RiskNav: Safety-Readable Latent World Models for Crowd Navigation

Anonymous authors
Paper under double-blind review

Abstract

Safe crowd navigation requires anticipating how candidate robot actions affect future human–robot interactions under partial observation. Reactive methods do not explicitly evaluate these futures, whereas predictive planners that optimize multi-agent trajectories online can become costly as crowds grow. Latent world models offer compact prediction, but their representations alone do not provide explicit safety quantities for risk-aware action selection. We present RiskNav, a risk-supervised, action-conditioned latent world model for crowd navigation. For each candidate action sequence, RiskNav rolls out future latent representations conditioned on the observation history and local waypoint. This latent rollout avoids explicitly predicting individual pedestrian trajectories or optimizing multi-agent trajectories online. A risk evaluator predicts static and dynamic safety margins and collision probabilities from each representation, with gradients from the resulting risk loss propagating through the predictor into the encoder. The planner incorporates these predicted safety quantities into a horizon-wise risk term for receding-horizon planning. Across indoor, open, and out-of-distribution settings, RiskNav achieves the highest strict success rates and the fewest robot-responsible contacts among nine baselines, with strict success rates of 0.76, 0.57, and 0.50, respectively. Replanning stays near 42 ms as the crowd grows from 1 to 21 pedestrians. Ablations show that RiskNav's advantage comes from coupling safety-supervised latent prediction with risk-aware action selection, as strict success falls from 0.750 to 0.350 without representation shaping and to 0.125 without the planning-time risk term. RiskNav makes action-conditioned future risk directly usable for planning, improving both safety and success in crowd navigation.

1 Introduction

When a robot navigates through a moving crowd, a currently open gap may close before the robot reaches it. Each candidate maneuver, such as continuing forward, slowing down to wait, or detouring laterally, produces different future separations from nearby pedestrians. Choosing among these alternatives from partial observations and within the time budget of each control cycle is a central challenge in crowd navigation (Mavrogiannis et al., 2023; Scofano et al., 2025; Singamaneni et al., 2024). From a planning perspective, the robot must predict the future interactions induced by each candidate action sequence and select one that makes progress while minimizing predicted risk.

Both hand-designed local methods (Helbing & Molnár, 1995; Fox et al., 1997; van den Berg et al., 2011) and learned navigation policies (Chen et al., 2017b; 2019; Liu et al., 2021; 2023; Xie & Dames, 2023; Liu et al., 2026) operate at low online cost but generally do not explicitly roll out and compare the multi-step safety consequences of alternative robot action sequences. A common predict-then-plan pipeline uses forecasting models that capture pedestrian–pedestrian interactions but do not condition their predictions on each candidate robot action sequence (Alahi et al., 2016; Gupta et al., 2018; Poddar et al., 2023). Such forecasts can support collision checking against candidate robot trajectories, but they cannot represent how pedestrian motion may change in response to those candidates. Interaction-aware approaches address this limitation in different ways. Interacting Gaussian Processes model a joint distribution over cooperative human–robot trajectories (Trautman et al., 2015), Trajectron++ can condition pedestrian forecasts on future ego-agent plans (Salzmann et al., 2020), and SICNav jointly optimizes robot and pedestrian motion online (Samavi et al., 2025a;b). Although these methods account for action-dependent interaction, they represent future interactions through

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

explicit agent-wise trajectories, whether predicted probabilistically or optimized online. This agent-wise representation grows with both crowd size and prediction horizon, increasing the online burden of repeatedly evaluating candidate actions. This leaves a central gap: the planner needs a compact, action-conditioned representation of future robot–pedestrian interactions that supports direct risk evaluation without explicitly rolling out and optimizing individual trajectories online.

Latent world models encode observations into compact latent states and learn action-conditioned dynamics over those states (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020). TD-MPC, TD-MPC2, and DINO-WM use learned latent dynamics to optimize action sequences at decision time (Hansen et al., 2022; 2024; Zhou et al., 2025). NavRep and NavDreams instead use world-model representations to train reinforcement-learning navigation policies (Dugas et al., 2021; 2022), while NavThinker incorporates action-conditioned world-model predictions into on-policy reinforcement learning (Hu et al., 2026). A joint-embedding predictive architecture (JEPA) predicts target representations from context representations rather than reconstructing targets in observation space (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024). However, low latent prediction error does not guarantee that futures on opposite sides of a safety boundary are distinguishable in latent space. Even if this distinction is preserved, explicit risk-aware planning still requires a mapping from the latent rollout to safety quantities that can score candidate action sequences. A more comprehensive discussion of related work is provided in Appendix A.

Taken together, these limitations motivate a central question: how can a robot compactly predict the future robot–crowd interaction under each candidate action sequence while retaining the information needed to assess the resulting risk in real time? We address this question with RiskNav, a crowd-navigation framework built around a risk-supervised, action-conditioned latent world model. For each candidate action sequence, RiskNav rolls out the corresponding future robot–crowd interaction in a fixed-dimensional latent space. This fixed-dimensional rollout avoids explicitly predicting individual pedestrian trajectories or optimizing multi-agent trajectories online, ensuring that the representation evaluated by the planner does not scale with the number of pedestrians. At each rollout step, a jointly trained risk evaluator maps the predicted representation to static and dynamic safety margins and their corresponding collision probabilities. The safety loss also updates the predictor and encoder, encouraging the latent dynamics to retain the risk-relevant information needed for safe planning. The planner aggregates these quantities over the prediction horizon and combines the resulting risk term with the navigation objective for receding-horizon planning. Across indoor, open, and out-of-distribution crowd-flow evaluations, RiskNav achieves the highest strict success rates and the fewest robot-responsible contacts among nine baselines. In a separate scalability test, replanning latency remains near 42 ms as the crowd grows from 1 to 21 pedestrians. Ablations show that RiskNav's performance gains rely on safety supervision shaping the latent representation and on the resulting risk predictions guiding the planner, with strict success dropping sharply when either mechanism is removed. Without real-world fine-tuning, RiskNav reaches the goal in all ten repeated trials on the same route using a physical humanoid, providing initial evidence of zero-shot transfer to real pedestrian interactions. Our contributions are threefold:

- We introduce RiskNav, an action-conditioned latent world model that rolls out fixed-dimensional representations of future crowd–robot interactions for candidate robot action sequences without reconstructing future observations or predicting individual pedestrian trajectories.
- We make these rollouts safety-readable by using per-step safety losses to shape representation learning and the resulting safety-margin and collision-probability predictions to guide receding-horizon action selection.
- We demonstrate the highest strict success rates and the fewest robot-responsible contacts across indoor, open, and out-of-distribution evaluations, nearly constant replanning latency as crowds grow, and zero-shot deployment on a humanoid robot.

2 Method

At the core of RiskNav is a safety-readable, action-conditioned latent world model for crowd navigation. Given the recent observation history, a local waypoint, and a candidate action sequence, the world model predicts the corresponding multi-step future in latent space. From each predicted representation, the risk evaluator estimates the static and dynamic safety margins and the corresponding predicted collision probabilities, which enter the planning objective as stage-wise risk penalties

over the prediction horizon. Offline, the encoder, predictor, and risk evaluator are trained jointly on navigation trajectories (Figure 1(a)), with future observations providing the latent prediction targets and simulator ground-truth states providing the risk and navigation labels. Gradients from the risk and navigation losses propagate through the predicted representations into the predictor and encoder, thereby encouraging the latent space to retain planning-relevant safety information. During online planning, all learned parameters remain fixed (Figure 1(b)). At each control cycle, a receding-horizon optimizer evaluates candidate action sequences using the predicted navigation and risk quantities, executes the first action of the lowest-cost sequence, and replans upon receiving the next observation.

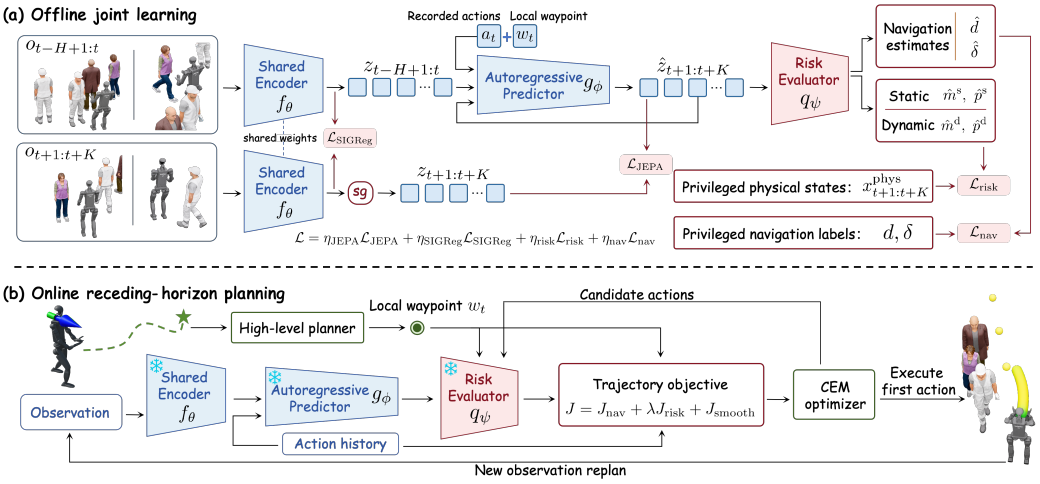


Figure 1: Overview of RiskNav. (a) A shared encoder maps observations to latents. The predictor rolls out future states conditioned on actions and local waypoints, and the risk evaluator extracts safety and navigation metrics. (b) With modules frozen, CEM samples and evaluates candidate action sequences, then executes the first action of the lowest-cost sequence.

2.1 Action-Conditioned Latent Prediction for Crowd Navigation

Different candidate robot-action sequences can lead to different future robot–pedestrian relative configurations and therefore different safety outcomes. The planner must therefore evaluate the action-dependent future risks associated with different candidate sequences. RiskNav models this action dependence by predicting the corresponding future observation representations in latent space for each candidate action sequence.

Action-conditioned latent rollout. Let $\mathbf{o}_i$ denote the local bird's-eye-view (BEV) observation supplied to the encoder at time step i, which is constructed from the observed static geometry and pedestrian states. Let $\mathbf{a}_i$ denote the robot action issued after observing $\mathbf{o}_i$, which is applied until the next observation $\mathbf{o}_{i+1}$ is obtained. Let $\mathbf{w}_t$ denote the local waypoint supplied by the global path planner at planning time t, which is treated as a fixed task-conditioning command within the ensuing K-step rollout. Let H and K denote the observation-history length and rollout horizon, respectively. At time t, let $\mathbf{U}_t = (\mathbf{a}_t, \ldots, \mathbf{a}_{t+K-1})$ denote a candidate action sequence. Within this rollout, $\mathbf{a}_i$ denotes an executed action for $i < t$ and a candidate action from $\mathbf{U}_t$ for $i \geq t$.

The encoder $f_{\boldsymbol{\theta}}$ maps each observed frame to a latent representation $\mathbf{z}_i = f_{\boldsymbol{\theta}}(\mathbf{o}_i)$. The action and the waypoint are encoded by two lightweight encoders, E_{a} and E_{w}, respectively, and combined into

$$\mathbf{c}_i = E_{\text{a}}(\mathbf{a}_i) + E_{\text{w}}(\mathbf{w}_t), \quad i = t - H + 1, \ldots, t + K - 1.$$

The current waypoint $\mathbf{w}_t$ is shared across the history and rollout as a common task condition. Following prior action-conditioned JEPA designs (Assran et al., 2025; Rao et al., 2026), the predictor $g_{\boldsymbol{\phi}}$ is implemented as a causal Transformer that takes the latest H latent representations as input tokens and outputs the latent representation of the next step, with the conditioning sequence injected into each Transformer layer through adaptive layer normalization, yielding:

$$\hat{\mathbf{z}}_{t+1} = g_{\boldsymbol{\phi}}\big(\mathbf{z}_{t-H+1:t};\ \mathbf{c}_{t-H+1:t}\big). \tag{1}$$

Applying Eq. (1) recursively, which involves feeding each prediction back into the latent window and using the next action in $\mathbf{U}_t$, yields the K-step latent rollout $\hat{\mathbf{z}}_{t+1:t+K}$.

Latent prediction and regularization. During training, $\mathbf{U}_t$ is taken from the recorded action sequence in each offline trajectory and used to generate the autoregressive latent rollout $\hat{\mathbf{z}}_{t+1:t+K}$. For each $k \in \{1, \dots, K\}$, the corresponding future observation $\mathbf{o}_{t+k}$ is encoded by the shared encoder to form the target representation $\mathbf{z}_{t+k} = f_{\boldsymbol{\theta}}(\mathbf{o}_{t+k})$. The stop-gradient operator $\mathrm{sg}[\cdot]$ is applied only to the target branch when computing the multi-step latent prediction loss:

$$\mathcal{L}_{\text{JEPA}} = \sum_{k=1}^{K} \left\| \hat{\mathbf{z}}_{t+k} - \mathrm{sg}\left[f_{\boldsymbol{\theta}}(\mathbf{o}_{t+k}) \right] \right\|_2^2. \tag{2}$$

Thus, $\mathrm{sg}[\cdot]$ blocks gradients through the future target encoding, while gradients from $\mathcal{L}_{\text{JEPA}}$ can still reach $f_{\boldsymbol{\theta}}$ through the context representations used by $g_{\boldsymbol{\phi}}$. To prevent representation collapse, we apply Sketched Isotropic Gaussian Regularization (Balestriero & LeCun, 2025) to the non-detached encoder representations, following Maes et al. (2026). This regularizer encourages the empirical distribution of the encoder representations to approach the standard isotropic Gaussian $\mathcal{N}(0, I)$. Predicting future representations avoids the need to reconstruct future observations or explicitly predict pedestrian trajectories, and the risk and navigation objectives introduced next further encourage the rollout representations to retain information relevant to safe planning.

2.2 Safety-Supervised Latent Representation Learning

The action-conditioned rollout predicts a latent future for each candidate action sequence. To compare these futures during planning, RiskNav maps each predicted representation to explicit safety and navigation quantities through a jointly trained readout module $q_{\boldsymbol{\psi}}$:

$$\left(\hat{m}^{\mathrm{s}}_{t+k},\ \hat{p}^{\mathrm{s}}_{t+k},\ \hat{m}^{\mathrm{d}}_{t+k},\ \hat{p}^{\mathrm{d}}_{t+k}, \hat{d}_{t+k}, \hat{\delta}_{t+k}\right) = q_{\boldsymbol{\psi}}(\hat{\mathbf{z}}_{t+k}), \quad k = 1, \dots, K \tag{3}$$

where $\hat{m}^{\mathrm{s}}_{t+k}$ and $\hat{m}^{\mathrm{d}}_{t+k}$ denote the predicted safety margins with respect to static scene geometry and pedestrians, respectively. A negative margin indicates that the corresponding safety buffer has been crossed. The quantities $\hat{p}^{\mathrm{s}}_{t+k}$ and $\hat{p}^{\mathrm{d}}_{t+k}$ denote the predicted probabilities of collision with static scene geometry and pedestrians, respectively. The remaining outputs, $\hat{d}_{t+k}$ and $\hat{\delta}_{t+k}$, denote the predicted distance to the local waypoint and the predicted single-step goal progress, respectively.

During training, labels for the safety margins, collision events, waypoint distance, and single-step progress are computed from simulator ground-truth states. The risk and navigation losses are

$$\begin{aligned} \mathcal{L}_{\text{risk}} &= \sum_{k=1}^{K} \sum_{b \in \{\mathrm{s},\mathrm{d}\}} \left[\ell_{\text{reg}}\left(\hat{m}^{b}_{t+k}, m^{b}_{t+k}\right) + \ell_{\text{cls}}\left(\hat{p}^{b}_{t+k}, c^{b}_{t+k}\right) \right], \\ \mathcal{L}_{\text{nav}} &= \sum_{k=1}^{K} \left[\ell_{\text{reg}}\left(\hat{d}_{t+k}, d_{t+k}\right) + \ell_{\text{reg}}\left(\hat{\delta}_{t+k}, \delta_{t+k}\right) \right] \end{aligned} \tag{4}$$

where m^b, c^b, d, and δ are the corresponding ground-truth labels, with $c^b \in \{0, 1\}$, and ℓ_{reg} and ℓ_{cls} denote the regression loss and binary classification loss, respectively.

Joint training objective. Combining the risk and navigation losses with the latent prediction and regularization losses introduced in Section 2.1 gives

$$\mathcal{L} = \eta_{\text{JEPA}}\mathcal{L}_{\text{JEPA}} + \eta_{\text{SIGReg}}\mathcal{L}_{\text{SIGReg}} + \eta_{\text{risk}}\mathcal{L}_{\text{risk}} + \eta_{\text{nav}}\mathcal{L}_{\text{nav}} \tag{5}$$

where $\eta_{\text{JEPA}}, \eta_{\text{SIGReg}}, \eta_{\text{risk}}, \eta_{\text{nav}} \geq 0$ control the relative contributions of the four loss terms. The gradients of $\mathcal{L}_{\text{risk}}$ and $\mathcal{L}_{\text{nav}}$ are backpropagated through $q_{\boldsymbol{\psi}}$ and $g_{\boldsymbol{\phi}}$ to $f_{\boldsymbol{\theta}}$. Thus, these losses not only train $q_{\boldsymbol{\psi}}$, but also encourage the learned latent representations to retain information relevant to safety assessment and navigation.

2.3 Risk-Aware Receding-Horizon Planning

During online planning, all learned modules are held fixed (Figure 1(b)). To incorporate the latest observation into each control decision, RiskNav adopts a receding-horizon MPC scheme (Williams

et al., 2017; Hansen et al., 2022). At each control cycle, the planner evaluates candidate action sequences of length K, executes the first action of the selected sequence, and repeats the optimization after receiving the next observation:

$$\begin{aligned} \mathbf{U}_t^* &= \arg\min_{\mathbf{U}_t \in \mathcal{U}^K} J(\mathbf{U}_t), \\ J(\mathbf{U}_t) &= J_{\text{nav}}(\mathbf{U}_t) + \lambda J_{\text{risk}}(\mathbf{U}_t) + J_{\text{smooth}}(\mathbf{U}_t) \end{aligned} \tag{6}$$

where $\mathcal{U}$ denotes the admissible action set, while λ controls the contribution of the risk cost during planning. For each candidate sequence $\mathbf{U}_t$, J_{nav} combines an analytic waypoint-tracking term, which is obtained by integrating the candidate sequence through the known robot kinematics, with the waypoint-distance and goal-progress predictions obtained through Eqs. (1) and (3). Meanwhile, J_{risk} accumulates the corresponding safety predictions over the planning horizon, and J_{smooth} penalizes changes between consecutive actions. Specifically, the components in Eq. (6) are defined as:

$$\begin{aligned} J_{\text{nav}}(\mathbf{U}_t) &= \lambda_{\text{trk}} \left\| \hat{\mathbf{x}}_{t+K}(\mathbf{U}_t) - \mathbf{w}_t \right\|_2 + \lambda_{\text{d}}\, \hat{d}_{t+K} - \lambda_\delta \sum_{k=1}^{K} \hat{\delta}_{t+k}, \\ J_{\text{risk}}(\mathbf{U}_t) &= \sum_{k=1}^{K} \gamma^{k-1} \sum_{b \in \{\text{s},\text{d}\}} \left[\text{sp}_\beta \left(\tau^b - \hat{m}_{t+k}^b \right) + \hat{p}_{t+k}^b \right], \\ J_{\text{smooth}}(\mathbf{U}_t) &= \lambda_{\text{u}} \sum_{i=t-1}^{t+K-2} \left\| \mathbf{a}_{i+1} - \mathbf{a}_i \right\|_2^2 \end{aligned} \tag{7}$$

where $\hat{\mathbf{x}}_{t+K}(\mathbf{U}_t)$ is the robot position after K steps obtained by integrating $\mathbf{U}_t$ through the kinematic model, τ^b is the safety-margin threshold, $\text{sp}_\beta(x) = \log(1+\exp(x/\beta))$ is the softplus margin penalty with $\beta = 0.2$ m, γ is the temporal discount factor, $\mathbf{a}_{t-1}$ is the currently executed action, and λ_{trk}, λ_{d}, λ_δ, $\lambda_{\text{u}} \geq 0$ are weighting coefficients. By accumulating risk at every rollout step, J_{risk} distinguishes candidate sequences with similar terminal navigation outcomes but different intermediate safety profiles. Following the planning strategy based on the cross-entropy method (CEM) used in DINO-WM (Zhou et al., 2025), we approximately solve Eq. (6) using CEM (Rubinstein, 1999; Pinneri et al., 2021). At each CEM iteration, candidate action sequences are sampled from the current distribution and evaluated using J. A fixed fraction of the lowest-cost sequences is selected as the elite set, and the sampling distribution is refitted to these elites. After a fixed number of CEM iterations, the planner returns the lowest-cost sequence from the final iteration. In summary, Algorithm 1 details the complete closed-loop navigation framework.

Algorithm 1 RiskNav: Offline Training and Online Receding-Horizon Planning

Input: Trajectory dataset $\mathcal{D}$; history length H; rollout horizon K; number of training steps T_{train}; admissible action set $\mathcal{U}$; CEM sample size M, iterations N, elite fraction ρ.
Initialize: Trainable modules $f_{\boldsymbol{\theta}}$, $g_{\boldsymbol{\phi}}$, $q_{\boldsymbol{\psi}}$.
▷ **Phase 1: Offline Training**
for $n = 1, \ldots, T_{\text{train}}$ **do**
  Sample a training window; roll out by Eq. (1) and read out by Eq. (3)
  Compute Eqs. (2), (4), (5); update $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}$
end for
▷ **Phase 2: Online Planning** ($f_{\boldsymbol{\theta}}, g_{\boldsymbol{\phi}}, q_{\boldsymbol{\psi}}$ frozen)
for each control cycle t **do**
  Encode the latest H observations, executed-action history, and current waypoint $\mathbf{w}_t$;
  Warm-start (μ, σ)
  for $j = 1, \ldots, N$ **do**
    Sample $\{\mathbf{U}_t^{(m)}\}_{m=1}^{M} \sim \mathcal{N}(\mu, \sigma^2)$ in $\mathcal{U}^K$; Roll out by Eq. (1) and read out by Eq. (3);
    Evaluate J by Eqs. (6), (7); record $\widehat{\mathbf{U}}_t^{(j)}$ and refit (μ, σ) using the lowest-cost ρ fraction
  end for
  Set $\mathbf{U}_t^* \leftarrow \widehat{\mathbf{U}}_t^{(N)}$ and execute its first action
end for

3 Experiments

Our simulation experiments evaluate two central claims: (i) RiskNav improves navigation performance in crowded environments, and (ii) safety supervision makes the action-conditioned latent

rollouts informative about future risk and useful for planning. To address the first claim, Section 3.2 evaluates our approach across an indoor crowd-flow scenario with static obstacles and an open-field scenario without static obstacles. The second claim is jointly investigated in Sections 3.3 and 3.4. Specifically, Section 3.3 assesses future safety-margin prediction and dynamic-collision discrimination from the predicted latent representations, while Section 3.4 analyzes how safety supervision structures the latent space and separately ablates representation shaping and the use of the risk term in the planning objective. Finally, Section 3.5 examines the deployment of the simulation-trained model on a humanoid robot navigating among pedestrians in a real-world environment.

3.1 Evaluation Setup

Environments. We evaluate RiskNav in a continuous two-dimensional crowd-navigation simulator across two layouts: an indoor environment containing static walls and an open field without static obstacles. In both layouts, the robot traverses a fixed multi-segment route by tracking local waypoints supplied by a global planner. The open-field layout includes head-on, co-flowing, and mixed pedestrian-flow configurations. Relative to the direction of the corresponding route segment, pedestrians move in the opposite direction in head-on flow, in the same direction in co-flowing scenarios, and in both directions in mixed flow. Each flow configuration is evaluated across multiple crowd densities, with the main results averaged across all such configurations and densities.

Pedestrian dynamics. The training set contains 34,500 episodes spanning mixed maps, seven pedestrian policies, five robot data-collection policies, and crowds of up to 12 pedestrians. The pedestrian-policy mixture includes robot-responsive behaviors but does not include the invisible-robot ORCA setting used for benchmark evaluation. Evaluation with more than 12 pedestrians therefore also tests extrapolation beyond the training crowd size. For benchmark evaluation, we follow CrowdNav's invisible-robot protocol (Chen et al., 2019) modeling pedestrian–pedestrian avoidance with ORCA (van den Berg et al., 2011) while excluding the robot from pedestrian motion updates. Candidate robot actions nevertheless produce different future robot–pedestrian relative configurations and safety outcomes through the robot's own motion. Robot-aware ORCA results are reported in Appendix D.1. We additionally replace ORCA with Social Force Model (SFM) (Helbing & Molnár, 1995) while retaining the invisible-robot protocol to evaluate transfer across pedestrian dynamics.

Metrics. Drawing on standard navigation evaluation practices (Anderson et al., 2018) and safety guidelines for social navigation (Francis et al., 2025), we report two primary metrics to jointly assess task completion and interaction safety: strict success rate (Succ) and the number of robot-responsible contacts per episode (RC). Specifically, Succ is the fraction of episodes in which the robot reaches the goal without any robot-responsible contacts. For RC, a contact is detected when the surface-to-surface distance between the robot and a pedestrian is ≤ 0.05 m, and responsibility is attributed based on each agent's contribution to reducing their separation. Table 2 also reports Reach (the fraction of episodes where the robot reaches the goal regardless of contacts) and Freeze (the fraction of time steps where its speed falls below $0.05\,\mathrm{m/s}$). Across all simulation tables, the $\pm$ values denote 95% paired-bootstrap confidence intervals. Due to space constraints, the main text focuses on Succ and RC as the two most representative metrics, while additional evaluation criteria, episode counts, and latency settings are detailed in Appendix C.

3.2 Navigation Performance and OOD Generalization

Averaged over all crowd-size levels and flow directions, RiskNav achieves the highest Succ and lowest RC across all three evaluation settings: indoor, open, and OOD (Table 1). Specifically, in the indoor scenario, RiskNav yields a Succ of 0.76 and an RC of 0.31, outperforming the strongest baseline, SARL (Succ: 0.64, RC: 0.38). Similarly, in the open scenario, our method surpasses the top-performing baseline, SICNav (Succ: 0.38, RC: 1.65), by achieving a Succ of 0.57 and an RC of 1.15. These improvements translate to absolute success-rate gains of 12 and 19 percentage points, and reductions of 0.07 and 0.50 in contacts per episode for indoor and open scenarios, respectively. Notably, in the densest indoor level (about 19 pedestrians with head-on traffic in a narrow corridor), SARL fails due to timeout in all 40 episodes, whereas RiskNav successfully completes 21 of the 40 episodes (Succ = 0.525). When subjected to the OOD shift in pedestrian dynamics (from ORCA to SFM), RiskNav's Succ decreases from 0.57 to 0.50, and its RC increases from 1.15 to 1.70. Nev-

Table 1: Comparison with nine baselines across indoor, open, and OOD scenarios, averaged over crowd densities and flow directions. Baselines are grouped into analytical planners (top), retrained RL methods (middle), and zero-shot transfers using official weights (bottom). CrowdNav++ is omitted in indoor scenarios due to its lack of static wall observations.

Method	Indoor crowd flow		Open crowd flow		OOD	
	Succ↑	RC↓	Succ↑	RC↓	Succ↑	RC↓
DWA (Fox et al., 1997)	0.59 ±0.06	0.79 ±0.19	0.18 ±0.03	8.25 ±0.89	0.15 ±0.03	8.73 ±0.83
ORCA (van den Berg et al., 2011)	0.45 ±0.06	0.81 ±0.14	0.14 ±0.03	6.49 ±0.69	0.10 ±0.02	7.50 ±0.68
SICNav (Samavi et al., 2025a)	0.42 ±0.06	0.88 ±0.13	0.38 ±0.04	1.65 ±0.21	0.27 ±0.04	2.31 ±0.18
SARL (Chen et al., 2019)	0.64 ±0.05	0.38 ±0.11	0.10 ±0.02	5.98 ±0.60	0.07 ±0.02	6.74 ±0.70
DS-RNN (Liu et al., 2021)	0.63 ±0.06	1.03 ±0.35	0.00 ±0.00	16.43 ±2.50	0.00 ±0.00	12.62 ±1.26
NavThinker (Hu et al., 2026)	0.53 ±0.06	0.78 ±0.12	0.01 ±0.01	12.30 ±1.02	0.00 ±0.01	10.04 ±0.98
CrowdNav++ (Liu et al., 2023)	–	–	0.19 ±0.03	3.79 ±0.29	0.14 ±0.03	5.47 ±0.33
DRL-VO (Xie & Dames, 2023)	0.40 ±0.06	0.80 ±0.10	0.25 ±0.03	3.32 ±0.32	0.27 ±0.03	2.11 ±0.14
HEIGHT (Liu et al., 2026)	0.38 ±0.05	1.53 ±0.26	0.16 ±0.02	4.24 ±0.26	0.16 ±0.03	2.61 ±0.19
RiskNav	**0.76** ±0.05	**0.31** ±0.08	**0.57** ±0.03	**1.15** ±0.13	**0.50** ±0.03	**1.70** ±0.12

ertheless, its OOD Succ of 0.50 substantially exceeds the second-best performance (0.27, shared by SICNav and DRL-VO) by 23 percentage points, while its RC of 1.70 is 0.41 lower than the next-best value recorded by DRL-VO (2.11). Figure 2 complements these quantitative results with a qualitative episode from the open scenario. As shown in panel (b), the selected rollout shifts laterally as the predicted collision probability increases near an oncoming pedestrian. Furthermore, in panel (d), the selected rollout changes when a pedestrian approaching from behind enters the prediction horizon. This episode clearly illustrates how the predicted interaction risk guides proactive replanning.

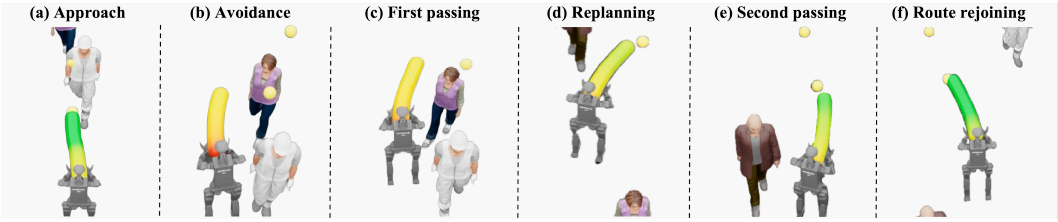


Figure 2: Receding-horizon planning with RiskNav. The colored line represents the selected 1.6 s action rollout, with colors indicating predicted collision probability (green: low; yellow: elevated). Yellow spheres denote local waypoints.

3.3 Action-Conditioned Safety Prediction for Online Planning

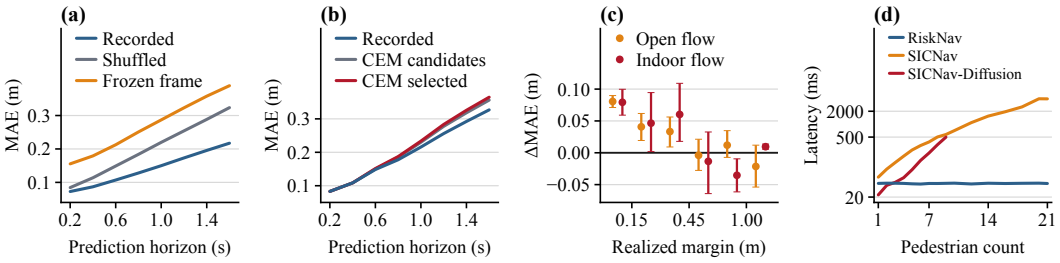


Figure 3: Reliability and efficiency of online planning. (a) Dynamic-margin readout error vs. prediction horizon for recorded, shuffled, and frozen-frame actions. (b) Readout error of counterfactual actions in closed-loop planning (all CEM candidates vs. selected ones). (c) Error difference between selected counterfactual and executed actions, where ΔMAE is the counterfactual-action MAE minus the executed-action MAE. (d) Replanning latency vs. pedestrian count. SICNav-Diffusion is shown up to 9 pedestrians due to a 24 GB compilation memory limit.

On the held-out validation set, latent rollouts conditioned on the executed action sequences yield a dynamic-margin mean absolute error (MAE) of 0.141 m, averaged over the 1.6 s prediction horizon.

Reusing the current-frame representation at every rollout step or temporally shuffling the actions produces larger errors, with the gaps increasing over the prediction horizon (Figure 3(a)). These controls indicate that the safety-margin readout depends on both predicted temporal evolution and action order, rather than only on the current observation. At each control cycle, CEM scores multiple candidate action sequences but executes only the first action of the selected sequence. The unselected candidates are therefore counterfactual to the observed rollout. For offline evaluation, we branch the simulator from the corresponding planning states and execute each candidate sequence to obtain a reference dynamic-margin trajectory. Over the first 0.4 s, the readout errors for executed and counterfactual action sequences are comparable (Figure 3(b)). At longer horizons, the error gap is concentrated in close encounters, defined by a ground-truth dynamic margin below 0.45 m (Figure 3(c)). Across the evaluated range of 1 to 21 pedestrians, RiskNav's replanning latency remains near 42 ms, whereas SICNav's latency grows rapidly to 575 ms at 9 pedestrians, and SICNav-Diffusion (Samavi et al., 2025b) reaches 498 ms at the same crowd size (Figure 3(d)). Consequently, the latent rollout supports fast, action-sensitive safety evaluation of alternative plans online.

3.4 Safety-Aware Representations and Ablation Study

We first analyze the ablations in Table 2 to assess whether safety-supervised representation learning and the use of the resulting safety predictions in planning both contribute to navigation performance. Table 2 isolates these two components separately and jointly. RiskNav is the only configuration that reaches every goal (Reach = 1.000), achieving the highest Succ (0.750), the lowest RC (0.250), and the lowest Freeze rate (0.053). Blocking safety-loss gradients from reaching the encoder and predictor reduces Succ to 0.350. Removing J_{risk} from the planning objective reduces Succ to 0.125, while the variant without the complete safety pathway yields a Succ of 0.025. The same performance ordering is observed for Reach, RC, and Freeze.

Table 2: Ablation of the risk evaluator. (i) w/o Shaping applies a stop-gradient to the predicted representation, restricting safety supervision to train the evaluator alone; (ii) w/o Risk cost retains the full model but omits the risk cost during planning; (iii) w/o Risk evaluator removes both, reducing the system to the LeWM baseline (Maes et al., 2026) without safety supervision.

Method	Risk evaluator			Succ↑	Reach↑	RC↓	Freeze↓
	Supervision	Shaping	Risk cost				
w/o Shaping	✓	×	✓	0.350 ±0.150	0.675 ±0.150	0.850 ±0.238	0.673 ±0.083
w/o Risk cost	✓	✓	×	0.125 ±0.100	0.250 ±0.138	1.075 ±0.312	0.858 ±0.062
w/o Risk evaluator	×	×	×	0.025 ±0.037	0.100 ±0.088	1.200 ±0.362	0.943 ±0.025
RiskNav	✓	✓	✓	**0.750** ±0.125	**1.000** ±0.000	**0.250** ±0.125	**0.053** ±0.021

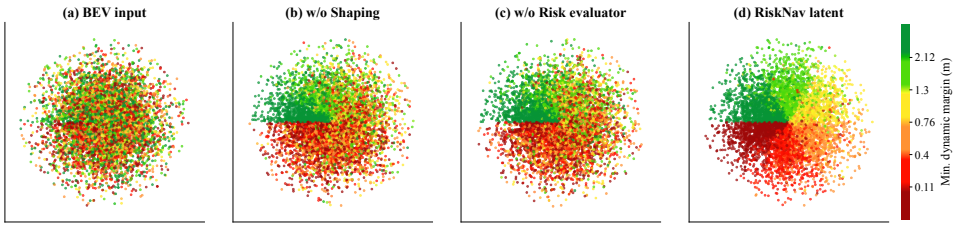


Figure 4: PaCMAP latent space visualizations. Colors indicate future safety margins (red: low; green: high) for (a) BEV, and encoders from (b) w/o Shaping, (c) w/o Risk evaluator, and (d) RiskNav.

We next examine how safety supervision affects the organization of learned encoder representations with respect to future dynamic safety margins. For the same validation samples, Figure 4 compares the PaCMAP projections of the BEV inputs and the encoder representations produced by the two ablations and RiskNav, colored by the minimum dynamic safety margin over the subsequent 1.6 s. In the BEV projection, samples with different margin values are intermingled, with no clear ordering. In the two ablation projections, high-margin samples form a distinct region, but lower- and intermediate-margin samples remain largely intermingled. By contrast, the RiskNav projection shows a clearer progression from low to high margins. The "w/o Risk cost" variant is omitted because it only changes the planning objective and uses the same trained encoder as RiskNav. Taken together, these projections suggest that backpropagated safety supervision makes states with different

future dynamic safety margins more readily distinguishable in the learned representation, which is consistent with the performance drop of the "w/o Shaping" variant in Table 2.

3.5 Real-Robot Deployment

We deploy RiskNav on a Unitree H1-2 humanoid using model weights trained entirely in simulation, without fine-tuning on real-robot data. A head-mounted Livox Mid-360 LiDAR provides 3D point clouds for online perception. The robot localizes against a prebuilt map, and an A* planner generates the global path. A perception pipeline based on point-cloud clustering and multi-object tracking first estimates local static obstacles and pedestrian tracks from the LiDAR point cloud. These estimates are then converted into the BEV observation $\mathbf{o}_i$ using the same construction as in simulation before being supplied to the RiskNav encoder. Based on this input, RiskNav replans at 5 Hz with a 1.6 s prediction horizon and sends the resulting linear and angular velocity commands to the manufacturer-provided locomotion controller. We evaluate the deployed system in ten trials along the same route, and RiskNav reaches the goal in all of them (Table 3). Across these trials, the robot spends 8.6 ± 2.5 s per trial within 1 m of the nearest pedestrian, and the mean trial-wise minimum center distance is 0.62 ± 0.06 m. Figure 5 presents representative behaviors under four real-world pedestrian configurations. In the top row, RiskNav moves aside for an oncoming pedestrian and passes behind two pedestrians crossing its path. In the bottom row, it navigates among three pedestrians moving in different directions and past three stationary pedestrians. Across these configurations, the robot avoids contact while continuing toward the local waypoint. Together, the quantitative results and qualitative examples support the feasibility of transferring RiskNav from simulation to closed-loop crowd navigation on a physical humanoid without real-world fine-tuning.

Table 3: Real-robot results over ten trials along the same route. Continuous metrics are reported as mean ± standard deviation across trials.

Metric	Value
Reached	10/10
Time (s)	38.5 ±3.5
Mean speed (m/s)	0.28 ±0.01
Stop fraction (%)	8.4 ±3.6
Min. center distance (m)	0.62 ±0.06
Time within 1 m (s)	8.6 ±2.5

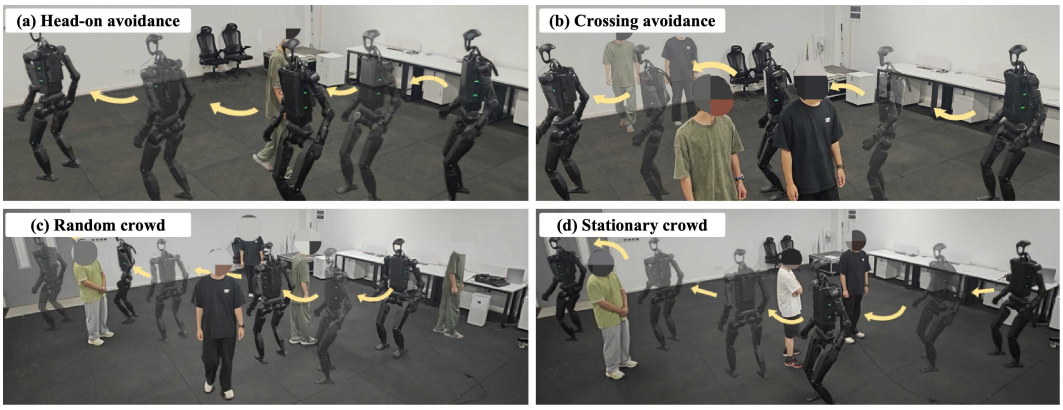


Figure 5: Zero-shot real-world deployment of RiskNav on the Unitree H1-2 humanoid across four pedestrian configurations. Each panel overlays snapshots from successive time steps, with yellow arrows indicating the robot's motion.

4 Conclusion

RiskNav shows that action-conditioned latent prediction can support crowd navigation when its predicted representations encode future safety information in a form that the planner can use. Across indoor, open, and out-of-distribution evaluations, RiskNav achieves the highest strict success rates and the fewest robot-responsible contacts among the evaluated methods. Its replanning latency remains nearly constant as crowd size grows. Ablations show that these gains arise from using safety information to shape future representations and guide action selection. Ten zero-shot real-robot trials further support feasibility beyond simulation. Although RiskNav transfers zero-shot to the real robot, it remains less reliable when comparing candidate actions in close encounters, and learning its safety-relevant representations still depends on simulator-derived supervision.

AI Use Statement

Large language models were used as assistants in this work to help draft and polish text, check references, and assist with writing code for the implementation and data analysis. The research questions, method design, experimental protocols, and conclusions were determined entirely by the authors. The authors reviewed and verified all model-assisted text, code, and citations, and take full responsibility for the content of this paper.

Ethics Statement

Our real-robot experiments involve human participants walking in the vicinity of a humanoid robot. All participants were members of our research group and gave written informed consent before taking part. To minimize risk, the robot operated at reduced speed, an operator could halt it at any time with an emergency stop, a safety supervisor monitored every trial, and the experiments were conducted in a cleared area without bystanders. Faces in the real-world photographs shown in this paper are blurred or otherwise not identifiable. The simulation experiments involve no human subjects.

Reproducibility Statement

To ensure full reproducibility, we detail the model architecture, loss functions, training configurations, and evaluation protocols in the appendix. We also provide our complete codebase—including the latent world model, risk evaluator, CEM planner, simulator, evaluation scripts, model checkpoints, and ROS 2 implementation—at https://anonymous.4open.science/r/submission-3B02/. All baseline methods were either re-evaluated from official checkpoints or reproduced under identical environments and seeds.

References

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971, 2016. doi: 10.1109/cvpr.2016.110.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. doi: 10.48550/arXiv.1807.06757.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023. doi: 10.1109/cvpr52729.2023.01499.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. doi: 10.48550/arXiv.2506.09985.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. TD-JEPA: Latent-predictive representations for zero-shot reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=SzXDuBN8M1.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. doi: 10.48550/arXiv.2511.08544.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801, 2025. doi: 10.1109/cvpr52734.2025.01472.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=QaCCuDfBk2.

Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6015–6022, 2019. doi: 10.1109/icra.2019.8794134.

Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P. How. Socially aware motion planning with deep reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, 2017a. doi: 10.1109/iros.2017.8202312.

Yu Fan Chen, Miao Liu, Michael Everett, and Jonathan P. How. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 285–292, 2017b. doi: 10.1109/icra.2017.7989037.

Matthieu Destrade, Oumayma Bounou, Quentin Le Lidec, Jean Ponce, and Yann LeCun. Value-guided action planning with JEPA world models. *arXiv preprint arXiv:2601.00844*, 2026. doi: 10.48550/arXiv.2601.00844.

Daniel Dugas, Juan I. Nieto, Roland Siegwart, and Jen Jen Chung. NavRep: Unsupervised representations for reinforcement learning of robot navigation in dynamic human environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7829–7835, 2021. doi: 10.1109/icra48506.2021.9560951.

Daniel Dugas, Olov Andersson, Roland Siegwart, and Jen Jen Chung. NavDreams: Towards camera-only RL navigation among humans. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2504–2511, 2022. doi: 10.1109/iros47612.2022.9982045.

Michael Everett, Yu Fan Chen, and Jonathan P. How. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3052–3059, 2018. doi: 10.1109/iros.2018.8593871.

Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997. doi: 10.1109/100.580977.

Anthony Francis, Claudia Pérez-D'Arpino, Chengshu Li, Fei Xia, Alexandre Alahi, Rachid Alami, Aniket Bera, Abhijat Biswas, Joydeep Biswas, Rohan Chandra, Hao-Tien Lewis Chiang, Michael Everett, Sehoon Ha, Justin Hart, Jonathan P. How, Haresh Karnan, Tsang-Wei Edward Lee, Luis J. Manso, Reuth Mirsky, Sören Pirk, Phani Teja Singamaneni, Peter Stone, Ada V. Taylor, Peter Trautman, Nathan Tsoi, Marynel Vázquez, Xuesu Xiao, Peng Xu, Naoki Yokoyama, Alexander Toshev, and Roberto Martín-Martín. Principles and guidelines for evaluating social robot navigation algorithms. *ACM Transactions on Human-Robot Interaction*, 14(2):1–65, 2025. doi: 10.1145/3700599.

Zeying Gong, Tianshuai Hu, Ronghe Qiu, and Junwei Liang. From cognition to precognition: A future-aware framework for social navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9122–9129, 2025. doi: 10.1109/icra55743.2025.11127910.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2255–2264, 2018. doi: 10.1109/cvpr.2018.00240.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406, 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51 (5):4282–4286, 1995. doi: 10.1103/PhysRevE.51.4282.

Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. SACSoN: Scalable autonomous control for social navigation. *IEEE Robotics and Automation Letters*, 9(1):49–56, 2024. doi: 10.1109/lra.2023.3329626.

Tianshuai Hu, Zeying Gong, Lingdong Kong, XiaoDong Mei, Yiyi Ding, Qi Zeng, Ao Liang, Rong Li, Yangyi Zhong, and Junwei Liang. NavThinker: Action-conditioned world models for coupled prediction and planning in social navigation. *arXiv preprint arXiv:2603.15359*, 2026. doi: 10.48550/arXiv.2603.15359.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf.

Shuijing Liu, Peixin Chang, Weihang Liang, Neeloy Chakraborty, and Katherine Driggs-Campbell. Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3517–3524, 2021. doi: 10.1109/icra48506.2021.9561595.

Shuijing Liu, Peixin Chang, Zhe Huang, Neeloy Chakraborty, Kaiwen Hong, Weihang Liang, D. Livingston McPherson, Junyi Geng, and Katherine Driggs-Campbell. Intention aware robot crowd navigation with attention-based interaction graph. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12015–12021, 2023. doi: 10.1109/icra48891.2023.10160660.

Shuijing Liu, Haochen Xia, Fatemeh Cheraghi Pouria, Kaiwen Hong, Neeloy Chakraborty, Zichao Hu, Joydeep Biswas, and Katherine Driggs-Campbell. HEIGHT: Heterogeneous interaction graph transformer for robot navigation in crowded and constrained environments. *IEEE Transactions on Automation Science and Engineering*, 23:1211–1230, 2026. doi: 10.1109/tase.2025.3646588.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026. doi: 10.48550/arXiv.2603.19312.

Christoforos I. Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core challenges of social robot navigation: A survey. *ACM Transactions on Human-Robot Interaction*, 12(3):1–39, 2023. doi: 10.1145/3583741.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning (CoRL)*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1049–1065, 2021. URL https://proceedings.mlr.press/v155/pinneri21a.html.

Sriyash Poddar, Christoforos I. Mavrogiannis, and Siddhartha S. Srinivasa. From crowd motion prediction to robot navigation in crowds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6765–6772, 2023. doi: 10.1109/iros55552.2023.10341464.

Sébastien Racanière, Théophane Weber, David P. Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/9e82757e9a1c12cb710ad680db11f6f1-Abstract.html.

Pratyaksh Rao, Wancong Zhang, Randall Balestriero, Yann LeCun, and Giuseppe Loianno. Sky-JEPA: Learning long-horizon world models for zero-shot sim-to-real control of quadrotors. *arXiv preprint arXiv:2606.23444*, 2026. doi: 10.48550/arXiv.2606.23444.

Arun Kumar Reddy, Vaibhav Malviya, and Rahul Kala. Social cues in the autonomous navigation of indoor mobile robots. *International Journal of Social Robotics*, 13(6):1335–1358, 2021. doi: 10.1007/s12369-020-00721-1.

Reuven Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999. doi: 10.1023/A:1010091220143.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision (ECCV)*, pp. 683–700, 2020. doi: 10.1007/978-3-030-58523-5_40.

Sepehr Samavi, James R. Han, Florian Shkurti, and Angela P. Schoellig. SICNav: Safe and interactive crowd navigation using model predictive control and bilevel optimization. *IEEE Transactions on Robotics*, 41:801–818, 2025a. doi: 10.1109/tro.2024.3484634.

Sepehr Samavi, Anthony Lem, Fumiaki Sato, Sirui Chen, Qiao Gu, Keijiro Yano, Angela P. Schoellig, and Florian Shkurti. SICNav-Diffusion: Safe and interactive crowd navigation with diffusion trajectory predictions. *IEEE Robotics and Automation Letters*, 10(9):8738–8745, 2025b. doi: 10.1109/lra.2025.3585713.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4.

Luca Scofano, Alessio Sampieri, Tommaso Campari, Valentino Sacco, Indro Spinelli, Lamberto Ballan, and Fabio Galasso. Following the human thread in social navigation. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=M8OGl34Pmg.

Phani Teja Singamaneni, Pilar Bachiller-Burgos, Luis J. Manso, Anaís Garrell, Alberto Sanfeliu, Anne Spalanzani, and Rachid Alami. A survey on socially aware robot navigation: Taxonomy and future challenges. *The International Journal of Robotics Research*, 43(10):1533–1572, 2024. doi: 10.1177/02783649241230562.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 49126–49162, 2025. doi: 10.52202/085713-1465.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=cHZn5Gdh8e.

Pete Trautman, Jeremy Ma, Richard M. Murray, and Andreas Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research*, 34(3):335–356, 2015. doi: 10.1177/0278364914557874.

Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 797–803, 2010. doi: 10.1109/iros.2010.5654369.

Jur van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1928–1935, 2008. doi: 10.1109/robot.2008.4543489.

Jur van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *International Symposium on Robotics Research (ISRR)*, pp. 3–19, 2011. doi: 10.1007/978-3-642-19457-3_1.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic MPC for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721, 2017. doi: 10.1109/icra.2017.7989202.

Zhanteng Xie and Philip Dames. DRL-VO: Learning to navigate through crowded dynamic scenes using velocity obstacles. *IEEE Transactions on Robotics*, 39(4):2700–2719, 2023. doi: 10.1109/tro.2023.3257549.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M. Kitani. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9793–9803, 2021. doi: 10.1109/iccv48922.2021.00967.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135, 2025. URL `https://proceedings.mlr.press/v267/zhou25t.html`.

Appendix

A Related Work **15**

B Implementation Details **16**

C Experimental Details **17**
C.1 Simulation Setup and Evaluation Protocol 17
C.2 Baselines 18
C.3 Real-Robot Setup 18

D Additional Results **19**
D.1 Detailed Navigation Benchmark Results 19
D.2 Ablations and Latent Representation Analysis 21
D.3 Real-World Humanoid Robot Deployment 22

A Related Work

Evaluating candidate actions in crowd navigation. The central distinction among crowd-navigation methods is whether the online decision process explicitly compares the multi-step safety consequences induced by alternative robot action sequences. Hand-designed local methods (Helbing & Molnár, 1995; Fox et al., 1997; van den Berg et al., 2008; 2011; Reddy et al., 2021) and learned navigation policies (Chen et al., 2017b;a; Everett et al., 2018; Chen et al., 2019; Liu et al., 2021; 2023; Xie & Dames, 2023; Liu et al., 2026; Hirose et al., 2024; Scofano et al., 2025) operate at low online cost and can encode future-aware behavior through objectives, value functions, recurrence, or attention (Gong et al., 2025). However, they select actions without rolling out alternative action sequences and comparing their resulting safety consequences at planning time. Predict-then-plan pipelines forecast pedestrian trajectories to capture pedestrian–pedestrian interactions, then check candidate robot trajectories for collisions (Alahi et al., 2016; Gupta et al., 2018; Yuan et al., 2021; Poddar et al., 2023). Because the forecast is fixed independently of each candidate, it cannot represent how pedestrian motion changes in response to that candidate (Trautman & Krause, 2010). Interaction-aware methods incorporate robot plans directly: Interacting Gaussian Processes model joint distributions over cooperative human–robot trajectories (Trautman & Krause, 2010; Trautman et al., 2015), Trajectron++ conditions forecasts on a future ego plan (Salzmann et al., 2020), and SICNav and SICNav-Diffusion couple robot planning with explicit pedestrian trajectories (Samavi et al., 2025a;b). Their agent-wise trajectory representations grow with crowd size and prediction horizon, increasing the cost of repeatedly evaluating candidate actions online.

Latent world models and safety-readable planning. Latent world models encode observations into compact latent states and learn action-conditioned dynamics over those states (Ha & Schmidhuber, 2018; Racanière et al., 2017; Hafner et al., 2019; 2020; Schrittwieser et al., 2020). Recent methods optimize action sequences within latent feature spaces (Hansen et al., 2022; 2024; Zhou et al., 2025) or train policies using latent representations (Dugas et al., 2021; 2022; Hu et al., 2026). Navigation World Models generate future egocentric frames with a conditional diffusion transformer, so part of their model capacity and per-candidate computation goes to reconstructing details unrelated to collision risk (Bar et al., 2025). Joint-embedding predictive architectures (JEPAs) avoid reconstructing observations by predicting target representations (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025; Sobal et al., 2025; Bagatella et al., 2026; Maes et al., 2026; Terver et al., 2026; Destrade et al., 2026; Rao et al., 2026). Yet low latent prediction error does not guarantee safety readability: safety-critical changes can be small relative to the aggregate prediction error, so futures on opposite sides of a safety boundary need not remain distinguishable. Even when this distinction is preserved, risk-aware planning still requires an explicit mapping from latent states to physical safety quantities. RiskNav addresses both limitations by jointly training a risk evaluator that maps predicted latents directly to static and dynamic safety margins and collision probabilities. Backpropagating the safety loss shapes the latent space to preserve risk-relevant features, and the evaluator's outputs provide the CEM planner with stage-wise risk terms.

B Implementation Details

Observation, action, and waypoint representation. Each observation is formatted as a three-channel, world-aligned Bird's-Eye-View (BEV) raster covering a $20 \times 20\,\mathrm{m}$ region at 256×256 resolution. Static obstacles, pedestrians, and the ego robot are rendered separately across these channels, with the robot's heading explicitly encoded; global paths and local waypoints are omitted from the visual observations. A centered crop corresponding to the latest robot position is extracted from the temporal observation history and resized to 256×256 without rotation, followed by standard ImageNet normalization. Finally, the robot control action is defined as $\mathbf{a}_i = (v_i, \omega_i)$, and the local waypoint is expressed in the robot body frame, remaining fixed throughout each rollout horizon. Key dimensions, action bounds, and temporal configurations are summarized in Table B.1.

Network architecture. The visual encoder $f_{\boldsymbol{\theta}}$ is a ViT-Tiny architecture trained from scratch, operating on 16×16 patches with 192-dimensional token embeddings, 12 Transformer layers, 3 attention heads, and an FFN expansion width of 768. Its CLS token is mapped through a $192 \to 2048 \to 192$ projection head to extract the frame-level latent representation. A projection head with identical dimensions projects the predictor outputs into the target representation space.

The action encoder comprises a pointwise convolution followed by an MLP, while the local waypoint is scaled by $3.2\,\mathrm{m}$ and processed by a dedicated MLP. Their element-wise sum conditions $g_{\boldsymbol{\phi}}$, a 6-layer causal Transformer featuring 16 attention heads, an FFN width of 2048, and Adaptive Layer Normalization Zero (AdaLN-Zero) layers. The risk evaluator $q_{\boldsymbol{\psi}}$ applies layer normalization followed by two fully connected layers of width 256 to predict static and dynamic safety margins, collision logits, waypoint distance, and single-step goal progress.

Safety labels and training objectives. Let $\mathbf{x}_i$ denote the robot position, r_{r} its radius, and $\mathbf{p}_{i,j}$ and r_j represent the position and radius of pedestrian j, respectively. Let $d(\mathbf{x}_i, \mathcal{O})$ be the signed distance from the robot center to the nearest static obstacle boundary. The safety labels are defined as

$$\begin{aligned} m_i^{\mathrm{s}} &= \mathrm{clip}(d(\mathbf{x}_i, \mathcal{O}) - r_{\mathrm{r}} - b_0,\ -5,\ 5)\,, \\ m_i^{\mathrm{d}} &= \mathrm{clip}\left(\min_j \left[\|\mathbf{x}_i - \mathbf{p}_{i,j}\|_2 - r_{\mathrm{r}} - r_j\right] - b_0,\ -5,\ 5\right), \\ c_i^b &= \mathbf{1}\{m_i^b \le 0\}\,, \qquad b \in \{\mathrm{s}, \mathrm{d}\}. \end{aligned} \tag{B.1}$$

We set $b_0 = 0.02\,\mathrm{m}$ and $r_{\mathrm{r}} = r_j = 0.3\,\mathrm{m}$. Missing static obstacles or pedestrians default to a margin of $5\,\mathrm{m}$. Collision probabilities are derived via a sigmoid function. Navigation supervision encompasses local waypoint distance and single-step goal progress. All labels are synthesized from privileged simulator states for offline training.

Regression uses Smooth L1 loss (transition parameter 1.0), and collision classification uses binary cross-entropy with a positive-class weight of 10.0. Loss weights are set as follows: latent prediction (1.0), SIGReg (0.09), static/dynamic margin regression (0.3/0.5), static/dynamic collision classification (0.5/1.0), and waypoint distance/progress (0.5/0.5).

Target representations in $\mathcal{L}_{\mathrm{JEPA}}$ are generated via the shared encoder with a stop-gradient operation applied solely to the target branch. SIGReg operates on non-detached features. Gradients from risk and navigation losses propagate through $q_{\boldsymbol{\psi}}$ and $g_{\boldsymbol{\phi}}$ into $f_{\boldsymbol{\theta}}$.

Offline training. The offline dataset comprises 34,500 simulated episodes of 160 steps across diverse map layouts, crowd configurations, pedestrian behaviors, and robot controllers (up to 12 pedestrians per scene; speeds 0.25–0.8 m/s). No spatial data augmentation is applied to the BEV inputs. Control actions are standardized using training statistics. All learnable modules are optimized jointly with AdamW. Optimization settings are detailed in Table B.1.

Receding-horizon planning. Candidate action sequences are forward-propagated using a planar unicycle kinematic model:

$$\begin{aligned} \theta_{i+1} &= \theta_i + \omega_i \Delta t, \\ x_{i+1} &= x_i + v_i \cos(\theta_{i+1}) \Delta t, \\ y_{i+1} &= y_i + v_i \sin(\theta_{i+1}) \Delta t. \end{aligned} \tag{B.2}$$

The planning objective follows Eqs. (6)–(7) in the main text. Action smoothness J_{smooth} is computed relative to the previously executed command. While the supervised safety margin incorporates a

0.02 m geometric buffer, the dynamic planning penalty penalizes pedestrian proximity prior to actual boundary encroachment.

The Cross-Entropy Method (CEM) samples control sequences from a diagonal Gaussian distribution, evaluates their costs, retains elite samples, and refits the distribution iteratively. The first planning cycle uses a fixed prior; subsequent cycles warm-start the mean by shifting the previously selected sequence forward by one step while resetting the standard deviations. Samples are clipped to within admissible action bounds. RiskNav executes the first action of the lowest-cost sampled sequence after the final iteration. Complete planning parameters are cataloged in Table B.1.

Table B.1: Key implementation, training, and planning hyperparameters of RiskNav.

Hyperparameter	Value	Hyperparameter	Value
History length H	3	Rollout horizon K	8
BEV crop	9.6×9.6 m	Latent dimension	192
Control interval Δt	0.2 s	Batch size	32
Learning rate	5×10^{-5}	Weight decay	10^{-3}
Training epochs	3	Warmup updates	4,366
AdamW (β_1, β_2)	$(0.9, 0.999)$	Gradient clip	1.0
Training seed	101	Precision	bf16
SIGReg points	17	Random projections	1024
Linear velocity	$[0, 1.0]$ m/s	Angular velocity	± 1.5 rad/s
CEM samples M	64	CEM iterations N	3
Elite fraction ρ	0.1 (6)	Minimum std.	10^{-3}
Initial mean (v, ω)	$(0.35, 0)$	Initial std. (v, ω)	$(0.35, 0.75)$
Tracking weight λ_{trk}	6.0	Distance weight λ_d	1.5
Progress weight λ_δ	1.5	Smoothness weight λ_u	0.1
Planning discount γ	0.97	Static threshold τ^{s}	0 m
Dynamic threshold τ^{d}	0.6 m	Softplus scale β	0.2 m

C Experimental Details

C.1 Simulation Setup and Evaluation Protocol

Scenario construction. We evaluate navigation performance across three distinct environmental setups: 1) *Indoor*: This environment features a fixed map layout with static boundary walls across six crowd-density levels, containing approximately 4–5, 6–7, 8–9, 10–11, 12, and 19 pedestrians, respectively. The first five levels progressively increase pedestrian crowd density along the predefined trajectory, whereas the most crowded setting introduces dense head-on pedestrian traffic along a narrow corridor segment.

2) *Open*: This setup employs an unconstrained environment without static obstacles, incorporating three pedestrian flow patterns: head-on, co-flowing, and mixed traffic. Each flow condition is evaluated with 7, 14, 24, and 42 pedestrians, yielding 12 distinct evaluation settings.

3) *OOD*: This variant retains the identical 12 open configurations while replacing the underlying ORCA pedestrian motion model with the Social Force Model (SFM). Each condition consists of 40 paired evaluation episodes, totaling 240 indoor, 480 open, and 480 OOD test scenarios. To ensure rigorous comparability, all evaluated baselines within a specific condition share identical initial scene states, pedestrian realization seeds, and pseudo-random sequences.

Notably, since our training dataset contains at most 12 pedestrians per scene, the dense indoor setting and the open/OOD environments featuring 14, 24, and 42 pedestrians explicitly benchmark OOD generalization to unseen crowd densities.

Pedestrian dynamics and episode protocol. The indoor and open benchmark evaluations utilize ORCA pedestrian dynamics (van den Berg et al., 2011) under the unreactive ("invisible-robot") evaluation protocol: pedestrians execute mutual collision avoidance but do not react to the ego robot. The OOD suite preserves the initial poses, flow topologies, agent counts, and evaluation random seeds, while transitioning pedestrian motion generation to SFM (Helbing & Molnár, 1995).

All approaches are benchmarked on strictly paired episodes sharing identical initial conditions. The master random seed for evaluation is fixed to 20260702. An episode is marked as successful if the ego robot reaches a $0.4\,\mathrm{m}$ tolerance radius around the final waypoint within a $480\,\mathrm{s}$ time budget. Controller-specific execution frequencies follow their original specifications. RiskNav replans at a rate of $5\,\mathrm{Hz}$ ($0.2\,\mathrm{s}$ per control step); in open and OOD simulations, pedestrian motion is integrated at $20\,\mathrm{Hz}$, with each planned RiskNav action zero-order-held across four simulator steps.

Navigation metrics. We evaluate system performance using Strict Success Rate (Succ), Reach Rate (Reach), Robot-Responsible Contacts per episode (RC), Freeze Rate (Freeze), Minimum Surface Separation (MinSep), Elapsed Navigation Time, and Success weighted by Path Length (SPL). Reach denotes the proportion of episodes reaching the goal, while Succ strictly requires zero robot-responsible collision events. Freeze represents the fraction of active navigation steps where the robot's linear speed falls below $0.05\,\mathrm{m/s}$. MinSep records the minimum boundary-to-boundary distance between the robot and any pedestrian within an episode. SPL measures path efficiency relative to the optimal reference path computed by an A* global planner. Timed-out episodes are assigned the full execution time limit of $480\,\mathrm{s}$.

A contact event is logged when the nearest robot-to-pedestrian surface distance drops to $\leq 0.05\,\mathrm{m}$; contiguous steps within a continuous contact are aggregated into a single event. Defining the contact normal vector as pointing from the robot center to the pedestrian center, a contact is explicitly attributed to the robot if its velocity component along this normal is positive, its linear speed exceeds $0.05\,\mathrm{m/s}$, and its longitudinal closing speed contribution exceeds that of the pedestrian. Contacts driven predominantly by aggressive pedestrian motion are excluded from the robot's penalty, whereas ambiguous contacts lacking positive forward movement from either party are conservatively assigned as robot-responsible.

Statistical reporting. To estimate statistical uncertainty, we apply $10{,}000$ condition-stratified paired bootstrap resamples, enforcing strictly identical resampled episode indices across all compared algorithms. Tabulated simulation results report the mean accompanied by half the width of the 95% percentile confidence interval. Physical real-world trials report the standard deviation calculated across independent experimental runs.

C.2 Baselines

Baseline categorizations. We compare RiskNav against nine representative navigation baselines, categorized by their deployment paradigms: 1) Analytical and optimization-based planners: DWA (Fox et al., 1997), ORCA (van den Berg et al., 2011), and SICNav (Samavi et al., 2025a). 2) Retrained learning-based models: SARL (Chen et al., 2019), DS-RNN (Liu et al., 2021), and NavThinker (Hu et al., 2026), each retrained within our simulation platform using their official network architectures and loss functions. 3) Zero-shot deployed models: CrowdNav++ (Liu et al., 2023), DRL-VO (Xie & Dames, 2023), and HEIGHT (Liu et al., 2026), deployed directly using their publicly released model checkpoints without environment-specific fine-tuning.

Method-specific configurations. SICNav considers the 6 nearest pedestrians during the navigation benchmark runs. CrowdNav++ is omitted from indoor evaluations because its published observation space lacks representations for static wall geometry. HEIGHT and DRL-VO retain their original action spaces and velocity smoothing filters. For DRL-VO, the main benchmark disables its heuristic wall-safety reflex while retaining its terminal goal-reaching controller.

Runtime benchmark setup. The computational latency experiment presented in Figure 3(d) of the main text is executed independently from the closed-loop navigation benchmark. In this latency test, SICNav is evaluated without capping its pedestrian tracking to the 6 nearest individuals. Additionally, SICNav-Diffusion (Samavi et al., 2025b) is included solely for computational benchmark profiling; its latency curve terminates at 9 pedestrians due to memory allocation limits ($> 24\,\mathrm{GB}$ GPU VRAM during compilation). All execution latencies are profiled on a dedicated workstation equipped with an NVIDIA RTX 4090 GPU and an Intel i7-14700KF CPU.

C.3 Real-Robot Setup

Hardware platform, perception, and BEV generation. Physical experiments are conducted on a Unitree H1-2 humanoid robot using the simulation-trained RiskNav model weights without any real-

world adaptation or fine-tuning. Perception relies on a head-mounted Livox Mid-360 3D LiDAR. The robot localizes within a prebuilt point-cloud map, and an online A* planner generates global reference paths and local target waypoints. Point-cloud clusters identified as human pedestrians are dynamically tracked using a Kalman filter and rendered as moving cylinders; unclassified clusters are merged into the static environment map. Combined with real-time pose estimates, these entities generate an identical three-channel, world-aligned 256×256 BEV tensor used during training, followed by the $9.6 \times 9.6\,\mathrm{m}$ spatial cropping described in Appendix B. Simulator ground-truth states and motion-capture feedback are strictly excluded from the planning loop.

Closed-loop physical deployment. RiskNav executes online replanning at $5\,\mathrm{Hz}$ with a temporal rollout horizon of $K = 8$ (1.6 s lookahead). For real-world hardware safety, the CEM action sampling bounds are constrained to:

$$v \in [0, 0.5]\,\mathrm{m/s}, \qquad \omega \in [-0.6, 0.6]\,\mathrm{rad/s}. \tag{C.1}$$

The navigation controller node enforces these explicit linear and angular velocity constraints and smooths the selected control commands using an exponential moving average (EMA) filter with a smoothing coefficient of 0.6 prior to dispatching them to the low-level locomotion controller. Aligning the CEM sampling bounds with these execution constraints ensures that all sampled candidate actions lie within the physical action limits.

Physical evaluation protocol. Real-world trials consist of 10 repeated navigations along a standardized test track featuring staged human interaction scenarios: an oncoming pedestrian, two crossing pedestrians, a three-person dynamic crowd flow, and three stationary pedestrians. Goal completion is declared when the robot enters a $0.4\,\mathrm{m}$ radius around the destination waypoint. Completion times and velocities are recorded via onboard odometry. The reported minimum human–robot distance reflects the minimum center-to-center distance measured by the onboard pedestrian tracking module, distinguishing it from the surface-to-surface MinSep metric computed in simulation. External video logs are independently reviewed to verify physical contact events. All physical runs are executed using identical neural weights and control parameters without per-trial tuning.

D Additional Results

D.1 Detailed Navigation Benchmark Results

Robot-aware pedestrians. The benchmark in Table 1 of the main text uses pedestrians that do not react to the robot (Appendix C.1). We additionally evaluate a robot-aware setting by including the robot as an agent in the ORCA simulation, so that pedestrians account for the robot when computing their collision-avoidance velocities. The robot itself remains controlled by the evaluated navigation method rather than by ORCA. All other evaluation settings remain unchanged. Each method is evaluated on 240 episodes with matched scene initializations and random seeds. CrowdNav++ is omitted because its observation space does not represent static walls.

Table D.1 reports the pooled results. RiskNav achieves the highest strict success rate (0.83), the fewest robot-responsible contacts (0.19 per episode), and the largest minimum separation (0.221 m), while reaching the goal in all 240 episodes. Under paired McNemar tests, its strict success rate is significantly higher than that of all baselines except SARL.

Why strict success remains below one. Robot-aware pedestrians reduce, but do not eliminate, contact failures. ORCA determines the pedestrians' avoidance velocities, whereas the robot continues to execute actions selected by each evaluated navigation method; therefore, the reciprocal collision-avoidance assumption does not provide a hard safety guarantee for the joint system. Contacts can also occur in dense corridor configurations where pedestrians are simultaneously constrained by surrounding agents, walls, and speed limits. Moreover, the benchmark registers a contact when the surface distance falls below 0.05 m, making the strict-success criterion more conservative than geometric collision alone.

Table D.1: Navigation results with robot-aware pedestrians in the indoor setting. Results are pooled over the six crowd-density levels (240 episodes total).

Method	Succ↑	Reach↑	RC↓	Freeze↓	MinSep (m)↑	Time (s)↓	SPL↑
DWA	0.72 ±0.06	1.00 ±0.00	0.33 ±0.08	0.00 ±0.00	0.090 ±0.015	38.9 ±0.2	0.99 ±0.00
ORCA	0.58 ±0.06	1.00 ±0.00	0.51 ±0.09	0.01 ±0.00	0.056 ±0.012	36.9 ±0.1	1.00 ±0.00
SICNav	0.33 ±0.06	1.00 ±0.00	0.99 ±0.10	0.01 ±0.00	0.038 ±0.010	44.9 ±4.2	1.00 ±0.00
SARL	0.79 ±0.05	0.99 ±0.01	0.23 ±0.06	0.02 ±0.01	0.162 ±0.021	41.7 ±4.7	0.97 ±0.01
DS-RNN	0.74 ±0.06	1.00 ±0.00	0.28 ±0.06	0.02 ±0.00	0.117 ±0.016	37.2 ±0.2	0.99 ±0.00
NavThinker	0.66 ±0.06	1.00 ±0.00	0.50 ±0.10	0.19 ±0.01	0.140 ±0.021	54.2 ±3.5	0.86 ±0.02
CrowdNav++	–	–	–	–	–	–	–
DRL-VO	0.07 ±0.03	0.95 ±0.03	1.80 ±0.11	0.08 ±0.03	0.011 ±0.007	96.9 ±11.3	0.93 ±0.03
HEIGHT	0.15 ±0.04	0.81 ±0.02	2.43 ±0.47	0.03 ±0.00	0.026 ±0.010	183.9 ±7.5	0.66 ±0.02
RiskNav	**0.83** ±0.05	**1.00** ±0.00	**0.19** ±0.05	0.04 ±0.01	**0.221** ±0.027	47.9 ±0.9	0.95 ±0.01

Tables D.2–D.4 report additional navigation metrics for the indoor, open-field, and OOD settings, complementing the strict success rate (Succ) and robot-responsible contacts per episode (RC) reported in Table 1 of the main text. In addition to Succ and RC, we report reach rate (Reach), freeze rate (Freeze), minimum surface separation (MinSep), navigation time, and success weighted by path length (SPL). All metrics are computed according to the definitions and evaluation protocol in Appendix C.1. The indoor evaluation spans six crowd-density levels (240 episodes), and the open-field evaluation spans three pedestrian-flow patterns and four crowd sizes (480 episodes). The OOD evaluation contains 480 episodes under the same open-field configurations, with ORCA replaced by SFM.

Table D.2: Comprehensive navigation performance in the indoor setting, aggregated over six crowd-density levels (40 paired episodes per setting; 240 total).

Method	Succ↑	Reach↑	RC↓	Freeze↓	MinSep (m)↑	Time (s)↓	SPL↑
DWA	0.59 ±0.06	1.00 ±0.00	0.79 ±0.19	0.03 ±0.01	0.068 ±0.015	42.9 ±1.4	0.97 ±0.00
ORCA	0.45 ±0.06	1.00 ±0.00	0.81 ±0.14	0.06 ±0.01	0.039 ±0.012	39.8 ±0.5	1.00 ±0.00
SICNav	0.42 ±0.06	1.00 ±0.01	0.88 ±0.13	0.10 ±0.01	0.022 ±0.009	52.9 ±5.3	1.00 ±0.01
SARL	0.64 ±0.05	0.82 ±0.01	0.38 ±0.11	0.21 ±0.01	0.174 ±0.026	118.0 ±6.3	0.80 ±0.01
DS-RNN	0.63 ±0.06	0.95 ±0.03	1.03 ±0.35	0.13 ±0.03	0.102 ±0.015	73.1 ±13.1	0.90 ±0.03
NavThinker	0.53 ±0.06	1.00 ±0.00	0.78 ±0.12	0.20 ±0.01	0.102 ±0.021	53.1 ±2.3	0.86 ±0.01
CrowdNav++	–	–	–	–	–	–	–
DRL-VO	0.40 ±0.06	0.93 ±0.03	0.80 ±0.10	0.11 ±0.03	0.010 ±0.005	107.7 ±13.3	0.90 ±0.03
HEIGHT	0.38 ±0.05	0.83 ±0.03	1.53 ±0.26	0.04 ±0.00	0.020 ±0.010	182.7 ±7.1	0.66 ±0.02
RiskNav	**0.76** ±0.05	**1.00** ±0.00	**0.31** ±0.08	**0.05** ±0.01	**0.181** ±0.029	48.6 ±0.7	0.93 ±0.01

Table D.3: Comprehensive navigation performance in the open setting, aggregated over head-on, co-flowing, and mixed flow patterns across 7, 14, 24, and 42 pedestrians (480 total paired episodes).

Method	Succ↑	Reach↑	RC↓	Freeze↓	MinSep (m)↑	Time (s)↓	SPL↑
DWA	0.18 ±0.03	0.99 ±0.01	8.25 ±0.89	0.03 ±0.00	0.048 ±0.013	33.2 ±1.7	0.82 ±0.01
ORCA	0.14 ±0.03	0.99 ±0.01	6.49 ±0.69	0.14 ±0.01	0.037 ±0.012	28.3 ±1.0	0.94 ±0.01
SICNav	0.38 ±0.04	0.99 ±0.01	1.65 ±0.21	0.15 ±0.01	0.050 ±0.012	39.9 ±1.3	0.93 ±0.01
SARL	0.10 ±0.02	0.60 ±0.04	5.98 ±0.60	0.32 ±0.03	−0.009 ±0.014	135.7 ±15.2	0.53 ±0.04
DS-RNN	0.00 ±0.00	0.00 ±0.00	16.43 ±2.50	0.12 ±0.02	0.046 ±0.016	472.1 ±5.4	0.00 ±0.00
NavThinker	0.01 ±0.01	0.02 ±0.01	12.30 ±1.02	0.48 ±0.01	0.006 ±0.014	412.9 ±12.9	0.00 ±0.00
CrowdNav++	0.19 ±0.03	0.61 ±0.04	3.79 ±0.29	0.04 ±0.00	0.000 ±0.015	119.4 ±11.3	0.45 ±0.03
DRL-VO	0.25 ±0.03	0.91 ±0.02	3.32 ±0.32	0.12 ±0.01	0.005 ±0.009	89.1 ±3.6	0.54 ±0.02
HEIGHT	0.16 ±0.02	0.53 ±0.03	4.24 ±0.26	0.09 ±0.00	0.002 ±0.011	287.1 ±10.3	0.24 ±0.01
RiskNav	**0.57** ±0.03	**0.97** ±0.01	**1.15** ±0.13	**0.03** ±0.00	**0.147** ±0.018	34.0 ±1.1	0.77 ±0.01

Table D.4: Comprehensive navigation performance in the out-of-distribution (OOD) setting, replacing ORCA pedestrians with SFM pedestrians (480 total paired episodes).

Method	Succ↑	Reach↑	RC↓	Freeze↓	MinSep (m)↑	Time (s)↓	SPL↑
DWA	0.15 ±0.03	1.00 ±0.01	8.73 ±0.83	0.01 ±0.00	0.039 ±0.011	24.8 ±0.3	0.89 ±0.01
ORCA	0.10 ±0.02	1.00 ±0.00	7.50 ±0.68	0.11 ±0.00	0.030 ±0.009	22.9 ±0.2	0.99 ±0.00
SICNav	0.27 ±0.04	0.99 ±0.01	2.31 ±0.18	0.14 ±0.01	0.036 ±0.009	39.1 ±0.9	0.95 ±0.01
SARL	0.07 ±0.02	0.60 ±0.04	6.74 ±0.70	0.26 ±0.03	−0.019 ±0.012	127.8 ±15.0	0.54 ±0.04
DS-RNN	0.00 ±0.00	0.00 ±0.00	12.62 ±1.26	0.07 ±0.02	0.048 ±0.015	473.1 ±5.0	0.00 ±0.00
NavThinker	0.00 ±0.01	0.00 ±0.01	10.04 ±0.98	0.50 ±0.01	0.004 ±0.012	425.0 ±12.3	0.00 ±0.00
CrowdNav++	0.14 ±0.03	0.65 ±0.03	5.47 ±0.33	0.02 ±0.00	−0.013 ±0.015	90.7 ±10.1	0.51 ±0.03
DRL-VO	0.27 ±0.03	0.85 ±0.03	2.11 ±0.14	0.10 ±0.00	−0.017 ±0.009	68.5 ±1.6	0.53 ±0.02
HEIGHT	0.16 ±0.03	0.45 ±0.02	2.61 ±0.19	0.08 ±0.00	−0.018 ±0.006	286.0 ±11.6	0.21 ±0.01
RiskNav	**0.50** ±0.03	**0.96** ±0.02	**1.70** ±0.12	**0.02** ±0.00	**0.121** ±0.015	28.8 ±0.8	0.79 ±0.02

D.2 Ablations and Latent Representation Analysis

Ablation evaluation protocol. The component-wise ablation studies reported in Table 2 of the main text evaluate a standalone indoor configuration comprising 8–9 pedestrians across 40 paired test episodes, independent of the aggregated benchmark in Table 1. This distinction explains the numeric variations in full-model success rates between the two tables.

The w/o Shaping variant detaches risk loss gradients, preventing them from backpropagating into the world model predictor g_ϕ and visual encoder f_θ, while maintaining the supervised risk evaluator q_ψ and the trajectory planning risk cost. The w/o Risk cost variant utilizes the fully trained network architecture but disables J_{risk} during online receding-horizon planning. The w/o Risk evaluator variant removes all risk-supervised objectives during training alongside the corresponding risk cost term during planning. All remaining hyperparameters remain strictly controlled.

Readout accuracy of dynamic safety. Table D.5 reports dynamic safety-margin prediction errors on 1,200 held-out rollout windows, complementing the horizon-wise analysis in Figure 3(a) of the main text. We compare RiskNav with the w/o Shaping ablation, which retains supervision of the risk evaluator but blocks the resulting risk-loss gradients from updating the predictor and encoder.

Table D.5: Dynamic safety-readout accuracy on 1,200 held-out prediction windows.

Metric	RiskNav	w/o Shaping
Dynamic margin MAE (m)↓	**0.141**	0.216
Dynamic collision AUPRC↑	**0.806**	0.709

The w/o Risk cost variant is excluded from Table D.5 because it shares identical weights with RiskNav, differing solely in its online execution objective. Conversely, the w/o Risk evaluator variant has no risk evaluator; its representation utility is instead probed via kNN evaluation in Table D.6.

Quantitative readability of latent representations. To quantitatively validate the latent structure visualized in Figure 4 of the main text, we evaluate learned features across 15,000 validation states. Specifically, we train kNN probes on high-dimensional features and low-dimensional manifolds to predict the minimum dynamic safety margin and impending contact events over a 1.6 s horizon.

Table D.6: Quantitative representation readability analysis across 15,000 validation states (corresponding to Figure 4 of the main text).

Metric	Raw BEV (PCA-50)	w/o Shaping	w/o Risk evaluator	RiskNav
2D margin R^2↑	−0.067	0.175	0.101	**0.736**
2D contact AUC↑	0.504	0.522	0.528	**0.921**
Full-dim. margin R^2↑	−0.042	0.583	0.519	**0.849**
Full-dim. contact AUC↑	0.548	0.649	0.638	**0.927**

As demonstrated in Table D.6, safety-supervised gradients dramatically enhance representation readability in both full-dimensional and low-dimensional projected spaces, corroborating the qualitative trends illustrated in Figure 4.

D.3 Real-World Humanoid Robot Deployment

Figure D.1 shows representative closed-loop interaction sequences from RiskNav's deployment on the Unitree H1-2 humanoid. Across the illustrated sequences, RiskNav replans from successive observations, updates the selected action rollout as the local pedestrian configuration changes, and continues toward the waypoint after each interaction.

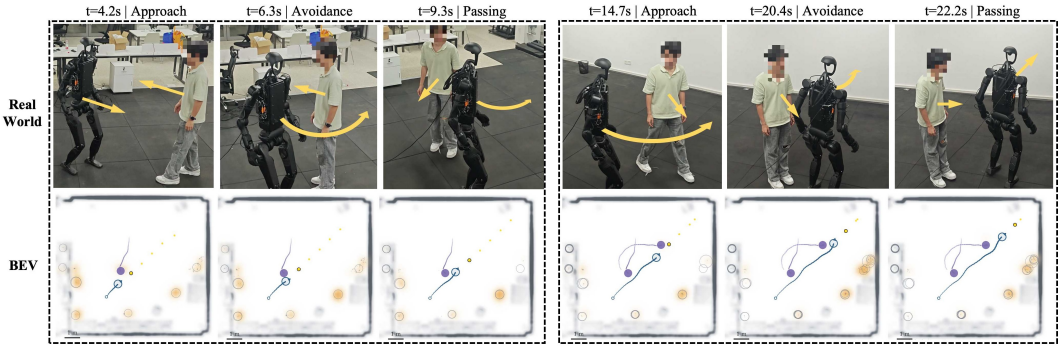


Figure D.1: Real-world collision avoidance sequences on a physical humanoid robot platform. Top: Third-person physical snapshots paired with corresponding onboard BEV spatial representations (bottom) at key interaction timestamps ($t = 4.2\,\text{s}, 6.3\,\text{s}, 9.3\,\text{s}$ and $t = 14.7\,\text{s}, 20.4\,\text{s}, 22.2\,\text{s}$). The sequences depict the full navigation progression from approach, reactive avoidance, to passing. Yellow arrows indicate the robot's motion.