LEARNING WHEN TO ADAPT: REACTIVITY-CONDITIONED RESIDUAL ADAPTATION UNDER HUMAN–ROBOT INTERACTION SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Pedestrians differ in whether they respond to a mobile robot, and these responses can change during navigation. Policies trained under a single response assumption may therefore be unnecessarily conservative or rely on cooperation that does not occur. We propose reactivity-conditioned residual adaptation (RCRA), which uses explicit estimates of pedestrian reactivity to modulate a learned residual around a frozen conservative navigation policy. A Transformer estimates per-pedestrian reactivity from motion history and robot context, while spatial aggregation determines the residual strength without online gradient updates. Across five ORCA-based interaction regimes, the estimator achieves 84.78% scenario-averaged classification accuracy. In heterogeneous and changing local interactions, replacing a Bayesian reactivity estimator with our learned estimator reduces mean collision rate from 6.56% to 4.85% and mean path length by 3.6%, with navigation components otherwise unchanged. Across all five regimes, RCRA reduces mean navigation time by 16.8% and path length by 16.7% relative to the conservative backbone, while mean success rate decreases from 98.35% to 96.05%. On recorded PeRoI trajectories, our estimator improves balanced accuracy over the Bayesian reactivity estimator by 4.2 percentage points. Controlled robot trials with prescribed reactivity labels validate the residual-modulation mechanism on hardware. These results support explicit reactivity estimation as a useful signal for policy modulation under heterogeneous and changing local interactions.

1 INTRODUCTION

Pedestrians respond heterogeneously to mobile robots: some adjust their motion in response to the robot, while others continue along their nominal paths. We study *human–robot interaction distribution shift*: the pedestrian response dynamics encountered at deployment differ from those used to train a navigation policy, and can vary across pedestrians, locations, and time. Policies trained under non-reactive pedestrian assumptions tend to preserve safety, but they can remain overly conservative when pedestrians actually make space. Conversely, policies adapted to fully cooperative pedestrians can move efficiently but may collide when deployed around pedestrians who do not yield.

Uncertainty-based navigation methods can respond to prediction mismatch by increasing safety margins (Yao et al., 2024; 2025). However, in interaction-sensitive crowds, not all prediction errors indicate danger: a pedestrian who yields may deviate from a non-reactive forecast in a way that creates more free space for the robot. Increasing margins alone can therefore overlook opportunities created by robot-directed responses. This motivates estimating how pedestrians respond to the robot and using that estimate to adjust navigation behavior.

We investigate whether explicit estimates of pedestrian reactivity provide a useful signal for modulating a pretrained navigation policy as local interaction conditions change. We propose reactivity-conditioned residual adaptation (RCRA), which couples two components: a learned estimator of latent robot-directed reactivity and a policy residual modulated by those estimates. The estimator uses human motion history and robot context to infer which pedestrians are likely to respond. Spatial aggregation converts these estimates into a continuous scale κ_t that modulates a residual learned

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.

from reactive crowds around a frozen conservative policy. This lets the robot adjust to local interactions without online gradient updates or a supplied scenario label. Our contributions are threefold:

- We introduce RCRA, a reactivity-conditioned residual adaptation framework for human–robot interaction distribution shift. It modulates an interaction-specific residual online while preserving a frozen conservative policy, allowing navigation behavior to adjust to local pedestrian responses without online gradient updates.
- We learn per-pedestrian robot-directed reactivity beliefs from partial observation histories. A Transformer-based estimator captures interaction context, and DAgger-style refinement addresses the observations induced by the adaptive policy. Spatial aggregation connects these beliefs to residual modulation.
- We validate RCRA through large-scale simulation and controlled real-robot trials using prescribed reactivity labels, demonstrating improved safety–efficiency trade-offs across heterogeneous crowd scenarios.

2 Related Work

Robot Crowd Navigation. Classical methods such as Social Force, velocity obstacles, and reciprocal collision avoidance generate local motion using structured assumptions about pedestrian behavior (Helbing & Molnar, 1995; Fiorini & Shiller, 1998; van den Berg et al., 2011), while cooperative planners explicitly account for human responses to robot motion (Trautman & Krause, 2010; Trautman et al., 2015). Learning-based methods capture crowd interactions through multi-agent reinforcement learning, attention-based pedestrian representations, and spatio-temporal interaction models (Chen et al., 2017; Everett et al., 2018; Chen et al., 2019; Liu et al., 2021). Trajectory prediction further enables policies to anticipate pedestrian motion (Liu et al., 2023), while constrained reinforcement learning incorporates safety costs and prediction uncertainty (Yao et al., 2025). However, the resulting policies can depend on the human–robot interaction patterns encountered during training: policies are often trained under the assumption that pedestrians do not react to the robot, encouraging safe but conservative behavior. Recent approaches such as Human–Human & Human–Robot Interaction Transformer (Shen et al., 2026) train across varied pedestrian-response conditions, but do not explicitly adapt the policy online based on estimated pedestrian reactivity. Model-based approaches such as Behavior-Driven Danger Signaling (BDDS) (Hosseinzadeh et al., 2023) address uncertainty about human responses through Bayesian inference, but require a human goal to evaluate their behavioral cost model. Such goals may be unknown or change during real-world interactions, limiting the applicability of this assumption. We instead estimate robot-directed reactivity from observed interaction history and use these estimates to modulate a learned residual on a frozen conservative policy as interaction conditions change.

Prediction uncertainty and conformal safety. Pedestrian trajectory prediction is central to social navigation, with representative models including Social-LSTM, Social-GAN, and Trajectron++ (Alahi et al., 2016; Gupta et al., 2018; Salzmann et al., 2020). Broader surveys further categorize human-motion prediction methods by interaction modeling, uncertainty handling, and dataset assumptions, highlighting why prediction remains difficult under distribution shift (Rudenko et al., 2020). Conformal prediction provides distribution-free coverage under assumptions such as exchangeability; standard guarantees do not automatically hold under arbitrary distribution shift (Angelopoulos & Bates, 2021). Recent crowd-navigation methods use adaptive or interaction-sensitive conformal uncertainty to regulate planning under out-of-distribution pedestrian behavior (Yao et al., 2025; Huang et al., 2024). These methods regulate navigation using prediction uncertainty. Our approach additionally estimates robot-directed reactivity, because a deviation caused by yielding may create usable free space and warrant a different navigation response.

Policy adaptation and parameter-efficient tuning. Robotic policies can be robustified or adapted through domain randomization, meta-learning, online reinforcement learning, imitation-learning correction, and efficient policy modulation (Tobin et al., 2017; Peng et al., 2018; Finn et al., 2017; Ross et al., 2011; Xu et al., 2023; Schmied et al., 2023). However, adapting the entire policy can require longer training and greater computational resources. To reduce this overhead, parameter-efficient tuning freezes most pretrained parameters and trains a small set of additional parameters,

as in residual adapters, neural-network adapters, low-rank adaptation (LoRA), and adapters for navigation and robot learning (Rebuffi et al., 2017; Houlsby et al., 2019; Hu et al., 2022; Qiao et al., 2023; Liu et al., 2024; Team et al., 2024).

3 Reactivity-Conditioned Residual Adaptation

System Framework and Problem Formulation. We formulate crowd navigation as a partially observable Markov decision process (POMDP), because robot-directed reactivity is not directly observed. Let x_t^r denote the robot state at timestep t, including its position, velocity, heading, and goal. For pedestrian i, let $x_{i,t}^h = (p_{i,t}^h, v_{i,t}^h, z_{i,t})$, where $p_{i,t}^h$ and $v_{i,t}^h$ are planar position and velocity, and $z_{i,t} \in \{0, 1\}$ is a latent indicator of robot-directed reactivity. In simulation, both types follow ORCA: reactive pedestrians ($z_{i,t} = 1$) include the robot in their interaction set, whereas non-reactive pedestrians ($z_{i,t} = 0$) do not. Both types respond to other pedestrians. A reactive label describes the response rule and does not require visible yielding at every timestep.

We write the state compactly as $s_t = (x_t^r, \{x_{i,t}^h\}_{i=1}^N)$, with pedestrian goals and fixed agent attributes implicit in this notation; N is the total number of pedestrians. The robot observes its own state and tracks $N_t \leq N$ pedestrians within sensing range, but does not observe their reactivity labels. We construct the augmented observation

$$o_t = \left(x_t^r, \{p_{i,t}^h\}_{i=1}^{N_t}, \{\hat{\tau}_{i,t+1:t+K|t}\}_{i=1}^{N_t}, \{\hat{\delta}_{i,t+1:t+K|t}\}_{i=1}^{N_t} \right), \tag{1}$$

where K is the prediction horizon in timesteps, $\hat{\tau}_{i,t+1:t+K|t}$ is the trajectory predicted at time t (Huang et al., 2021), and $\hat{\delta}_{i,t+1:t+K|t}$ denotes the associated prediction-uncertainty features used by GenSafeNav (Yao et al., 2025). Position histories provide motion information, with pedestrian identities maintained by tracking. The robot's belief about reactivity is

$$b_{i,t} = P(z_{i,t} = 1 \mid o_{1:t}). \tag{2}$$

Here, $o_{1:t}$ denotes the observation history. The continuous belief $b_{i,t} \in [0, 1]$ represents uncertainty about a binary response mode, rather than a directly supervised degree of reactivity. Interaction distribution shift changes how these modes are distributed and evolve, and consequently changes pedestrian motion in response to the robot.

This setting presents two coupled tasks: infer latent reactivity from partial histories and use those estimates to adjust navigation under changing interaction dynamics. As illustrated in Fig. 1, RCRA combines a residual policy family, a learned belief estimator, and spatial aggregation into the continuous adaptation scale κ_t. The navigation objective and training costs follow GenSafeNav (Appendix B). We first describe the policy family, then the estimator and the rule connecting them.

A Reactivity-Conditioned Policy. We represent navigation behavior using a frozen conservative backbone and a learned interaction-specific parameter residual. We first train the backbone on non-reactive crowds and freeze its parameters θ_0. We then learn a residual $\Delta\theta$ from reactive-crowd rollouts while keeping θ_0 fixed. We parameterize the residual with LoRA. For an adapted linear layer ℓ with frozen weight $W_{\ell,0} \in \mathbb{R}^{m \times n}$, the learnable update is $\Delta W_\ell = A_\ell B_\ell$, where $A_\ell \in \mathbb{R}^{m \times r}$, $B_\ell \in \mathbb{R}^{r \times n}$, and $r \ll \min(m, n)$. Here, m and n are the layer's output and input dimensions, and r is the adaptation rank. The fixed LoRA gain is absorbed into these factors. We adopt the GenSafeNav architecture (Yao et al., 2025), which combines human–human attention, human–robot attention, a GRU, and actor–critic heads. We inject LoRA only into the human–robot attention layers and actor–critic heads, since latent reactivity primarily affects how the robot interprets human–robot interaction and selects actions. Human–human attention remains frozen to preserve the learned crowd-interaction representation. Thus, $\Delta\theta$ contains the low-rank updates for these adapted layers and zeros for all remaining parameters. Appendix E compares the computational costs of this implementation with gated full fine-tuning and action-space interpolation.

At runtime, a reactivity-dependent scale $\kappa_t \in [0, 1]$ controls the residual strength, giving the active parameters $\theta_t = \theta_0 + \kappa_t \Delta\theta$. Each adapted layer uses $W_{\ell,t} = W_{\ell,0} + \kappa_t A_\ell B_\ell$ for $\ell \in \mathcal{L}_{\text{adapt}}$, while all other layers retain $W_{\ell,t} = W_{\ell,0}$. Here, $\mathcal{L}_{\text{adapt}}$ denotes the set of LoRA-adapted layers. This yields the policy family

$$\pi_{\kappa_t}(a_t \mid o_t) = \pi(a_t \mid o_t; \theta_0 + \kappa_t \Delta\theta), \tag{3}$$

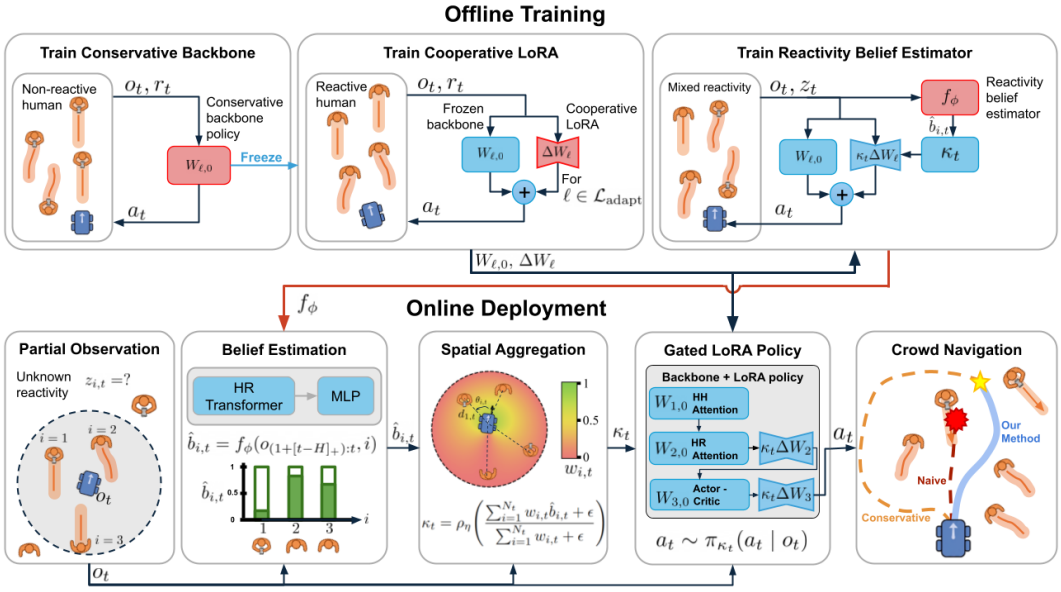

Figure 1: **Overview of RCRA.** Offline, a conservative backbone is trained on non-reactive pedestrians and then frozen while LoRA residual modules are learned for reactive-crowd behavior. A reactivity belief estimator is trained on mixed scenarios. Online, the robot estimates per-human reactivity beliefs from partial observations, aggregates them using spatial relevance weights, and uses the resulting scale κ_t to gate the LoRA residual to adapt the navigation safety–efficiency operating point.

where a_t is the robot's velocity command. We suppress the policy's GRU hidden state in this notation; the policy also depends on its recurrent history. Setting $\kappa_t = 0$ disables the residual and uses the conservative parameters, while $\kappa_t = 1$ activates the full learned residual. Intermediate values continuously interpolate the parameters; they do not guarantee monotonic changes in actions or safety. We obtain κ_t from estimated per-pedestrian beliefs and spatial relevance, as described next.

Reactivity Belief Estimation. The inference problem is to estimate the reactivity belief $b_{i,t}$ from a finite history of the robot's observations. We estimate this using a learned reactivity estimator

$$\hat{b}_{i,t} = f_\phi(o_{(1+[t-H]_+):t}, i) \in [0, 1], \tag{4}$$

where $[x]_+ = \max(0, x)$, H is the history length in timesteps, and i identifies the pedestrian track. This window provides temporal context for identifying motion adjustments in response to the robot while limiting the influence of outdated behavior.

To process observation histories, each pedestrian's augmented state and the robot state are embedded with MLPs at each timestep. A multi-head Transformer encoder with padding masks captures interactions among nearby pedestrians, and an MLP outputs $\hat{b}_{i,t}$ for each observed pedestrian. We train f_ϕ using the current simulator labels $z_{i,t}$ with binary cross-entropy and refine it with DAgger (Ross et al., 2011) to reduce closed-loop covariate shift. At each DAgger-style aggregation iteration, we roll out the current adaptive policy, label visited observations using the simulator's reactivity labels, and update f_ϕ on the aggregated data. The policy backbone and learned residual remain fixed. This addresses the estimator's policy-induced covariate shift, which is distinct from the changing pedestrian response dynamics targeted by RCRA.

Spatial Reactivity Aggregation. A global average of $\hat{b}_{i,t}$ may obscure locally important differences in mixed crowds: distant reactive pedestrians can dilute the influence of a nearby non-reactive pedestrian. We therefore weight reactivity beliefs by distance and bearing relative to the robot's heading, emphasizing nearby pedestrians ahead of the robot when determining the residual strength. This spatial aggregation provides a tractable control heuristic rather than a solution to the

Algorithm 1 Reactivity-conditioned residual adaptation

Input: frozen parameters θ_0, residual $\Delta\theta$, estimator f_ϕ, prediction horizon K, history length H, sensing radius R, bias-control parameter $\eta \in (0, 1)$, and $\epsilon > 0$
for each control step t **do**
  Construct o_t from robot state, tracked pedestrians, predictions, and uncertainty features; update the observation history
  for each observed pedestrian $i = 1, \ldots, N_t$ **do**
    Estimate reactivity belief $\hat{b}_{i,t}$ using Eq. (4)
    Compute spatial relevance $w_{i,t}$ using Eq. (5)
  end for
  Aggregate continuous beliefs into κ_t using Eq. (6)
  Set adapted weights $W_{\ell,t} = W_{\ell,0} + \kappa_t \Delta W_\ell$ for all $\ell \in \mathcal{L}_{\text{adapt}}$
  Sample action $a_t \sim \pi_{\kappa_t}(a_t \mid o_t)$
  Execute a_t
end for

full POMDP. For each observed pedestrian, we define a spatial relevance weight

$$w_{i,t} = \left[1 - \frac{d_{i,t}}{R}\right]_+ \left[\cos\left(\frac{\psi_{i,t}}{2}\right)\right]_+, \tag{5}$$

where $R > 0$ is the sensing radius, $d_{i,t}$ is the robot–pedestrian distance, and $\psi_{i,t} \in [-\pi, \pi]$ is the pedestrian's bearing relative to the robot's heading. These weights approximate spatial relevance to near-term navigation.

The runtime adaptation scale is then computed as

$$\kappa_t = \rho_\eta\left(\frac{\sum_{i=1}^{N_t} w_{i,t}\hat{b}_{i,t} + \epsilon}{\sum_{i=1}^{N_t} w_{i,t} + \epsilon}\right), \qquad \rho_\eta(\bar{\kappa}) = \frac{\eta\bar{\kappa}}{\eta\bar{\kappa} + (1-\eta)(1-\bar{\kappa})}, \quad \kappa_t, \bar{\kappa} \in [0, 1]. \tag{6}$$

where $\epsilon > 0$ prevents division by zero and biases the aggregation toward full residual activation when total spatial relevance is small. The bias-control parameter $\eta \in (0, 1)$ adjusts the safety–efficiency preference: $\eta = 0.5$ preserves the aggregated reactivity score, $\eta < 0.5$ reduces residual activation, and $\eta > 0.5$ increases it. When no pedestrian is spatially relevant, $\sum_i w_{i,t} = 0$ and Eq. (6) gives $\kappa_t = 1$, so the robot defaults to efficient free-space behavior.

Training and Deployment. The framework is trained in three stages. First, we train the conservative backbone policy on non-reactive pedestrians and freeze its weights. Second, with the backbone frozen, we train the low-rank residual modules on reactive-crowd rollouts to learn cooperative behavior without overwriting the conservative prior. Third, we train the reactivity estimator using simulator labels and refine it with DAgger to reduce closed-loop covariate shift. The backbone and residual parameters remain fixed throughout this stage. DAgger refinement improves closed-loop reactivity-estimation accuracy from 77.27% to 84.78%. At deployment, no online gradient update is performed. The robot repeatedly estimates local human reactivity, computes κ_t, evaluates the reactivity-conditioned policy in Eq. (3), and executes the velocity command. Algorithm 1 summarizes deployment; Appendices B and C provide implementation and estimator details.

4 Experiments

We evaluate whether RCRA adapts navigation to interaction distribution shift and whether its learned estimator supplies a useful control signal. Simulation tests the complete pipeline across five response regimes; recorded human trajectories test estimation, and controlled robot trials isolate residual adaptation.

Experimental Setup and Baselines. We use the CrowdNav simulator (Chen et al., 2019) with 20 pedestrians in a $12\,\text{m} \times 12\,\text{m}$ environment with randomized start and goal positions. The robot has radius $0.3\,\text{m}$, pedestrian radii are sampled from $[0.3, 0.5]\,\text{m}$, and all agents have maximum

speed $1.0\,\mathrm{m/s}$. We report Success Rate (SR), Collision Rate (CR), Navigation Time (NT), and Path Length (PL), defined in Appendix A. Each episode ends in success, collision, or timeout. We omit Timeout Rate (TR) from the table since $\mathrm{TR} = 100\% - \mathrm{SR} - \mathrm{CR}$. All simulated pedestrians follow the ORCA collision-avoidance policy: reactive pedestrians include the robot in their collision-avoidance calculations, whereas non-reactive pedestrians respond only to other pedestrians. We test five interaction scenarios: *Non-reactive* and *Reactive* contain homogeneous pedestrian types, *Mixed* randomly distributes equal numbers of both types, and *Spatial Clusters* separates the types into distinct regions. Reactivity remains fixed within each episode in these four scenarios. *Time-varying* starts from Mixed, with each pedestrian independently switching reactivity with 5% probability per timestep.

Baselines include the reactive local method ORCA (van den Berg et al., 2011), the game-theoretic planner BRNE (Muchen Sun et al., 2025), and learning-based policies comprising CrowdNav++ (Liu et al., 2023) and three GenSafeNav (Yao et al., 2025) variants. CrowdNav++ and GenSafeNav (Cons.) are trained in non-reactive crowds, with GenSafeNav (Cons.) serving as our frozen backbone; GenSafeNav (Naive) is trained in reactive crowds; and GenSafeNav (Rand.) is trained across all five interaction scenarios through domain randomization. Our LoRA residual is trained on reactive-crowd rollouts with the backbone frozen. Fixed scales $\kappa_t = 0.2$ and 0.8 provide controlled comparisons with RCRA, which selects κ_t online from estimated reactivity. For the bias adjustment $\rho_\eta(\cdot)$, we use $\eta = 0.5$ for *Adaptive*, $\eta = 0.3$ for *Adaptive-Safe*, and $\eta = 0.7$ for *Adaptive-Efficient*, biasing the policy toward neutral, more conservative, and more efficient residual modulation, respectively. The evaluation examines the safety–efficiency trade-off of reactivity-conditioned adaptation, the estimator's accuracy and impact on navigation, and performance relative to the baselines. Navigation results cover $1{,}250$ test episodes per scenario across five random seeds and are reported as mean $\pm$ standard deviation across seeds.

Reactivity-Conditioned Adaptation and the Safety–Efficiency Trade-off. Fig. 2 isolates the adaptation mechanism using ground-truth reactivity. Across the LoRA, gated full fine-tuning, and MPC sweeps, increasing the fixed scale from $\kappa_t = 0$ toward $\kappa_t = 1$ generally shortens paths at the cost of lower SR, exposing the safety–efficiency trade-off associated with a fixed adaptation strength. For LoRA, $\kappa_t = 0$ retains the conservative backbone, achieving the highest SR in the fixed-scale sweep but producing long paths even in reactive crowds, whereas $\kappa_t = 1$ fully activates the residual, improving efficiency but substantially reducing SR when pedestrians do not react. Adaptive scaling achieves a favorable balance by remaining conservative near non-reactive pedestrians and increasing residual strength when local pedestrian responses permit more efficient motion. The trend across mechanisms shows that reactivity-conditioned adaptation is not specific to LoRA: the same online conditioning principle extends to gated full fine-tuning and to a planning framework, MPC. Additional fixed-scale LoRA results and adaptive comparisons with full fine-tuning, action-space interpolation, and MPC are provided in Appendix D, with scalability and implementation details in Appendix E.

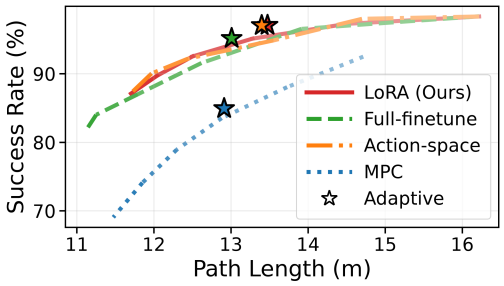

Figure 2: **Scenario-averaged adaptation trade-offs.** Fixed-scale sweeps run from light ($\kappa = 0$) to dark ($\kappa = 1$), with monotonically decreasing PL and SR. Stars denote adaptation using ground-truth reactivity. Results use a backbone with 2.5M parameters.

Table 1 compares two fixed LoRA scales with RCRA, using learned reactivity estimation. In Non-reactive crowds, RCRA achieves higher SR than the conservative fixed $\kappa_t = 0.2$ policy, achieving 95.52% versus 93.84%, while in Reactive crowds it is more efficient than the fixed $\kappa_t = 0.8$ policy, reducing PL from 11.15 m to 11.01 m. In Mixed, Spatial Clusters, and Time-varying scenarios, RCRA operates between the two fixed scales in both safety and efficiency, indicating that it adjusts the residual strength according to the local interaction conditions rather than committing to a single operating point. Together, these results show that RCRA adapts the safety–efficiency trade-off across all five interaction scenarios, preserving conservative behavior in non-reactive crowds, exploiting cooperation in reactive crowds, and balancing safety and efficiency under heterogeneous and changing responses.

Table 1: **Fixed versus reactivity-conditioned adaptation.** We compare fixed residual scales with adaptive residual modulation driven by BDDS or our learned reactivity estimator. Adaptive-Safe and Adaptive-Efficient are omitted in homogeneous scenarios.

Scenario	Conditioning Strategy	SR (%) ↑	CR (%) ↓	NT (s) ↓	PL (m) ↓
Non-reactive	Fixed ($\kappa_t = 0.2$)	<u>93.84</u> ± 0.78	<u>6.16</u> ± 0.78	<u>15.01</u> ± 0.34	<u>14.94</u> ± 0.34
	Fixed ($\kappa_t = 0.8$)	77.36 ± 2.92	22.64 ± 2.92	**12.66** ± 0.26	**12.64** ± 0.26
	BDDS (Hosseinzadeh et al., 2023) (Adaptive)	93.60 ± 0.63	6.40 ± 0.63	15.08 ± 0.34	15.02 ± 0.33
	Learned estimator (Adaptive)	**95.52** ± 0.69	**4.48** ± 0.69	15.86 ± 0.45	15.80 ± 0.43
	Ground Truth (Adaptive)	*97.04 ± 0.36*	*2.96 ± 0.36*	*16.08 ± 0.41*	*16.01 ± 0.44*
Reactive	Fixed ($\kappa_t = 0.2$)	**99.92** ± 0.18	**0.08** ± 0.18	14.34 ± 0.43	14.28 ± 0.42
	Fixed ($\kappa_t = 0.8$)	99.52 ± 0.52	0.48 ± 0.52	<u>11.16</u> ± 0.13	<u>11.15</u> ± 0.13
	BDDS (Hosseinzadeh et al., 2023) (Adaptive)	<u>99.68</u> ± 0.33	<u>0.32</u> ± 0.33	11.84 ± 0.18	11.80 ± 0.19
	Learned estimator (Adaptive)	99.28 ± 0.33	0.72 ± 0.33	**11.02** ± 0.12	**11.01** ± 0.12
	Ground Truth (Adaptive)	*99.20 ± 0.28*	*0.80 ± 0.28*	*10.87 ± 0.11*	*10.87 ± 0.11*
Mixed	Fixed ($\kappa_t = 0.2$)	**98.00** ± 0.00	**2.00** ± 0.00	14.88 ± 0.28	14.81 ± 0.28
	Fixed ($\kappa_t = 0.8$)	89.60 ± 1.13	10.40 ± 1.13	**12.19** ± 0.15	**12.17** ± 0.15
	BDDS (Hosseinzadeh et al., 2023) (Adaptive)	93.60 ± 1.44	6.40 ± 1.44	14.10 ± 0.46	14.07 ± 0.39
	Learned estimator (Adaptive)	<u>96.08</u> ± 1.58	<u>3.92</u> ± 1.58	<u>13.68</u> ± 0.43	<u>13.64</u> ± 0.37
	Ground Truth (Adaptive-Safe)	*97.36 ± 0.67*	*2.64 ± 0.67*	*14.26 ± 0.41*	*14.22 ± 0.40*
	Ground Truth (Adaptive)	*96.72 ± 1.21*	*3.28 ± 1.21*	*13.82 ± 0.37*	*13.78 ± 0.38*
	Ground Truth (Adaptive-Efficient)	*95.28 ± 1.53*	*4.72 ± 1.53*	*13.08 ± 0.33*	*13.06 ± 0.33*
Spatial Clusters	Fixed ($\kappa_t = 0.2$)	**96.00** ± 0.40	**4.00** ± 0.40	13.76 ± 0.40	13.71 ± 0.40
	Fixed ($\kappa_t = 0.8$)	90.64 ± 1.08	9.36 ± 1.08	**11.87** ± 0.25	**11.86** ± 0.25
	BDDS (Hosseinzadeh et al., 2023) (Adaptive)	93.28 ± 1.04	6.72 ± 1.04	13.60 ± 0.38	13.56 ± 0.41
	Learned estimator (Adaptive)	<u>94.88</u> ± 1.00	<u>5.12</u> ± 1.00	<u>13.01</u> ± 0.33	<u>12.98</u> ± 0.36
	Ground Truth (Adaptive-Safe)	*96.08 ± 1.11*	*3.92 ± 1.11*	*13.28 ± 0.32*	*13.26 ± 0.32*
	Ground Truth (Adaptive)	*95.68 ± 0.87*	*4.32 ± 0.87*	*13.02 ± 0.30*	*13.00 ± 0.29*
	Ground Truth (Adaptive-Efficient)	*95.52 ± 1.48*	*4.48 ± 1.48*	*12.90 ± 0.31*	*12.89 ± 0.31*
Time-varying	Fixed ($\kappa_t = 0.2$)	**98.64** ± 0.73	**1.36** ± 0.73	14.85 ± 0.17	14.78 ± 0.16
	Fixed ($\kappa_t = 0.8$)	91.44 ± 0.46	8.56 ± 0.46	**12.39** ± 0.25	**12.38** ± 0.25
	BDDS (Hosseinzadeh et al., 2023) (Adaptive)	93.44 ± 1.59	6.56 ± 1.59	14.72 ± 0.44	14.66 ± 0.43
	Learned estimator (Adaptive)	<u>94.48</u> ± 1.63	<u>5.52</u> ± 1.63	<u>14.19</u> ± 0.36	<u>14.13</u> ± 0.39
	Ground Truth (Adaptive-Safe)	*97.76 ± 1.00*	*2.24 ± 1.00*	*14.35 ± 0.30*	*14.27 ± 0.29*
	Ground Truth (Adaptive)	*96.88 ± 0.95*	*3.12 ± 0.95*	*13.78 ± 0.35*	*13.71 ± 0.31*
	Ground Truth (Adaptive-Efficient)	*94.16 ± 1.22*	*5.84 ± 1.22*	*12.99 ± 0.18*	*12.93 ± 0.18*

Note: Performance rankings (1st** and <u>2nd</u>) are calculated excluding the ground truth.*

We additionally vary η to bias the adaptive scale toward more conservative or more efficient behavior. Adaptive-Safe and Adaptive-Efficient are omitted in the homogeneous Non-reactive and Reactive scenarios, where ground-truth reactivity labels yield effectively the same behavior as the neutral Adaptive policy. In heterogeneous scenarios, Adaptive-Safe shifts the adaptive policy toward a more conservative operating point, increasing SR at the cost of longer NT and PL, whereas Adaptive-Efficient shifts it toward greater efficiency with lower NT and PL but reduced SR. The neutral Adaptive variant remains between these two operating points, showing that varying η changes the safety–efficiency bias while preserving online reactivity-conditioned adaptation.

Reactivity Belief Estimation under Interaction Shift. We compare our learned estimator with the passive inference component of BDDS, keeping the conservative backbone, learned residual, and spatial weighting rule fixed and changing only the supplied reactivity signal. Appendix C details the baseline adaptation and evaluation protocol. Across all five scenarios, our estimator achieves **84.78%** scenario-averaged classification accuracy, compared with 72.16% for BDDS. These more accurate estimates support a better empirical safety–efficiency trade-off in heterogeneous and changing crowds. From Table 1, averaging equally across Mixed, Spatial Clusters, and Time-varying, our learned estimator reduces CR from 6.56% to 4.85% (26.0% relative reduction) and PL from 14.10 to 13.58 m (3.6% reduction), while increasing SR from 93.44% to 95.15%. These controlled comparisons demonstrate the downstream navigation benefit of our learned reactivity estimates. In Non-reactive, the false-positive share decreases from 27.4% to 12.6%, accompanied by an increase in navigation SR from 93.60% to 95.52%. In Reactive, the false-negative share decreases from 27.5% to 9.6%, while NT decreases from 11.84 to 11.02 s and PL from 11.80 to 11.01 m. Ground-truth conditioning provides a reference for evaluating how closely learned estimates reproduce navigation performance with exact reactivity labels. Across the five scenarios, learned conditioning achieves 96.05% mean SR compared with 97.10% for ground-truth conditioning, a difference of approxi-

Table 2: **Interaction-shift benchmark.** We compare all methods across non-reactive, reactive, mixed, spatial-cluster, and time-varying scenarios over 1,250 test episodes per scenario across five random seeds. RCRA improves the empirical safety–efficiency trade-off.

Scenario	Methods	SR (%) ↑	CR (%) ↓	NT (s) ↓	PL (m) ↓
Non-reactive	*ORCA* (van den Berg et al., 2011)	*66.40 ± 3.88*	*29.20 ± 2.71*	*23.68 ± 0.82*	*15.03 ± 0.35*
	BRNE (Muchen Sun et al., 2025)	*23.04 ± 1.56*	*76.00 ± 1.17*	*17.07 ± 0.54*	*16.58 ± 0.53*
	CrowdNav++ (Liu et al., 2023)	*79.20 ± 2.35*	*20.80 ± 2.35*	*14.42 ± 0.28*	*13.53 ± 0.22*
	GenSafeNav (Cons.) (Yao et al., 2025)	**97.20** ± 0.40	**2.80** ± 0.40	16.59 ± 0.47	16.52 ± 0.47
	GenSafeNav (Naive) (Yao et al., 2025)	*59.52 ± 2.47*	*40.48 ± 2.47*	*11.61 ± 0.20*	*11.58 ± 0.21*
	GenSafeNav (Rand.) (Yao et al., 2025)	91.92 ± 1.45	7.84 ± 1.28	**15.16** ± 0.35	**15.05** ± 0.34
	Ours (RCRA)	<u>95.52</u> ± 0.69	<u>4.48</u> ± 0.69	<u>15.86</u> ± 0.45	<u>15.80</u> ± 0.43
Reactive	ORCA (van den Berg et al., 2011)	93.12 ± 1.11	0.72 ± 0.18	23.32 ± 0.72	15.39 ± 0.72
	BRNE (Muchen Sun et al., 2025)	86.80 ± 1.67	2.40 ± 0.57	18.26 ± 0.42	17.75 ± 0.41
	CrowdNav++ (Liu et al., 2023)	94.24 ± 0.46	5.44 ± 0.46	12.57 ± 0.17	11.83 ± 0.16
	GenSafeNav (Cons.) (Yao et al., 2025)	**100.00** ± 0.00	**0.00** ± 0.00	16.24 ± 0.47	16.19 ± 0.47
	GenSafeNav (Naive) (Yao et al., 2025)	99.20 ± 0.40	0.80 ± 0.40	**10.69** ± 0.07	**10.66** ± 0.08
	GenSafeNav (Rand.) (Yao et al., 2025)	<u>99.52</u> ± 0.52	<u>0.16</u> ± 0.36	14.01 ± 0.24	13.92 ± 0.24
	Ours (RCRA)	99.28 ± 0.33	0.72 ± 0.33	<u>11.02</u> ± 0.12	<u>11.01</u> ± 0.12
Mixed	ORCA (van den Berg et al., 2011)	83.68 ± 1.91	11.84 ± 1.49	23.87 ± 0.41	15.46 ± 0.39
	BRNE (Muchen Sun et al., 2025)	*43.36 ± 5.20*	*53.92 ± 4.93*	*16.52 ± 0.78*	*16.05 ± 0.76*
	CrowdNav++ (Liu et al., 2023)	85.44 ± 1.25	14.56 ± 1.25	13.71 ± 0.21	<u>13.59</u> ± 0.14
	GenSafeNav (Cons.) (Yao et al., 2025)	**98.56** ± 0.73	**1.36** ± 0.73	16.61 ± 0.58	16.55 ± 0.57
	GenSafeNav (Naive) (Yao et al., 2025)	82.40 ± 4.79	17.60 ± 4.79	**11.36** ± 0.20	**11.34** ± 0.17
	GenSafeNav (Rand.) (Yao et al., 2025)	<u>96.72</u> ± 0.66	<u>3.20</u> ± 0.75	14.74 ± 0.35	14.64 ± 0.35
	Ours (RCRA)	96.08 ± 1.58	3.92 ± 1.58	<u>13.68</u> ± 0.43	13.64 ± 0.37
Spatial Clusters	ORCA (van den Berg et al., 2011)	89.60 ± 2.24	6.88 ± 1.88	18.94 ± 0.51	14.14 ± 0.66
	BRNE (Muchen Sun et al., 2025)	*65.12 ± 4.27*	*30.80 ± 3.97*	*17.56 ± 0.42*	*17.03 ± 0.41*
	CrowdNav++ (Liu et al., 2023)	85.60 ± 1.57	14.40 ± 1.57	13.31 ± 0.12	13.13 ± 0.29
	GenSafeNav (Cons.) (Yao et al., 2025)	<u>97.12</u> ± 0.77	<u>2.88</u> ± 0.77	15.47 ± 0.68	15.40 ± 0.67
	GenSafeNav (Naive) (Yao et al., 2025)	82.88 ± 2.42	17.12 ± 2.42	**10.83** ± 0.16	**10.80** ± 0.16
	GenSafeNav (Rand.) (Yao et al., 2025)	**97.60** ± 0.28	**2.40** ± 0.28	14.79 ± 0.34	14.65 ± 0.35
	Ours (RCRA)	94.88 ± 1.00	5.12 ± 1.00	<u>13.01</u> ± 0.33	<u>12.98</u> ± 0.36
Time-varying	ORCA (van den Berg et al., 2011)	82.56 ± 2.41	12.40 ± 1.41	23.66 ± 1.04	15.11 ± 0.53
	BRNE (Muchen Sun et al., 2025)	*47.84 ± 2.81*	*49.68 ± 3.18*	*17.50 ± 0.79*	*17.01 ± 0.77*
	CrowdNav++ (Liu et al., 2023)	85.28 ± 2.07	14.72 ± 2.07	14.28 ± 0.18	<u>14.04</u> ± 0.17
	GenSafeNav (Cons.) (Yao et al., 2025)	**98.88** ± 0.77	**1.12** ± 0.77	16.50 ± 0.53	16.44 ± 0.53
	GenSafeNav (Naive) (Yao et al., 2025)	81.84 ± 4.20	18.16 ± 4.20	**11.60** ± 0.22	**11.55** ± 0.21
	GenSafeNav (Rand.) (Yao et al., 2025)	<u>96.96</u> ± 1.08	<u>2.72</u> ± 1.00	14.74 ± 0.29	14.65 ± 0.29
	Ours (RCRA)	94.48 ± 1.63	5.52 ± 1.63	<u>14.19</u> ± 0.36	14.13 ± 0.39

Note: Performance rankings (1st** and <u>2nd</u>) are calculated among methods that achieve SR $> 80\%$.*

mately 1.1 percentage points, while scenario-averaged NT and PL each differ by less than 0.3%. This proximity indicates that the learned estimator provides a useful conditioning signal, enabling the adaptive policy to approach the aggregate navigation performance of ground-truth conditioning.

We further evaluate reactivity estimation on recorded PeRoI (Agrawal et al., 2026) moving-robot trajectories, using avoidance and neutrality annotations as proxies for reactive and non-reactive behavior. Our estimator achieves **73.0%** balanced accuracy, compared with 68.8% for tuned BDDS, a gain of **4.2** percentage points. Our estimator learns temporal motion patterns directly from observation histories without requiring an intended pedestrian goal, whereas BDDS interprets observed actions through a fixed goal-and-safety cost model. These results support the use of learned temporal features for reactivity estimation from real human motion. Details are provided in Appendix C.

Comparison with Navigation Baselines. GenSafeNav (Cons.) and (Naive) expose the safety–efficiency trade-off between conservative and cooperative navigation. Compared with GenSafeNav (Cons.), RCRA reduces scenario-averaged NT by **16.8%** and PL by **16.7%**, while mean SR decreases from 98.35% to 96.05%. The benefit is most pronounced in Reactive, where RCRA reduces NT by **32.14%** and PL by **32.00%** while maintaining 99.28% SR, approaching GenSafeNav (Naive) in efficiency (11.01 m versus 10.66 m) with comparable SR (99.28% versus 99.20%). Conversely, in Non-reactive, RCRA remains close to GenSafeNav (Cons.) in SR (95.52% versus 97.20%), whereas GenSafeNav (Naive) drops to 59.52%. Thus, a single policy can move between conservative and cooperative behavior by conditioning its residual strength on locally observed interactions,

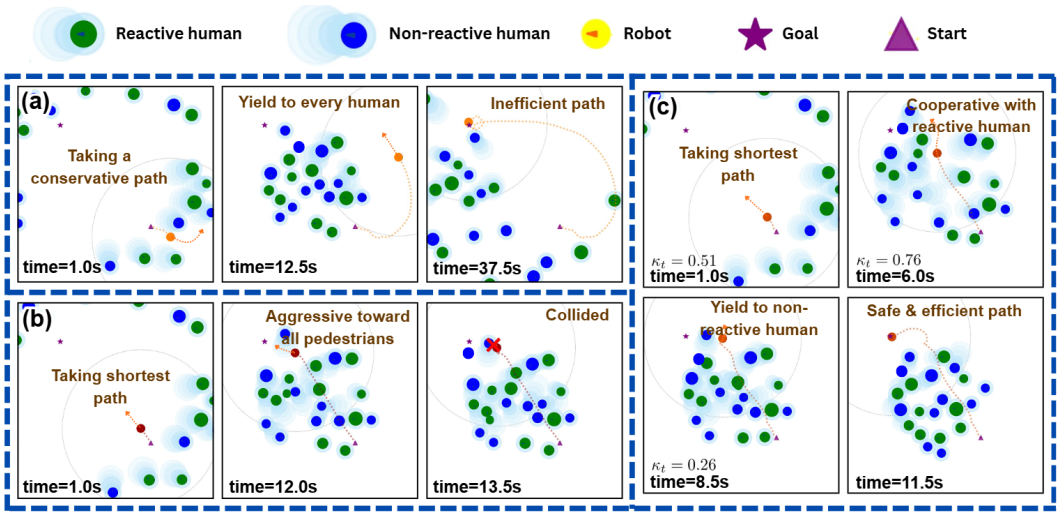

Figure 3: **Representative mixed-crowd trajectories,** comparing GenSafeNav (Cons.), GenSafeNav (Naive), and RCRA. (a) The conservative policy treats all pedestrians as non-reactive and takes a long detour. (b) The naive policy pursues a shorter route but fails to account for non-reactive pedestrians, leading to collision. (c) RCRA adapts to local robot-directed reactivity, yielding near non-reactive pedestrians while moving more assertively near reactive pedestrians.

without online gradient updates. The remaining baselines illustrate the importance of this conditioning. ORCA and BRNE achieve lower SR in Non-reactive crowds, consistent with their interaction models anticipating collision-avoidance responses from pedestrians. CrowdNav++ lacks an explicit safety-cost constraint and reaches a worst-case SR of 79.20%. GenSafeNav (Rand.) broadens its training distribution across all five interaction scenarios but does not explicitly estimate pedestrian reactivity; it achieves 91.92% SR in Non-reactive and a longer path in Reactive (13.92 m versus 11.01 m for RCRA), despite comparable SR. Together, these results support the usefulness of explicit reactivity conditioning for adjusting navigation behavior.

Fig. 3 illustrates how RCRA adapts its navigation behavior within a mixed scenario. The conservative policy in Fig. 3(a) remains safe but detours around the entire crowd, while the naive policy in Fig. 3(b) takes a direct route but collides with a non-reactive pedestrian. In contrast, RCRA in Fig. 3(c) adjusts its behavior online through the adaptation scale κ_t. Starting from an intermediate value ($\kappa_t = 0.51$), the scale increases to 0.76 when reactive pedestrians become locally relevant, allowing the robot to take a more direct path, and subsequently decreases to 0.26 as a non-reactive pedestrian becomes critical, shifting the policy toward the conservative backbone, producing a safe and efficient trajectory within the same episode. Appendix F evaluates the same online reactivity-conditioned adaptation using prescribed ground-truth reactivity labels in real-robot experiments. RCRA achieves 20 successful runs out of 20 trials, compared with 16 for GenSafeNav (Naive), while reducing NT by 15.4% and PL by 26.5% relative to GenSafeNav (Cons.).

5 Discussion

Our results support explicit pedestrian-reactivity estimation as a useful signal for modulating a pre-trained navigation policy under heterogeneous and changing local interactions. RCRA spatially aggregates these estimates to control a learned residual around a frozen conservative backbone. With navigation components held fixed, learned conditioning reduces collision rate and path length over BDDS conditioning in heterogeneous and time-varying crowds. **Limitations.** The reactivity estimator relies on binary supervision and motion history, which may miss ambiguous real-world reactivity or noise-induced motion changes. Moreover, the simulator represents reactivity through binary response modes, and the real-robot experiments use prescribed reactivity labels to isolate the adaptation mechanism. Future work will evaluate the learned estimator in closed-loop deployment and incorporate visual cues such as body orientation and gaze.

AI USE STATEMENT

We used generative AI tools to assist with language editing, manuscript organization, literature search, code development and debugging. The authors reviewed and revised AI-assisted text for technical accuracy and consistency with the implemented methods and reported results, inspected and tested AI-assisted code, and verified references against original sources. The authors take full responsibility for the final content, including text, claims, code, figures, and reported results.

REFERENCES

Subham Agrawal, Nico Ostermann-Myrau, Nils Dengler, and Maren Bennewitz. PeRoI: A pedestrian-robot interaction dataset for learning avoidance, neutrality, and attraction behaviors in social navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://arxiv.org/abs/2503.16481.

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*, pp. 961–971, 2016. doi: 10.1109/CVPR.2016.110.

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468, 2016.

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*, 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*, pp. 1343–1350, 2017. doi: 10.1109/IROS.2017.8202312.

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*, pp. 3052–3059, 2018. doi: 10.1109/IROS.2018.8593871.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, 2017.

Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998. doi: 10.1177/027836499801700706.

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

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

Mehdi Hosseinzadeh, Bruno Sinopoli, and Aaron F Bobick. Toward safe and efficient human–robot interaction via behavior-driven danger signaling. *IEEE Transactions on Control Systems Technology*, 32(1):214–224, 2023.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Zhe Huang, Ruohua Li, Kazuki Shin, and Katherine Driggs-Campbell. Learning sparse interaction graphs of partially detected pedestrians for trajectory prediction. *IEEE Robotics and Automation Letters*, 7(2):1198–1205, 2021.

Zhe Huang, Tianchen Ji, Heling Zhang, Fatemeh Cheraghi Pouria, Katherine Driggs-Campbell, and Roy Dong. Interaction-aware conformal prediction for crowd navigation. In *International Workshop on the Algorithmic Foundations of Robotics*, pp. 377–395. Springer, 2024.

Dan Jia, Alexander Hermans, and Bastian Leibe. DR-SPAAM: A spatial-attention and auto-regressive model for person detection in 2D range data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10270–10277. IEEE, 2020.

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*, pp. 3517–3524, 2021.

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*, pp. 12015–12021, 2023.

Zuxin Liu, Jesse Zhang, Kavosh Asadi, Yao Liu, Ding Zhao, Shoham Sabach, and Rasool Fakoor. Tail: Task-specific adapters for imitation learning with large pretrained models. In *International Conference on Learning Representations*, volume 2024, pp. 16330–16353, 2024.

Max Muchen Sun, Francesca Baldini, Katie Hughes, Peter Trautman, and Todd Murphey. Mixed strategy nash equilibrium for crowd navigation. *The International Journal of Robotics Research*, 44(7):1156–1185, 2025.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation*, pp. 3803–3810, 2018. doi: 10.1109/ICRA.2018.8460528.

Yanyuan Qiao, Zheng Yu, and Qi Wu. Vln-petl: Parameter-efficient transfer learning for vision-and-language navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15443–15452, 2023.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, 2011.

Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M. Kitani, Dariu M. Gavrila, and Kai O. Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020. doi: 10.1177/0278364920917446.

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

Tim Salzmann, Hao-Tien Lewis Chiang, Markus Ryll, Dorsa Sadigh, Carolina Parada, and Alex Bewley. Robots that can see: Leveraging human pose for trajectory prediction. *IEEE Robotics and Automation Letters*, 8(11):7090–7097, 2023.

Thomas Schmied, Markus Hofmarcher, Fabian Paischer, Razvan Pascanu, and Sepp Hochreiter. Learning to modulate pre-trained models in RL. *Advances in Neural Information Processing Systems*, 36:38231–38265, 2023.

Ao Shen, Kaixi Chen, Shiwei Liu, Fang Deng, and Chen Chen. Human-human & human-robot interaction transformer (h2int) for robot navigation in dense and uncertain crowds. *arXiv preprint arXiv:2609.05300*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 23–30, 2017. doi: 10.1109/IROS.2017.8202133.

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

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.

Jur van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics Research*, volume 70 of *Springer Tracts in Advanced Robotics*, pp. 3–19. Springer, 2011. doi: 10.1007/978-3-642-19457-3_1.

Mengdi Xu, Yuchen Lu, Yikang Shen, Shun Zhang, Ding Zhao, and Chuang Gan. Hyper-decision transformer for efficient online policy adaptation. *arXiv preprint arXiv:2304.08487*, 2023.

Jianpeng Yao, Xiaopan Zhang, Yu Xia, Zejin Wang, Amit K Roy-Chowdhury, and Jiachen Li. Sonic: Safe social navigation with adaptive conformal inference and constrained reinforcement learning. *arXiv preprint arXiv:2407.17460*, 2024.

Jianpeng Yao, Xiaopan Zhang, Yu Xia, Zejin Wang, Amit Roy-Chowdhury, and Jiachen Li. Towards generalizable safety in crowd navigation via conformal uncertainty handling. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4206–4225, 2025.

A Evaluation Metrics

Each episode terminates in exactly one of three outcomes: success, collision, or timeout. Let N_{total}, N_{success}, and $N_{\text{collision}}$ denote the corresponding total and outcome counts. We report:

- **Success Rate (SR):** percentage of episodes that reach the goal before collision or timeout, $\text{SR} = 100\%\, N_{\text{success}}/N_{\text{total}}$.
- **Collision Rate (CR):** percentage of episodes terminated by collision, $\text{CR} = 100\%\, N_{\text{collision}}/N_{\text{total}}$. The timeout percentage is $\text{TR} = 100\% - \text{SR} - \text{CR}$.
- **Navigation Time (NT):** mean duration in seconds over successful episodes only, $\text{NT} = N_{\text{success}}^{-1} \sum_{k=1}^{N_{\text{success}}} T_k$, where T_k is the completion time of successful episode k.
- **Path Length (PL):** mean traveled distance in meters over successful episodes only, $\text{PL} = N_{\text{success}}^{-1} \sum_{k=1}^{N_{\text{success}}} L_k$, where L_k is the total distance traveled by the robot until reaching the goal in successful episode k.

We compute scenario-averaged metrics as unweighted means across the five scenarios; percentage reductions compare these means. NT and PL are computed over successful episodes, whose composition may differ across methods, and should therefore be interpreted alongside SR and CR.

B Implementation Details

Base architecture and simulator. We build on the GenSafeNav (Yao et al., 2025) crowd-navigation framework, which uses GST trajectory prediction, prediction-uncertainty features, human–human attention, human–robot attention, temporal fusion, and actor–critic layers. We retain GenSafeNav's reward and safety-cost formulations and PPO-Lagrangian training settings, with a cost limit of 0.4. We use a prediction horizon of $K = 5$ timesteps, a spatial weighting radius of $R = 5\,\text{m}$, and $\epsilon = 10^{-6}$ in the adaptation scale.

Three-stage training pipeline. Our framework is trained in three stages. First, we initialize the conservative backbone from a pretrained GenSafeNav checkpoint trained for 2×10^7 environment steps and freeze all backbone parameters. Second, we fine-tune only the LoRA residual modules on reactive-crowd rollouts for 2×10^7 environment steps, using the same PPO-Lagrangian hyperparameters as GenSafeNav. During this stage, the residual branch is fully activated with $\kappa_t = 1$. Third, we train the reactivity estimator using simulator-provided reactivity labels and refine it with DAgger. All simulation experiments are run on an NVIDIA RTX PRO 6000 Blackwell GPU.

LoRA adaptation. LoRA modules are inserted into the human–robot cross-attention feature projections and the linear layers of the actor and critic MLPs, while the remaining GenSafeNav backbone remains frozen. The LoRA adapters use rank $r = 1$ and $\alpha = 128$. The fixed gain α/r is absorbed into the factors in our weight-update notation. The backbone contains approximately 2.5M parameters, while LoRA introduces approximately 3.5K trainable parameters, equivalent to 0.14% of the backbone. At deployment, the effective adapted weights are controlled by the online scale κ_t, allowing the policy to interpolate between the frozen conservative backbone and the learned reactive-crowd residual.

C Reactivity Estimation: Baseline and Evaluation Details

Our reactivity estimator is inspired by the Human Scene Transformer (HST) (Salzmann et al., 2023), but adapts its architecture to reactivity classification rather than trajectory forecasting. We remove the visual cues used by HST and replace the original scene context with the robot state. The estimator takes a short history of human motion together with the robot state. Feature-level attention first encodes human motion features, followed by per-agent temporal alignment and joint self-attention over agent–time tokens to capture human–human and human–robot interactions. We use four attention heads, six latent modes, dropout 0.1, and a learned temporal embedding. The hidden dimension is 128 in simulation and 32 for PeRoI, with a feed-forward dimension of 128 in both settings. The representation at each pedestrian's last valid observation is used for classification; latent-mode

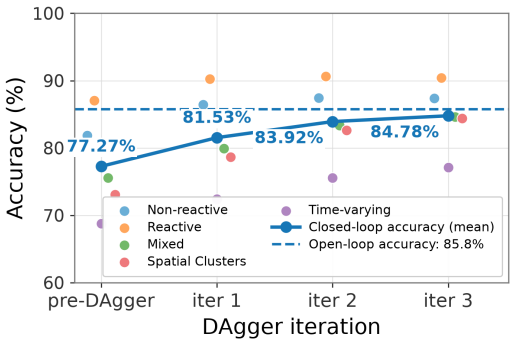

Figure 4: **DAgger refinement.** Reactivity estimation accuracy across aggregation iterations.

Table 3: **Reactivity estimation accuracy.** BDDS (Hosseinzadeh et al., 2023) versus our learned estimator. FP and FN are percentages of all evaluated pedestrian–timestep pairs.

Scenario	**BDDS** Acc./FP/FN (%)	**Ours** Acc./FP/FN (%)
Non-reactive	72.6/27.4/0.0	**87.4**/12.6/0.0
Reactive	72.5/0.0/27.5	**90.4**/0.0/9.6
Mixed	72.8/12.1/15.1	**84.6**/7.0/8.4
Spatial Clusters	72.1/12.4/15.5	**84.4**/7.1/8.5
Time-varying	70.8/13.3/15.9	**77.1**/10.4/12.5
Mean	72.16/13.04/14.80	**84.78**/7.42/7.80

representations are averaged before a LayerNorm–MLP classification head produces one logit per pedestrian. A sigmoid maps this logit to the continuous reactivity belief $\hat{b}_{i,t} \in [0, 1]$. The estimator uses an approximately 2 s observation history in both domains: $H = 8$ samples in simulation at $\Delta t = 0.25$ s and $H = 30$ samples for PeRoI at $\Delta t \approx 0.0667$ s. We train for 30 epochs with class-weighted binary cross-entropy, AdamW with learning rate 3×10^{-4} and weight decay 10^{-4}, a cosine learning-rate schedule, batch size 32, and gradient clipping at 1.0.

Evaluation metrics. For classification, both estimators predict reactive when their continuous score exceeds 0.5 and non-reactive otherwise. True positives (TP) and true negatives (TN) denote correctly classified reactive and non-reactive samples, respectively. False positives (FP) are non-reactive samples incorrectly classified as reactive, while false negatives (FN) are reactive samples incorrectly classified as non-reactive. Simulation accuracy is the percentage of correctly classified pedestrian–timestep pairs, averaged equally across the five scenarios. For class-imbalanced PeRoI evaluation and BDDS parameter selection, we use balanced accuracy, $50\%\,[\mathrm{TP}/(\mathrm{TP}+\mathrm{FN}) + \mathrm{TN}/(\mathrm{TN}+\mathrm{FP})]$. The classification threshold is used only for reporting and does not enter the continuous navigation scale in Eq. (6).

Behavior-Driven Danger Signaling (BDDS). We adapt the Bayesian estimator of Hosseinzadeh et al. (2023), which updates a belief over whether pedestrian actions minimize a goal cost alone or a combined goal and robot-safety cost. We evaluate its passive inference component without emitting danger signals. We denote its posterior score by $\hat{b}^{\mathrm{BDDS}}_{i,t}$, using our belief notation. This is the probability of the model's robot-safety-responsive hypothesis given the observed human–robot motion history, and serves as a proxy for our reactivity label. We extend the paper's one-dimensional example to planar kinematics, $p^h_{i,t+1} = p^h_{i,t} + u^h_{i,t}\Delta t$, where $u^h_{i,t}$ is a candidate planar velocity and Δt is the sampling interval. The action grid contains a stop action and speeds $\{0.5, 1, 1.5, 2\}v_H$ at seven headings from $-90°$ to $90°$ in $30°$ increments relative to the current pedestrian heading, giving 29 actions. Here, v_H is the reference pedestrian speed. Observed velocity is assigned to the nearest action, and costs are evaluated at the candidate next position. We use $(v_H, \Delta t) = (1.0\,\mathrm{m/s}, 0.25\,\mathrm{s})$ in simulation and $(1.41\,\mathrm{m/s}, 0.0667\,\mathrm{s})$ on PeRoI, retaining the published cost weights and random-action mixture weight $\omega_H = 0.1$. We select the Boltzmann sharpness γ by maximizing balanced accuracy on the training split and fix it for held-out evaluation: $\gamma^*_{\mathrm{sim}} = 1.44$ and $\gamma^*_{\mathrm{PeRoI}} = 0.30$.

Simulation data splits and DAgger protocol. Initial estimator training, DAgger data collection, and final evaluation use disjoint episode sets. The offline accuracy reference is computed over 250 evaluation episodes per scenario. At each DAgger iteration, we collect 250 episodes per scenario using the current adaptive policy, label the visited observations with simulator ground truth, and aggregate them with the existing training data to update the estimator. Final estimator and navigation results are reported on held-out episodes excluded from training and DAgger collection. Fig. 4 shows that the pre-DAgger estimator reaches only 77.27% scenario-averaged closed-loop accuracy, indicating a substantial distribution mismatch. After three DAgger iterations, accuracy improves to

84.78%, a gain of 7.51 percentage points, approaching the offline evaluation reference of 85.80%. Table 3 reports the corresponding scenario-wise accuracy and error breakdown.

Real-Human Trajectory Evaluation. PeRoI (Pedestrian-Robot Interaction) contains recorded pedestrian trajectories from two outdoor sites, annotated as avoidance, neutrality, or attraction under absent, stationary, and moving-robot conditions (Agrawal et al., 2026). We use the moving-robot recording and map avoidance to reactive ($z_{i,t} = 1$) and neutrality to non-reactive ($z_{i,t} = 0$) throughout each annotated track. These track-level behavior labels provide a proxy for reactivity rather than timestep-specific causal labels. Although attraction is a robot-directed response, approaching the robot differs from the yielding or avoidance behavior targeted by our adaptation signal. We therefore exclude the 18 attraction tracks (approximately 8.0% of 226 labeled tracks), leaving 208 tracks: 149 neutral and 59 avoidance.

We use a stratified 70/30 train–validation split by pedestrian track, keeping all windows from each track in the same split. Velocities are derived from timestamps and positions, and tracking gaps split the sequences. Because the dataset does not provide intended pedestrian goals, BDDS uses each contiguous track's final observed position as a goal proxy, giving it access to future endpoint information in this offline evaluation. Both estimators are evaluated at the same window endpoints against the track-level annotations, using the 0.5 classification threshold. The validation split is excluded from estimator training and BDDS γ selection.

D Adaptability and the Safety–Efficiency Trade-off

We examine how the policy adjusts its navigation behavior to robot-directed reactivity by comparing fixed LoRA scales and several adaptive mechanisms. Table 4 provides the fixed-scale LoRA sweep together with adaptive LoRA, gated full fine-tuning, action-space interpolation, and adaptive MPC. A fixed global scale κ remains constant across pedestrians, timesteps, and scenarios. The adaptive variants use the runtime adaptation scale under ground-truth reactivity labels, isolating the adaptation mechanism from estimation errors. Fig. 2 summarizes the scenario-averaged trade-offs: $\kappa = 0$ uses the conservative backbone and $\kappa = 1$ fully activates the residual, with intermediate scales tracing an empirically monotonic decrease in average path length and SR.

Fixed-scale sweep. Holding $\kappa_t = 0$ throughout a rollout disables the residual and uses the GenSafeNav (Cons.) policy, with the same recurrent-state initialization. Increasing the fixed scale progressively activates the residual learned from reactive-crowd rollouts, reducing NT and PL in every scenario. Because the backbone remains frozen, $\kappa_t = 1$ denotes the fully activated low-rank residual on the conservative backbone; it is distinct from the separately trained GenSafeNav (Naive) policy. The sweep reveals that the safety cost of more efficient behavior depends strongly on pedestrian reactivity. In Non-reactive, increasing the scale from 0 to 1 reduces NT from 16.59 to 12.35 s and PL from 16.52 to 12.33 m, but SR drops from 97.20% to 73.12%. In Reactive, the same change reduces NT from 16.24 to 10.87 s and PL from 16.19 to 10.87 m, while SR changes only from 100.00% to 99.20%. Thus, a scale that efficiently exploits reactive pedestrian responses can be unsafe when pedestrians do not respond to the robot.

Online reactivity-conditioned adaptation. In Non-reactive, the adaptive policy remains close to the conservative operating point ($\kappa_t = 0$), with nearly unchanged SR (97.04% versus 97.20%), while reducing NT and PL by 3.1%. In Reactive, it matches the fully activated residual ($\kappa_t = 1$), achieving the same 99.20% SR, 10.87 s NT, and 10.87 m PL. In Mixed, the adaptive policy instead operates between these two extremes, maintaining 96.72% SR while reducing NT by 16.8% and PL by 16.7% relative to $\kappa_t = 0$; in contrast, always using $\kappa_t = 1$ reduces SR to 86.08%. In Time-varying, oracle adaptation achieves 96.88% SR with 13.7 m PL, compared with 88.24% SR and 11.8 m PL at fixed $\kappa = 1$, retaining substantially higher success while still improving efficiency over the conservative endpoint. Together, these results show that RCRA adapts the residual strength to the local interaction regime.

Adaptive MPC. To evaluate whether reactivity conditioning extends beyond learned policies, we use adaptation κ_t to modulate MPC's clearance around predicted pedestrian trajectories:

$$d^{\min}_{i,t+k|t} = \hat{\delta}_{i,t+k|t} + 0.5\,\mathrm{m}\,(1 - \kappa_t), \tag{7}$$

where $\hat{\delta}_{i,t+k|t}$ is the prediction-uncertainty radius. Fig. 2 shows that MPC also exhibits a monotonic safety–efficiency trade-off, with increasing κ reducing both PL and SR. Table 4 shows MPC's adaptive operating point is less safe than the learned adaptive variants, indicating that the underlying planner remains important. Nevertheless, the result shows that reactivity-conditioned adaptation is not limited to the RL policy backbone and can also be incorporated into a model-based planner by modulating its safety constraints online.

E SCALABILITY

We compare the computational scalability of three implementations of the same reactivity-conditioned adaptation rule: LoRA, gated full fine-tuning, and action-space interpolation. LoRA is particularly attractive because it preserves the conservative backbone and expresses adaptation through a small gated residual, which can better retain the pretrained policy's behavior than full-parameter adaptation (Schmied et al., 2023). Gated full fine-tuning trains all policy parameters from the same conservative initialization on reactive-crowd rollouts and scales the resulting parameter residual:

Table 4: **Adaptation mechanism ablations.** Fixed residual scales expose a safety–efficiency trade-off; adaptive LoRA, full fine-tuning, action-space interpolation, and MPC use ground-truth reactivity to isolate online conditioning.

Scenario	Setting	SR (%) ↑	CR (%) ↓	NT (s) ↓	PL (m) ↓
Non-reactive	LoRA ($\kappa_t = 0.2$)	<u>93.84</u> ± 0.78	<u>6.16</u> ± 0.78	15.01 ± 0.34	14.94 ± 0.34
	LoRA ($\kappa_t = 0.6$)	83.12 ± 1.78	16.88 ± 1.78	<u>13.25</u> ± 0.25	<u>13.23</u> ± 0.25
	LoRA ($\kappa_t = 1.0$)	73.12 ± 3.99	26.88 ± 3.99	**12.35** ± 0.25	**12.33** ± 0.25
	LoRA (Adaptive)	**97.04** ± 0.36	**2.96** ± 0.36	16.08 ± 0.41	16.01 ± 0.44
	Full fine-tuning (Adaptive)	*97.04 ± 0.61*	*2.96 ± 0.61*	*16.01 ± 0.45*	*15.93 ± 0.45*
	Action-space (Adaptive)	*97.04 ± 0.46*	*2.96 ± 0.46*	*16.00 ± 0.47*	*15.93 ± 0.46*
	MPC (Adaptive)	*69.44 ± 2.31*	*30.56 ± 2.31*	*14.81 ± 0.38*	*14.22 ± 0.21*
Reactive	LoRA ($\kappa_t = 0.2$)	**99.92** ± 0.18	**0.08** ± 0.18	14.34 ± 0.43	14.28 ± 0.42
	LoRA ($\kappa_t = 0.6$)	<u>99.68</u> ± 0.33	<u>0.32</u> ± 0.33	<u>11.73</u> ± 0.27	<u>11.71</u> ± 0.27
	LoRA ($\kappa_t = 1.0$)	99.20 ± 0.28	0.80 ± 0.28	**10.87** ± 0.11	**10.87** ± 0.11
	LoRA (Adaptive)	99.20 ± 0.28	0.80 ± 0.28	**10.87** ± 0.11	**10.87** ± 0.11
	Full fine-tuning (Adaptive)	*99.20 ± 0.40*	*0.80 ± 0.40*	*10.70 ± 0.08*	*10.70 ± 0.08*
	Action-space (Adaptive)	*99.20 ± 0.28*	*0.80 ± 0.28*	*10.92 ± 0.12*	*10.92 ± 0.12*
	MPC (Adaptive)	*100.00 ± 0.00*	*0.00 ± 0.00*	*12.75 ± 0.06*	*12.40 ± 0.06*
Mixed	LoRA ($\kappa_t = 0.2$)	**98.00** ± 0.00	**2.00** ± 0.00	14.88 ± 0.28	14.81 ± 0.28
	LoRA ($\kappa_t = 0.6$)	94.16 ± 1.15	5.84 ± 1.15	<u>12.66</u> ± 0.27	<u>12.64</u> ± 0.27
	LoRA ($\kappa_t = 1.0$)	86.08 ± 2.79	13.92 ± 2.79	**11.85** ± 0.18	**11.84** ± 0.18
	LoRA (Adaptive)	<u>96.72</u> ± 1.21	<u>3.28</u> ± 1.21	13.82 ± 0.37	13.78 ± 0.38
	Full fine-tuning (Adaptive)	*92.80 ± 1.20*	*7.20 ± 1.20*	*12.97 ± 0.25*	*12.93 ± 0.24*
	Action-space (Adaptive)	*96.56 ± 1.22*	*3.44 ± 1.22*	*13.56 ± 0.32*	*13.51 ± 0.32*
	MPC (Adaptive)	*81.36 ± 1.59*	*18.64 ± 1.59*	*13.09 ± 0.08*	*12.69 ± 0.04*
Spatial Clusters	LoRA ($\kappa_t = 0.2$)	**96.00** ± 0.40	**4.00** ± 0.40	13.76 ± 0.40	13.71 ± 0.40
	LoRA ($\kappa_t = 0.6$)	92.32 ± 1.73	7.68 ± 1.73	<u>12.18</u> ± 0.31	<u>12.17</u> ± 0.30
	LoRA ($\kappa_t = 1.0$)	88.80 ± 1.85	11.20 ± 1.85	**11.61** ± 0.24	**11.60** ± 0.24
	LoRA (Adaptive)	<u>95.68</u> ± 0.87	<u>4.32</u> ± 0.87	13.02 ± 0.30	13.00 ± 0.29
	Full fine-tuning (Adaptive)	*94.56 ± 1.37*	*5.44 ± 1.37*	*12.53 ± 0.27*	*12.50 ± 0.27*
	Action-space (Adaptive)	*95.84 ± 1.28*	*4.16 ± 1.28*	*12.98 ± 0.28*	*12.94 ± 0.28*
	MPC (Adaptive)	*86.16 ± 0.67*	*13.84 ± 0.67*	*12.94 ± 0.16*	*12.73 ± 0.10*
Time-varying	LoRA ($\kappa_t = 0.2$)	**98.64** ± 0.73	**1.36** ± 0.73	14.85 ± 0.17	14.78 ± 0.16
	LoRA ($\kappa_t = 0.6$)	93.36 ± 1.15	6.64 ± 1.15	<u>12.78</u> ± 0.32	<u>12.76</u> ± 0.31
	LoRA ($\kappa_t = 1.0$)	88.24 ± 1.19	11.76 ± 1.19	**11.81** ± 0.22	**11.80** ± 0.22
	LoRA (Adaptive)	<u>96.88</u> ± 0.95	<u>3.12</u> ± 0.95	13.78 ± 0.35	13.71 ± 0.31
	Full fine-tuning (Adaptive)	*92.40 ± 1.26*	*7.60 ± 1.26*	*13.02 ± 0.14*	*12.98 ± 0.14*
	Action-space (Adaptive)	*96.72 ± 1.21*	*3.28 ± 1.21*	*13.74 ± 0.36*	*13.69 ± 0.36*
	MPC (Adaptive)	*86.72 ± 1.04*	*13.28 ± 1.04*	*12.90 ± 0.06*	*12.62 ± 0.04*

Note: Performance rankings (1st** and <u>2nd</u>) are calculated only among the LoRA variants.*

Table 5: **Computational scalability.** FT denotes full fine-tuning. P95 and memory refer to policy inference latency and peak inference memory. Action interpolation shares the trained endpoints.

Backbone	Method	Train (h)	P95 (ms)	Memory (MB)
2.5M	LoRA	6.30	1.90	47.2
	Gated FT	6.44	2.20	56.6
	Action-interpolation	Shared	3.06	89.7
25M	LoRA	6.80	1.97	287
	Gated FT	7.58	2.23	398
	Action-interpolation	Shared	3.03	510
250M	LoRA	9.26	2.52	2774
	Gated FT	15.92	3.20	3784
	Action-interpolation	Shared	4.05	4688
500M	LoRA	9.32	3.18	5614
	Gated FT	22.52	4.16	7641
	Action-interpolation	Shared	5.63	9459

$$\Delta\theta_{\mathrm{FT}} = \theta_{\mathrm{FT}} - \theta_0, \qquad \theta_t^{\mathrm{FT}} = \theta_0 + \kappa_t \Delta\theta_{\mathrm{FT}}. \tag{8}$$

Action-space interpolation evaluates the conservative and fully activated LoRA endpoint policies on the same observation and combines their velocity commands, denoted $a_t^{(0)}$ and $a_t^{(1)}$, respectively:

$$a_t^{\mathrm{interp}} = (1 - \kappa_t) a_t^{(0)} + \kappa_t a_t^{(1)}. \tag{9}$$

Scaling setup. We scale the backbone by increasing the hidden-layer width, yielding 2.5M, 25M, 250M, and 500M parameter models. Each model is trained for 2×10^7 environment steps on an NVIDIA RTX PRO 6000 Blackwell GPU. LoRA trains 0.14%, 0.085%, 0.030%, and 0.021% of the backbone parameter count, respectively, whereas full fine-tuning updates all backbone parameters. Action-space interpolation reuses the trained LoRA endpoint policies and requires two policy evaluations per control step, executed in parallel in the measured implementation. P95 denotes the 95th percentile of policy inference latency. The latency and peak inference memory measurements concern the policy implementation, excluding perception, trajectory prediction, and reactivity estimation.

Scaling results. Table 5 shows that LoRA and full fine-tuning have similar training times at 2.5M parameters (6.30 versus 6.44 h), while their costs diverge as the backbone grows. At 500M, LoRA requires 9.32 h compared with 22.52 h for full fine-tuning, corresponding to a $2.42\times$ training speedup. Its P95 inference latency is 3.18 ms versus 4.16 ms, and peak inference memory is 5,614 MB versus 7,641 MB. Action-space interpolation requires 5.63 ms and 9,459 MB at the same scale. For a square layer of width d, a full parameter residual contains d^2 parameters, whereas rank-r factors contain $2dr$; at fixed rank, their relative parameter cost decreases with width. Together, these measurements show the computational advantage of the low-rank implementation for larger backbones. These are implementation-specific resource measurements. Frozen-backbone computation, rollout collection, and residual evaluation still incur costs, so trainable-parameter savings do not imply proportional speedups. Fig. 2 shows similar safety–efficiency trends for LoRA and gated full fine-tuning at 2.5M parameters under ground-truth reactivity conditioning, although their operating points differ. The larger backbones exhibit navigation performance similar to their respective 2.5M counterparts; these results are omitted for brevity. These results suggest that the behavioral benefit of reactivity-conditioned adaptation is not specific to LoRA.

F Controlled Real-Robot Validation

We deploy the navigation stack on a Clearpath Jackal running ROS2 with an onboard NVIDIA RTX 4000 GPU. Pedestrians are detected from 10 Hz Velodyne LiDAR scans using DR-SPAAM (Jia et al., 2020), tracks are maintained using SORT (Bewley et al., 2016), and future trajectories are

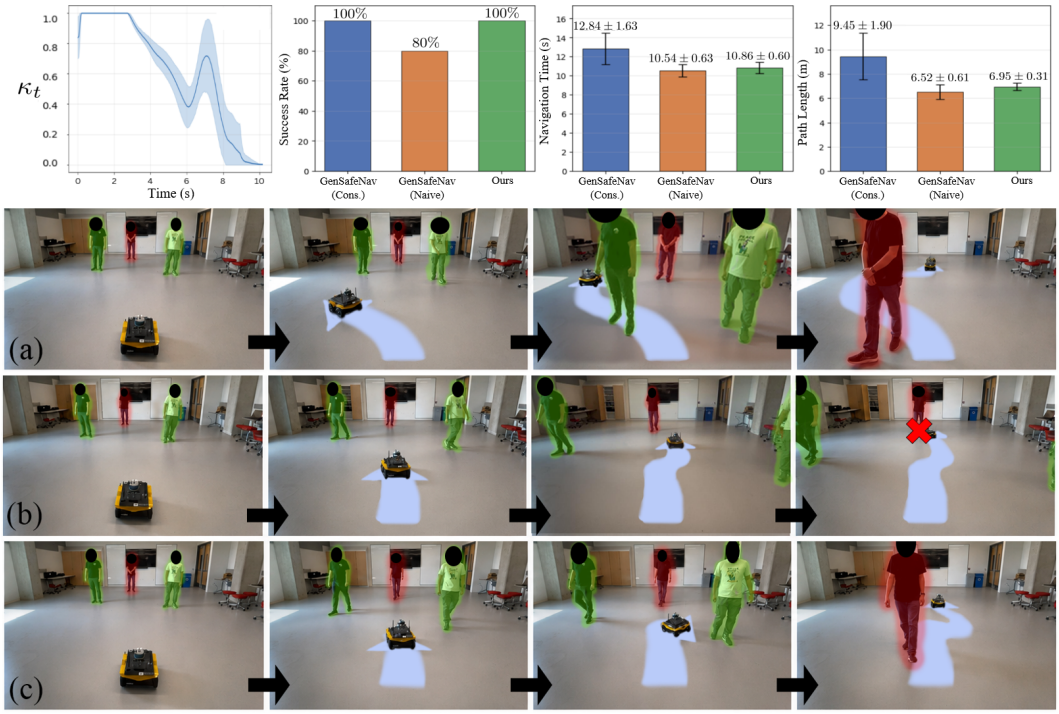

Figure 5: **Controlled real-robot validation using prescribed reactivity labels.** Top: adaptation scale κ_t and performance over 20 trials. Bottom: representative executions of (a) GenSafeNav (Cons.), (b) GenSafeNav (Naive), and (c) RCRA, illustrating a conservative detour, a collision, and locally adapted avoidance, respectively.

predicted using GST (Huang et al., 2021). The real-world experiment uses four scripted pedestrians over 20 trials per policy: two non-reactive pedestrians and two reactive pedestrians. These labels are defined by instructions to maintain nominal motion or respond to the robot, respectively, while walking at approximately 1 m/s. We use the prescribed labels directly to isolate the adaptation mechanism from reactivity-estimation errors under noisy real-world observations. The robot is tasked with navigating through the crowd toward a target located 5 m in front of its initial position. We compare three deployed policies: GenSafeNav (Cons.), GenSafeNav (Naive), and RCRA.

Fig. 5 shows that GenSafeNav (Cons.) takes a long detour, whereas GenSafeNav (Naive) follows a direct route but collides with the non-reactive pedestrian. RCRA yields near that pedestrian while using the gap created by reactive pedestrians. It achieves 20 successful runs out of 20 trials, compared with 16 for GenSafeNav (Naive), and reduces NT by 15.4% and PL by 26.5% relative to GenSafeNav (Cons.). Initially, $\kappa_t \approx 1$ activates the reactive-crowd residual; as the non-reactive pedestrian becomes relevant, the scale decreases toward zero, shifting behavior toward the conservative backbone. These trials validate adaptation with prescribed labels; PeRoI separately evaluates learned reactivity estimation (Appendix C). Future work may combine motion history with body orientation and gaze before testing learned estimation in closed-loop robot navigation. Additional experimental details and videos are provided in the supplementary material.