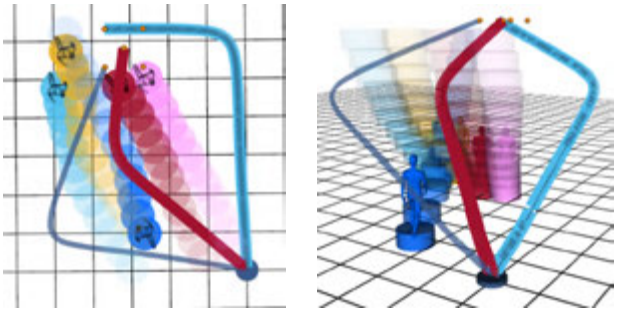
Learning Social Homologies for Navigation

Diego Martinez-Baselga*, Oscar de Groot†, Luzia Knoedler†,
Luis Riazuelo*, Javier Alonso-Mora† and Luis Montano*
*University of Zaragoza, Spain
†Delft University of Technology, Netherlands

***Abstract*—Navigating mobile robots in social environments remains a challenging task due to the intricacies of human-robot interactions. Most of the motion planners designed for crowded and dynamic environments focus on reaching the goal while avoiding collisions or local social interactions, but do not explicitly consider the high-level navigation behavior (avoiding through the left or right side, letting others pass or passing before others, etc.). In this work, we present a novel motion planner that incorporates topology distinct paths representing diverse navigation strategies around humans. The planner uses a deep neural network model trained on real-world human motion data to estimate how well a topology classes imitate the human behavior. Then, selects the best topology class using it, ensuring socially intelligent and contextually aware navigation. Our system refines the chosen path through an optimization-based local planner in real time, ensuring adherence to desired social behaviors. We evaluate the prediction accuracy of the network with real-world data and the navigation capabilities in both simulation and a real-world platform. Our method demonstrates socially desirable behaviors, smooth and remarkable performance compared to other planners.**



(a) Top view (b) State space (time in z-axis)

Fig. 1: A scenario of the ETH/UCY dataset. The robot is the blue disk and the humans future are disks with increasing transparency. There are three possible topology distinct trajectories. Our method chooses the light blue while the one with minimum length is the red one.

I. Introduction

Motion planning in social environments involves global and local planning. Traditional global planners compute a plan that only considers static obstacles and a local planner tries to follow it. Thus, the local planner may find itself stuck in locally optimal plans that overlook the presence of dynamic obstacles beyond its planning horizon. This can lead to different dynamic behaviors that were not considered by the initial plan, as passing before obstacles or letting them pass.

In addition, social navigation poses the challenge of selecting the desired robot dynamic behavior. While manually defining a cost based on heuristics is an option, evaluating social norms and adapting to every possible situation is a very complex problem. In some scenarios, the robot must make high-level decisions such as avoiding uncomfortable situations, not blocking others' paths, or understanding preferences. For example, in Fig. 1, the robot faces a crowded scenario where the decisions are encoded in three topological classes of trajectories: Light blue is less intrusive with the blue pedestrian and avoids the pink and the red through the right side; red avoids them through the left and waits for the blue to pass, while the other blue trajectory passes after every human

Optimization-based planners designed for social environments typically fail in local optima [3], do not consider the dynamism of the environment [5], are designed for non-social scenarios [4] or are only based on social heuristics [13], and learning-based approaches that predict the topology class of trajectories either assume the environment is static or are not prepared for navigation [16, 18]. Other deep learning methods, such as imitation learning and deep reinforcement learning [6, 9, 12], only consider local interactions and may fail with out-of-distribution observations, making them unsafe.

We propose a new supervised learning approach to learn human navigation preferences in crowded environments. It assesses how closely the topology class of a queried trajectory matches human choices, distinguishing between social and non-social navigation decisions. We use it in a new motion planner called **SHINE** (**S**ocial **H**omology **I**dentification for **N**avigation in crowded **E**nvironments), which infers social high-level navigation decisions to initialize an optimization-based planner (LMPCC [3]). In addition, SHINE may reactively replan with new social decisions if the scenario changes unexpectedly.

II. Problem formulation

We consider scenarios in which a robot must navigate to a goal, $\boldsymbol{p}_{goal} = (x_{goal}, y_{goal})$, while ensuring collision avoidance with humans, represented as disks. We model the robot's motion by the deterministic discrete-time non-linear dynamics $\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$, where $\boldsymbol{x}_k$ is the state and $\boldsymbol{u}_k$ the control input at time k. The state contains its position $\boldsymbol{p}_k = (x_k, y_k) \in \mathbb{R}^2$. We will use the superscript T when referring to the collection of variables over T discrete time steps. For example, $\boldsymbol{p}_k^T = [\boldsymbol{p}_{k-T}, \dots, \boldsymbol{p}_k]$ denotes the robot's positions over the past T time steps.

The position of the human j is denoted as $\boldsymbol{o}_{j,k} = (x_{j,k}, y_{j,k}) \in \mathbb{R}^2$ and the position of the set of M humans is denoted as $\mathcal{O}_k = [\boldsymbol{o}_{0,k}, \dots, \boldsymbol{o}_{M,k}]$. We assume that $\mathcal{O}_k^T$ and $\boldsymbol{p}_k^T$ are known. The state space that we consider for planning is composed by the workspace and time: $\mathcal{X} = \mathbb{R}^2 \times [0, T]$. A trajectory is a continuous path through the state space: $\boldsymbol{\tau} : [0, 1] \to \mathcal{X}$. The collision-free space describes the state space without the space occupied by the obstacles.

A. Homotopy and Homology Classes

We are interested in the high-level behavior of trajectories in collision-free space to learn this behavior from humans. The direction in which a trajectory passes obstacles is captured by *homotopy* classes, a topology concept. Two trajectories are in the same homotopy class if they can be continuously deformed into each other without intersecting obstacles, while keeping the endpoints fixed [1]. An example of homotopically distinct trajectories is shown in Fig. 1. While homotopy classes provide useful information, they are difficult to compute. Therefore, we use the almost equivalent concept of homology classes, as done in previous works [1, 15].

Contrary to homotopy classes, homology classes are comparable in practice through their H-signature. In the 3-dimensional state space like $\mathcal{X}$, the H-signature can be computed by assuming that the time dimension progresses linearly (e.g., discrete-time), and using a wire (or skeleton) representing obstacles and their future motion through the state space. The homology classes of two trajectories $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ connected to the same starting and ending points may be compared using their H-signature, by using the loop formed by $\boldsymbol{\tau}_1 \sqcup -\boldsymbol{\tau}_2$:

$$h_i(\boldsymbol{\tau}_1 \sqcup -\boldsymbol{\tau}_2) = \int_{\boldsymbol{\tau}_1 \sqcup -\boldsymbol{\tau}_2} \boldsymbol{B}(\boldsymbol{l}) d\boldsymbol{l}, \tag{1}$$

where $\boldsymbol{B}(\boldsymbol{l})$ is the magnetic field vector and $d\boldsymbol{l}$ an infinitesimal vector element along the loop path. We refer the reader to [1] for more information.

B. Reference H-signature

We define our own version of the H-signature using a reference trajectory $\boldsymbol{\tau}_{ref}$ and Eq. 1. We use a reference trajectory to be able to assess the topology of a trajectory in isolation, consistently and invariantly. Having a trajectory $\boldsymbol{\tau}$ with initial and final times t_0 and t_f, whose origin is $\boldsymbol{\tau}^0 = (x_o, y_o, t_0)$ and final point is $\boldsymbol{\tau}^f = (x_f, y_f, t_N)$; the reference trajectory is defined by the straight line that connects the points $\boldsymbol{\tau}^0$ and $\boldsymbol{\tau}^f$. It is the trajectory on the minimal energy, and it forms a loop with $\boldsymbol{\tau}$. If we assume that the magnitude of the current in the conductor in Eq. 1 is 1, $h_i(\boldsymbol{\tau}_{ref} \sqcup -\boldsymbol{\tau})$ has the value 0 if the obstacle i is not inside the loop ($\boldsymbol{\tau}$ avoids i in the same way as $\boldsymbol{\tau}_{ref}$), or ± 1 otherwise. Considering M humans as dynamic obstacles, we define the new signature as:

$$\mathcal{H}_{ref}(\boldsymbol{\tau}) = [h_0(\boldsymbol{\tau}_{ref} \sqcup -\boldsymbol{\tau}), \dots, h_M(\boldsymbol{\tau}_{ref} \sqcup -\boldsymbol{\tau})]^T \tag{2}$$

III. Navigation system

We aim to develop a navigation system that follows the homology class selection of humans. It first derives homology distinct trajectories, then selects one of them, and finally uses a local motion planner to track it, repeating iteratively the process.

A. Guidance trajectories proposal

We use the algorithm proposed in [4] to compute a sparse representation of paths (guidance trajectories) that go from the initial position of the robot to the goal, and use the H-signature to filter the trajectories that belong to the same homology class. The method is built on Visibility-PRM and works in the same state space introduced in the problem formulation, composed by the workspace and time $\mathcal{X} = \mathbb{R}^2 \times [0, T]$. The result is a graph with a set of trajectories $\mathcal{T}^*$ where $\mathcal{H}_{ref}(\boldsymbol{\tau}_1) \neq \mathcal{H}_{ref}(\boldsymbol{\tau}_2), \forall (\boldsymbol{\tau}_1, \boldsymbol{\tau}_2) \in \mathcal{T}^*$. The resulting trajectories are topologically distinct and represent different homology classes that the robot may follow to reach the goal.

B. Homology class selection

While humans are constantly choosing a suitable topological path when navigating, manually defining a function that considers all factors involved in the decision-making process is impossible. We propose using a neural network with parameters θ to estimate a cost, $\mathcal{J}_\theta$, that encodes the difference between the homology class of the trajectory followed by a hypothetical human, $\mathcal{H}_{ref}(\boldsymbol{\tau}^h)$, and the homology class of each of the guidance trajectories found at time k, $\mathcal{T}_k^*$. Thus, the selected guidance trajectory is the one with the minimum cost:

$$\boldsymbol{\tau}_k = \underset{\boldsymbol{\tau}_{i,k} \in \mathcal{T}_k^*}{\arg\min} \mathcal{J}_\theta(\mathcal{H}_{ref}(\boldsymbol{\tau}_{i,k}), \boldsymbol{p}_k^T, \mathcal{O}_k^T), \tag{3}$$

The network is trained to estimate the difference in H-signature (in a Euclidean sense) between the trajectory $\boldsymbol{\tau}^g$ and the human trajectory $\boldsymbol{\tau}^h$:

$$y^g = MSE(\mathcal{H}_{ref}(\boldsymbol{\tau}^g), \mathcal{H}_{ref}(\boldsymbol{\tau}^h)), \tag{4}$$

Real-world UCY [11] and ETH [14] datasets of humans navigating in crowds are used to train it. Each training sample consists of a planning problem for one of the humans where his trajectory is used as ground truth ($\boldsymbol{\tau}^h$ in Eq. 4). The network parameters θ are optimized to fit the data:

$$\min_\theta \sum_{k=1}^{N_s} \sum_{g=1}^{N_h} \mathscr{L}_{pred}(\hat{y}_k^g, y_k^g), \tag{5}$$

where $\mathscr{L}_{pred}$ is MSE, N_s are the number of scenarios in the dataset, N_h the number of homology classes found in scenario k and $\mathcal{J}_\theta(\mathcal{H}_{ref}(\boldsymbol{\tau}^g), \boldsymbol{p}_k^T, \mathcal{O}_k^T) = \hat{y}_k^g$ the network output.

The network encoding part is based on the architecture proposed in [17] and [10]. It shares the same encoding architecture with modifications to include the homology information. The states of the robot and the surrounding humans are first extended by linearly interpolating the velocity $(v_{x,k}, v_{y,k})$,

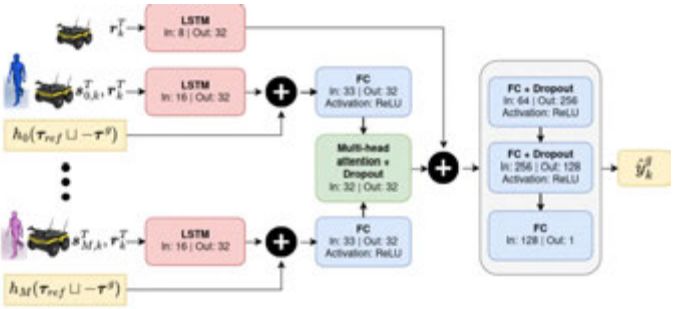


Fig. 2: Diagram with the layers of the network.

acceleration $(a_{x,k}, a_{y,k})$, and the heading angle α from current and previous positions. Thus, the considered full state of the robot is $\boldsymbol{r}_k = [x_k, y_k, v_{x,k}, v_{y,k}, a_{x,k}, a_{y,k}, \sin\alpha_k, \cos\alpha_k]$ and the state of the human j is $\boldsymbol{s}_{j,k} = [x_{j,k}, y_{j,k}, v_{j,x,k}, v_{j,y,k}, a_{j,x,k}, a_{j,y,k}, \sin\alpha_{j,k}, \cos\alpha_{j,k}]$. The network is represented in Fig. 2.

C. Local Optimization-based Planner

The selected guidance trajectory identifies the most suitable homology class but is not guaranteed to be dynamically feasible or collision free. To obtain a high-quality trajectory, we initialize and track the guidance trajectory with a local optimization-based planner that refines the trajectory in the same state space, similar to [4]. The local planner is LMPCC [3], which solves a trajectory optimization, enforcing dynamic and collision avoidance constraints and returns a smooth trajectory near the guidance trajectory.

At every time step k, new guidance trajectories are sampled and a new homology class is selected, reacting to changes in environments with fast replanning capabilities, accounting for estimation errors or changes in the behavior of the pedestrians. Nevertheless, fast switching between trajectories in different topology classes can lead to indecisive and ultimately non-social navigation, so we include a small consistency weight to keep the same homology class when others' cost is similar.

IV. Evaluation

A. Prediction evaluation

The network's ability to select the correct guidance trajectory is evaluated using the ETH/UCY dataset. We designed an accuracy score to measure how often the homology class chosen by the human matches the lowest cost estimated by the network. To test its generalization, we trained the network on four of the five scenarios (ETH, Hotel, University, Zara1, and Zara2) and evaluated it on the fifth (Leave One Out). As baselines, we used three heuristic-based hand-crafted costs, taking N_τ samples of positions $\boldsymbol{p}_i$ and accelerations $\boldsymbol{a}_i$ at constant intervals along the trajectories:

- Minimum length: The shortest guidance trajectory has the lowest cost: $\mathcal{J}_{len} = \sum_{i \in N_\tau} ||\boldsymbol{p}_i - \boldsymbol{p}_{i-1}||$
- Minimum acceleration: The smoothest trajectory has the minimum cost: $\mathcal{J}_{acc} = \sum_{i \in N_\tau} \alpha^i ||\boldsymbol{a}_i||$, where $\alpha \approx 1$ to discount accelerations in time.
- Mixed cost: Combine previous costs: $\mathcal{J}_{mix} = \mathcal{J}_{len} + \mathcal{J}_{acc}$

The accuracy results in Table I show that our method significantly outperforms the baselines. We noted that once a human decides how to pass others, they typically stick to their initial decision, explaining why all methods have good accuracy (> 0.5). Our method accurately selects the homology class the human follows and estimates the decision at any time step. The network's consistent accuracy across scenarios suggests it effectively generalizes. Fig. 1 shows a scenario where the network chose the same trajectory as the prerecorded human, opting for a less intrusive path different from the one with the minimum $\mathcal{J}_{len}$.

TABLE I: Accuracy metrics of the predictions.

Cost	ETH	Hotel	Univ	Zara1	Zara2	Avg.
$\mathcal{J}_{len}$	0.770	0.808	0.730	0.786	0.767	0.772
$\mathcal{J}_{acc}$	0.623	0.617	0.625	0.516	0.645	0.605
$\mathcal{J}_{mix}$	0.730	0.797	0.729	0.779	0.758	0.759
$\mathcal{J}_\theta$ (ours)	**0.984**	**0.951**	**0.942**	**0.955**	**0.925**	**0.951**

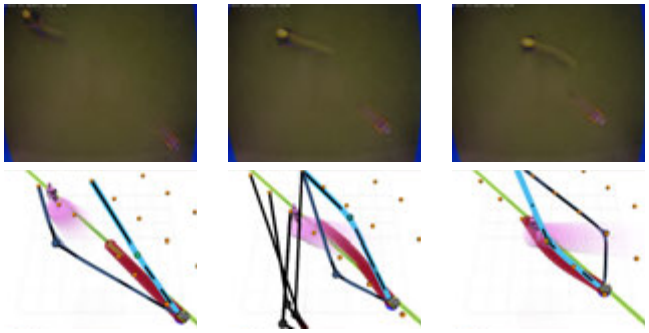
B. Simulation experiments

We conducted 50 simulation scenarios, gathering navigation metrics similar to previous studies [2, 7], in a corridor environment with 12 pedestrians using the Social Force model [8], where the robot navigates 25 meters with differential-drive restrictions. Metrics were collected for a differential-drive Social Force version, a social DRL-based planner [12], an MPC-based planner (LMPCC)[3], and SHINE. Metrics are relative to Social Force, with results in Table II.

TABLE II: Social metrics of the different planners.

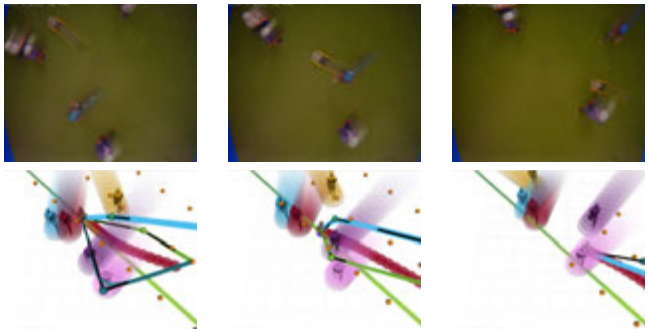
Metric	Social Force	DRL	LMPCC	SHINE
Success rate ↑	0.74	0.90	**0.98**	**0.98**
Collision rate ↓	0.26	0.10	**0.02**	**0.02**
Avg. path length ↓	**1.000**	1.014	1.010	1.009
Avg. time to goal ↓	1.000	**0.997**	1.052	1.003
Avg. speed ↑	1.000	1.005	0.968	**1.010**
Path irregularity ↓	1.000	2.093	**0.793**	0.814
Avg. ω ↓	1.000	2.662	0.389	**0.311**
Avg. acceleration ↓	1.000	1.537	1.116	**0.805**
Avg. jerk ↓	1.000	1.673	1.080	**0.760**

In these scenarios, both LMPCC and SHINE are the safest, showing higher success rates and lower collision rates. The DRL planner, trained in a different simulator, might improve in corridor scenarios with specific training, but this would indicate limited effectiveness across different settings. There is a noticeable difference in path irregularity and angular velocity between MPC-based planners and reactive ones. MPC-based planners, which plan the entire path in advance, produce smoother trajectories, beneficial for social navigation. Additionally, our planner differs in average acceleration and jerk, as it can escape local minima and smoothly replan when the environment changes.



(a) (b) (c)

Fig. 3: Video frames (top) and visualization (bottom) showing a robot adapting to human intentions. In the visualization: the light green line is the robot's reference path, the transparent shadow predicts the human's trajectory, orange dots are potential subgoals, and colored lines show possible guidance trajectories. The chosen path is light blue, and red circles represent the LMPCC-optimized path.



(a) (b) (c)

Fig. 4: A sequence of the robot in a crowded environment.

C. Real world experiments

We conducted experiments in a Clearpath Jackal platform, with a marker-based tracking system to determine robot and pedestrians positions and a Kalman filter to estimate their velocities. A constant velocity assumption is used to estimate humans' future trajectories. We conducted experiments with Social Force, the previous DRL planner, LMPCC, Guidance-MPCC [4], and SHINE. While the former three exhibit different local behavior, Guidance-MPCC is similar to SHINE but without social high-level considerations (it also chooses among homology classes but with a cost based on heuristics).

We empirically tested SHINE in challenging scenarios where other planners failed. Particularly, in case of Social Force, scenarios where pedestrians and the goal had opposing forces, for DRL out-of-distribution scenarios and for LMPCC sudden changes in pedestrians trajectories that made it fall into local optima. SHINE succeeded in all of them. An example of SHINE in the latter is shown in Fig. 3. There is a head-on scenario where, at first, the human's intention is avoiding the robot on the right side, and the guidance chosen by the robot avoids the human on the right side too (light blue). Then, the human decides to avoid the robot through the left. The robot perceives his intentions and changes the homology class, following a smooth avoidance maneuver.

We also conducted experiments with five pedestrians using different navigation algorithms, without revealing which one was active. Participants rated their comfort after each interaction in short interviews. They found Social Force and DRL too reactive, leading to some collisions, while SHINE was rated higher than Guidance-MPCC due to fewer disturbances. We compared SHINE and Guidance-MPCC based on their obstacle avoidance behavior in Table III, noting how often they avoided pedestrians by moving right or left, and whether they passed before or after them. Both methods tended to avoid obstacles on the right, likely due to cultural habits. Guidance-MPCC preferred passing before pedestrians to maintain speed, while SHINE preferred passing after to avoid interrupting paths, aligning with socially desirable behavior.

TABLE III: Passing behavior of Guidance-MPCC and SHINE.

Method	Left	Right	Before	After
G-MPCC	0.437	0.563	0.596	0.404
SHINE	0.463	0.538	**0.329**	**0.671**

SHINE exhibits learned behaviors derived from real human data rather than manual programming. For instance, in Fig. 4, the robot encounters two humans at a crossing—one moving quickly and the other slowly. SHINE chooses to pass behind the fast-moving pedestrian and ahead of the slow-moving one, minimizing disruption.

V. Conclusion

This work presented a novel motion planner for dynamic environments that, unlike most of the learning-based social navigation approaches, is capable of making human-like discrete high-level decisions to navigate. It demonstrated a safe and smooth behavior in a simulated and a real-world crowded environments. Furthermore, the prediction framework shows impressive results in predicting the homology class chosen by the humans, which could be used in other problems and open new research lines. The main limitation of the work is computational time in low-cost devices, as it involves an optimization and a neural network inference, while further work will involve introducing a social local planner.

Acknowledgements

This work was supported by the Ministry of Science and Innovation of Spain through the research project PID2022-139615OB-I00/MCIN/AEI/10.13039/501100011033/FEDER-UE, from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101017008, and from the European Union (ERC, INTERACT, 101041863). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. Topological constraints in search-based robot path planning. *Autonomous Robots*, 33:273–290, 2012.

[2] Abhijat Biswas, Allan Wang, Gustavo Silvera, Aaron Steinfeld, and Henny Admoni. Socnavbench: A grounded simulation testing framework for evaluating social navigation. *ACM Transactions on Human-Robot Interaction (THRI)*, 11(3):1–24, 2022.

[3] Bruno Brito, Boaz Floor, Laura Ferranti, and Javier Alonso-Mora. Model predictive contouring control for collision avoidance in unstructured dynamic environments. *IEEE Robotics and Automation Letters*, 4(4): 4459–4466, 2019.

[4] Oscar de Groot, Laura Ferranti, Dariu Gavrila, and Javier Alonso-Mora. Globally guided trajectory planning in dynamic environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10118–10124. IEEE, 2023.

[5] Zhiyu Ding, Jie Liu, Wenzheng Chi, Jiankun Wang, Guodong Chen, and Lining Sun. PRTIRL based socially adaptive path planning for mobile robots. *International Journal of Social Robotics*, 15(2):129–142, 2023.

[6] Michael Everett, Yu Fan Chen, and Jonathan P How. Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access*, 9:10357–10377, 2021.

[7] Anthony Francis, Claudia Pérez-d'Arpino, Chengshu Li, Fei Xia, Alexandre Alahi, Rachid Alami, Aniket Bera, Abhijat Biswas, Joydeep Biswas, Rohan Chandra, et al. Principles and guidelines for evaluating social robot navigation algorithms. *arXiv preprint arXiv:2306.16740*, 2023.

[8] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[9] Zhengxi Hu, Yingli Zhao, Sen Zhang, Lei Zhou, and Jingtai Liu. Crowd-comfort robot navigation among dynamic environment based on social-stressed deep reinforcement learning. *International Journal of Social Robotics*, 14(4):913–929, 2022.

[10] Boris Ivanovic, James Harrison, and Marco Pavone. Expanding the deployment envelope of behavior prediction via adaptive meta-learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7786–7793, 2023.

[11] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[12] Diego Martinez-Baselga, Luis Riazuelo, and Luis Montano. Improving robot navigation in crowded environments using intrinsic rewards. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9428–9434, 2023.

[13] Christoforos Mavrogiannis, Krishna Balasubramanian, Sriyash Poddar, Anush Gandra, and Siddhartha S Srinivasa. Winding Through: Crowd navigation via topological invariance. *IEEE Robotics and Automation Letters*, 8(1):121–128, 2022.

[14] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009.

[15] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems*, 88:142–153, 2017.

[16] Christoph Rösmann, Malte Oeljeklaus, Frank Hoffmann, and Torsten Bertram. Online trajectory prediction and planning for social robot navigation. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1255–1260. IEEE, 2017.

[17] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

[18] Jennifer Wakulicz, Ki Myung Brian Lee, Teresa Vidal-Calleja, and Robert Fitch. Topological trajectory prediction with homotopy classes. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6930–6936, 2023.