HEIGHT: HEterogeneous Interaction GrapH Transformer for Robot Navigation in Crowded and Constrained Environments

Shuijing Liu*, Haochen Xia†, Fatemeh Cheraghi Pouria†, Kaiwen Hong,
Neeloy Chakraborty, Zichao Hu, Joydeep Biswas, and Katherine Driggs-Campbell

***Abstract*—We study the problem of robot navigation in dense and interactive crowds with environmental constraints such as corridors and furniture. Previous methods fail to consider all types of interactions among agents and obstacles, leading to unsafe and inefficient robot paths. In this article, we leverage a graph-based representation of crowded and constrained scenarios and propose a structured framework to learn robot navigation policies with deep reinforcement learning. We first split the representations of different components in the environment, and propose HEIGHT, a novel navigation policy network architecture with different components to capture heterogeneous interactions among entities through space and time. HEIGHT utilizes attention mechanisms to prioritize important interactions and a recurrent network to track changes in the dynamic scene over time, encouraging the robot to avoid collisions adaptively. Through extensive simulation and real-world experiments, we demonstrate that HEIGHT outperforms state-of-the-art baselines in terms of success, efficiency, and generalization capability when the densities of humans and obstacles change. More videos are available at https://sites.google.com/view/crowdnav-height/home.**

I. Introduction

Robots are increasingly prevalent in human-centric environments. We study robot navigation to a destination without colliding with humans **and** obstacles, a crucial ability for applications such as last-mile delivery and household robots. For example, Fig. 1 shows a navigation scenario with abundant subtle interactions among the robot, humans, and obstacles. These interactions are heterogeneous, dynamic, and difficult to reason, making navigation in such environments challenging.

Previous works have explored model-based and learning-based approaches for crowd navigation in an open space without obstacles [1]–[4]. However, static obstacles such as furniture, walls, and untraversable regions are common in the real-world. To this end, other works use groups of circles or raw images or point clouds to represent both humans and static obstacles. Nevertheless, their scene representations or navigation algorithms do not differentiate between dynamic and static obstacles, and thus the robot has difficulties taking adaptive strategies to avoid collisions [5]–[8]. To address these limitations, we ask the following research question: *How can a robot navigation policy represent and reason about diverse interactions in crowded and constrained environments to adaptively avoid collisions?*

*Corresponding author. Email: shuijing.liu@utexas.edu

This material is based upon work supported by the National Science Foundation under Grant No. 2143435.

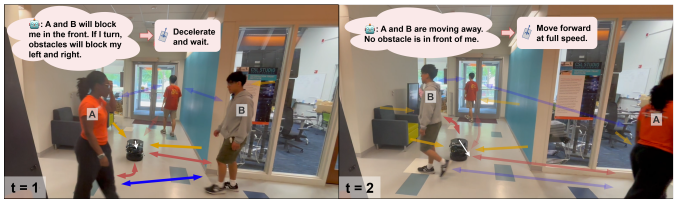

Fig. 1: **A heterogeneous graph aids spatio-temporal reasoning when a robot navigates in a crowded and constrained environment.** The colored arrows denote robot-human (RH), human-human (HH), and obstacle-agent (OA) interactions. The opaque arrows are the more important interactions while the transparent arrows are the less important ones.

To answer this question, we propose a framework that leverages the heterogeneity of interactions in crowded and constrained scenarios. First, we split the environment into human and obstacle representations with information that is essential to navigation. The split representations are processed and fed separately into the reinforcement learning (RL)-based navigation pipeline. Then, inspired by recent breakthroughs in spatio-temporal (st) networks for crowd navigation [8]–[11], we decompose the scenario into a heterogeneous spatio-temporal (st) graph with different types of edges to represent different types of interactions among the robot, observed and untracked humans, and observed obstacles, as shown in the colored arrows in Fig. 1. Finally, we convert the heterogeneous st-graph into a **HE**terogeneous **I**nteraction **G**rap**H** **T**ransformer (HEIGHT), a robot policy network consisting of different modules to parameterize the heterogeneous spatio-temporal interactions. In the rapidly changing scenario in Fig. 1, HEIGHT injects structures to and captures the synergy between scene representation and network architectures. By reasoning about the heterogeneous interactions among different components through space and time, the robot is able to avoid collisions and approach its goal in an efficient manner. In summary, the main contributions of this article are as follows.

1) We propose a split input representation that treats humans and obstacles separately, enabling structured modeling of crowded and constrained environments.
2) We introduce HEIGHT, a heterogeneous graph transformer that models robot-human, human-human, and obstacle-agent interactions for effective spatio-temporal reasoning and navigation policy learning.
3) We demonstrate that our method outperforms prior approaches in simulation and generalizes well to out-of-distribution environments, with a successful sim-to-real deployment in challenging scenarios.

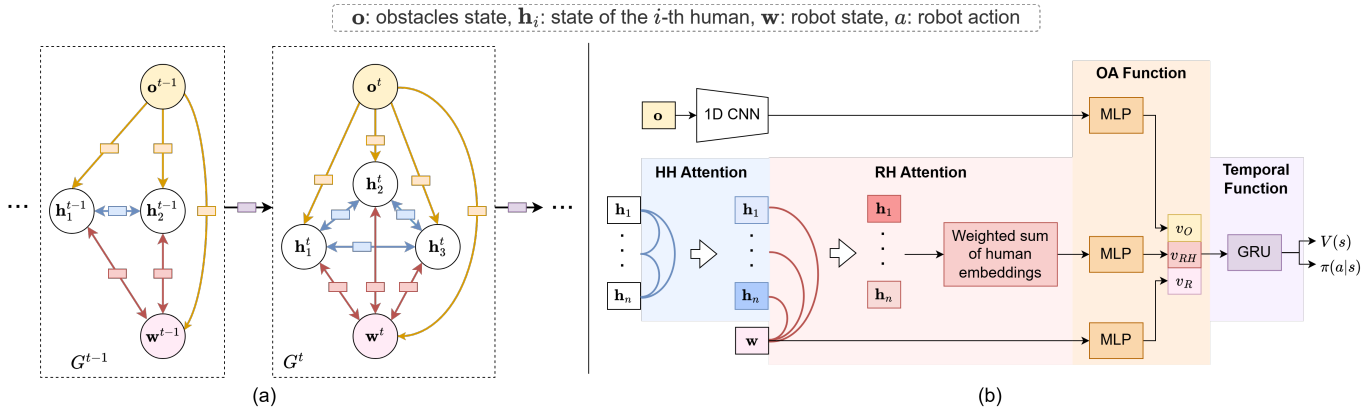

Fig. 2: **The heterogeneous st-graph and the HEIGHT network architecture.** (a) Graph representation of crowd navigation. The robot node is w (pink), the i-th human node is h_i (white), and the obstacle node is o (yellow). HH edges and HH functions are in blue, OA edges and OA functions are in orange, and RH edges and RH functions are in red. The temporal function is in purple. (b) HEIGHT network. Two attention mechanisms are used to model the HH and RH interactions. We use MLPs and a concatenation for obstacle-agent interactions, and a GRU for the temporal function. The superscript t that indicates the timestep and the human mask M is eliminated for clarity.

II. Preliminaries

Problem formulation: We formulate constrained crowd navigation as a Markov Decision Process (MDP) defined by $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, \mathcal{S}_0 \rangle$. The robot state w^t includes its position, velocity, heading, and goal. Each detected and untracked human h_i^t includes position and velocity, and static obstacles o^t are represented as a 2D point cloud. The full state is $s^t = [w^t, o^t, h_1^t, \ldots, h_n^t]$, where n varies by timestep and is determined by the number of detected humans. At each timestep, the robot selects an action $a^t = [a^t_{\text{trans}}, a^t_{\text{rot}}]$ from a discrete space, where a^t_{trans} and a^t_{rot} are the desired translational and rotational accelerations. The it transitions to s^{t+1} based on $\mathcal{P}(\cdot|s^t, a^t)$ while receiving a reward r^t. The goal is to maximize expected return $R^t = \mathbb{E}[\sum_{i=t}^{T} \gamma^{i-t} r^i]$. Episodes terminate upon the robot's goal arrival, collision, or timeout.

State representation: At each timestep t, we decompose the scene into two components: a set of observed humans $h_1^t, \ldots, h_n^t$ and a static obstacle point cloud o^t.

- *Human representation:* Each human is represented by a low-dimensional state vector extracted via off-the-shelf detectors [12]–[14]. This abstraction omits appearance and gait, which are difficult to simulate accurately and may introduce sim-to-real gaps.
- *Obstacle representation:* To reduce noise from dynamic entities and perception artifacts, we generate a 2D point cloud of static obstacles from a known map and robot pose using SLAM. This synthetic observation is consistent across simulation and real-world environments and avoids relying on noisy real-time sensor data.

This scene representation enables the robot to gain a structured view of the environment that is both generalizable and robust to domain shifts, and serves as input to our heterogeneous interaction graph.

Reward function: The reward is the sum of 3 components:

$$r(s^t, a^t) = r_{\text{main}}(s^t, a^t) + r_{\text{spin}}(s^t, a^t) + r_{\text{time}}. \tag{1}$$

In Eq. 1, the main reward r_{main} encourages reaching the goal and avoiding collisions:

$$r_{\text{main}}(s^t, a^t) = \begin{cases} 20, & \text{if } d^t_{\text{goal}} \le \rho_{\text{robot}} \\ -20, & \text{if } d^t_{\text{min}} \le 0 \\ d^t_{\text{min}} - 0.25, & \text{if } 0 < d^t_{\text{min}} < 0.25 \\ 4(d^{t-1}_{\text{goal}} - d^t_{\text{goal}}), & \text{otherwise} \end{cases} \tag{2}$$

where d^t_{goal} is the robot's distance to the goal, d^t_{min} is the minimum distance from the robot to any human, and ρ_{robot} is the robot radius.

The spin penalty r_{spin} discourages excessive rotation: $r_{\text{spin}}(s^t, a^t) = -0.05\|\omega^t\|_2^2$, where ω^t is the robot's rotational velocity. Finally, a small time penalty r_{time} encourages faster task completion: $r_{\text{time}} = -0.025$.

Intuitively, the robot gets a high reward when it approaches the goal with a high speed and a short and smooth path, while maintaining a safe distance from dynamic and static obstacles.

III. Methodology

We present a structured policy framework for robot navigation in crowded and constrained environments. Our approach decomposes the environment into a heterogeneous spatio-temporal graph (st-graph), capturing different types of interactions among agents and obstacles. This graph structure informs the design of our transformer-based policy network, which enables joint spatial and temporal reasoning. An overview of the st-graph and the network architecture is shown in Fig. 2.

Heterogeneous Spatio-Temporal Graph: In Fig. 2(a), at each timestep t, we construct a graph $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$ where the node set $\mathcal{V}^t$ includes the robot w^t, detected humans $h_1^t, ..., h_n^t$, and a single obstacle node o^t representing the static environment. The edge set $\mathcal{E}^t$ models three interaction types: human-human (HH), robot-human (RH), and obstacle-agent (OA). RH edges model direct interactions that affect robot decisions, HH edges capture indirect social influence, and OA edges reflect static environmental constraints. These edge types are associated with different neural modules and share parameters within each type to maintain scalability across varying numbers of agents. To model dynamics over time, we connect graphs across adjacent timesteps using a temporal

TABLE I: Baseline comparison results with different human and obstacle densities in unseen environments

Environment	Method	Success↑	Collision↓			Timeout↓	Nav Time↓	Path Len↓
			Overall	w/ Humans	w/ Obstacles			
Training distribution	A*+CNN [7]	0.64	0.29	0.28	**0.01**	0.07	25.72	12.30
5-9 humans	DRL-VO [15]	0.59	0.41	0.34	0.07	**0.00**	21.45	10.36
8-12 obstacles	HEIGHT (ours)	**0.88**	**0.12**	**0.09**	0.03	**0.00**	**18.31**	**10.34**
More crowded	A*+CNN [7]	0.47	0.42	0.39	**0.03**	0.11	27.33	12.47
10-14 humans	DRL-VO [15]	0.50	0.49	0.41	0.09	0.01	22.72	10.26
8-12 obstacles	HEIGHT (ours)	**0.78**	**0.22**	**0.19**	**0.03**	**0.00**	**19.69**	10.39
More constrained	A*+CNN [7]	0.48	0.29	0.23	**0.06**	0.23	27.28	13.11
5-9 humans	DRL-VO [15]	0.55	0.40	0.23	0.07	0.05	21.87	10.22
13-17 obstacles	HEIGHT (ours)	**0.84**	**0.15**	**0.07**	0.08	**0.01**	**18.79**	10.65

function that enables reasoning over motion continuity and partial observability.

HEIGHT Architecture: The network shown in Fig. 2(b) is derived from the heterogeneous st-graph. HH and RH interactions are modeled using multi-head attention. HH attention computes the importance of interactions among each pair of observed humans. Then, the humans are weighted again with RH attention based on their relevance to the robot. We apply binary masks to attention scores to handle varying visibility due to occlusion and limited sensor range. For obstacles, a 1D-CNN encodes the point cloud into a fixed-dimensional embedding. Robot states are processed through a linear layer. All embeddings—from RH attention, the robot, and obstacles—are concatenated and passed into a GRU to capture temporal dependencies. The GRU output is fed into fully connected layers that produce the state value $V(s^t)$ and policy logits $\pi(a^t|s^t)$.

Training: The network is trained end-to-end with Proximal Policy Optimization (PPO) in simulation. At each timestep, the policy outputs an action and value given the current state. During training, actions are sampled; during evaluation, the most likely action is taken. While our method does not require supervision, it can optionally incorporate imitation learning to improve sample efficiency and learning stability.

IV. Simulation Experiments

We evaluate our method in a simulation environment to answer two core questions: (1) How does our structured scene representation impact performance in crowded and constrained environments? (2) How important is the heterogeneous st-graph design in improving generalization?

Environment and Setup: We use a PyBullet-based simulator in a 12 m×12 m arena with randomized poses for the robot, static obstacles, and moving humans. Humans are controlled by ORCA, with a mix of reactive and non-reactive behaviors. Each episode runs up to 491 steps. Human and obstacle densities are varied to create a range of difficulty levels, including challenging out-of-distribution (OOD) scenarios.

We compare HEIGHT with two representative RL-based baselines that reflect common design choices for scene representation and policy architecture:

- **A*+CNN** [7]: A hybrid approach combining A* global planning with a CNN-based local RL policy. The inputs are a 2D LiDAR point cloud (which implicitly includes humans), the robot state, and A* waypoints. Humans are not explicitly detected or distinguished from obstacles.
- **DRL-VO** [15]: A hybrid method with a pure pursuit algorithm as the global planner and an RL local planner. The inputs are an occupancy map (OM) for humans, a 2D LiDAR point cloud, and the robot state. The input embeddings are fused without graph structures.

We report success rate, collision rate (human and obstacle), timeout rate, navigation time, and path length as metrics.

Results: Table I summarizes our comparison with prior RL-based methods under both training and challenging OOD settings. Our method consistently achieves the highest success rate and lowest collision rate across environments. In particular, HEIGHT demonstrates robust behavior in difficult scenes with high human or obstacle density. Fig. 3 shows that HEIGHT selects safer and more efficient paths to reach the goal and avoid dense human flows and static obstacles.

- *Effectiveness of Scene Representation:* A*+CNN treat humans and obstacles in a unified way as point clouds. This leads to ambiguity between dynamic and static objects, causing more collisions and poor generalization in unseen environments. The OM representation in DRL-VO is sparse and high-dimensional, which causes underfitting with the same training budget. However, since the size of OM does not change with number of humans, DRL-VO exhibits stronger generalization in OOD environments. In contrast to baselines, our split representation—with separate human detections and static-only obstacle point clouds—offers clear structural cues that help the policy generalize across different crowd and layout densities.
- *Effectiveness of Heterogeneous st-Graph:* Compared to approaches without an interaction graph, HEIGHT explicitly models RH, HH, and OA relationships using separate modules. This structure allows the policy to reason about different types of influence: yielding to nearby humans (RH), anticipating social dynamics (HH), and avoiding static obstacles and walls (OA). As shown in Fig. 3(f), this enables the robot to adjust its path to avoid emerging congestion, maintain safe distances, and exploit efficient routes in tight spaces—capabilities that are missing in unstructured baselines.

V. Real-World Experiments

In sim2real transfer, the robot is tested in a hallway and a lounge in a university building. The policy is directly trans-

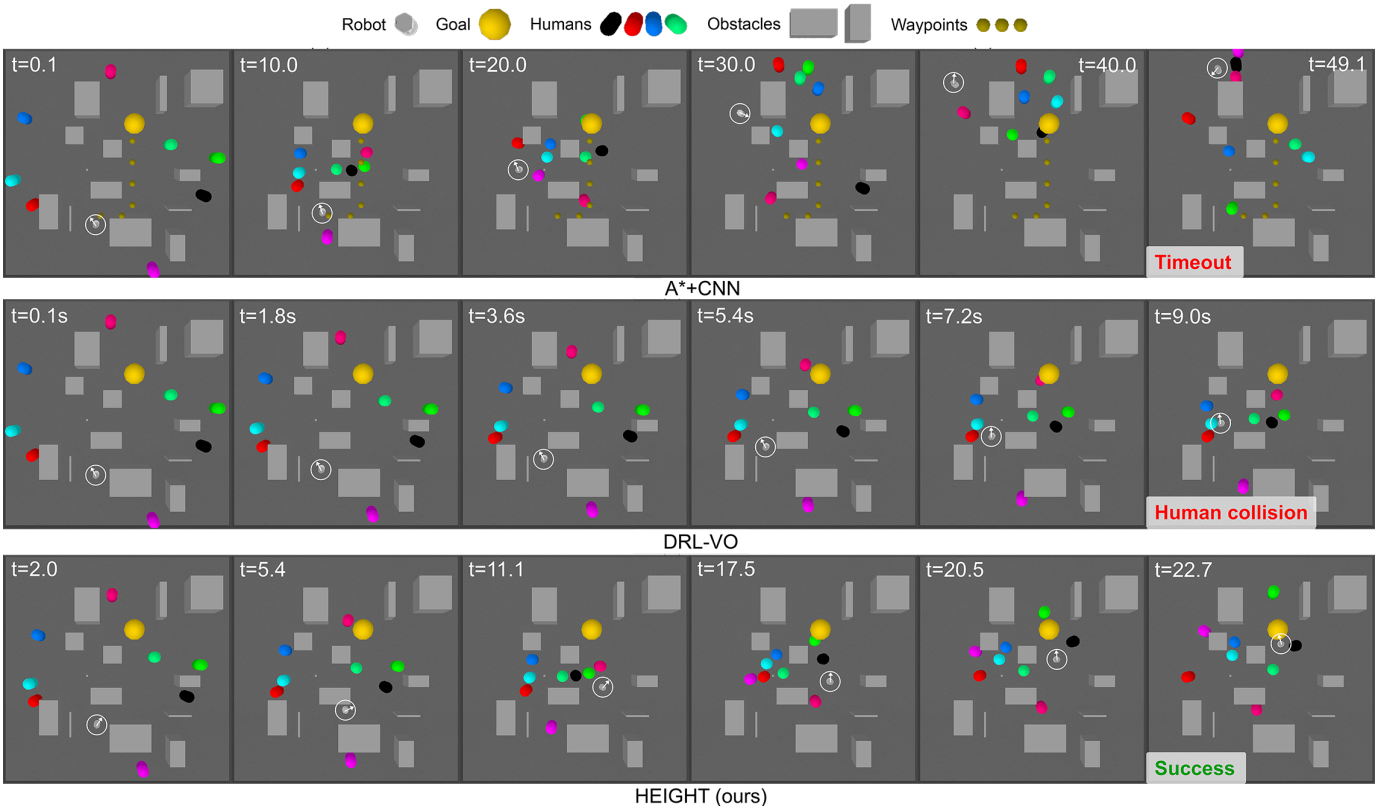

Fig. 3: **Comparison of different methods in the same testing episode in *More Constrained* environment.** The robot is centered in white circles and its orientation is denoted by white arrows. More qualitative results can be found in the video attachment and at https://sites.google.com/view/crowdnav-height/home.

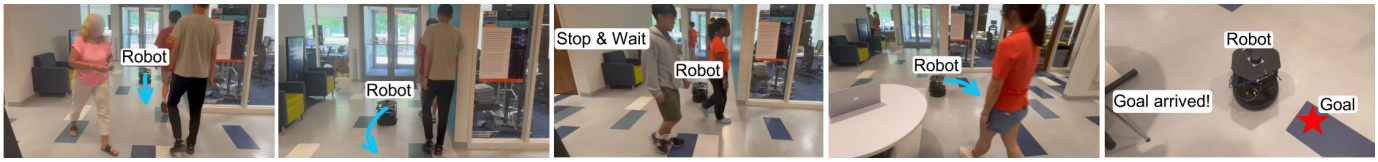

Fig. 4: **A testing episode of our method in the real Lounge environment.** The turtlebot avoids multiple groups of people who pass each other in different heading directions, avoids the walls and furnitures, and arrives at the goal.

TABLE II: Real-world results in two everyday environments

Environment	Method	Success↑	Nav Time↓
Hallway (1–2 humans)	Navigation Stack	0.72	**16.71**
	HEIGHT (ours)	**1.00**	22.36
Lounge (1–6 humans)	Navigation Stack	0.83	32.00
	HEIGHT (ours)	0.83	**30.71**

ferred from a low-fidelity PyBullet simulation. Our baseline is ROS Navigation Stack, which uses A* for global planning and Dynamic Window Approach (DWA) [5] as the local planner. Both humans and obstacles are treated as groups of circles in this baseline. In the **Hallway** environment, the baseline often times out due to aggressive re-planning and spinning behavior in tight spaces, while our method succeeds in all trials. In the **Lounge** environment, both methods achieve comparable success, but Navigation Stack takes longer by relying on stop-and-wait strategies. In contrast, HEIGHT reasons about dynamic and static interactions to generate more efficient motion. Fig. 4 shows an example episode where the robot successfully navigates across the lounge while avoiding multiple groups of pedestrians crossing in different directions. These results show that our structured policy can reason about dynamic and static interactions in real time. The successful deployment highlights the robustness of our input design and the effectiveness of sim2real learning with low-cost simulation and simple perception.

VI. Conclusion

In this article, we proposed HEIGHT, a structured robot navigation framework in dynamic and constrained environments. By leveraging the graphical nature and decomposability of navigation scenarios, we introduce a structured scene representation and RL policy network. These allows the robot to effectively reason about the different geometrics and dynamics of humans and obstacles, improving its ability to navigate complex environments. Our simulation experiments show that the HEIGHT model outperforms other learning-based methods in terms of collision avoidance, navigation efficiency, and generalization with varied human and obstacle densities. In real-world environments, HEIGHT is seamlessly deployed to everyday indoor navigation scenarios. Our work highlights the significance of uncovering the inherent structure of complex problems and injecting these structures into learning frameworks to solve the problems in a principled manner.

REFERENCES

[1] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[2] C. Mavrogiannis, K. Balasubramanian, S. Poddar, A. Gandra, and S. S. Srinivasa, "Winding through: Crowd navigation via topological invariance," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 121–128, 2023.

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

[4] Y. Yang, J. Jiang, J. Zhang, J. Huang, and M. Gao, "St^2: Spatial-temporal state transformer for crowd-aware autonomous navigation," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 912–919, 2023.

[5] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[6] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational graph learning for crowd navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[7] C. Pérez-D'Arpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot navigation in constrained pedestrian environments using reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1140–1146.

[8] 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)*, 2021, pp. 3517–3524.

[9] W. Wang, R. Wang, L. Mao, and B.-C. Min, "Navistar: Socially aware robot navigation with hybrid spatio-temporal graph transformer and preference learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 11 348–11 355.

[10] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. L. McPherson, J. Geng, and K. Driggs-Campbell, "Intention aware robot crowd navigation with attention-based interaction graph," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 015–12 021.

[11] B. Chen, H. Zhu, S. Yao, S. Lu, P. Zhong, Y. Sheng, and J. Wang, "Socially aware object goal navigation with heterogeneous scene representation learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 6792–6799, 2024.

[12] D. Jia, A. Hermans, and B. Leibe, "DR-SPAAM: A Spatial-Attention and Auto-regressive Model for Person Detection in 2D Range Data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 270–10 277.

[13] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[14] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *2017 IEEE international conference on image processing (ICIP)*, 2017, pp. 3645–3649.

[15] Z. Xie and P. Dames, "Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2700–2719, 2023.