From Crowd Motion Prediction to Robot Navigation in Crowds

Sriyash Poddar[1], Christoforos Mavrogiannis[2], Siddhartha S. Srinivasa[2]

***Abstract*— We focus on robot navigation in crowded environments. To navigate safely and efficiently within crowds, robots need models for crowd motion prediction. Building such models is hard due to the high dimensionality of multiagent domains and the challenge of collecting or simulating interaction-rich crowd-robot demonstrations. While there has been important progress on models for offline pedestrian motion forecasting, transferring their performance on real robots is nontrivial due to close interaction settings and novelty effects on users. In this paper, we investigate the utility of a recent state-of-the-art motion prediction model (S-GAN) for crowd navigation tasks. We incorporate this model into a model predictive controller (MPC) and deploy it on a self-balancing robot which we subject to a diverse range of crowd behaviors in the lab. We demonstrate that while S-GAN motion prediction accuracy transfers to the real world, its value is not reflected on navigation performance, measured with respect to safety and efficiency; in fact, the MPC performs indistinguishably even when using a simple constant-velocity prediction model, suggesting that substantial model improvements might be needed to yield significant gains for crowd navigation tasks. Footage from our experiments can be found at https://youtu.be/mzFiXg8KsZ0.**

I. Introduction

Large-scale deep learning methods [5, 9, 14, 28, 32, 34, 45] have dramatically improved the state-of-the-art in prediction accuracy across standard benchmarks [20, 29]. While these models have been the foundational in recent real-world robot demonstrations [4, 6, 7, 10, 16, 22], scaling their performance to complex environments like pedestrian domains, warehouses, or hospitals is challenging as these environments feature close interaction settings, large space of behavior, and limited rules.

To address these challenges, many approaches involve training deep-learning models on simulated crowd-robot interactions [6, 7, 10, 22]. While typical crowd simulators [15, 40] produce realistic behaviors, some of their core assumptions limit their relevance to crowd-navigation tasks. For instance, Fraichard and Levesy [12] showed that the assumptions of omniscience and homogeneity of existing crowd simulators give rise to behaviors that would be unsafe to execute on a real robot. Further, Mavrogiannis et al. [26] showed that a non-reactive, non-collision-avoiding agent is safer than ORCA-simulated agents in an ORCA-simulated world [40] due to the overly submissive behaviors this model may exhibit.

[1]Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, Kharagpur, India. Email: poddarsriyash@iitkgp.ac.in

[2]Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, USA. Email: {cmavro, siddh}@cs.washington.edu

This work was (partially) funded by the Honda Research Institute USA, the National Science Foundation NRI (#2132848) and CHS (#2007011), DARPA RACER (#HR0011-21-C-0171), the Office of Naval Research (#N00014-17-1-2617-P00004 and #2022-016-01 UW), and Amazon.

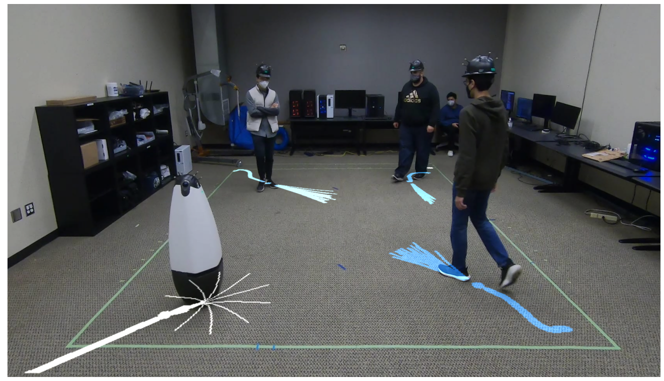


Fig. 1: Honda's experimental ballbot [17] navigates next to three users in our lab. Agents' past trajectories and distribution of future actions are shown. In this paper, we approach the question of how human motion prediction accuracy translates into robot navigation performance in crowded environments.

Other approaches use pedestrian datasets to train and validate models for crowd motion prediction [23, 37]. However, the pedestrian datasets most commonly used [20, 29] feature well-structured, goal-directed, and cooperative motion. These settings represent a very narrow subset of the behavior that a robot would encounter in the real world. This behavior is so prevalent in those datasets that according to Schöller et al. [36], even constant-velocity (CV) prediction, a very simple, analytical model, performs comparably to recent state-of-the-art (SOTA) deep models. Therefore, while the SOTA in human motion prediction keeps improving, it is unclear what its relevance is for robot navigation in crowds.

Inspired by these observations, we ask the question:

> *To what extent does crowd motion prediction accuracy translate to robot navigation performance in crowd navigation tasks?*

To approach this question, we investigate the transfer of a recent SOTA model (S-GAN [14]) from offline datasets to onboard performance and its implications on navigation performance. We integrate the S-GAN as a dynamics model into a standard MPC architecture and deploy it on a self-balancing robot, which we subject to a series of diverse crowd-robot interactions in the lab. Overall, we use a general framework for control in social navigation to evaluate the trajectory prediction models in crowds. We find that while the onboard prediction accuracy of S-GAN is superior to a simple CV baseline, the MPC navigation performance (measured in terms of safety and time efficiency) is indistinguishable, suggesting that substantial prediction model improvements may be needed to achieve improved navigation performance.

II. Related Work

We discuss related work from the human motion prediction and crowd navigation literature.

A. Crowd Motion Prediction

The goal of modeling interactions among crowds in pedestrian domains has motivated much recent work in human motion prediction [32]. Recent works have used a variety of architectures including recurrent [1, 8, 45, 46] and convolutional [27] neural networks, spatiotemporal graphs [21, 31, 34], state-refinement modules [35], explicit probability maps [23], normalizing flows [2], and Gaussian processes [37, 38]. Some state-of-the-art methods based on generative adversarial networks (GAN) [14, 33] are particularly applicable to motion prediction due to their ability to model multimodality and diversity in crowd navigation.

Inspired by the effectiveness of GAN-based approaches, we build our crowd motion prediction architecture around S-GANs [14]. While prior work has used S-GANs primarily for motion tracking on offline datasets [14] and simulated environments [42], in this work we deploy a S-GAN-based architecture on a real robot navigating under a variety of crowd conditions. Our implementation enables real-time performance capable of handling dynamic environments.

B. Crowd Navigation

In recent years, several crowd navigation algorithms have been deployed on real robots [26]. Some approaches incorporate explicit models of human motion prediction into receding-horizon reactive controllers [4, 7, 18, 19, 25, 35, 37, 38, 42, 47]. Others learn end-to-end navigation policies using techniques like deep reinforcement learning [6, 7, 10, 22].

Our approach falls into the former category: similar to several recent works [4, 25, 42], we integrate a crowd motion prediction model into a MPC architecture. In our prior work, we showed that CV-based motion prediction can empower a MPC to outperform recent end-to-end approaches [25]. In this work, we explore the utility of the recent state-of-the-art architectures like S-GANs for crowd navigation tasks.

C. Benchmarking in Crowd Navigation

A challenge in crowd navigation research is benchmarking and validation [26]. Observing limitations of widely adopted practices reported in recent literature [12, 26, 36], some works have developed new simulation environments [3, 13, 39], real-world datasets [43], and experimental protocols [24, 25, 30] to improve the validation of future frameworks.

In this work, we also contribute towards these efforts by developing a series of benchmarking experiments designed to subject a navigation system to diverse crowd conditions. Unlike prior work, which typically focuses on navigation under cooperative, goal-directed settings, in this paper, we also develop benchmarking scenarios in non-cooperative settings, where humans are *aggressive* or *distracted* during navigation.

III. Problem Statement

We consider a workspace $\mathcal{W} \subseteq \mathbb{R}^2$ where a robot navigates among n human agents. We denote by $s \in \mathcal{W}$ the robot state and by $s^i \in \mathcal{W}$ the agent state $i \in \mathcal{N} = \{1, \dots n\}$. The robot is navigating from a state s_0 towards a goal state g whereas agent $i \in \mathcal{N} = \{1, \dots n\}$ is navigating from s_0^i towards a destination g^i. The robot is unaware of g^i but we assume that it is fully observing the world state $(s_t, s_t^{1:n})$ at every timestep t. Maintaining a history of states for all agents, the robot predicts their future trajectories using a model f. In this paper, our goal is to investigate whether the prediction accuracy of f translates to robot navigation performance. As a proxy for navigation performance, we consider metrics capturing safety and efficiency of robot motion.

IV. Human Motion Prediction

We treat human motion prediction as trajectory prediction over a horizon T given a past trajectory of horizon h.

A. Probabilistic Modeling

We denote by $s^i_{t-h:t} \in \mathcal{W}^h$ the partial trajectory of an agent $i \in \mathcal{N}$ of horizon h and by $s^i_{t:t+T} \in \mathcal{W}^T$ the future trajectory until time T. Consider a joint state prediction model $f : \mathcal{W}^{n\times h} \to \mathcal{W}^{n\times T}$, which takes as input the joint states of the agents $\boldsymbol{s}^{1:n}_{t-h:t}$ and predicts the future states $\hat{\boldsymbol{s}}^{1:n}$.

$$f\left(s^1_{t-h:t}, \dots s^n_{t-h:t}\right) = \left(\hat{s}^1_{t:t+T}, \dots \hat{s}^n_{t:t+T}\right) = \hat{\boldsymbol{s}}^{1:n}$$

We denote the distribution of future states for an agent $i \in \mathcal{N}$ as $p(\hat{s}^i_{t:t+T})$, and the joint distribution of states is represented as $p(\hat{\boldsymbol{s}}^{1:n})$. The prediction model $f : \mathcal{W}^{t\times n} \times \mathcal{W}^{T\times n} \to [0, 1]$ is a conditional distribution; denoting the distribution of the future trajectories given past trajectories of all the agents i.e f corresponds to $p(\hat{\boldsymbol{s}}^{1:n} | s^{1:n}_{t-h:t})$.

B. Probabilistic Trajectory Prediction using S-GAN

In this paper, we adopt a probabilistic trajectory prediction mechanism f using Social GAN (S-GAN), a state-of-the-art model from Gupta et al. [14]. A GAN consists of two neural networks: a generator $\mathcal{G}$ that estimates the data distribution and a discriminator $\mathcal{D}$ that classifies examples as real or fake (generated by $\mathcal{G}$). $\mathcal{D}$ and $\mathcal{G}$ are trained via a min-max game:

$$\begin{aligned}&\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{G}, \mathcal{D}) = \\ &\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log \mathcal{D}(x)] + \mathbb{E}_{z\sim p_{(z)}}[\log(1 - \mathcal{D}(\mathcal{G}(z)))]\end{aligned} \tag{1}$$

Given training data distribution p, and latent variable $z \sim \mathcal{N}(0, 1)$, $\mathcal{G}$ takes input z and outputs a sample in the training distribution i.e. $\mathcal{G}(z) \sim p$. This formulation can be extended to conditional distributions such that given condition c and $z \sim \mathcal{N}(0, 1)$ as input to $\mathcal{G}$, the output is $\mathcal{G}(z, c) \sim p(\cdot|c)$.

S-GAN [14] is conditioned on the past states of all the agents, $s^{1:n}_{t-h:t}$. The generator $\mathcal{G}$ comprises an *Encoder*: a recurrent network that takes as input $s^i_{t-h:t}, i \in \mathcal{N}$ and generates latent representations, a *Pooling* module: which uses these representations and agents' relative positions, and generates a pooled representation incorporating multiagent

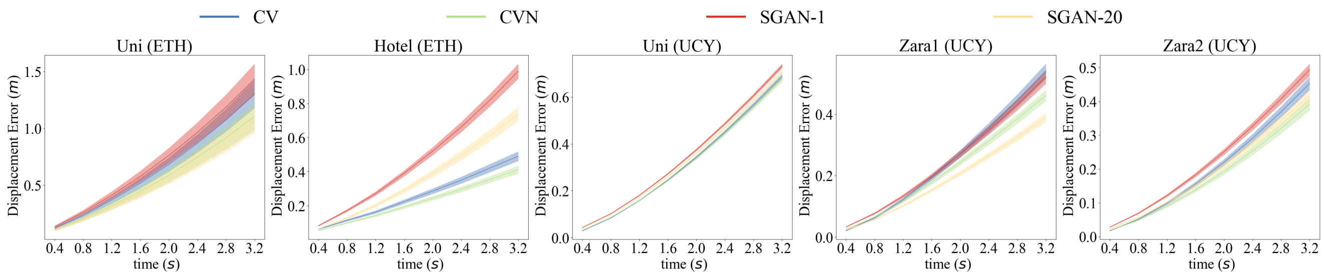


Fig. 2: Error in trajectory prediction of humans on the ETH [29] and UCY [20] datasets. Baselines are referred to from [36]. Error bars indicate 95% confidence intervals, and the line represents the minimum displacement error across the samples.

interaction, a *Decoder*: a recurrent network which takes a latent variable z, the latent and pooled representations and generates a sample from the future state distribution $p(\hat{s}^i_{t:t+T})\forall i \in \mathrm{N}$. Using $\mathcal{G}$ and $z \sim \mathcal{N}(0,1)$, given the past states of all humans $s^{1:n}_{t-h:t}$ we generate samples for the future states $\hat{\boldsymbol{s}}^{1:n}$ by approximating the distribution $f(s^{1:n}_{t-h:t}) = \mathcal{G}(s^{1:n}_{t-h:t}, z) \sim p(\hat{\boldsymbol{s}}^{1:n}|s^{1:n}_{t-h:t})$. In order to model the distribution of trajectories and diversity in samples from the generator, S-GAN adds an auxiliary variety loss [11, 14].

C. Offline Prediction Performance

Schöller et al. [36] compared the Average Displacement Error (ADE) and the Final Displacement Error (FDE) of S-GAN-based prediction against CV prediction and CV prediction with added noise (CVN), showing that the latter ones perform comparably across the scenes in the ETH [29] and UCY [20] datasets. In Fig. 2, we compare their *multistep* prediction performance (i.e., the L2-norm between the predicted position and ground truth at each timestep of prediction), which is informative for navigation tasks.

Similarly to the observations of Schöller et al. [36], we see that S-GAN's performance is mixed. While it exhibits lower error on Zara1, it ties with CVN and CV on ETH-Uni and Zara2 and it is outperformed by them in Hotel, whereas on UCY-Uni all models perform comparably. It should be noted that the human behavior featured in these datasets mostly consists of linear segments that can be well approximated by CV/CVN whereas the S-GAN models promise a better generalization to more complex, nonlinear behavior.

V. MPC with Probabilistic Multiagent Trajectory Prediction

We integrate prediction models from Sec. IV into an MPC for crowd navigation.

A. MPC for Navigation in Crowds

We employ a discrete MPC formulation for navigation in a multiagent environment:

$$\begin{aligned} \boldsymbol{u}^* &= \arg\min_{\boldsymbol{u}\in\boldsymbol{\mathcal{U}}} \mathcal{J}(\boldsymbol{s}, \hat{\boldsymbol{s}}, \hat{\boldsymbol{s}}^{1:n}) \\ &\quad s.t.\ s_{t+1} = g(s_t, u_t) \\ &\qquad (\hat{\boldsymbol{s}}, \hat{\boldsymbol{s}}^{1:n}) = f(s^{1:n}_{t-h:t}, s_{t-h:t}),\ i \in \mathcal{N} \end{aligned}, \tag{2}$$

where: $\boldsymbol{s} = (s_1, \ldots, s_T)$ is a state rollout, acquired by passing a control trajectory $\boldsymbol{u} = (u_0, \ldots, u_{T-1})$ drawn from a space of controls $\mathcal{U}$ through the dynamics g; $\hat{\boldsymbol{s}}^i = (\hat{s}^i_1, \ldots, \hat{s}^i_T)$ is a trajectory prediction for agent i, extracted using f, which takes as input a state history of h timesteps in the past for all the agents, and $\hat{\boldsymbol{s}}^{1:n} = (\hat{\boldsymbol{s}}^1, \ldots, \hat{\boldsymbol{s}}^n)$; $\mathcal{J}$ is a cost expressing considerations of safety, efficiency, and human comfort. We use the trajectory predictions $\hat{\boldsymbol{s}}^{1:n}$ to evaluate the control trajectory $\boldsymbol{u}$. The accuracy of the cost J depends on the model f performance, so, using better prediction models gives more accurate estimates of human safety, and efficiency for robot actions, overall improving the cost function and control.

B. MPC with Probabilistic Prediction

We use the model from Sec. IV-B, to jointly estimate the future states of all agents (including the robot) conditioned on their state histories. We integrate this model of into the MPC framework through the following composite cost:

$$\begin{aligned} \mathcal{J}^{exp}(\boldsymbol{s}, \hat{\boldsymbol{s}}^{1:n}) &= a_g\mathcal{J}_g(\boldsymbol{s}) + \\ &\mathbb{E}\big[a_d\mathcal{J}_d(\boldsymbol{s}, \hat{\boldsymbol{s}}^{1:n}) + a_p\mathcal{J}_p(\boldsymbol{s}, \hat{\boldsymbol{s}}^{1:n}) + a_c\mathcal{J}_c(\boldsymbol{s}, \hat{\boldsymbol{s}}^{1:n})\big] \end{aligned}, \tag{3}$$

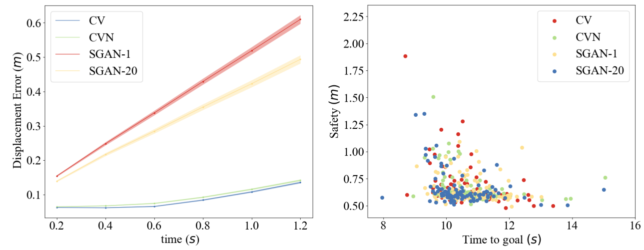
where: the expectation is taken over the distribution $(\hat{\boldsymbol{s}}, \hat{\boldsymbol{s}}^{1:n}) \sim p(\hat{\boldsymbol{s}}, \hat{\boldsymbol{s}}^{1:n}|s_{t-h:t}, s^{1:n}_{t-h:t})$; the functions J_g, J_d account respectively for *progress to goal*, *respect of users' personal space* (see our prior work [25] for detailed definitions), and *prediction consistency*; a_d, a_p, a_c are weights.

Prediction inconsistency cost. Minimizing the cost:

$$\mathcal{J}_c(\boldsymbol{s}, \hat{\boldsymbol{s}}) = \mathbb{E}\left[\|\boldsymbol{s} - \hat{\boldsymbol{s}}\|\right], \tag{4}$$

matches in expectation the model prediction about the robot motion, $\hat{\boldsymbol{s}}$, given its past interactions with the crowd. Since the prediction model is trained to jointly predict multiagent interactions, by staying close to predictions about its own motion, the robot can be more confident about the consistency of its predictions about *users*' motion. In practice, this motivates the robot to avoid maneuvers that could surprise users, forcing them to unexpected reactions that would also be hard to predict using the model.

Overall, the expected cost $\mathcal{J}^{exp}$ enables the controller to probabilistically reason about the quality of candidate trajectories, incorporating a notion of uncertainty over the future human behavior given the robot's intended behavior.



(a) (b)

Fig. 3: Simulation results. (a) Human motion prediction error over time. (b) Safety vs Time to goal. Lines represent minimum displacement errors across the samples and error bands indicate 95% confidence intervals.

C. Simulated Experiments

As a first step towards understanding the impact of prediction accuracy on navigation performance, we instantiated Honda's experimental ballbot [17, 25, 44] (see Fig. 1) in a simulated Gazebo world where human agents were controlled using the ORCA [40] model. We considered a setting in which three human agents and the robot move across the diagonals of a $3.6 \times 4.5m^2$ workspace (see Table II, top left). We evaluated navigation performance in terms of *Safety*, defined as the minimum distance between the robot and human agents (minus the assumed radii of the robot and human agents, both set to $0.3m$) throughout a trial, and *Time to goal*, defined as the time taken by the robot to reach its goal.

Algorithms. We instantiated four different MPC variants, each using a different mechanism for motion prediction:

MPC with CV prediction: This baseline approximates the transition function f as a CV model, i.e., $\boldsymbol{s}^i_{t+1} = \boldsymbol{s}^i_t + v^i_t \cdot dt$ for agent i, where dt represents a timestep; this approximation ignores possible reactions to the motion of other agents.

MPC with CVN prediction: This baseline from Schöller et al. [36] generates noisy samples from a CV model.

MPC with S-GAN-1 prediction: This baseline uses a single-sample estimate extracted from the S-GAN [14] model. We used the best performing model trained on the ETH dataset.

MPC with S-GAN-20 prediction: This baseline uses a 20-sample estimate extracted from the same S-GAN model.

Implementation. We follow an MPC implementation similar to Brito et al. [4], using a set $\mathcal{U}$ of robot control trajectories extracted by propagating the robot with constant velocity towards 10 subgoals, placed around the robot at fixed orientation intervals of $\frac{\pi}{5}$ and distance of $10m$. The robot control trajectories and the human motion prediction are generated for 6 timesteps of size $0.2s$. This parametrization enabled timely response to the dynamic environment: our control loop closed with a frequency of $10Hz$. We tuned all MPC variants through parameter sweeps balancing *Safety* and *Time*. The code for our implementation on a 2d-planar robot can be found at https://github.com/sriyash421/Pred2Nav.

Results. Fig. 3a depicts multistep displacement errors across models. We see that the error of S-GAN models is consistently higher than CV/CVN. We suspect that this is because the behavior of ORCA agents often comprises perfectly linear segments which can be effectively approximated using CV-based models; in contrast, the S-GAN models, trained on real-world datasets are less accurate on ORCA agents. However, we see that the superiority of the CV prediction does not translate to superiority in navigation: a scatter plot for safety vs. time to goal for all trials (Fig. 3b) does not show a clear winner, similarly to conducted pairwise U-tests.

TABLE I: Average (ADE) and final (FDE) displacement error (m) across real-world experiments.

	Cooperative		Aggressive		Distracted	
Prediction Model	ADE	FDE	ADE	FDE	ADE	FDE
S-GAN-20	**0.257**	**0.388**	**0.276**	**0.423**	**0.345**	**0.531**
S-GAN-1	0.357	0.575	0.393	0.637	0.504	0.827
CV	0.389	0.644	0.334	0.539	0.462	0.753

VI. Real-World Experiments

As discussed in Sec. I, benchmarking in a simulated environment –while a widely adopted practice in crowd navigation research– comes with limitations [12, 26]. In this section, we investigate the relationship between prediction and navigation under realistic settings in the lab.

A. Experimental Setup

We used Honda's experimental ballbot [17, 25, 44] (see Fig. 1), and deployed it into a rectangular workspace of area $3.6 \times 4.5m^2$, mirroring out simulation setup.

Conditions. We designed three experimental conditions (shown on the left column of Table II) involving robot navigation under different crowd behaviors that a robot could encounter in a crowded space:

Cooperative: Three users and the robot move between the corners of the workspace. Users were instructed to navigate naturally with a normal walking speed.

Aggressive: One user and the robot exchange corners. The user was instructed to move straight with normal walking speed without accounting for collisions, i.e., forcing the robot to assume complete responsibility for collision avoidance.

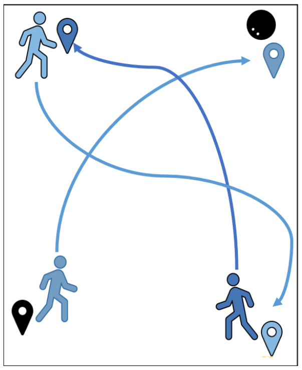
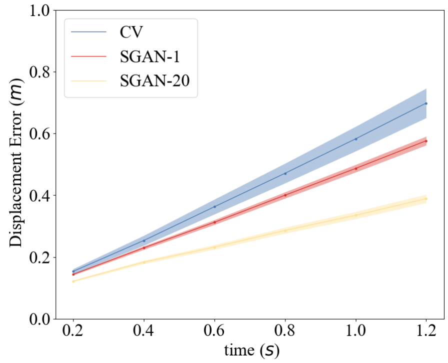
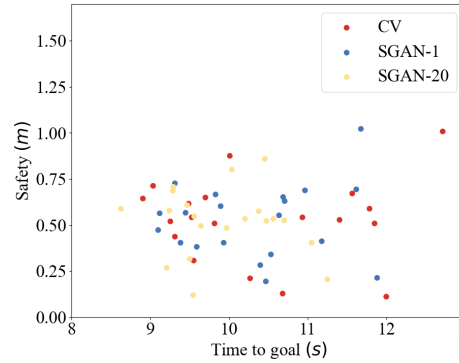
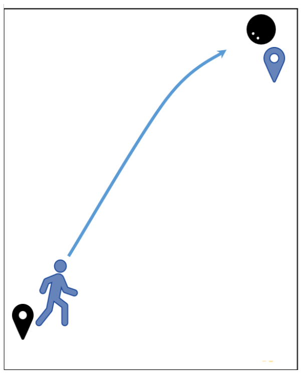
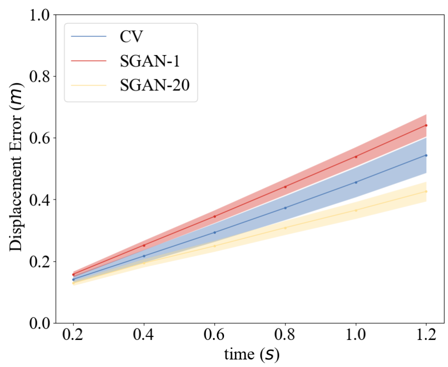
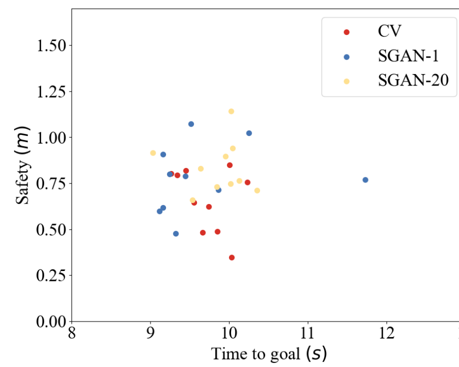
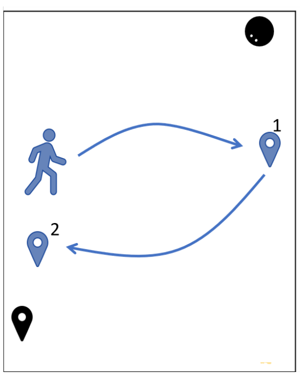
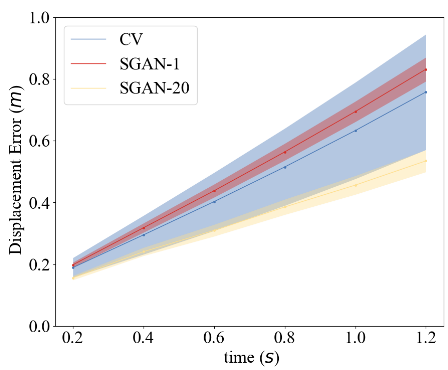
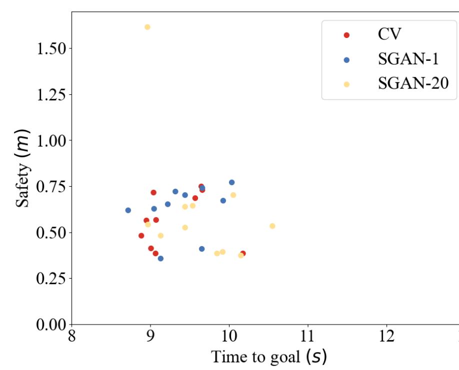
Distracted: One user, starting from the left side, first moves to the right but quickly shifts back to their initial position.

Across all conditions, the robot moves between the start and goal points, fixed at $(0, 0)$ and $(3.6, 4.5)$ respectively. The preferred speed for the robot is set to $0.8m/s$ which was empirically observed to be a natural speed for users during pilot trials. We conducted real-world experiments under all conditions (20 trials per algorithm for the cooperative condition and 10 trials per algorithm for the rest).

Algorithms. Across conditions, we compared the performance of the same MPC architecture under three different motion prediction models: CV, S-GAN-1, and S-GAN-20. We did not instantiate a baseline based on CVN since it was shown to perform comparably with CV in simulation.

Hypotheses. While S-GAN models performed worse than CV in simulation, their prediction accuracy on real-world datasets [14] (Fig. 2) appeared promising for real-world operation. Thus, we expected S-GAN models to outperform baselines and enable improved navigation performance. We formalized these expectations into the following hypotheses:

TABLE II: Real-world experiments. Each row shows a different experimental condition: an illustration of the crowd behavior under each condition is shown on the left (users and their goals are shown in blue, whereas the robot and its goal are shown in black color); the multistep prediction error across trials is shown in the middle (error bands indicate 95% confidence intervals); a scatter plot of *Safety* against *Time to goal* is shown on the right.

Condition	Prediction Error	*Safety* vs *Time to goal*
Cooperative 		
Aggressive 		
Distracted 		

H1: S-GAN-based prediction is more accurate than CV prediction across all conditions.

H2: S-GAN-based prediction enables the MPC to achieve higher navigation performance across all conditions.

H3: Lower prediction error generally enables the MPC to achieve higher navigation performance.

B. Results

Table II shows the multistep prediction error (aggregated values are listed in Table I) and the navigation performance distribution per condition. Fig. 5 relates average prediction error per trial to navigation performance. Fig. 4 connects proximity to the robot to prediction performance. Finally, Fig. 6 shows how different algorithms make predictions and action decisions in the same scene. Footage from our experiments can be found at https://youtu.be/mzFiXg8KsZ0.

H1. We see that S-GAN-20 outperforms CV and S-GAN-1 in terms of ADE and FDE across all conditions (Table I), and exhibits consistently lower multistep prediction error (Table II, 2nd column). S-GAN-1 also outperforms CV under the cooperative condition but not under aggressive and distracted conditions. Thus, H1 holds for a strong model like S-GAN-20.

H2. From the right column of Table II, we see that for the cooperative condition, S-GAN-20 is mostly on the left, corresponding to a good time efficiency, and usually higher than the $0.5m$ *Safety* line whereas the other algorithms are more dispersed all over the graph. In the aggressive condition, no major differences are observed in terms of efficiency; S-

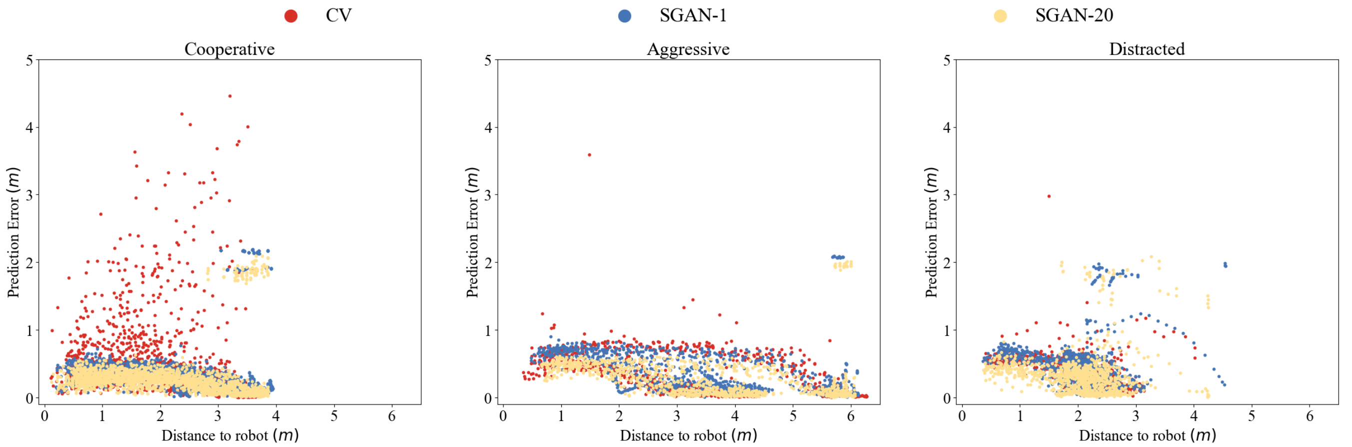
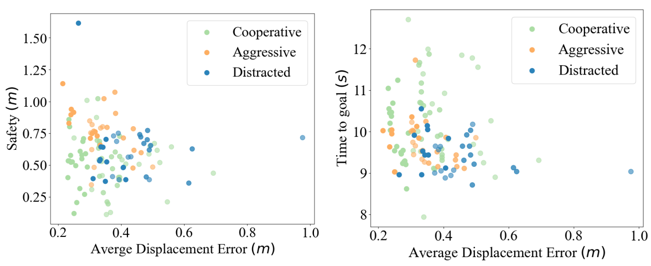


Fig. 4: Average human trajectory prediction error against distance from human at the time of the prediction.



(a) *Safety*

(b) *Time to goal*

Fig. 5: Relationship between prediction performance and navigation performance per trial in the real world.

GAN-20 is often but not consistently safer than baselines. In the distracted condition, algorithms are close to each other. None of these relationships appeared to be statistically significant (pairwise U-tests). Thus, we find no support that the clear superiority in SGAN-20 predictions (H1) translates to superiority in navigation, and therefore H2 is rejected.

H3. Fig. 5 shows scatter plots for *Safety* and *Time to goal* against Displacement Error per trial and condition. Across conditions, we see a pattern connecting lower errors to higher safety and lower time to goal. However, this pattern is not definitive: datapoints are scattered across large regions for both navigation metrics. Further, as shown in Table II, prediction rankings do not transfer clearly to navigation rankings. Thus, we find no support that lower prediction error correlates with improved navigation and H3 is rejected.

VII. Discussion

Model Transfer. The high-quality predictions of S-GAN transferred from offline datasets to onboard robot performance: S-GAN-20 was consistently more accurate across real-world conditions(H1), which shows the efficacy of this generative machinery in modeling multiagent interactions. However, S-GANs struggled with out-of-distribution behaviors encountered in the ORCA-simulated trials (Sec. V-C). While ORCA behaviors are less representative of real pedestrians, this observation highlights the sensitivity of the model to the modes of interaction found in the training dataset. Inducing structure through interaction representations [23, 31, 37] might improve transfer across a wider range of behavior.

Robot and crowd motion are entangled. Across models, prediction performance did not clearly map to navigation performance (H3). In a tight space like our lab workspace, robot motion is coupled with crowd motion. We accounted for that with a *joint prediction* model, capturing the close unfolding crowd-robot interactions. However, when the MPC forces the robot to deviate from the model's ego-prediction, the resulting robot action likely violates the validity of the crowd motion prediction. While the prediction inconsistency cost (see Sec. V-B) motivated the MPC to stay close to the ego prediction, the other costs may conflict on some occasions, leading to states outside the model's confidence. An exciting direction for future work is incorporating explicit formalisms of prediction model confidence into online decision-making.

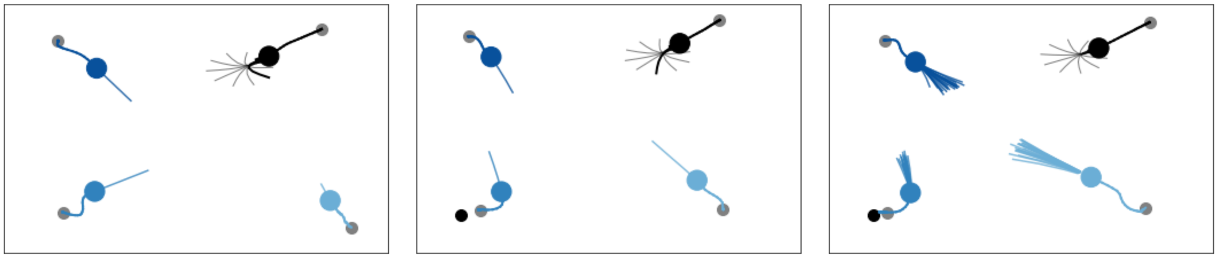
Beyond the Safety-Efficiency tradeoff. After the lab experiments, users shared that MPC with S-GAN was predictable, safer, and more comfortable, but these values are not reflected in the evaluation metrics. While Safety and Efficiency are extensively used for evaluation in social navigation [26], they miss important attributes of interaction such as human comfort, satisfaction, and smoothness. Ongoing work has looked at different aspects of comfort [24, 41] but future work is needed on the design of validated metrics of interaction. Observations such as human gaze, expressions, and gestures can be used to recover these properties of the interaction.

References

[1] S. Becker, R. Hug, W. Hübner, and M. Arens. Red: A simple but effective baseline predictor for the TrajNet benchmark. In *Workshops of the European Conference on Computer Vision*, 2018.

[2] A. Bhattacharyya, M. Hanselmann, M. Fritz, B. Schiele, and C.-N. Straehle. Conditional flow variational autoencoders for structured sequence prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni. SocNavBench: A grounded simulation



(a) CV. (b) S-GAN-1. (c) S-GAN-20.

Fig. 6: Instances from lab trials under the cooperative condition for all algorithms. The black dot is the robot, and the others are users. Solid lines represent agents' state histories and faded lines are trajectory predictions under each prediction model. The robot's rollouts are shown in gray and the selected one is shown in black.

testing framework for evaluating social navigation. *Transactions on Human-Robot Interaction*, 11(3), 2022.

[4] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora. Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments. *IEEE Robotics and Automation Letters*, 6(3):4616–4623, 2021.

[5] S. Casas, W. Luo, and R. Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2018.

[6] C. Chen, Y. Liu, S. Kreiss, and A. Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6015–6022, 2019.

[7] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva. Relational graph learning for crowd navigation. In *Proceesings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10007–10013, 2020.

[8] H. Chen and X. Zhang. Path planning for intelligent vehicle collision avoidance of dynamic pedestrian using att-lstm, msfm, and mpc at unsignalized crosswalk. *IEEE Transactions on Industrial Electronics*, 69(4):4285–4295, 2022.

[9] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, and J. G. Schneider. Motion prediction of traffic actors for autonomous driving using deep convolutional networks. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[10] M. Everett, Y. F. Chen, and J. P. How. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3052–3059, 2018.

[11] H. Fan, H. Su, and L. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2017.

[12] T. Fraichard and V. Levesy. From crowd simulation to robot navigation in crowds. *IEEE Robotics and Automation Letters*, 5(2):729–735, 2020.

[13] F. Grzeskowiak, D. Gonon, D. Dugas, D. Paez-Granados, J. J. Chung, J. Nieto, R. Siegwart, A. Billard, M. Babel, and J. Pettré. Crowd against the machine: A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3879–3885, 2021.

[14] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2255–2264, 2018.

[15] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1995.

[16] S. Hoermann, M. Bach, and K. Dietmayer. Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, page 2056–2063, 2018.

[17] Honda. Honda P.A.T.H. Bot, 2019. URL https://global.honda/innovation/CES/2019/path_bot.html.

[18] Y. Kobayashi, T. Sugimoto, K. Tanaka, Y. Shimomura, F. J. A. Garcia, C. H. Kim, H. Yabushita, and T. Toda. Robot navigation based on predicting of human interaction and its reproducible evaluation in a densely crowded environment. *International Journal of Social Robotics*, 14(2):373–387, may 2021.

[19] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11):1289–1307, 2016.

[20] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

[21] C. Liu, Y. Chen, M. Liu, and B. E. Shi. AVGCN: Trajectory prediction using graph convolutional networks guided by human attention. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 14234–14240, 2021.

[22] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell. Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3517–3524, 2021.

[23] K. Mangalam, Y. An, H. Girase, and J. Malik. From goals, waypoints & paths to long term human trajectory forecasting. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15213–15222, 2021.

[24] C. Mavrogiannis, P. Alves-Oliveira, W. Thomason, and R. A. Knepper. Social momentum: Design and evaluation of a framework for socially competent robot navigation. *Transactions on Human-Robot Interaction*, 11(2), 2022.

[25] C. Mavrogiannis, K. Balasubramanian, S. Poddar, A. Gandra, and S. S. Srinivasa. Winding through: Crowd navigation via topological invariance. *IEEE Robotics and Automation Letters*, 8(1):121–128, 2023.

[26] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh. Core Challenges of Social Robot Navigation: A Survey. *Transactions on Human-Robot Interaction*, 12(3), 2023.

[27] N. Nikhil and B. T. Morris. Convolutional neural network for trajectory prediction. In *Workshops of the European Conference on Computer Vision (ECCV)*, 2018.

[28] H. S. Park, J.-J. Hwang, Y. Niu, and J. Shi. Egocentric future localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4697–4705, 2016.

[29] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 261–268, 2009.

[30] S. Pirk, E. Lee, X. Xiao, L. Takayama, A. Francis, and A. Toshev. A Protocol for Validating Social Navigation Policies. *arXiv e-prints*, 2022.

[31] J. Roh, C. Mavrogiannis, R. Madan, D. Fox, and S. Srinivasa S. Multimodal trajectory prediction via topological invariance for navigation at uncontrolled intersections. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2020.

[32] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras. Human motion trajectory prediction: a survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020.

[33] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–700, 2020.

[35] A. J. Sathyamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, and D. Manocha. DenseCAvoid: Real-time navigation in dense crowds using anticipatory behaviors. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11345–11352, 2020.

[36] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll. What the constant velocity model can teach us about pedestrian. *IEEE Robotics and Automation Letters*, 5 (2):1696–1703, 2020.

[37] M. Sun, F. Baldini, P. Trautman, and T. Murphey. Move Beyond Trajectories: Distribution Space Coupling for Crowd Navigation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2021.

[38] P. Trautman, J. Ma, R. M. Murray, and A. Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation. *International Journal of Robotics Research*, 34(3):335–356, 2015.

[39] N. Tsoi, M. Hussein, O. Fugikawa, J. D. Zhao, and M. Vázquez. An approach to deploy interactive robotic simulators on the web for HRI experiments: Results in social robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 7528–7535, 2021.

[40] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics Research*, pages 3–19. Springer Berlin Heidelberg, 2011.

[41] N. Walker, C. Mavrogiannis, S. S. Srinivasa, and M. Cakmak. Influencing behavioral attributions to robot motion during task execution. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2021.

[42] A. Wang, C. Mavrogiannis, and A. Steinfeld. Group-based motion prediction for navigation in crowded environments. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2021.

[43] A. Wang, A. Biswas, H. Admoni, and A. Steinfeld. Towards Rich, Portable, and Large-Scale Pedestrian Data Collection. *arXiv e-prints*, Mar. 2022.

[44] K. Yamane and C. Kurosu. Stable balance controller, March 2020. US Patent 16/375,111.

[45] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng. SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12077–12086, 2019.

[46] H. Zhu, F. M. Claramunt, B. Brito, and J. Alonso-Mora. Learning interaction-aware trajectory predictions for decentralized multi-robot motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 6(2):2256–2263, 2021.

[47] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3931–3936, 2009.