waypoint filter with a learned score function that maintains the full multimodal posterior.

Learned goal recognition methods cast the problem as discriminative classification [3, 6, 22, 23]: a model maps $\mathbf{Z}_{1:k}$ to a posterior $p(g \mid \mathbf{Z}_{1:k})$ over candidate goals, bypassing symbolic domain models at the cost of requiring labeled trajectory data. By learning $p(g \mid \mathbf{Z}_{1:k})$ directly without modeling the generative process $p(\mathbf{Z}_{1:k} \mid g)$, these methods are brittle when observations are partial or the goal set shifts at test time. WoRD instead models $p(\mathbf{Z}_{1:k} \mid g)$ via a learned trajectory prior, recovering the goal posterior by Bayes' rule and avoiding the train-time overconfidence bias of cross-entropy.

Diffusion Models: Score-based generative models learn a data distribution by training a neural network to estimate the score $\nabla_x \log p(x)$ and then draw samples via Langevin dynamics [25, 26]. Ho et al. [10] show that this framework is equivalent to *denoising diffusion probabilistic models* (DDPMs), unifying the two formulations under a common objective. Conditional generation is typically achieved either through classifier guidance [5], which injects the gradient of a separately trained classifier $p(y \mid x_t)$ at sampling time, or through classifier-free guidance [9], which integrates the conditioning into the score network at training time. WoRD belongs to the classifier-guidance family, but replaces the learned classifier with an analytic gradient of the observation likelihood through the measurement model h, so no auxiliary classifier is trained or evaluated at inference. The score network itself plays the role of a process model: filter-based observers specify the dynamics analytically, whereas WoRD's prior is learned from trajectory data.

In planning, Janner et al. [11] treat trajectory optimization as conditional denoising: a DDPM trained on expert trajectories is guided at inference time toward goal-satisfying samples. MotionDiffuser [12] extends this to joint distributions over interacting agent trajectories, recovering multimodal collision-avoidance behaviors that unimodal models cannot represent. In trajectory prediction, conditional diffusion models generate distributions over plausible futures given past motion [1, 7], but the latent goal is never explicitly inferred. WoRD inverts this direction: rather than generating trajectories from a known goal or predicting futures from past states, it infers the latent waypoint $w^\star$ from a partial trajectory $\mathbf{Z}_{1:k}$ via score-based guidance over a joint trajectory-waypoint distribution.

Motion Forecasting: Map-conditioned forecasters such as QCNet [32], HPNet [28], and DeMo [31] predict multimodal future trajectories. Since every goal induces a trajectory, forecasting and goal recognition are closely related, yet they report different quantities: a forecaster returns a distribution over future trajectories, whereas a goal recognizer returns a posterior over a given candidate set, which mode probabilities yield only after aggregation over the candidates. The evaluation metrics differ accordingly: FDE, miss rate, and Brier-minFDE [30] score the best of K predicted modes, so none of them penalizes a belief that commits to a wrong candidate before the observed prefix disambiguates the goal. Deriving the goal from forecast trajectories is a predict-then-classify strategy; our Reverse Diffuser baseline (section 5.1) is a controlled instance of it that shares WoRD's score network.

3 Problem Formulation

We consider an agent navigating under closed-loop control in a bounded planar environment $\mathcal{E} \subseteq \mathbb{R}^2$ with static obstacles toward an unknown target $w^* \in \mathcal{W}$, where $\mathcal{W} = \{w^1, \ldots, w^N\}$ is a finite set of candidate waypoints known to the observer. At time step t, the agent's state is $\mathbf{x}_t = [\mathbf{p}_t^\top, \mathbf{v}_t^\top, \theta_t]^\top \in \mathcal{X} \subseteq \mathbb{R}^5$, with position $\mathbf{p}_t \in \mathbb{R}^2$, velocity $\mathbf{v}_t \in \mathbb{R}^2$, and heading $\theta_t \in [0, 2\pi)$.

The observer cannot access $\mathbf{x}_t$ directly. At each step k it receives a noisy measurement

$$\mathbf{z}_k = h(\mathbf{x}_k) + \boldsymbol{\eta}_k, \qquad \boldsymbol{\eta}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \tag{1}$$

where $h : \mathcal{X} \to \mathcal{Z} \subseteq \mathbb{R}^m$ is the measurement map and $\mathbf{R} \in \mathbb{R}^{m \times m}$ is a known, time-invariant, symmetric positive-definite noise covariance. The observation history up to time k is $\mathbf{Z}_{1:k} = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$. The online goal recognition objective is to compute the posterior over candidate goals,

$$p(w^* = w^j \mid \mathbf{Z}_{1:k}), \quad \forall\, w^j \in \mathcal{W}, \tag{2}$$

required both to converge to w^* once the agent's behavior is distinctive and to remain multimodal while the prefix is consistent with multiple goals. The observer is online: at each k, it must produce eq. (2) from $\mathbf{Z}_{1:k}$ alone, without access to future observations.

We note that optimal plans toward distinct goals may share a long non-distinctive prefix whenever the worst-case distinctiveness (WCD) [14] is large. Within such a prefix the true posterior is multimodal, and an observer that collapses to a single mode is wrong in a way the argmax cannot detect, even when the eventual argmax is correct.

Assumptions. We make the following assumptions: (1) The setting is keyhole: the agent's policy is observer-agnostic, with its actions independent of the observer's presence, and the observer takes no action to influence the agent or environment [18]; (2) the environment is static: obstacles and passages do not change across the observation horizon; (3) the observer is trained or configured in the same environment in which it is deployed; no map is provided as input at inference; (4) $\mathcal{W}$ is finite and known to the observer; (5) h is differentiable; section 4.3 uses this to compute likelihood gradients $\nabla_{\mathbf{x}} \log p(\mathbf{z}_k \mid \mathbf{x}_k)$ analytically; (6) the agent is intentional, navigating toward some $w^* \in \mathcal{W}$, though its trajectory may not be optimal due to obstacles or kinematic constraints; (7) observations arrive with known timestamps and bounded noise.

WORD shares assumptions (1)–(4), (6), and (7) with OWR [8]. OWR additionally requires a linear-Gaussian process model for its UKF prediction step, whereas WORD requires only that h admits a gradient and substitutes a learned score function for the Kalman prediction step.

4 Waypoint Recognition with Diffusion (WORD)

WORD learns the motion through which agents reach their goals from a dataset of demonstrations, removing the need for a hand-specified dynamics model. At inference time we invert the learned prior: at each observation step, one denoising chain per candidate goal is steered toward the observed trajectory, and the chain endpoints are reweighted by Bayes' rule into a posterior.

Given a partial observation history $\mathbf{Z}_{1:k}$ and a candidate set $\mathcal{W} = \{w^j\}_{j=1}^N$, WORD returns a posterior $p(w^j \mid \mathbf{Z}_{1:k})$ over waypoints. WORD introduces a latent trajectory $\boldsymbol{\tau} = [\mathbf{x}_0, \ldots, \mathbf{x}_{T-1}] \in \mathbb{R}^{T \times d}$ over a fixed horizon T. The score network denoises $\boldsymbol{\tau}$ and marginalizes to yield a likelihood per candidate w^j; d is the agent state dimension. Throughout, t indexes physical time and i indexes diffusion step; $\boldsymbol{\tau}_i(t)$ is the value at physical time t of the trajectory at diffusion step i.

For each candidate w^j, WORD models the joint distribution $p(\boldsymbol{\tau}, \mathbf{Z}_{1:k} \mid w^j)$ as the product of two terms: a learned *conditional trajectory prior* $p(\boldsymbol{\tau} \mid w^j)$ realized by a DDPM, and an analytic *observation likelihood* $p(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau})$ obtained by differentiating the measurement model h (eq. (1)). At inference, WORD runs one denoising chain per candidate, guides each reverse step with the analytic observation-likelihood gradient, and reweights the resulting trajectories by Bayes' rule (eq. (9)) to produce the waypoint posterior. Figure 2 illustrates the full inference loop.

4.1 Conditional trajectory prior

We model the conditional trajectory distribution $p(\boldsymbol{\tau} \mid w^j)$ for each candidate $w^j \in \mathcal{W}$, treating $w^j \in \mathbb{R}^2$ as a continuous conditioning vector rather than a discrete index, allowing arbitrary waypoint

coordinates without architectural changes (architecture in section 4.2). The network thus learns the generative direction $p(\boldsymbol{\tau} \mid w)$ and never the discriminative $p(w \mid \boldsymbol{\tau})$: it has no output indexed by candidate, and the posterior over $\mathcal{W}$ is assembled only at inference by Bayes' rule (eq. (9)), so the candidate set can change at inference without retraining (section E.1). We adopt a diffusion prior because observation guidance must act online: guidance enters as an additive gradient on the prior score, which requires both a tractable score and an iterative sampler in which to apply it.

Forward process. The DDPM [10] forward process corrupts a clean trajectory $\boldsymbol{\tau}_0$ over N steps using a prescribed variance schedule $\{\beta_i\}_{i=1}^{N} \subset (0, 1)$:

$$q(\boldsymbol{\tau}_i \mid \boldsymbol{\tau}_{i-1}) = \mathcal{N}\Big(\boldsymbol{\tau}_i;\ \sqrt{1-\beta_i}\,\boldsymbol{\tau}_{i-1},\ \beta_i \mathbf{I}\Big)\,, \tag{3}$$

with closed-form marginal $q(\boldsymbol{\tau}_i \mid \boldsymbol{\tau}_0) = \mathcal{N}(\boldsymbol{\tau}_i;\ \sqrt{\bar{\alpha}_i}\,\boldsymbol{\tau}_0,\ (1-\bar{\alpha}_i)\mathbf{I})$, where $\bar{\alpha}_i = \prod_{s=1}^{i}(1-\beta_s)$.

Reverse process. The reverse process denoises $\boldsymbol{\tau}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ conditioned on goal w^j and observation history $\mathbf{Z}_{1:k}$. The conditional mean $\boldsymbol{\mu}_\theta$ is the noise-corrected estimate of $\boldsymbol{\tau}_{i-1}$: a learned noise predictor $\boldsymbol{\epsilon}_\theta$ estimates the Gaussian perturbation added to $\boldsymbol{\tau}_{i-1}$ during the forward process at step i, and $\boldsymbol{\mu}_\theta$ subtracts a scaled version of that prediction from $\boldsymbol{\tau}_i$, with coefficients fixed by the noise schedule [10]:

$$\begin{aligned} p_\theta(\boldsymbol{\tau}_{i-1} \mid \boldsymbol{\tau}_i, w^j, \mathbf{Z}_{1:k}) &= \mathcal{N}\big(\boldsymbol{\tau}_{i-1};\ \boldsymbol{\mu}_\theta,\ \sigma_i^2 \mathbf{I}\big)\,, \\ \boldsymbol{\mu}_\theta &= \frac{1}{\sqrt{\alpha_i}}\left(\boldsymbol{\tau}_i - \frac{\beta_i}{\sqrt{1-\bar{\alpha}_i}}\,\boldsymbol{\epsilon}_\theta\big(\boldsymbol{\tau}_i, w^j, i, \mathbf{Z}_{1:k}\big)\right). \end{aligned} \tag{4}$$

4.2 Training

We train the score network in two stages that mirror the score decomposition of eq. (6): stage 1 fits the trajectory prior $p(\boldsymbol{\tau} \mid w^j)$, and stage 2 fits the observation-conditional $p(\boldsymbol{\tau} \mid w^j, \mathbf{Z}_{1:k})$. Both stages use the same dataset $\mathcal{D} = \{(\boldsymbol{\tau}_0^{(i)}, w^{(i)})\}_{i=1}^{|\mathcal{D}|}$ of trajectories paired with goal waypoints, drawn from the target goal-conditional distribution. End-to-end training tends to be unstable: the network can learn to regress observations directly and bypass the prior, yielding low training loss but poor generalization beyond the observed prefix. Pretraining stage 1 stabilizes optimization and lets each score component be evaluated and ablated independently.

Stage 1: Trajectory prior. The noise predictor is trained with the observation channel masked,

$$\mathcal{L}_{\text{prior}} = \mathbb{E}_{(\boldsymbol{\tau}_0, w^j)\sim\mathcal{D},\,\boldsymbol{\epsilon},\,i}\Big[\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}_i, w^j, i, \emptyset)\big\|_2^2\Big]\,. \tag{5}$$

Stage 2: Observation conditioning. The pretrained network is fine-tuned end-to-end with the observation channel populated, optimizing the same denoising-score-matching objective with $\mathbf{Z}_{1:k}$ in place of $\emptyset$. At each training step we sample a prefix length $k = \lfloor \rho T \rfloor$ with ρ drawn uniformly from a fixed range (in our experiments $\rho \sim \mathcal{U}(0.3,\ 0.9)$) and synthesize $\mathbf{Z}_{1:k}$ from the clean trajectory $\boldsymbol{\tau}_0$ via the observation model of eq. (1).

Architecture. The score network is a 2.2M-parameter Temporal U-Net [11] with FiLM-based waypoint conditioning [19]; full architecture, hyperparameters, and the stage 2 data-augmentation schedule are in appendix C.

4.3 Observation guidance

The observation-conditioned score decomposes via Bayes' rule [5] (derivation in appendix A):

$$\nabla_{\boldsymbol{\tau}_i} \log p(\boldsymbol{\tau}_i \mid w^j, \mathbf{Z}_{1:k}) = \underbrace{\nabla_{\boldsymbol{\tau}_i} \log p(\boldsymbol{\tau}_i \mid w^j)}_{\text{learned prior}} + \underbrace{\nabla_{\boldsymbol{\tau}_i} \log p(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau}_i)}_{\text{observation guidance}}\,. \tag{6}$$

In practice we evaluate the leading term with the stage-2 conditional network (section 4.2) rather than the bare prior, as $-\boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}_i, w^j, i, \mathbf{Z}_{1:k})/\sqrt{1-\bar{\alpha}_i}$ via the noise-prediction-to-score equivalence [10]. We replace the Bayes-rule weight on the guidance term by a tunable scale λ [5], which increases observation fidelity beyond what the stage-2 conditioning already captures; sensitivity is reported in

appendix E. For the guidance term, the Gaussian observation model from eq. (1) yields a closed-form log-likelihood, differentiable in $\boldsymbol{\tau}_i$ through h:

$$\log p(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau}_i) = -\frac{1}{2} \sum_{t=1}^{k} \|\mathbf{z}_t - h(\boldsymbol{\tau}_i(t))\|^2_{\mathbf{R}^{-1}} . \tag{7}$$

The guidance gradient $\mathbf{g}^j = \nabla_{\boldsymbol{\tau}_i^j} \log p(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau}_i^j)$ is computed by backpropagation through h, which assumption 5 (section 3) makes differentiable, so no learned likelihood model or classifier head [5] is needed. Together with the network's learned observation conditioning (section 4.2), the analytic gradient provides an exact likelihood signal at every diffusion step.

4.4 Online inference with warm-start

Denoising from pure noise at every observation is incompatible with online operation. The previous estimate $\hat{\boldsymbol{\tau}}_{k-1}$ (cached at step 4 below) already lies near the data manifold, so we re-noise it to an intermediate step $i^* \ll N$ via the closed-form marginal,

$$\boldsymbol{\tau}_{i^*}^j \sim q(\boldsymbol{\tau}_{i^*} \mid \hat{\boldsymbol{\tau}}_{k-1}) = \mathcal{N}\big(\sqrt{\bar{\alpha}_{i^*}}\, \hat{\boldsymbol{\tau}}_{k-1},\, (1-\bar{\alpha}_{i^*})\mathbf{I}\big) , \tag{8}$$

and reverse from i^* to 0, cutting per-update cost by a factor of N/i^*. At each new observation $\mathbf{z}_k$, WoRD performs four steps per candidate w^j:

1. **Re-noise**: corrupt $\hat{\boldsymbol{\tau}}_{k-1}^j$ to step i^*.
2. **Guided reverse**: denoise from i^* to 0, adding $\lambda \mathbf{g}^j$ to the prior score at each step.
3. **Reweight**: update the posterior via eq. (9).
4. **Store**: cache $\hat{\boldsymbol{\tau}}_k^j = \boldsymbol{\tau}_0^j$ for the next warm-start.

To reduce variance, we draw S trajectories per candidate from the shared warm-start and estimate the marginal likelihood as

$$\hat{p}(\mathbf{Z}_{1:k} \mid w^j) \;=\; \frac{1}{S} \sum_{s=1}^{S} p\Big(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau}_0^{j,(s)}\Big) . \tag{9}$$

Combining these per-candidate marginal likelihoods with the candidate prior via Bayes' rule yields the posterior $p(w^j \mid \mathbf{Z}_{1:k})$. Appendix B states the full inference procedure.

5 Experiments

We evaluate WoRD on five domains and five baselines. The domains split into two settings by whether goal-directed behavior is distinctive in the early prefix: *non-complex*, with optimal trajectories that diverge early, and *complex*, with prefixes that stay ambiguous for many timesteps. The central claim under test is that WoRD preserves multimodal beliefs through these ambiguous prefixes, which yields better performance when premature unimodal commitment misrepresents the true posterior, while still converging onto w^* once the trajectory becomes distinctive. The formulation also transfers across agent types without method-level changes, consistent with the data-source-agnostic training procedure of section 4.2.

5.1 Setup

Domains. We created two simulated domains by procedurally generating 2D worlds: **Sim 3WP** and **Sim 4WP** use three and four waypoints, respectively, while **Sim 4WP-complex** (**Sim 4WP-c**) uses a four-waypoint complex variant with longer ambiguous prefixes. Obstacle layouts and trajectories are randomized to produce ambiguous prefixes, the setting in which multimodal belief preservation matters. For real-world data, we use two human-driven datasets. From the rounD dataset [15] we extracted the vehicle subset **rounD vehicles** (**rounD-V**); from the inD dataset [2] we extracted the pedestrian-agent subset **inD pedestrians** (**inD-Ped**). For each real-world domain, we retrain the score network with the same two-stage procedure described in section 4.2. Locations covered, goal extraction, clustering hyperparameters, per-domain sampling rates, observation models, and episode counts are in appendix D. **Sim 3WP** and **Sim 4WP** are the non-complex domains; **Sim 4WP-complex**, **rounD vehicles**, and **inD pedestrians** are the complex domains with extended ambiguous prefixes.

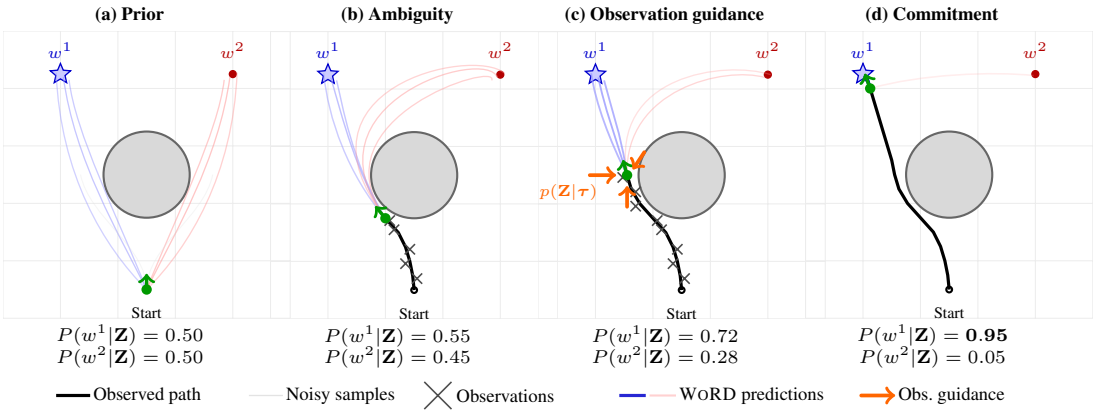


Figure 2: **WoRD online inference on a 2-goal synthetic scenario.** An agent (green) starts at the bottom of a 10×10 world and must reach one of two goals separated by a circular obstacle; WoRD maintains one denoising chain per candidate goal and updates a posterior as observations arrive. **(a)** *Prior.* Before any observation, the score network ϵ_θ shapes Gaussian noise into two obstacle-aware trajectory clusters, one towards each waypoint, and the goal prior is uniform. **(b)** *Ambiguity.* The agent has skirted the lower edge of the obstacle but its heading is still consistent with both goals; both clusters survive and the posterior remains nearly balanced. A filtering baseline would already have collapsed onto one mode here. **(c)** *Observation guidance.* Once the agent commits left of the obstacle, the score-guidance term $\nabla_{\boldsymbol{\tau}} \log p(\mathbf{Z} \mid \boldsymbol{\tau})$ (orange) pins each chain to the observed path; the right cluster must arc awkwardly over the obstacle to remain reachable, and posterior mass shifts toward w^1. **(d)** *Commitment.* As the agent approaches w^1, the surviving chain dominates and the posterior collapses. Pseudocode is in algorithm 1.

Baselines. We compare WoRD against five baselines spanning the main families of online goal-recognition methods, in the order of the failure-mode discussion in section 1: (1) **Naive**, a distance-and-heading geometric heuristic. It serves as a floor: since it ignores dynamics, observation history, and goal structure, a recognizer that fails to outperform it is mismatched to the domain. (2) **Particle Filter (PF)**, structurally identical to OWR but replacing the per-waypoint Gaussian filter with a particle-based posterior, using dynamics propagation and observation-weighted resampling. This isolates the contribution of the unimodal Gaussian assumption from the rest of the OWR pipeline. (3) **Online Waypoint Recognizer (OWR)** [8], a UKF-per-waypoint filter whose precomputed reference trajectories are generated from a hand-specified dynamic model of the agent. (4) **GRU Classifier (GRU)** [4], trained with cross-entropy on the same trajectory data, the discriminative-learning alternative for goal recognition [3, 22] that sees the same data but commits to a single softmax per timestep with no generative model. (5) **Reverse Diffuser (RD)** [11], which uses the same WoRD model but applies observation guidance *post-hoc* rather than during denoising and re-initializes each chain from noise at every update; RD is thus a controlled instance of the predict-then-classify strategy.

Metrics. A goal recognizer faces two failure modes that argmax accuracy alone cannot separate. When a distinctive path is present, the optimal posterior peaks sharply on w^* and the recognizer should converge to it. When the prefix is non-distinctive (optimal plans toward different goals overlap), the true posterior is multimodal; unimodal collapse is the failure of placing $\max_j p_t(w^j)$ on a single waypoint before the trajectory disambiguates the goal, locking the recognizer onto a commitment that may be wrong. We use four metrics chosen to expose both modes. **Top-1** (↑, fraction of timesteps with $\arg\max_j p_t(w^j) = w^*$) and **Brier** (↓, $\frac{1}{T}\sum_{t,j}(p_t(w^j) - \mathbb{1}[w^j{=}w^*])^2$) test convergence and full-posterior calibration. **CVaR$_{0.1}$** (↑, mean $p_t(w^*)$ on the worst 10% of (episode, timestep) pairs) and **CaW** (↓, fraction of pairs with $\max_j p_t(w^j) \geq 0.92$ on an incorrect waypoint) test the other failure mode: whether the recognizer keeps w^* in its support through ambiguous prefixes and avoids unimodal collapse, which argmax-style metrics underweight. Extended results, including per-scene inD/rounD breakdown, are in appendix F.

Protocol. For each method we run 5 training seeds and evaluate each seed on 50 held-out episodes per simulation domain and 100 per real-world domain, reporting the mean $\pm$ 1.96 $\times$ SE across seeds (Table 3). Real-world episodes come from a leave-one-location-out cross-validation, so the

Table 1: **Simulation results.** Bold marks the best entry in each column. **Sim 3WP** and **Sim 4WP** are non-complex domains; **Sim 4WP-complex** is the complex variant.

	Top-1 (%)↑			Brier↓ $[0, 2]$		
Method	3WP	4WP	4WPc	3WP	4WP	4WPc
Naive	64.2±2.1	55.6±2.4	38.4±3.2	0.41±0.03	0.49±0.03	0.59±0.04
OWR	81.4±1.6	75.2±1.9	58.7±2.7	0.17±0.02	0.24±0.03	0.34±0.03
PF	74.8±2.1	76.8±2.0	64.2±2.4	0.19±0.02	0.23±0.02	0.30±0.03
GRU	**88.4±1.3**	**82.9±1.4**	67.8±2.1	**0.12±0.01**	**0.16±0.02**	0.27±0.02
RD	84.3±1.4	80.7±1.6	70.4±1.9	0.14±0.01	0.18±0.02	0.23±0.02
WoRD	84.6±1.5	82.4±1.3	**78.2±1.6**	0.13±0.01	0.17±0.02	**0.19±0.02**
	$\text{CVaR}_{0.1}$↑			CaW (%)↓		
Method	3WP	4WP	4WPc	3WP	4WP	4WPc
Naive	0.06±0.01	0.04±0.01	0.03±0.01	19.8±2.0	28.4±2.7	39.7±3.5
OWR	0.18±0.02	0.13±0.02	0.08±0.01	11.4±1.6	17.6±2.0	25.2±2.6
PF	0.26±0.02	0.22±0.02	0.17±0.02	14.2±1.8	19.3±2.1	23.1±2.3
GRU	0.13±0.02	0.10±0.02	0.07±0.01	24.7±2.2	36.5±2.9	50.8±3.7
RD	0.32±0.02	0.27±0.02	0.22±0.02	7.0±1.3	14.6±1.8	17.2±2.0
WoRD	**0.48±0.02**	**0.43±0.02**	**0.32±0.02**	**4.6±0.9**	**8.5±1.3**	**11.4±1.6**

Table 2: **Real-world results.** Bold marks the best entry in each column. **rounD vehicles** aggregates the three roundabout locations; **inD pedestrians** aggregates the four intersections. Per-scene splits are in section F.2.

	Top-1 (%)↑		Brier↓ $[0, 2]$		$\text{CVaR}_{0.1}$↑		CaW (%)↓	
Method	rounD-V	inD-Ped	rounD-V	inD-Ped	rounD-V	inD-Ped	rounD-V	inD-Ped
Naive	32.8±5.5	27.4±6.2	0.63±0.06	0.67±0.08	0.02±0.02	0.02±0.02	44.6±5.8	50.1±6.6
OWR	53.1±4.7	31.4±6.2	0.42±0.06	0.48±0.07	0.06±0.02	0.05±0.02	29.8±4.4	47.8±5.9
PF	62.7±4.5	54.2±5.2	0.31±0.05	0.36±0.05	0.14±0.03	0.11±0.03	24.7±3.8	28.6±4.5
GRU	63.4±4.6	57.1±5.4	0.32±0.05	0.39±0.06	0.05±0.02	0.04±0.02	48.4±4.6	51.2±5.8
RD	64.9±4.4	58.6±5.1	0.27±0.05	0.31±0.05	0.18±0.03	0.14±0.03	22.4±3.4	24.4±3.4
WoRD	**71.3±3.7**	**65.7±5.1**	**0.22±0.04**	**0.25±0.04**	**0.34±0.02**	**0.26±0.04**	**14.1±2.9**	**17.0±3.2**

test location is never seen during training. All baselines see the same observation history $\mathbf{Z}_{1:k}$ and candidate set $\mathcal{W}$ at every timestep. Hardware, GRU retraining, ablation seeds, and per-method wall-clock are documented in appendices C, E and F.

5.2 Results

Per-domain numbers are reported in Table 1 (simulation) and Table 2 (real-world); the four-metric trade-off scatter plots (fig. 5) are in appendix F.

Overall. WoRD is best on every $\text{CVaR}_{0.1}$ and CaW cell and on the Top-1 and Brier cells of all three complex domains; three of the four non-complex Top-1 and Brier cells are statistical ties with GRU under overlapping 95% CIs, with **Sim 3WP** Top-1 the one cell where GRU leads outside the CI (Tables 1 and 2). The split between the argmax-style metrics and the multimodal-belief metrics follows from the mechanism: argmax wins require keeping w^* in the support throughout the prefix, which only matters when prefixes become ambiguous, while $\text{CVaR}_{0.1}$ and CaW penalize unimodal collapse independently of whether the eventual argmax is correct.

Non-complex setting. On **Sim 3WP** and **Sim 4WP**, goal-directed trajectories diverge early enough that an aggressive early commitment is usually correct: the clarifying signal arrives before a wrong hypothesis can compound into a persistent error. Top-1 and Brier consequently reduce to ties among methods that converge by the end of the prefix. GRU leads WoRD on Top-1 by 3.8 pp on **Sim 3WP** and 0.5 pp on **Sim 4WP** (Table 1); only the **Sim 3WP** gap falls outside the 95% CI. This is the one cell where a discriminative classifier cleanly beats WoRD, because **Sim 3WP** prefixes

disambiguate too early to reward belief preservation. Brier ties on both. CaW exposes the cost: GRU's confident-and-wrong rate on **Sim 3WP** is 24.7 ± 2.2, more than five times WORD's $\mathbf{4.6 \pm 0.9}$, because cross-entropy training commits confidently regardless of prefix ambiguity. Here the argmax is correct often enough to mask this miscalibration.

Complex setting. **rounD vehicles** most clearly shows why multimodal belief preservation matters. Vehicles entering a roundabout follow an identical circular arc for most of the trajectory; the exit maneuver is the first and only distinctive part of the trajectory. Throughout the shared arc the optimal posterior is multimodal: a well-calibrated recognizer should distribute belief over all plausible exits and commit only as the exit signal emerges. $\text{CVaR}_{0.1}$ directly tests whether a method maintains this distribution. GRU's $\text{CVaR}_{0.1}$ on **rounD vehicles** is 0.05 ± 0.02: the true exit receives almost no probability mass during the shared arc, because cross-entropy training has already driven the softmax to commit to one exit long before the trajectory disambiguates the goal. WORD's $\text{CVaR}_{0.1}$ is $\mathbf{0.34 \pm 0.02}$, a more than six-fold improvement, because per-step score guidance keeps probability spread across plausible exits and updates proportionally as the trajectory disambiguates the goal. CaW shows the downstream consequence: GRU emits a confident-and-wrong belief on 48.4 ± 4.6 of (episode, timestep) pairs, while WORD exceeds the 0.92 threshold on an incorrect hypothesis on only $\mathbf{14.1 \pm 2.9}$.

The same pattern holds across the full complex set. WORD leads GRU on Top-1 by $+10.4$ pp on **Sim 4WP-complex**, $+7.9$ pp on **rounD vehicles**, and $+8.6$ pp on **inD pedestrians**, gaps that are negligible in the non-complex domains and arise because correct final recognition requires keeping w^* in the support throughout the ambiguous prefix. On **inD pedestrians**, where pedestrians share entry corridors before diverging to distinct destinations, GRU's CaW reaches 51.2 ± 5.8, on par with Naive's 50.1 ± 6.6 despite GRU's access to the full observation history; cross-entropy training has amplified overconfidence to the point where GRU's safety-critical error rate matches a method with no observation model at all. WORD remains at $\mathbf{17.0 \pm 3.2}$, a 34 pp gap that Top-1 alone does not reveal.

Mechanism. The Reverse Diffuser (RD) shares architecture and training with WORD but applies the observation likelihood as post-hoc reweighting of unguided samples and re-initializes each denoising chain from noise at every update instead of warm-starting. The CaW gap between RD and WORD therefore measures the joint contribution of online guidance and warm-start. That gap widens substantially in the complex setting: 2.4 pp on **Sim 3WP**, 6.1 pp on **Sim 4WP**, 5.8 pp on **Sim 4WP-complex**, 8.3 pp on **rounD vehicles**, and 7.4 pp on **inD pedestrians**. $\text{CVaR}_{0.1}$ shows the same pattern: RD's worst-decile $p_t(w^*)$ on **rounD vehicles** is 0.18 ± 0.03 compared to WORD's $\mathbf{0.34 \pm 0.02}$. Both metrics tie the multimodal belief advantage to online inference rather than to the architecture or prior, which the two methods share. The λ rows of Table 5 isolate guidance alone: at $\lambda = 0$, Top-1 falls to chance.

Transfer. OWR's Top-1 drops by 50 pp from **Sim 3WP** to **inD pedestrians** (81.4 ± 1.6 to 31.4 ± 6.2) and PF's by 20 pp (74.8 ± 2.1 to 54.2 ± 5.2) as their hand-coded dynamics models meet real-world geometry, while WORD's drop over the same span is 19 pp (84.6 ± 1.5 to $\mathbf{65.7 \pm 5.1}$), comparable in magnitude; the loss is attributable to distribution shift in the learned prior, whereas OWR and PF lose to model misspecification.

Key finding. WORD is the only observer in our evaluation that improves on all four metrics jointly. It reduces the safety-critical confident-and-wrong rate by roughly 8 percentage points relative to the strongest baseline (Table 2), without regressing on Top-1 accuracy. The advantage is largest on domains with naturally ambiguous prefixes (intersections, roundabouts), where filter-based observers commit early and discriminative classifiers grow overconfident. The Reverse Diffuser, which shares WORD's score network but reweights unguided samples post-hoc, is the strongest baseline on CaW and trails WORD by 0.06 Brier and ~ 8 percentage points of confident-and-wrong rate. The safety-critical gain therefore comes from online inference rather than from the learned prior.

6 Conclusion

We present WORD, which casts online goal recognition as posterior inference under a learned diffusion prior: each candidate waypoint anchors an independent denoising chain, observation guidance enters analytically through the gradient of the measurement model, and warm-starting

keeps per-update cost compatible with online operation. Across three simulated and two real-world domains, WORD is the only observer that improves on all four metrics jointly. It preserves calibrated multimodal beliefs through ambiguous prefixes where filter- and classifier-based observers collapse to a single mode or grow overconfident, and it lowers the confident-and-wrong rate by roughly 8 percentage points relative to the strongest baseline on real-world data.

Limitations. Three assumptions limit WORD's applicability. The candidate waypoint set $\mathcal{W}$ must be specified in advance: the method recognizes goals from a known finite set and cannot discover new ones at inference time. This closed-set assumption is standard in goal recognition and shared with OWR; when it is violated, WORD degrades toward uncertainty, whereas OWR commits confidently to a wrong goal (section E.1). All real-world results hold out the test location, so deployment does not require training data from the target site. Symbolic goal recognition over discrete action domains is complementary to our continuous-trajectory setting and outside our scope. The measurement model h must be known and differentiable; sensor models with non-differentiable gates (rejected detections, hard occlusion masks) require a smoothed surrogate. Finally, although warm-starting reduces per-update cost by a factor of N/i^*, WORD remains more expensive per observation than a Kalman filter step (section C.1); on hardware-constrained platforms the appropriate operating point trades off the cost of confident-and-wrong predictions against inference latency.

Future Directions. WORD currently operates on planar position trajectories; extending to full 3D state with attitude and angular velocity is a question of training data and network capacity rather than method-level changes. Open-set goal discovery, in which $\mathcal{W}$ is augmented from observations, would relax the most restrictive limitation above; the conditional score network already supports continuous goal coordinates, so the remaining bottleneck is detecting new candidates. Latent diffusion would extend WORD to image- or video-based intent inference without changing the inference principle, but raises two open problems: guidance is informative only if progress toward a goal varies smoothly in the learned latent geometry, and calibration must survive the encoder. Finally, the multimodal posteriors that WORD maintains are the input that risk-sensitive planners need to reason about branching futures and deceptive agents.

Broader Impacts. Better-calibrated goal recognition supports safer autonomous planning: planners can defer commitment under ambiguity rather than collapse to a single hypothesis and replan when the agent diverges. The same framework applied to passive trajectory streams could enable behavioral surveillance, and a planner that anticipates pedestrian intent could in principle plan adversarially against humans. Responsible deployment therefore requires consent frameworks for the data sources that feed the score network, and oversight of the planners that consume WORD's posterior.

References

[1] Inhwan Bae, Young-Jae Park, and Hae-Gon Jeon. SingularTrajectory: Universal trajectory predictor using diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17890–17901, 2024. doi: 10.1109/CVPR52733.2024.01694.

[2] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections. In *IEEE Intelligent Vehicles Symposium (IV 2020)*, pages 1929–1934, 2020. doi: 10.1109/IV47402.2020.9304839.

[3] Mattia Chiari, Alfonso Emilio Gerevini, Francesco Percassi, Luca Putelli, Ivan Serina, and Matteo Olivato. Goal Recognition as a Deep Learning Task: The GRNet Approach. *Proceedings of the International Conference on Automated Planning and Scheduling*, 33(1):560–568, July 2023. ISSN 2334-0843, 2334-0835. doi: 10.1609/icaps.v33i1.27237. URL https://ojs.aaai.org/index.php/ICAPS/article/view/27237.

[4] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics, 2014.

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

[6] Osher Elhadad, Owen Morrissey, and Reuth Mirsky. General dynamic goal recognition using goal-conditioned and meta reinforcement learning. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2026, Paphos, Cyprus, May 25–29, 2026*, 2026. URL https://arxiv.org/abs/2505.09737.

[7] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17113–17122, 2022.

[8] Jia Guo, Sushrut Surve, Zilong He, Silvia Ferrari, and Sarah Keren. Online waypoint recognition of controlled agents in uncertain environments. In *IEEE International Conference on Robotics and Automation, ICRA 2025, Atlanta, GA, USA, May 19-23, 2025*, pages 12600–12606. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11128799. URL https://doi.org/10.1109/ICRA55743.2025.11128799.

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

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[11] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*, pages 9902–9915, 2022. URL https://proceedings.mlr.press/v162/janner22a.html.

[12] Chiyu Max Jiang, Andre Cornman, Cheolho Park, Ben Sapp, Yin Zhou, and Dragomir Anguelov. MotionDiffuser: Controllable multi-agent motion prediction using diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2023)*, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Jiang_MotionDiffuser_Controllable_Multi-Agent_Motion_Prediction_Using_Diffusion_CVPR_2023_paper.html.

[13] Gal Kaminka, Mor Vered, and Noa Agmon. Plan recognition in continuous domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.12097. URL https://ojs.aaai.org/index.php/AAAI/article/view/12097.

[14] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal Recognition Design. *Proceedings of the International Conference on Automated Planning and Scheduling*, 24:154–162, May 2014. ISSN 2334-0843, 2334-0835. doi: 10.1609/icaps.v24i1.13617. URL https://ojs.aaai.org/index.php/ICAPS/article/view/13617.

[15] Robert Krajewski, Tobias Moers, Julian Bock, Lennart Vater, and Lutz Eckstein. The round dataset: A drone dataset of road user trajectories at roundabouts in germany. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020. doi: 10.1109/ITSC45102.2020.9294728.

[16] Jean Massardi, Mathieu Gravel, and Éric Beaudry. Error-Tolerant Anytime Approach to Plan Recognition Using a Particle Filter. *Proceedings of the International Conference on Automated Planning and Scheduling*, 29:284–291, July 2019. ISSN 2334-0843, 2334-0835. doi: 10.1609/icaps.v29i1.3490. URL https://ojs.aaai.org/index.php/ICAPS/article/view/3490.

[17] Felipe Meneguzzi and Ramon Fraga Pereira. A survey on goal recognition as planning. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4524–4532, 2021. doi: 10.24963/ijcai.2021/616.

[18] Reuth Mirsky, Sarah Keren, and Christopher Geib. *Introduction to symbolic plan and goal recognition*, volume 16. Springer, 2021.

[19] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 3942–3951, 2018. doi: 10.1609/aaai.v32i1.11671.

[20] Miquel Ramırez and Hector Geffner. Plan recognition as planning. In *Proceedings of the 21st international joint conference on Artifical intelligence. Morgan Kaufmann Publishers Inc*, pages 1778–1783, 2009.

[21] Miguel Ramírez and Hector Geffner. Probabilistic Plan Recognition Using Off-the-Shelf Classical Planners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1): 1121–1126, July 2010. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v24i1.7745. URL https://ojs.aaai.org/index.php/AAAI/article/view/7745.

[22] Lorenzo Serina, Mattia Chiari, Alfonso Emilio Gerevini, Luca Putelli, and Ivan Serina. Towards efficient online goal recognition through deep learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*, pages 1895–1903. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2025. doi: 10.5555/3709347.3743826. URL https://dl.acm.org/doi/10.5555/3709347.3743826.

[23] Matan Shamir and Reuth Mirsky. GRAML: Goal recognition as metric learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2025)*, pages 8626–8634, 2025. doi: 10.24963/ijcai.2025/959. URL https://www.ijcai.org/proceedings/2025/959.

[24] Shirin Sohrabi, Anton V. Riabov, and Octavian Udrea. Plan recognition as planning revisited. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3258–3264, 2016. URL https://www.ijcai.org/Abstract/16/461.

[25] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[26] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations (ICLR 2021)*. OpenReview.net, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS. arXiv:2011.13456.

[27] Gita Sukthankar, Christopher W. Geib, Hung Hai Bui, David V. Pynadath, and Robert P. Goldman, editors. *Plan, Activity, and Intent Recognition: Theory and Practice*. Morgan Kaufmann, Waltham, MA, USA, 2014. ISBN 978-0-12-398532-3.

[28] Xiaolong Tang, Meina Kan, Shiguang Shan, Zhilong Ji, Jinfeng Bai, and Xilin Chen. HPNet: Dynamic trajectory forecasting with historical prediction attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15261–15270, 2024. doi: 10.1109/CVPR52733.2024.01445.

[29] Franz A. Van-Horenbeke and Angelika Peer. Activity, Plan, and Goal Recognition: A Review. *Frontiers in Robotics and AI*, 8:643010, May 2021. ISSN 2296-9144. doi: 10.3389/frobt.2021.643010. URL https://www.frontiersin.org/articles/10.3389/frobt.2021.643010/full.

[30] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[31] Bozhou Zhang, Nan Song, and Li Zhang. DeMo: Decoupling motion forecasting into directional intentions and dynamic states. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

[32] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17863–17873, 2023.

A Bayes' Rule on Observations

Bayes' rule applied to $p(\boldsymbol{\tau}_i \mid w^j, \mathbf{Z}_{1:k})$ gives

$$p(\boldsymbol{\tau}_i \mid w^j, \mathbf{Z}_{1:k}) \;=\; \frac{p(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau}_i, w^j)\, p(\boldsymbol{\tau}_i \mid w^j)}{p(\mathbf{Z}_{1:k} \mid w^j)}. \tag{10}$$

Taking the logarithm and then the gradient with respect to $\boldsymbol{\tau}_i$ removes the evidence term $\log p(\mathbf{Z}_{1:k} \mid w^j)$, which is constant in $\boldsymbol{\tau}_i$:

$$\nabla_{\boldsymbol{\tau}_i} \log p(\boldsymbol{\tau}_i \mid w^j, \mathbf{Z}_{1:k}) \;=\; \nabla_{\boldsymbol{\tau}_i} \log p(\boldsymbol{\tau}_i \mid w^j) \;+\; \nabla_{\boldsymbol{\tau}_i} \log p(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau}_i, w^j). \tag{11}$$

The measurement model of eq. (1) depends on the trajectory alone, so $p(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau}_i, w^j) = p(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau}_i)$, recovering eq. (6) with $\lambda = 1$. In practice, following classifier guidance [5], we treat λ as a tunable guidance scale: $\lambda > 1$ sharpens the posterior toward observation-consistent trajectories at the cost of prior coverage, while $\lambda < 1$ relaxes it. The Bayesian derivation no longer holds exactly under $\lambda \neq 1$; we report sensitivity in Appendix E.

B Online inference algorithm

Algorithm 1 shows the full WoRD online process.

Algorithm 1 WoRD Online Inference with Warm-Start

Input: Observations $\mathbf{Z}_{1:k}$, previous estimate $\hat{\boldsymbol{\tau}}_{k-1}$ (or $\emptyset$ if $k = 1$), noise predictor $\boldsymbol{\epsilon}_\theta$, guidance scale λ, warm-start step i^*

1: **if** $k = 1$ **then** $\boldsymbol{\tau}_N^j \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for all j, $i_{\mathrm{s}} \leftarrow N$
2: **else** $\boldsymbol{\tau}_{i^*}^j \sim q(\hat{\boldsymbol{\tau}}_{k-1}, i^*)$ for all j, $i_{\mathrm{s}} \leftarrow i^*$ {Eq. 8}
3: **for** $i = i_{\mathrm{s}}$ **downto** 1 **do**
4: **for** each $w^j \in \mathcal{W}$ **do**
5: $\boldsymbol{\epsilon} \leftarrow \boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}_i^j, w^j, i, \mathbf{Z}_{1:k})$ {noise prediction}
6: $\mathbf{g}^j \leftarrow \nabla_{\boldsymbol{\tau}_i^j} \log p(\mathbf{Z}_{1:k} \mid \boldsymbol{\tau}_i^j)$ {Eq. 7, via autograd}
7: $\boldsymbol{\mu}^j \leftarrow \mathrm{DDPMStep}(\boldsymbol{\tau}_i^j, \boldsymbol{\epsilon})$ {Eq. 4}
8: $\boldsymbol{\tau}_{i-1}^j \sim \mathcal{N}(\boldsymbol{\mu}^j + \sigma_i^2\, \mathbf{g}^j,\ \sigma_i^2 \mathbf{I})$
9: **end for**
10: **end for**
11: $p(w^j \mid \mathbf{Z}_{1:k}) \;\propto\; p(w^j) \cdot \exp\!\left(-\sum_{t=1}^{k} \left\| \mathbf{z}_t - h(\boldsymbol{\tau}_0^j(t)) \right\|^2_{\mathbf{R}^{-1}} \right)$ for all j {Eq. 9, normalize over j}
12: $\hat{\boldsymbol{\tau}}_k^j \leftarrow \boldsymbol{\tau}_0^j$ for all j {store per-goal; used as warm-start at step $k{+}1$}

Output: Posterior $\{p(w^j \mid \mathbf{Z}_{1:k})\}_{j=1}^N$ over the candidate set $\mathcal{W}$

The pseudocode maps line-by-line to the equations of section 4. Lines 1 and 2 implement the warm-start branch: at the first observation ($k{=}1$) every candidate's chain is initialized from pure noise and runs through the full N reverse steps, while at every subsequent step the cached estimate $\hat{\boldsymbol{\tau}}_{k-1}^j$ is re-noised to i^* via the closed-form forward marginal of eq. (8). Lines 3 through 10 form the guided reverse chain, repeated independently per candidate w^j: line 5 evaluates the stage-2 noise predictor $\boldsymbol{\epsilon}_\theta$ from section 4.2; line 6 computes the observation-likelihood gradient $\mathbf{g}^j$ of eq. (7) by backpropagation through the measurement model h, exploiting the differentiability assumption of section 3; line 7 forms the standard DDPM mean of eq. (4); and line 8 draws the next iterate from a Gaussian whose mean is shifted by the guidance term $\sigma_i^2 \mathbf{g}^j$, implementing the score modification of eq. (6). Line 11 reweights the candidates via Bayes' rule with a uniform candidate prior and the single-sample marginal-likelihood estimator of eq. (9) (the $S{=}1$ case), evaluating the observation likelihood at the denoised endpoint $\boldsymbol{\tau}_0^j$ and normalizing across j to produce a proper distribution. Line 12 caches this endpoint per candidate so that the next observation can re-noise from a near-data-manifold trajectory rather than from pure noise. The output is the full waypoint posterior $\{p(w^j \mid \mathbf{Z}_{1:k})\}_{j=1}^N$ over $\mathcal{W}$, which the multimodal-belief metrics of section 5 ($\mathrm{CVaR}_{0.1}$, CaW, Brier) consume directly rather than reduced to an argmax.

C Implementation details

Architecture. The score network is a Temporal U-Net [11] operating on trajectory tensors of shape (T, d_{in}) with T=100 and d_{in}=6 input channels per timestep: the diffused position $(\boldsymbol{\tau}_x, \boldsymbol{\tau}_y)$, the observed position $(\mathbf{z}_x, \mathbf{z}_y)$, the observation mask, and the heading θ. The encoder uses three downsampling scales with channel widths $[32, 64, 128]$ and two residual blocks per scale; the decoder mirrors the encoder with skip connections. Each residual block is a 1D convolution (kernel size 3) followed by GroupNorm (8 groups) and a SiLU nonlinearity, with FiLM conditioning [19] injected by a learned linear projection from the concatenation of the waypoint embedding $w^j \in \mathbb{R}^2$ and a sinusoidal diffusion-step embedding of dimension 64. The output head projects back to $(T, 2)$, predicting noise on the (x, y) channels only. The full network has 2.2M parameters.

Hyperparameters. Table 3 lists the hyperparameters used throughout the paper for both training and online inference. Inference values are the canonical defaults; ablations on i^*, λ, S, and the score-network architecture are reported in Appendix E.

Table 3: Implementation hyperparameters used in WoRD experiments.

Component	Hyperparameter	Value
Diffusion	Trajectory horizon T	100
	Number of steps N	100
	β schedule	linear, $\beta \in [10^{-4}, 2{\times}10^{-2}]$
Network	Architecture	Temporal U-Net
	Channels per scale	$[32, 64, 128]$
	Residual blocks per scale	2
	Time embedding dim	64
	Conditioning	FiLM on w^j
	Total parameters	2.2M
Training (both stages)	Optimizer	Adam, lr $= 10^{-4}$
	Batch size	64
	Gradient clip (ℓ_2)	1.0
	Epochs (stage 1 / stage 2)	200 / 100
Stage 2 augmentation	Prefix length sampling	$k = \lfloor \rho T \rfloor,\ \rho \sim \mathcal{U}(0.3,\ 0.9)$
	Observation noise std	0.1
Inference	Warm-start step i^*	20
	Guidance scale λ	1.0
	Samples per candidate S	50
	Posterior temperature τ	1
Evaluation	Training seeds	5
	Episodes per seed (sim.)	50
	Episodes per seed (real-world)	100

Compute. All training and evaluation use a single NVIDIA L40 GPU. Two-stage training completes in under one day on the simulation dataset and in a few hours per recording fold on inD and rounD. Online inference cost is dominated by the U-Net forward pass and scales linearly in the warm-start step i^* and the sample count S.

C.1 Runtime

Table 4 reports the per-update wall-clock of every method, averaged over 10,000 updates, with all methods of a scenario timed on the same GPU. The cost of the classical filters grows with the number of candidates, whereas both diffusion-based methods are dominated by a fixed per-update denoising cost that is insensitive to it.

Iterative denoising is more expensive per update than a Kalman or particle-filter step; WoRD's per-update cost of 6.8 to 12.2 ms is nonetheless of the same order as the 10 ms observation budget

Table 4: Per-update wall-clock in milliseconds, mean over 10,000 updates. All methods within a scenario are timed on the same GPU.

Method	2WP	3WP	4WP
Naive	0.036	0.036	0.038
OWR (UKF)	0.96	1.40	1.89
Particle Filter	0.85	1.26	1.69
GRU	2.70	2.72	2.72
WORD (ours)	6.8	10.3	12.2
Reverse Diffuser	111	169	199



(a) inD location 2 (Bendplatz, pedestrians). W=4 candidate goals.



(b) rounD location 1 (Thiergarten, motor vehicles). W=4 candidate goals.

Figure 3: **Representative locations from the two real-world domains, with DBSCAN-extracted candidate goals overlaid.** Stars mark cluster centroids of agent terminal positions. Aerial backgrounds are recordings rec28 (inD) and rec00 (rounD).

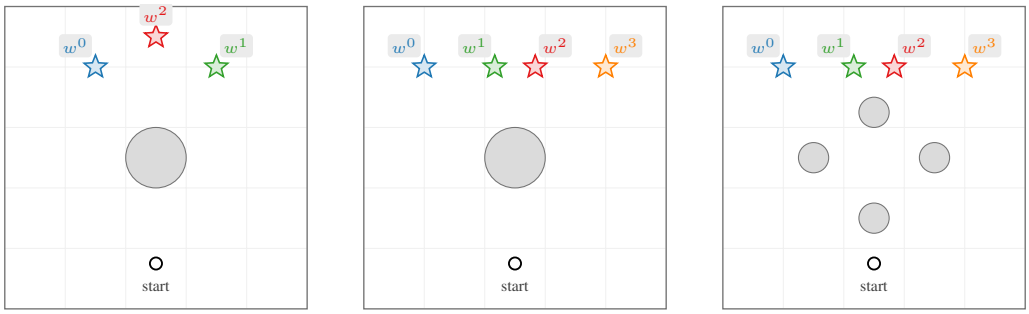
and therefore compatible with online operation. Goal recognition also tolerates a slower update rate than low-level control, since the destination of a pedestrian or vehicle remains stable over seconds and the belief is consumed by a planning or decision layer. The implementation is not optimized for inference speed: it runs in full precision with an uncompiled sampling loop and without step distillation, so table 4 is an upper bound on the cost of WORD.

WORD is roughly $16\times$ faster than the Reverse Diffuser, which re-initializes every denoising chain from noise at each update; warm-starting from the previous estimate removes this cost. In the 2WP ablation setting (table 5), a cold start ($i^* = 100$) is about $5\times$ slower than the default $i^* = 20$ and also less accurate. The cost can be reduced further at a modest loss in accuracy: relative to the default configuration, $S = 25$ samples per candidate cost about $0.6\times$ at 91.2 Top-1 (against 93.1), and the MLP score network costs about $0.7\times$ at 89.2 Top-1.

D Datasets

D.1 Real-World Datasets

Goal extraction. For both real-world domains (inD pedestrians and rounD vehicles), candidate goals are extracted from the terminal positions of agent trajectories using per-agent-group DBSCAN clustering, with ε=3.5 m and MIN_SAMPLES=5. Figure 3 shows one location from each domain with



(a) 3 goals (simple). One central obstacle.

(b) 4 goals (simple). Closely-spaced top-row candidates discriminated by x.

(c) 4 goals (complex). Two sequential center forks plus lateral flanks force compounded commitment decisions.

Figure 4: **Synthetic simulation scenarios for diverse-dataset training and candidate-count scaling.**

the recovered candidate set overlaid on the recording's aerial. Per-domain sampling rates, observation models, and episode counts are summarized below.

D.2 Simulation

Synthetic scenarios. The synthetic environments used for diverse-dataset training and the candidate-count scaling experiments are configurable 10×10 m worlds with a differential-drive agent and static circular obstacles. In each panel of fig. 4, stars denote the candidate goals $\{w^j\}$, the white circle marks the shared start at $(5,\ 1.5)$, and gray discs are the obstacles. The 3WP scenario (fig. 4a) places three top-row goals around a single central obstacle. The simple 4WP scenario (fig. 4b) retains that obstacle but compresses the goal set into a single horizontal row, isolating the effect of candidate count. The complex 4WP variant (fig. 4c) replaces the central obstacle with four scattered discs that induce two sequential lateral forks before the agent reaches the goal row, isolating the effect of obstacle layout. The full diverse training pool covers 50 environments sampled from this configuration space.

E Ablation studies

These ablations probe sensitivity to design choices; the primary evidence for our main claims is the five-domain evaluation in section 5. All runs use the 2WP toy environment: a 10×10 m world with two goals at $(3, 8)$ and $(7, 8)$, a single circular obstacle of radius 1 at $(5, 5)$, and a fixed start at $(5, 1.5)$. The symmetric two-goal layout admits a closed-form analytical posterior, which we use as a ground-truth reference to cross-check implementations before scaling to the full evaluation set. Each cell reports the mean and bootstrap 95% CI over (episode, timestep) pairs from $n = 5$ training seeds. The direction of each effect below transfers to the five primary domains, though the absolute magnitudes do not.

E.1 Sensitivity to the candidate set

The closed-set assumption of section 3 can fail at deployment in two ways: $\mathcal{W}$ may contain irrelevant candidates, or it may omit the agent's true destination. We test both cases on 2WP with the trained 2WP model and without retraining, which is possible because the network is conditioned on waypoint coordinates rather than on a fixed set of class indices (section 4.1). The GRU classifier cannot be evaluated in either case, since its output layer is fixed to the training goals.

Irrelevant candidates. We add 2 or 6 distractor candidates placed away from the paths to the two true goals. As the candidate set grows, the accuracy of WORD falls and its peak confidence falls with it (table 6): six distractors cost 18 points of accuracy and 31 points of peak confidence. This coupling is the property a downstream planner relies on, since a belief that loses accuracy while retaining its

Table 5: Ablation studies on WORD design choices, all runs on the 2WP environment. Default configuration shaded. CIs are bootstrap 95% over (episode, t) pairs from $n = 5$ seeds.

Factor	Value	Top-1 ↑	Brier ↓
Warm-start i^*	0 (no re-noise)	76.2±2.4	0.20±0.02
	10	88.7±1.8	0.10±0.01
	20 (default)	93.1±1.4	0.07±0.01
	100 (cold start)	89.4±1.7	0.09±0.01
Guidance λ	0.0 (prior only)	49.1±2.1	0.32±0.02
	0.5	81.6±2.0	0.13±0.02
	1.0 (default)	93.4±1.4	0.07±0.01
	5.0	58.7±2.3	0.13±0.02
Architecture	Temporal U-Net	93.1±1.4	0.07±0.01
	MLP	89.2±1.9	0.09±0.02
Samples S	10	86.4±2.0	0.12±0.02
	25	91.2±1.6	0.09±0.01
	50 (default)	93.1±1.4	0.07±0.01
	100	93.6±1.5	0.06±0.01

confidence would mislead it. The model-based baselines retain a confidence of 1.0 in this setting, because their hand-specified reference trajectories rule out off-path candidates entirely; the same mechanism makes them confidently wrong when the true goal is absent.

Table 6: Irrelevant distractor candidates added to the 2WP candidate set. Accuracy, CaW, and mean peak confidence of WORD.

Candidate set	Acc ↑	CaW ↓	Max-conf
2 goals (control)	0.93	0.00	0.91
+2 distractors (4)	0.82	0.24	0.63
+6 distractors (8)	0.75	0.30	0.60

True goal removed. For each test episode we remove the true goal, so that every observer receives the remaining goal and two off-path candidates and no candidate explains the observed motion. Since every candidate is wrong, any confidence is misplaced. Table 7 reports the confidence each observer places on its top candidate at the final step, over 200 episodes (100 per removed goal).

Table 7: True goal removed from the candidate set (2WP, 200 episodes). With no correct candidate, lower confidence and higher entropy are better. Values at the final step, averaged over the two removed-goal conditions. Entropy uses the natural logarithm, so its maximum for three candidates is $\ln 3 \approx 1.10$ nats.

Method	Mean max-conf ↓	Episodes with max-conf ≥ 0.9 ↓	Entropy (nats) ↑
WORD (ours)	0.80	3%	0.58
Particle Filter	0.82	38%	0.45
Naive	0.98	95%	0.04
OWR	1.00	100%	0.00

WORD is the least confident observer when the true goal is absent (table 7). OWR renormalizes its near-deterministic likelihoods over the remaining candidates and commits to a wrong goal with full confidence in every episode, whereas WORD weights each candidate by how well the learned prior explains the observed motion, so that no candidate receives confident mass when none fits. The same weighting explains the distractor result: implausible candidates are down-weighted without receiving zero mass, so adding them dilutes the confidence of WORD.

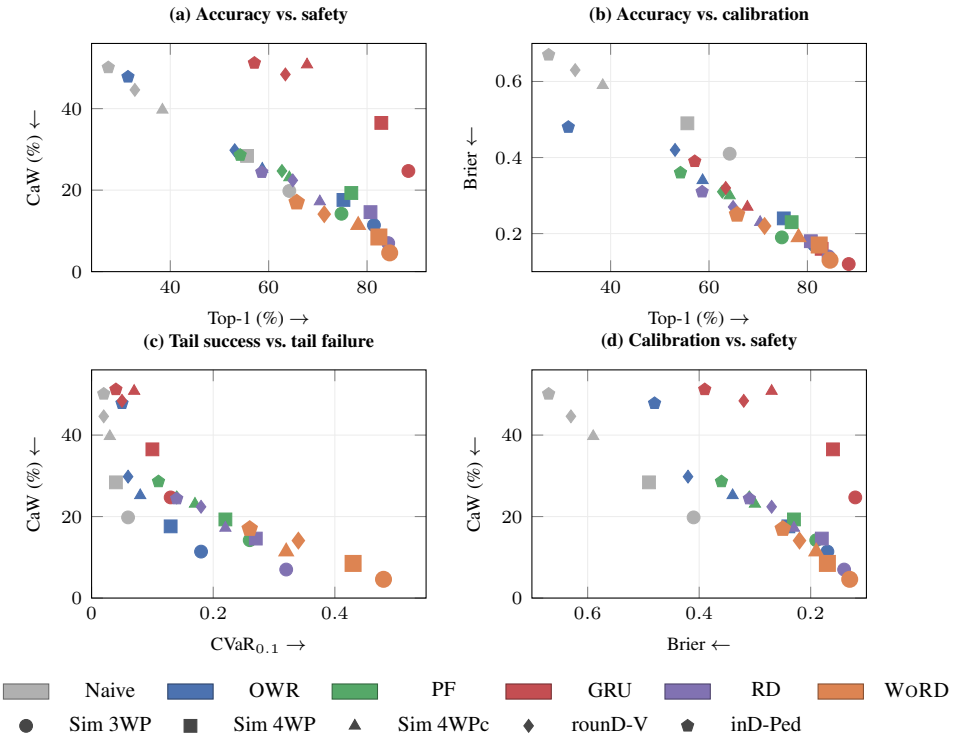


Figure 5: **Four metric trade-offs across the five evaluation domains.** Each panel scatters one point per (method, domain) for the six methods over the five domains. Colour identifies the method; marker shape identifies the domain (legend below). WoRD points carry a thicker outline. The desirable corner is bottom-right in every panel; panel (d) reverses the x-axis so lower Brier still points right.

F Extended experimental results

F.1 Pareto trade-offs across metrics

Figure 5 reads the per-cell rankings of Tables 1 and 2 as four metric trade-offs, with one point per (method, domain) on each panel. Color identifies the method; marker shape identifies the domain (legend below the figure). The four panels separate the two failure modes the metrics target: panels (a) and (b) ask whether a method converges and is calibrated when it does; panels (c) and (d) ask whether it preserves multimodal belief through ambiguous prefixes. Methods that win one pair but not the other, most visibly GRU on (b) but not (c), are the reason we report four metrics rather than Top-1 alone.

Panel (a), Top-1 vs. CaW (accuracy vs. safety): WoRD reaches the desirable corner on every domain, and the orange cluster sits well below the GRU red cluster on the three complex domains (triangle, diamond, pentagon). **Panel (b), Top-1 vs. Brier (accuracy vs. calibration):** on the non-complex sim domains (circle, square) GRU and WoRD are statistical ties; on the three complex domains WoRD dominates by a Brier margin that grows with prefix ambiguity. **Panel (c), CVaR$_{0.1}$ vs. CaW (tail success vs. tail failure):** the most direct test of multimodal belief preservation. WoRD sits uniformly to the right of every other method, and the gap is largest on rounD-V (diamond), where the shared circular arc keeps the optimal posterior multimodal for most of the trajectory. **Panel (d), Brier vs. CaW with reversed x-axis:** the bottom-right corner combines low calibration error with low unimodal collapse; the GRU cluster (red) sits high on CaW despite competitive Brier, the signature of cross-entropy overconfidence. Per-cell numerical values are in Tables 1 and 2.

Table 8: Top-1 accuracy decomposed by recording location for the two real-world domains. **Left:** inD pedestrians, four signalized-intersection locations. **Right:** rounD vehicle traffic, three roundabout locations.

inD pedestrians: Top-1 (%) ↑

Method	Loc 1	Loc 2	Loc 3	Loc 4	Avg
Naive	31.2	28.5	25.4	24.5	27.4
OWR	35.8	32.9	28.8	28.1	31.4
PF	58.1	55.7	51.8	51.2	54.2
GRU	61.4	58.8	54.5	53.7	57.1
RD	63.2	60.5	56.0	54.7	58.6
WoRD	**70.4**	**67.5**	**63.0**	**61.8**	**65.7**

rounD: Top-1 (%) ↑

Method	Loc 0	Loc 1	Loc 2	Avg
Naive	29.0	35.6	33.8	32.8
OWR	49.0	56.7	53.6	53.1
PF	58.8	66.4	62.9	62.7
GRU	59.5	67.1	63.6	63.4
RD	60.9	68.7	65.1	64.9
WoRD	**67.4**	**74.8**	**71.7**	**71.3**

F.2 Per-scene inD/rounD breakdown

Table 8 decomposes the headline real-world Top-1 numbers by recording location: inD pedestrians at four signalized-intersection locations (Loc 1–4 in standard inD nomenclature, left half) and rounD vehicle traffic at three roundabout locations (Loc 0–2, right half). The rightmost "Avg" column reproduces the headline Top-1 from Table 2; per-location standard deviations across the 5 training seeds range from ± 2.4 to ± 4.2 for WoRD and from ± 3.1 to ± 5.3 for the baselines. The method ordering, WoRD $>$ RD $>$ GRU $>$ PF $>$ OWR $>$ Naive, is preserved at every location, so the aggregate ranking is not driven by a single easy or hard scene. Rank-preserving variation across locations spans 6–10 Top-1 points within each dataset and reflects geometry density: rounD Loc 0 (the densest roundabout) and inD Loc 3–4 (the largest multi-arm intersections) are the hardest scenes for every method.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Abstract and section 1 state the four contributions (diffusion-inference formulation, warm-start algorithm, analytic observation guidance, and cross-domain evaluation), which map to sections 4 and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses three limitations: closed candidate set, known and differentiable measurement model, and per-update cost above a Kalman filter.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper presents a method, not theoretical results. The score decomposition (eq. (6)) is a Bayes-rule identity; its derivation and the assumption it relies on (observation depends on τ alone, eq. (1)) are given in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sections 4 and 5.1 specify the architecture, two-stage training, datasets, splits, baselines, and metrics, with full hyperparameters in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary material includes the anonymized source code, training and evaluation scripts, and configs needed to reproduce the main experimental results in Tables 1 and 2. The experiments rely on the publicly available inD [2] and rounD [15] datasets and the synthetic generator described in section 5.1 and Appendix C.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training objectives, two-stage procedure, and architecture are in sections 4 and 4.2; optimizer, learning rate, batch size, and the data-augmentation schedule are in Appendix C. Splits and the bootstrap protocol are described in section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main results report mean values with 95% bootstrap confidence intervals over episodes (section 5.1). Variability captured: random seeds across model training and per-episode noise; both contribute to the bootstrap distribution.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are reported in Appendix C: per-experiment wall-clock and total project compute are summarized in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses publicly released, pre-anonymized traffic datasets (inD, rounD) and synthetic trajectories; no human subjects, no deployment, and no decisions about identifiable individuals. Submission is anonymized.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Broader impacts are discussed in the Broader Impacts paragraph of section 6: positive (safer autonomous planning under calibrated uncertainty) and negative (passive surveillance, adversarial planning against humans), with mitigation hooks (consent frameworks for training data, oversight on downstream planners).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper releases no high-risk model: WoRD outputs a posterior over a known finite waypoint set, not generated content. Datasets used (inD, rounD) are publicly available under LevelXData's research license; no scraping or new data collection.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Datasets (inD [2], rounD [15]) are used under the non-commercial research license bundled with each dataset, with creators cited and license noted in section 5.1. Method-relevant prior work (DDPMs, classifier guidance, Diffuser, MotionDiffuser, FiLM) is cited at point of use in sections 2 and 4.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: No new assets are released with this submission; code, model weights, and goal-extraction annotations are planned for the camera-ready version (see Q5).

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: No crowdsourcing or human-subjects research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human-subjects research; IRB approval not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the method. LLM use was limited to writing assistance, which the policy exempts from declaration.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.