Socially Acceptable Bipedal Robot Navigation via Social Zonotope Network Model Predictive Control

Abdulaziz Shamsah, Krishanu Agarwal, Nigam Katta, Abirath Raju, Shreyas Kousik, *Member, IEEE*, and Ye Zhao, *Senior Member, IEEE*

***Abstract*— This study addresses the challenge of social bipedal navigation in a dynamic, human-crowded environment, a research area largely underexplored in legged robot navigation. We present a zonotope-based framework that couples prediction and motion planning for a bipedal ego-agent to account for bidirectional influence with the surrounding pedestrians. This framework incorporates a Social Zonotope Network (SZN), a neural network that predicts future pedestrian reachable sets and plans future socially acceptable reachable set for the ego-agent. SZN generates the reachable sets as zonotopes for efficient reachability-based planning, collision checking, and online uncertainty parameterization. Locomotion-specific losses are added to the SZN training process to adhere to the dynamic limits of the bipedal robot that are not explicitly present in the human crowds data set. These loss functions enable the SZN to generate locomotion paths that are more dynamically feasible for improved tracking. SZN is integrated with a Model Predictive Controller (SZN-MPC) for footstep planning for our bipedal robot Digit. SZN-MPC solves for collision-free trajectory by optimizing through SZN's gradients. Our results demonstrate the framework's effectiveness in producing a socially acceptable path, with consistent locomotion velocity, and optimality. The SZN-MPC framework is validated with extensive simulations and hardware experiments.**

***Note to Practitioners*—This paper is motivated by the challenge of navigating bipedal robots through dynamic, human-crowded environments in a socially acceptable manner. Existing methods for social navigation often only address obstacle avoidance and are demonstrated on a robot with simple dynamics. This paper proposes the Social Zonotope Network (SZN), a novel neural network that couples pedestrian future trajectory prediction and robot motion planning to facilitate socially aware navigation for bipedal robots such as Digit, designed by Agility Robotics. The social behaviors are learned from real open-sourced pedestrian data using the SZN, which outputs the future predictions as reachable sets for each agent in the environment. The SZN is then integrated into a trajectory optimization problem that takes into account personal space preferences and bipedal robot capabilities to design trajectories that are both collision-free and socially acceptable. This work also highlights the computational efficiency of the SZN design that makes it suitable for real-time integration with motion planners. The framework is validated through extensive simulations and hardware experiments. From a practical standpoint, this research provides a framework that can be applied to bipedal robots to improve automation in human-populated environments such as hospitals, shopping centers, and airports. The framework's ability to automatically adapt to surrounding human movement helps minimize disruptions and ensures that the robot's presence is not a hindrance to the flow of human traffic. Future work will focus on outdoor deployment, which will require onboard perception capabilities to detect surrounding pedestrians.**

***Index Terms*— Social navigation, bipedal robot locomotion, reachability, collision avoidance, path planning, neural networks.**

Received 15 October 2024; accepted 5 December 2024. This article was recommended for publication by Associate Editor Z. Ma and Editor D. Song upon evaluation of the reviewers' comments. This work was supported in part by the Office of Naval Research (ONR) under Grant N000142312223; in part by the National Science Foundation (NSF) under Grant IIS-1924978, Grant CMMI-2144309, and Grant FRR-2328254; and in part by United States Department of Agriculture (USDA) under Grant 2023-67021-41397. *(Shreyas Kousik and Ye Zhao are co-senior authors.) (Corresponding author: Ye Zhao.)*

Abdulaziz Shamsah is with George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA, and also with the Mechanical Engineering Department, College of Engineering and Petroleum, Kuwait University, Safat 13060, Kuwait (e-mail: ashamsah3@gatech.edu).

Krishanu Agarwal and Nigam Katta are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30308 USA.

Abirath Raju is with the Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology, Atlanta, GA 30308 USA.

Shreyas Kousik and Ye Zhao are with George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ye.zhao@me.gatech.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2024.3519012.

Digital Object Identifier 10.1109/TASE.2024.3519012

I. Introduction

BIPEDAL locomotion in complex environments has garnered substantial attention in the robotics community [4], [5], [6], [7], [8], [9], [10], [11]. Social navigation poses a particular challenge due to the inherent uncertainty of the environment, unknown pedestrian dynamics, and implicit social behaviours [12]. Recently, there has been an increasing focus on social navigation for mobile robots in human environments [13], [14], [15], [16], [17], [18], [19], [20]. Using mobile robots benefits from stable motion dynamics, which facilitate the investigation of high-level social path planners. In contrast, the exploration of social navigation for bipedal robots remains unexplored. This is largely attributed to the intricacies of the hybrid, nonlinear, and high degree-of-freedom dynamics associated with bipedal locomotion. The complexities inherent in stabilizing and controlling bipedal robots have led researchers to prioritize fundamental locomotion research in the past [21], [22], [23], [24], delaying advancements in deploying these robots into more complex dynamic environments such as human-populated ones (see Fig. 1).

In this study, we present an integrated framework for prediction and motion planning for socially acceptable bipedal navigation as shown in Fig. 2. We propose a Social Zonotope

Network (SZN) that both predicts pedestrians' future trajectories and plans a socially acceptable path for the bipedal robot Digit [25] with 28 degrees of freedom (DoFs). SZN is trained with locomotion-specific losses to improve full-body tracking of the desired socially acceptable paths.

Our SZN model outputs reachable sets for both pedestrians and the ego-agent represented as zonotopes, a class of convex symmetric polytopes. Zonotopes enable efficient and robust reachability-based planning, collision checking, and uncertainty parameterization [26], [27], [28], [29]. In this work, we detect and avoid collisions between zonotopes corresponding to the ego-agent and pedestrians. Additionally, the ego-agent's zonate for (1) robot modeling errors of the ego-agent's dynamics via Gaussian process (GP) regression, and (2) ego-agent's personal space for increased social acceptance in practice. Zonotopes provide a balance between geometric complexity and computational efficiency. We specifically take advantage of two facts: (1) the Minkowski sum of two zonotopes is again a zonotope, allowing us to easily augment the zonotopes output by a neural network; and (2) collision checking a pair of zonotopes can be differentiated for use in gradient-based motion planning methods.

Our framework integrates SZN in a model predictive controller (MPC) at the middle level as shown in Fig. 2. SZN-MPC optimizes over the output of the neural network by encoding it as reachability and collision avoidance constraints. We incorporate a novel cost function designed to encourage the generation of socially acceptable trajectories. SZN-MPC employs a reduced-order model (ROM), i.e., a Linear Inverted Pendulum (LIP) model, for the bipedal locomotion process, and then sends optimal commands, i.e., center of mass (CoM) velocity and heading change, to the low-level controller on Digit for full-body joint trajectory design and control.

A. Contributions and Outline

The main contributions of this study are as follows.

1) A reachability-based prediction and planning framework for bipedal navigation in a social environment: we introduce the Social Zonotope Network (SZN), a CVAE architecture for coupled pedestrian future trajectory prediction and ego-agent social planning both parameterized as zonotopes.
2) Learning locomotion safety: we encode locomotion safety into the learning module using signal temporal logic (STL) specifications, and design its robustness score as loss functions of the CVAE network during training.
3) Online zonotope refinements for social acceptability and model discrepancy compensation: we leverage the adaptable nature of zonotope parameterization to enhance social acceptability by incorporating personal space refinements and to efficiently adjust online for learned model discrepancies between the reduced-order model (ROM) and the full-order model of Digit.
4) SZN-MPC: we integrate the SZN with MPC, where the zonotopes outputted by SZN are encoded as constraints for reachability-based motion planning and collision checking.

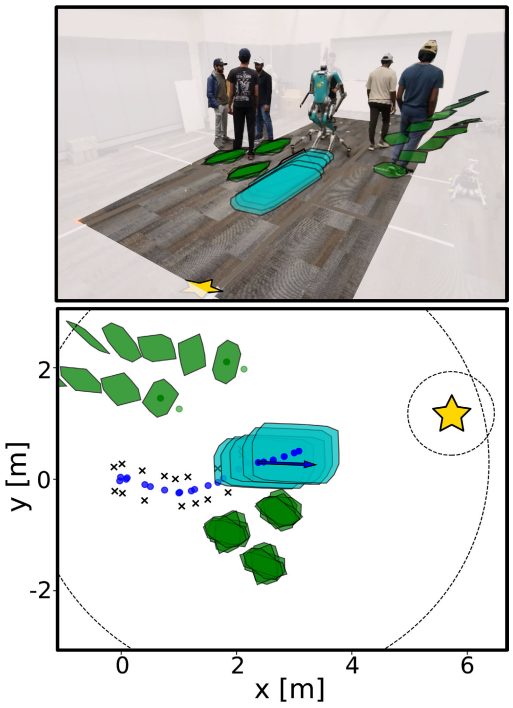


Fig. 1. (top) Snapshot of the proposed social path planner demonstrated on hardware with 5 pedestrians with pedestrian's prediction zonotopes (green), ego-agent's social zonotope (cyan), and goal location (yellow star) superimposed. (bottom) shows a top-down view of the ego-agent's path, pedestrians' prediction, and social zonotopes.

5) Social acceptability cost function design: a novel MPC cost function to plan trajectories for the ego-agent using learned paths from human data sets.
6) Experimental hardware evaluation: we implement the proposed framework on a bipedal robot Digit equipped with a low-level passivity controller for full-body joint control.

This article is organized as follows. Sec. II reviews related work. Sec. III introduces the robot dynamics, environment setup, and our problem statement. Zonotope preliminaries are in Sec. IV. Sec. V presents SZN's architecture and loss functions. Sec. VI provides zonotope refinements for social acceptability and uncertainty parameterization. Sec. VII formulates the problem as a MPC and introduces the social acceptability cost. Implementation details and results are in Sec. VIII. In Sec. IX we discuss limitations of the proposed framework. Finally, Sec. X concludes the article.

This paper expands upon a previous conference version [30]. The work presented here extends the SZN training to include additional locomotion-specific loss functions, incorporates a social acceptability cost function in the SZN-MPC, and introduces a coupled SZN-MPC for simultaneous pedestrian prediction and ego-agent planning. Additionally, this paper refines the zonotope parameterization by incorporating personal space modulation and uncertainty parameterization. We also benchmark the collision-avoidance performance of our method with a control barrier function baseline and validate our framework by extensive hardware experiments that were not included in the conference version.

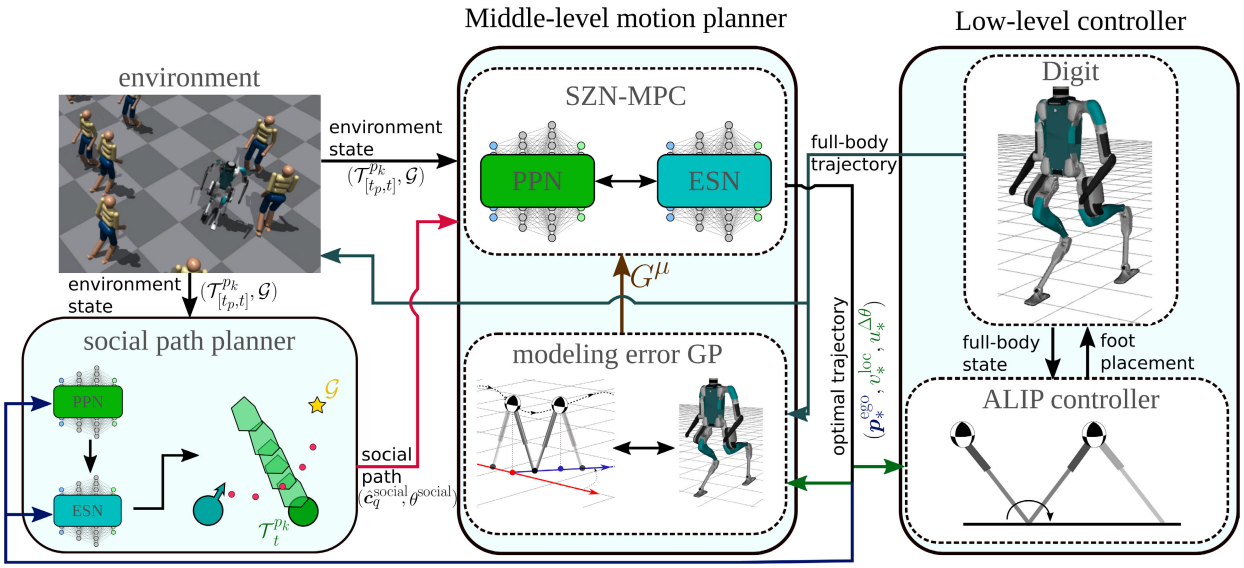


Fig. 2. Block diagram of the proposed framework. The framework is developed around the Social Zonotope Network (SZN) (Sec. V), which is composed of two sub-networks: the Pedestrian Prediction Network (PPN) and the Ego-agent Social Network (ESN) shown in green and cyan, respectively. Given an environment with observed pedestrians $\mathcal{T}_t^{p_k}$ (green circle) and a goal location $\mathcal{G}$ (yellow $\star$), the social path planner designs a social path for Digit (red dots) (Sec. VII-C). At the middle level, SZN-MPC optimizes through SZN to generate both collision-free and socially acceptable trajectories for Digit (Sec. VII). The optimal trajectory is then sent to the ALIP controller [1] to generate the desired foot placement for reduced-order optimal trajectory tracking. An ankle-actuated-passivity-based controller [2], [3] is implemented on Digit for full-body trajectory tracking. Digit current velocity and the optimal trajectory from SZN-MPC are used in the modeling error GP to compensate for the modeling uncertainty between ROM dynamics and full-order dynamics (Sec. VI-B).

II. Related Work

This work lies in the intersection of social navigation and bipedal locomotion. We now review these topics.

A. Social Navigation

Social navigation literature can be categorized into two main categories: coupled prediction and planning and decoupled prediction and planning [12]. We further discuss methods for predicting human motion, then the specific methods that we build upon.

1) Coupled Prediction and Planning: In coupled prediction and planning literature, the proposed frameworks recognize the influence of the robot motion on the surrounding pedestrians and vice versa [12]. Navigating an environment with humans in a socially compliant manner requires a proactive approach to motion planning [17], [19], [31], [32]. In [19], the authors use coupled opinion dynamics to proactively design motion plans for a mobile robot, without the need for human prediction models. This is an implicit approach, as it relies only on the observation of the approaching human position and orientation to form an opinion that alters the nominal path and avoids collisions with pedestrians. Another implicit approach is [33], which uses inverse reinforcement learning to learn robot motions that mimics human behavior.

On the other hand, the approaches in [17] and [31] explicitly predict the future trajectories of pedestrians. The authors in [31] propose a social interference metric based on Kullback-Leibler divergence to measure the interference of the robot's path plan on the surrounding humans' future trajectory, hypothesizing that minimizing the social interference metric will result in a socially acceptable trajectory for the ego-agent. Similarly, a gradient-based trajectory optimization is introduced in [17] to minimize the difference between the humans' future path prediction conditioned on the robot's plan and the nominal prediction. Both of these studies work under the assumption that a minimally-invasive robot trajectory, with minimal effect on surrounding humans' nominal trajectory, is socially acceptable (see Fig. 3(a)). In contrast, our work leverages a different notion: we aim to learn the socially acceptable trajectory of the ego-agent from human crowd datasets to avoid any heuristics or biases on prioritizing the minimal invasiveness of the pedestrians. Our method enables bidirectional influence between the pedestrians and the ego-agent, and allows the ego-agent to change surrounding pedestrians' paths (see Fig. 3(b)).

2) Decoupled Prediction and Planning: In decoupled prediction and planning, social navigation can be viewed as dynamic obstacle avoidance [12], [34], [35]. For example, the work in [36] ignores any effect the robot path has on the surrounding pedestrians. They propose a linear Kalman filter to predict the pedestrians' future paths, then use MPC for collision avoidance. As another example, a decoupled approach can be used for a robot to approach a group of pedestrians in a socially aware way [37]. While effective for a small number of pedestrians, ignoring the coupling effect between the robot and the pedestrians can result in two problems: first, reciprocal dance, meaning an oscillatory interaction between the robot and pedestrians [38]; and second, the so-called "freezing robot problem" [39], wherein the robot comes to a stop to avoid collisions.

To overcome the aforementioned challenges, our framework uses a coupled prediction and planning approach, as the pedestrians' future predictions are conditioned on the ego-agent's future planned motion. Our approach integrates both

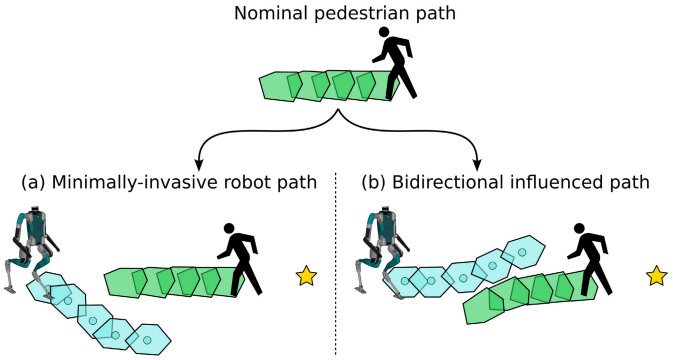


Fig. 3. Comparison between two different designs of socially acceptable paths. (a) shows a minimally invasive path design for the robot as in [31] and [17], i.e., the robot adjusts its path to not alter the pedestrian's original path or change it in a minimally-invasive way. (b) shows the bidirectionally influenced path between the robot and the pedestrian that our work employs, i.e., the robot and the pedestrian mutually react to each other and adjust their own paths accordingly.

explicit and implicit elements. Explicitly, we predict the future trajectories of pedestrians and utilize these predictions for collision avoidance. Implicitly, we leverage learned trajectories from pedestrian datasets to generate ego-agent trajectories that mimic human behavior.

3) Human Trajectory Prediction: Our framework is inspired by the human trajectory prediction community, such as Trajectron++ [40], SocialGAN [41], PECNet [42], Y-net [43], Sophie [44], and STAR [45] where we aim to design a socially acceptable trajectory for the ego-agent that mimics the path from human crowd datasets. The work in [46] proposes an obstacle avoidance learning method that uses a Conditional Variational Autoencoder (CVAE) framework to learn a temporary, near-horizon target distribution to avoid pedestrians actively. However, during training, the temporary targets are selected heuristically. In contrast, we aim to learn such temporary waypoints from human crowd datasets to capture a heuristic-free socially acceptable path. In [42], the authors develop a simple but accurate CVAE architecture based on Multi-Layer Perceptrons (MLP) networks to predict crowd trajectories conditioned on past observations and intermediate endpoints. Our SZN inherits a similar MLP-based CVAE architecture, where the ego-agent path is conditioned on the final goal location and surrounding pedestrians' future trajectories. In addition, our SZN is conditioned on the immediate change in the ego-agent state to be better integrated with MPC for planning. Importantly, we find that the MLP network architecture enables real-time gradient-based ego-agent motion planning coupled with pedestrian prediction, which is challenging with more advanced architectures such as LSTMs [26], [40] or Transformers [45].

4) Direct Predecessors to Our Work: Our work builds off of the methods in [17] and [26], which leverage neural network gradients in trajectory optimization (TO) for safe motion planning. The method in [17] employs [40] for generating multimodal probabilistic predictions and integrates the prediction model gradients in the cost function of the TO problem, to minimize the invasiveness of the robot's path to the surrounding pedestrians' paths (as previously discussed in Sec. II-A); similarly, we leverage neural network gradients in TO. Our prior work [26] presents a Zonotope Alignment of Prediction and Planning (ZAPP) that relies on zonotopes to enable continuous-time reasoning for planning, just as we do in this work. This method uses Trajectron++ [40] to predict obstacle trajectories via a Gaussian mixture model, then constructs a zonotope over this distributions to overapproximate the non-ego agents' reachable sets. The predictive model's gradients are used in TO for obstacle avoidance [26], rather than in the cost function as designed in [17].

Our key insight, which distinguishes this work from the prior work [17], [26], is to learn path prediction distributions directly as zonotopes, bypassing the initial step of predicting Gaussian distributions for pedestrian motion. This makes our approach more computationally efficient and facilitates real-time integration with MPC, where both [17] and [26] struggled to achieve real-time implementation. Our learned zonotopes also provide gradients for constraints in MPC, enabling reachability-based planning and collision checking.

B. Bipedal Locomotion

Bipedal locomotion has been extensively studied with a wide spectrum of model representations and methods—such as reduced-order models (ROM) [5], [6], [7], [47], single rigid body models [48], centroidal models [49], whole-body models [23], [24], [50], ROM-inspired reinforcement learning [51], and model-free reinforcement learning [52], [53]—to name a few. Generally, locomotion planning using whole-body models results in high computation cost [54], [55] and becomes much less efficient for navigation in complex environments that often involve long-horizon planning; therefore, in this work, we focus on ROM-based methods for CoM trajectory and foot placement planning.

Other methods also use ROM-based methods for planning. For example, the work in [5] uses an omnidirectional differential-drive wheeled robot model with a preference for sagittal movement to capture bipedal robots' behaviors accurately. They design a control Lyapunov function (CLF) to drive and orient that robot towards the goal and a control barrier function (CBF) for obstacle avoidance. The framework is demonstrated for indoor and outdoor navigation in environments with static obstacles only. In our work, we use a Linear Inverted Pendulum (LIP) [56] as the ROM for our bipedal robot. A similar model is used in [6], while [7] and [57] use an angular momentum-based LIP (ALIP) [1] for improved prediction of CoM velocities.

There also exist other approaches that seek to generate safe bipedal motion in complex scenes. For example, the method in [6] present discrete-time CBF (DCBF) [58] constraints with a sequential LIP-based MPC to plan trajectories for Digit in an environment with static and dynamic obstacles. However, the dynamic obstacles trajectories are assumed to be known in [6] and the framework is only demonstrated in simulation. The work in [57] similarly uses a DCBF-MPC to avoid static obstacles in simulation.

In contrast to the aforementioned studies, we develop a bipedal locomotion planner that generates safe paths while

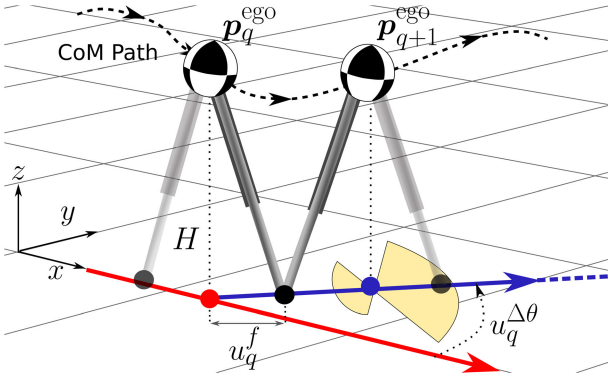


Fig. 4. Illustration of the Linear Inverted Pendulum model for two consecutive foot contact switching states $\boldsymbol{x}_q$ and $\boldsymbol{x}_{q+1}$. The shaded yellow regions indicate the kinematics constraint on the control input $\boldsymbol{u}$ detailed in Sec. VII-A.

predicting pedstrian motion. This enables considering social norms and can be efficiently solved in real-time, as we show via hardware implementation.

III. Problem Formulation

We seek to safely navigate a bipedal robot through a crowd. We now present our robot model and environment model, and then formalize this problem.

A. Walking Robot Model

We leverage a locomotion-specific reduced order model (ROM) to describe our bipedal robot dynamics in this study. Consider a bipedal ego-agent with discrete time dynamics $\boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q)$, where $\boldsymbol{x}_q$ and $\boldsymbol{u}_q$ are the state and control input respectively at the q^{th} walking step.[1] The state of the bipedal robot, i.e., the ego-agent, is $\boldsymbol{x} = (\boldsymbol{p}^{\text{ego}}, v^{\text{loc}}, \theta)$ where $\boldsymbol{p}^{\text{ego}} = (x, y)$ is the 2-D location in the world coordinate, v^{loc} is the sagittal velocity at the foot contact switching instant in the local coordinate, and θ is the heading. The control input is $\boldsymbol{u}_q = (u_q^f \quad u_q^{\Delta\theta})$, where u_q^f is the sagittal foot position relative to the CoM, and $u_q^{\Delta\theta}$ is the heading angle change between two consecutive walking steps. A schematic robot model is shown in Fig. 4.

ROM Dynamics: We use a linear inverted pendulum (LIP) model [56] as a ROM to design the 3-D walking motion of Digit For the LIP model we assume that each walking step has a fixed duration[2] T [6], [57]. Then we build our model on the discrete sagittal dynamics[3] $(\Delta x_q^{\text{loc}}, v_q^{\text{loc}})$, where x_q^{loc} is CoM position at the beginning of the q^{th} step, $\Delta x^{\text{loc}} = x_{q+1}^{\text{loc}} - x_q^{\text{loc}}$ is the local sagittal CoM position difference between two consecutive walking steps, and v_q^{loc} is the sagittal velocity at the local coordinate for the q^{th} walking step as shown in Fig. 4 (see Appendix. A for detailed derivation):

$$\Delta x^{\text{loc}}(u_q^f) = \left(v_q^{\text{loc}} \frac{\sinh(\omega T)}{\omega} + (1 - \cosh(\omega T)) u_q^f\right) \tag{1}$$

$$v_{q+1}^{\text{loc}}(u_q^f) = \cosh(\omega T) v_q^{\text{loc}} - \omega \sinh(\omega T) u_q^f \tag{2}$$

where $\omega = \sqrt{g/H}$, g is the gravitational constant, and H is the constant CoM height. The heading angle change is governed by $u_q^{\Delta\theta} = \theta_{q+1} - \theta_q$. Based on the sagittal dynamics (1) and (2), we introduce coordinate transformation based on the heading angle θ_q to control the LIP dynamics in 2-D Euclidean space. Therefore the full LIP dynamics in 2-D Euclidean space become:

$$x_{q+1} = x_q + \Delta x^{\text{loc}}(u_q^f)\cos(\theta_q) \tag{3a}$$

$$y_{q+1} = y_q + \Delta x^{\text{loc}}(u_q^f)\sin(\theta_q) \tag{3b}$$

$$v_{q+1}^{\text{loc}} = \cosh(\omega T) v_q^{\text{loc}} - \omega \sinh(\omega T) u_q^f \tag{3c}$$

$$\theta_{q+1} = \theta_q + u_q^{\Delta\theta} \tag{3d}$$

A detailed derivation of (3) is in Appendix. A. For notation simplicity, and hereafter, (3) will be referred to as:

$$\boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q) \tag{4}$$

Later on, (4) will be enforced as dynamic constraints in our proposed SZN-MPC in Sec. VII.

B. Environment Assumptions and Observations

In this work, we hypothesize that, in a social setting, a human determines their future path using three pieces of information: (i) their final destination $\mathcal{G} = (x^{\text{dest}}, y^{\text{dest}})$ (navigation intent), (ii) the surrounding pedestrians' past trajectory[4] $\mathcal{T}_{[t_p,t]}^{p_k} = \{x_q^{p_k}, y_q^{p_k}\}_{q=t_p}^{t}, \forall k$, where k indexes the k^{th} pedestrian, and (iii) their prior social experience, meaning assumptions on how to navigate the environment in a socially-acceptable manner. We treat the social experience as latent information that is implicit in human crowd datasets:

Assumption 1: Suppose we learn a model of the future trajectory of a human as a function of their final goal $\mathcal{G}$ and surrounding pedestrians' past trajectories $\mathcal{T}_{[t_p,t]}^{p_k}$. We assume that this model will implicitly represent each human's social experience.

In this work, to learn a socially acceptable future path for an ego-agent $\mathcal{T}_{[t,t_f]}^{\text{ego}} = \{x_q^{\text{ego}}, y_q^{\text{ego}}\}_{q=t}^{t_f}$, we use real human crowd datasets, and substitute a single human for the ego-agent.

Only the pedestrians within a prespecified radius of the ego-agent are observable, and we assume that their past trajectories were observable over a specified time interval from t_p to t.

C. Problem Setup and Statement

The ego-agent is tasked to navigate to a known goal location $\mathcal{G}$ in an open environment with $m \in \mathbb{N}$ observed pedestrians treated as dynamic obstacles. Denote the pedestrian state $\mathcal{T}_{[t_p,t]}^{p_k}$, which is the 2-D trajectory of k^{th} pedestrian observed

[1]The robot model used in our study represent step-by-step dynamics, i.e., $\boldsymbol{x}_q$ and $\boldsymbol{x}_{q+1}$ are the CoM state at the foot contact switching instant of two consecutive walking steps.

[2]Set to be equal to the timestep between frames in the dataset (0.4 s).

[3]The lateral dynamics are only considered in the ALIP model at the low level since they are periodic with a constant desired lateral foot placement (see Fig. 2).

[4]The subscripts t_p, t, and t_f represent discrete time indices denoting the past, current and future trajectories, respectively, where $t_p < t < t_f$.

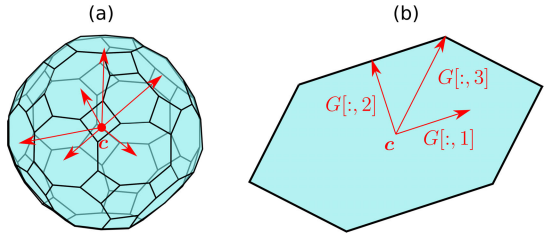


Fig. 5. An illustration of zonotopes: (a) a 3-D zontope ($n = 3$) with $n_G = 13$ (b) a 2-D zonotope ($n = 2$) with $n_G = 3$. Red arrows indicate the generators in G, with only 6 out of 13 generators are illustrated in (a). In this study, we will use the 2-D zonotopes for our reachability path design.

over a discrete time interval $[t_p, t]$, where $t_p < t$. The environment is partially observable, as only the pedestrians in a pre-specified sensory radius of the ego-agent are observed. The path the ego-agent takes should ensure locomotion safety and navigation safety while being socially acceptable.

Definition 2 (Locomotion Safety): For a bipedal robot, locomotion safety means maintaining balance dynamically throughout its locomotion process.

Definition 3 (Navigation Safety): Navigation safety means avoiding collisions with pedestrians. We model this as a constraint, $\|\boldsymbol{p}_t^{\text{ego}} - \mathcal{T}_t^{p_k}\| > d,\ \forall\, t, k \in m$, where $\boldsymbol{p}_t^{\text{ego}}$ denotes the ego-agent 2-D position and d represents a minimum allowable distance between the ego-agent and the pedestrians.

Definition 4 (Socially Acceptable Path for Bipedal Systems): A path that a bipedal ego-agent takes in a human-crowded environment is socially acceptable if it has an Average Displacement Error (ADE) $< \epsilon$ when compared to ground truth human data navigating in the same environment.[5]

Based on the aforementioned definitions and environment setup, we seek to solve the following problem:

Problem 5: Given a bipedal robot modeled by discrete dynamics $\boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q)$ and a partially-observable environment state $\mathcal{E} = (\mathcal{T}_{[t_p,t]}^{p_k}, \mathcal{G})$, find an optimal motion plan that ensures locomotion and navigation safety while promoting social acceptability.

IV. Zonotope Preliminaries

A zonotope $\mathcal{Z} \in \mathbb{R}^n$ is a convex, symmetrical polytope parameterized by a center $\boldsymbol{c} \in \mathbb{R}^n$ and a generator matrix $G \in \mathbb{R}^{n \times n_G}$ (see Fig. 5).

$$\mathcal{Z} = \mathscr{Z}(\boldsymbol{c}, G) = \{\boldsymbol{c} + G\beta \mid \|\beta\|_\infty \leq 1\} \tag{5}$$

The Minkowski sum of $\mathcal{Z}_1 = \mathscr{Z}(\boldsymbol{c}_1, G_1)$ and $\mathcal{Z}_2 = \mathscr{Z}(\boldsymbol{c}_2, G_2)$ is $\mathcal{Z}_1 \oplus \mathcal{Z}_2 = \mathscr{Z}(\boldsymbol{c}_1 + \boldsymbol{c}_2, [G_1\ G_2])$. To Check collisions between two zonotopes, [59, Lemma 5.1] is used:

Proposition 6 [59, Lemma 5.1]: $\mathcal{Z}_1 \cap \mathcal{Z}_1 = \emptyset$ iff $\boldsymbol{c}_1 \notin \mathscr{Z}(\boldsymbol{c}_2, [G_1\ G_2])$

Per [29, Theorem 2.1], zonotopes can be parameterized using a half-space representation $\mathcal{P} = \{x \mid Ax \leq b\}$, where $x \in \mathcal{P} \iff \max(Ax - b) \leq 0$ and $x \notin \mathcal{P} \iff \max(Ax - b) > 0$ (see Fig. 5(b)), which we show is useful for collision checking. In the special case of a 2-D zonotope, the center-generator representation to the half-space representation is given analytically as follows:

Proposition 7 [26, Proposition 2]: Let $C = \begin{bmatrix} -G[2,:] & G[1,:] \end{bmatrix}$ and $l_G[i]$ be the norm of the i^{th} generator $l_G[i] = \|G[:,i]\|_2$, the half-space representation of a 2-D zonotope:

$$A[i,:] = \frac{1}{l_G[i]} \cdot \begin{bmatrix} C \\ -C \end{bmatrix} \in \mathbb{R}^{2n_G \times 2} \tag{6}$$

$$b = A \cdot \boldsymbol{c} + |AG|\ 1_{m \times 1} \in \mathbb{R}^{2n_G} \tag{7}$$

where $i = 1, \ldots, n_G$ indexes the number of generators.

In this work, we use zonotopes to represent the social reachable set of each agents, i.e., the ego-agent and pedestrians. We seek to learn a sequence of social zonotopes for the ego-agent $\mathcal{Z}_q^{\text{ego}}$, each of which contains two consecutive waypoints of the ego-agent's future social trajectory $\mathcal{T}_{[t,t_f]}^{\text{ego}}$, thereby approximating the agent's continuous-time motion similar to [26].

Definition 8 (Social Zonotope $\mathcal{Z}_q^{\text{ego}}$): A social zonotope for the ego-agent's q^{th} step is $\mathcal{Z}_q^{\text{ego}} = \mathscr{Z}(\boldsymbol{c}_q, G_q)$, satisfying that the future traj $\ldots \mathcal{T}_{[t,t_f]}^{\text{ego}} \in \bigcup_{q=t}^{t_{f-1}} \mathcal{Z}_q^{\text{ego}}$.

V. Social Zonotope Network

This section introduces the Social Zonotope Network (SZN) architecture and the loss functions used during training, which are designed both for shaping the social zonotopes and for ensuring the physical viability of the path for bipedal locomotion. The key feature of our social zonotope network is to learn the zonotope representation directly as an output of the neural network enabling real-time reachability-based planning and collision avoidance in the MPC introduced later in Sec. VII.

A. Learning Architecture

We use a conditional variational autoencoder (CVAE) architecture to learn the ego-agent's future trajectory conditioned on the final destination goal, the immediate change in the ego-agent's state, and the surrounding pedestrians' past trajectories. The proposed architecture uses Multi-Layer Perceptrons (MLP) with ReLU non-linearity for all the sub-networks.

Our Social Zonotope Network (SZN) is comprised of two coupled neural networks to not only predict the reachable set of the surrounding pedestrians, but also learn the social reachable set of the ego-agent as shown in Fig. 6.

1) Pedestrian Prediction Network (PPN): Our pedestrian prediction network (shown in Fig. 6(a)) is inspired by PECNet [42], where the endpoint of the predicted pedestrian trajectory $\mathcal{T}_{t_f}^{p_k}$ ($t_f = 8$ indicating a 8-step horizon) is learned first, and then the future trajectory for the $[t, t_f]$ horizon is predicted. Our proposed network deviates from PECNet in three ways. First, the pedestrian future trajectory is also conditioned on the immediate change in the ego-agent's state $\mathcal{T}_{t+1}^{\text{ego}}$ (shown in red in Fig. 6(a)). This coupling of the pedestrian prediction and ego-agent planning networks is intended to capture the effect of the ego-agent's control

[5] ϵ represents the allowable deviation from the socially acceptable path. The Average Displacement Error denotes the average error between the planned path and the ground-truth path.

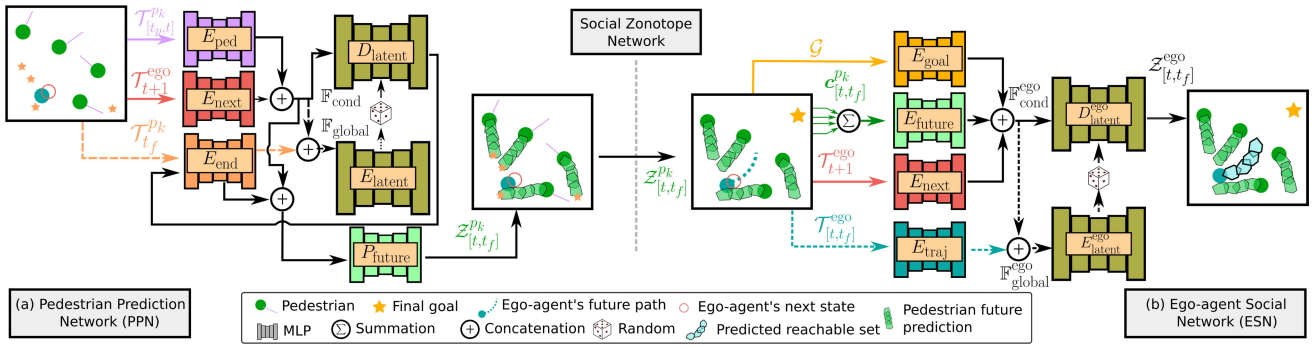


Fig. 6. The Social Zonotope Network (SZN) is comprised of two coupled neural networks to both predict the reachable set of the surrounding pedestrians and learn the social reachable set of the ego-agent. (a) shows the pedestrian prediction network, conditioned on the pedestrian endpoints ($\mathcal{T}_{t_f}^{p_k}$) and the immediate change in the ego-agent's state ($\mathcal{T}_{t+1}^{\text{ego}}$). (b) shows the ego-agent social network conditioned on the pedestrians' future prediction ($\boldsymbol{c}_{[t,t_f]}^{p_k}$), the immediate change in the ego-agent's state ($\mathcal{T}_{t+1}^{\text{ego}}$), and the ego-agent's goal location ($\mathcal{G}$). Dashed connections are used during training only. The input and output variables to the neural networks are color-coded with figures. Note that, one of the key features of these two networks is to output a sequence of predicted future zonotopes instead of waypoints (see the final output block of ESN on the right). This zonotope representation enables collision avoidance for navigation safety.

on the future trajectories of the surrounding pedestrians [12], [17]. Second, the output of the network is the pedestrian's future reachable set parameterized as zonotopes $\mathcal{Z}_{[t,t_f]}^{p_k}$ rather than point-based trajectories for robust collision checking and uncertainty parameterization [26], [27], [28]. Third, we replace PECNet's social pooling module with a simple ego-agent sensory radius threshold for computational efficiency, which simplifies integrating our SZN into MPC for a unified prediction and planning framework in Sec. VII.

The pedestrians' past trajectories $\mathcal{T}_{[t_p,t]}^{p_k}$ are encoded by a neural network E_{ped} (shown as the purple arrow in Fig. 6(a)), while the incremental change in the ego-agent state representing the ego-agent control is encoded by E_{next} (shown as the red arrow in Fig. 6(a)). This allows us to condition pedestrian trajectory predictions on the ego-agent's control. The resultant latent features $E_{\text{ped}}(\mathcal{T}_{[t_p,t]}^{p_k})$ and $E_{\text{next}}(\mathcal{T}_{t+1}^{\text{ego}})$ are then concatenated and used as the condition features $\mathbb{F}_{\text{cond}}$. The pedestrian's endpoint locations are encoded as E_{end} as seen by the orange arrows in Fig. 6(a). The resultant latent features $E_{\text{end}}(\mathcal{T}_{t_f}^{p_k})$ are then concatenated with $\mathbb{F}_{\text{cond}}$ as global features $\mathbb{F}_{\text{global}}$ and encoded by the latent encoder E_{latent}. We randomly sample features from a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ generated by the E_{latent} module, and concatenate them with $\mathbb{F}_{\text{cond}}$. This concatenated information is then passed into the latent decoder D_{latent}. Then D_{latent} outputs the predicted endpoint that is passed again through E_{end}. The output is concatenated again with $\mathbb{F}_{\text{cond}}$ and passed to another encoder P_{future} to output the predicted zonotopes of the pedestrians $\mathcal{Z}_{[t,t_f]}^{p_k}$.

2) Ego-Agent Social Network (ESN): Our ESN architecture is shown in Fig. 6(b). The surrounding pedestrians' future zonotope centers $\boldsymbol{c}_{[t,t_f]}^{p_k}$ are aggregated through summation to take into account the collective effect of surrounding pedestrians[6] while keeping a fixed-input-size architecture [40], [60], [61]. The summed pedestrian features are then encoded by E_{future} as seen by the green arrows in Fig. 6(b). The goal location for the ego-agent is encoded by E_{goal}, while the incremental change in the ego-agent state is encoded by

[6]Other human trajectory learning modules include a social module to take into account the surrounding pedestrians effect such as social non-local pooling mask [42], max-pooling [41], and sorting [44].

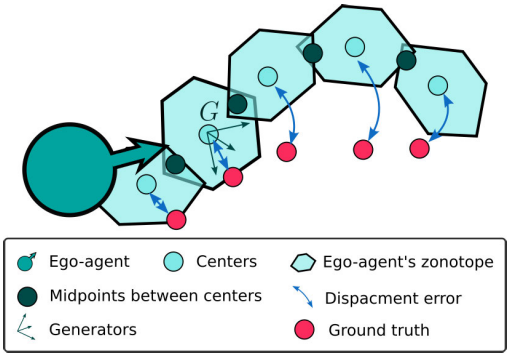


Fig. 7. Our zonotope shaping loss functions. The loss aims to learn interconnected zonotopes that engulf the ground truth path.

E_{next} as seen by the orange and red arrows respectively in Fig. 6(b). The resultant latent features $E_{\text{future}}(\sum_{k=1}^{m} \boldsymbol{c}_{[t,t_f]}^{p_k})$, $E_{\text{goal}}(\mathcal{G})$ and $E_{\text{next}}(\mathcal{T}_{t+1}^{\text{ego}})$ are then concatenated and used as the condition features $\mathbb{F}_{\text{cond}}^{\text{ego}}$ for the CVAE. The ground truth of the ego-agent's future trajectory $\mathcal{T}_{[t,t_f]}^{\text{ego}}$ is encoded by E_{traj} as shown by the cyan arrows in Fig. 6(b). The resultant latent features $E_{\text{traj}}(\mathcal{T}_{[t,t_f]}^{\text{ego}})$ are then concatenated with $\mathbb{F}_{\text{cond}}^{\text{ego}}$ as global features $\mathbb{F}_{\text{global}}^{\text{ego}}$ and encoded by the latent encoder $E_{\text{latent}}^{\text{ego}}$. Similarly, we randomly sample features from a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ generated by the $E_{\text{latent}}^{\text{ego}}$ module, and concatenate them with $\mathbb{F}_{\text{cond}}^{\text{ego}}$. This concatenated information is then passed into the latent decoder $D_{\text{latent}}^{\text{ego}}$, resulting in our prediction of the ego-agent's future reachable set $\mathcal{Z}_{[t,t_f]}^{\text{ego}}$.

Remark 1: Including the E_{next} encoder in both the PPN and ESN facilitates seamless integration with a step-by-step MPC, as one of the MPC's decision variables, i.e., $\Delta \boldsymbol{p}^{\text{ego}}$, is used as inputs to E_{next} as detailed in Sec. VII.

B. Zonotope Shaping Loss Functions

We propose novel zonotope shaping loss functions for both the PPN and the ESN to achieve three goals: (i) position the zonotope centers close to ground truth future trajectories; (ii) ensure intersection of the zonotopes representing consecutive walking steps to approximate continuous time; and (iii) reduce the size of the zonotopes to mitigate conservativeness from

overapproximation. Based on these goals, we implement the following losses, also illustrated inFig. 7.

1) Minimize average displacement error between the predicted centers and midpoint of the ground truth trajectory $\mathcal{T}_{\text{mid},i}$:
$$\mathcal{L}_{\text{ADE}} = \frac{\sum_{i=1}^{t_f-1} \|\mathcal{T}_{\text{mid},i} - \boldsymbol{c}_i\|}{t_f - 1}$$
2) Minimize final displacement error between the last predicted center and the final midpoint of the ground truth trajectory:
$$\mathcal{L}_{\text{FDE}} = \|\mathcal{T}_{\text{mid},t_f-1} - \boldsymbol{c}_{t_f-1}\|$$
3) Contain the midpoint between the current center and previous center $\boldsymbol{c}^p_{\text{mid},i}$ in the current zonotope:
$$\mathcal{L}_{\text{prev}} = \sum_{i=0}^{t_f-1} \text{ReLU}(A_i \cdot \boldsymbol{c}_{\text{mid},i-1} - b_i),$$
where $\boldsymbol{c}_{\text{mid},-1}$ is the initial point of the ground truth trajectory $\mathcal{T}_{[t,t_f]}$, i.e., the current location of the ego-agent.
4) Contain the midpoint between the current center and the next center $\boldsymbol{c}^n_{\text{mid},i}$ in the current zonotope:
$$\mathcal{L}_{\text{next}} = \sum_{i=0}^{t_f-1} \text{ReLU}(A_i \cdot \boldsymbol{c}_{\text{mid},i+1} - b_i),$$
where $\boldsymbol{c}_{\text{mid},t_f}$ is the endpoint of the ground truth trajectory $\mathcal{T}_{[t,t_f]}$.
5) Regulate the size of each output zonotope by penalizing the norm of the generators:
$$\mathcal{L}_G = \sum_{i=1}^{n_G} \|l_G[i] - d_i\|,$$
where d_i is the desired length for each i^{th} generator.

We sum the zonotope shaping losses listed above into a total loss term $\mathcal{L}_{\mathcal{Z}}$. Similar to PECNet [42], we use Kullback-Leibler divergence to train the output of the latent encoder, aiming to regulate the divergence between the encoded distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ and the standard normal distribution $\mathcal{N}(0, \boldsymbol{I})$:
$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}\big(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \parallel \mathcal{N}(0, \boldsymbol{I})\big)$$
Next, we introduce robot-specific losses for ESN to promote locomotion safety and ensure that the Digit robot is able to reach consecutive zonotopes in consecutive walking steps.

C. Incorporating Robot Safety Specifications

To deploy the learning-based social path planner on the Digit robot, it is essential to consider the features of bipedal locomotion such as kinematic and dynamic constraints. To this end, we introduce additional losses to ensure the learned path is viable for bipedal locomotion.

Signal Temporal Logic (STL) is a well-established temporal logic language to formally encode natural language into mathematical representation for control synthesis [62]. More importantly, the quantitative semantics of STL offer a measure of the robustness of an STL specification $\rho(s_t, \phi)$, thereby quantifying the satisfaction or violation of the specification ϕ given a specific signal s_t. Positive robustness values, i.e., $\rho(s_t, \phi) > 0$, indicate specification satisfaction, while negative robustness values indicate a violation. The authors in [63] present STLCG, a tool that transforms STL formulas into computational graphs to be used in gradient-based problems such as neural network learning. To this end, we leverage a similar technique to formally incorporate desired locomotion safety behaviors into our learning framework by encoding STL specifications as additional loss functions that penalize STL formula violation [63], [64].

We derive locomotion specifications based on our previously introduced Reduced-Order Model (ROM) safety theorems [3] and our empirical knowledge about the locomotion safety of Digit [25] during our experiments. To maintain balance, we must bound the ROM Center of Mass (CoM) velocity based on step length and heading change [3]. Therefore, we design STL specifications to regulate $\boldsymbol{c}^{\text{ego}}_{[t,t_f]}$ to limit the sagittal and lateral COM velocities as well as the heading change between consecutive walking steps, based on prespecified thresholds.

1) Locomotion Velocity Specification ϕ_{vel}: Let $s^{v_{\text{sag}}}_{[t+1,t_f]}$ and $s^{v_{\text{lat}}}_{[t+1,t_f]}$ be signals representing the velocity of $\boldsymbol{c}^{\text{ego}}_{[t,t_f]}$ (via finite difference) in the sagittal and lateral directions, respectively. The locomotion velocity specification has:
$$\begin{aligned}
\phi_{\text{sag}} &= \square_{[t+1,t_f]}(s^{v_{\text{sag}}}_{[t+1,t_f]} \le v_{\max} \wedge s^{v_{\text{sag}}}_{[t+1,t_f]} \ge v_{\min}),\\
\phi_{\text{lat}} &= \square_{[t+1,t_f]}(s^{v_{\text{lat}}}_{[t+1,t_f]} \le v_{\text{lat}} \wedge s^{v_{\text{lat}}}_{[t+1,t_f]} \ge -v_{\text{lat}}),\\
\phi_{\text{vel}} &= \phi_{\text{sag}} \wedge \phi_{\text{lat}},
\end{aligned} \tag{8}$$
where $\square_{[a,b]}(c)$ denotes that specification c must be satisfied for all $t \in [a, b]$. We represent this via a loss:
$$\mathcal{L}_{\phi_{\text{vel}}} = \underbrace{\text{ReLU}(-\rho((s^{v_{\text{sag}}}, s^{v_{\text{lat}}}), \phi_{\text{vel}}))}_{\textit{velocity violation}} \tag{9}$$

2) Heading Change Specification $\phi_{\Delta\theta}$: Let $s^{\Delta\theta}_{[t+1]}$ be a signal equal to the heading change between $\boldsymbol{c}^{\text{ego}}_t$ and $\boldsymbol{c}^{\text{ego}}_{t+1}$. The heading change specification is:
$$\phi_{\Delta\theta} = \square_{[t+1,t_f]}(s^{\Delta\theta}_{[t+1,t_f]} < \Delta\theta_{\max} \wedge s^{\Delta\theta}_{[t+1,t_f]} > -\Delta\theta_{\max})$$
Therefore, we propose the following loss:
$$\mathcal{L}_{\phi_{\Delta\theta}} = \underbrace{\text{ReLU}(-\rho(s^{\Delta\theta}, \phi_{\Delta\theta}))}_{\textit{heading change violation}} \tag{10}$$

We sum the STL locomotion losses into a single term $\mathcal{L}_{\text{STL}}$. The network is trained end to end using the following loss function:
$$\mathcal{L} = \alpha_1\mathcal{L}_{KL} + \alpha_2\mathcal{L}_{\mathcal{Z}} + \alpha_3\mathcal{L}_{\text{STL}} \tag{11}$$
where α_i are weighting coefficients.

VI. Zonotope Refinement for Social Acceptability and Uncertainty Parameterization

In this section, we introduce two types of zonotope refinements for the ego-agent based on (1) personal space preference for increased social acceptability, and (2) modeling error

compensation of robot dynamics discrepancy between the ROM and the full-order model of our Digit robot.

Remark 2: Our key insight is that, by using zonotopes as the output format for our neural networks, we can easily postprocess the network outputs to incorporate sources of error that would be either difficult for the neural network to learn directly, or may change at runtime when the network cannot be retrained.

A. Personal Space Modulation

One key consideration for social acceptability is the personal space surrounding every agent [65]. We leverage the adaptable nature of zonotope parameterization, which can be modulated in specific directions, to develop personal space generators. These generators are inspired by the sociological study of proxemics [65] and its application to social navigation [66], [67]. The personal space generator matrix, G^{PS}, is formulated as follows:

$$G^{\text{PS}} = \text{diag}(a, b) \cdot \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{12}$$

The parameters a and b represent the scalar distances that define the personal space along the sagittal and lateral axes, respectively. These distances are rotated by the angle θ to align the personal space with the ego-agent's current walking orientation in both directions as shown in Fig. 8(a). We then modify the ego-agent's zonotope in a straightforward way: $\hat{\mathcal{Z}}^{\text{ego}} = \mathscr{Z}(\boldsymbol{c}^{\text{ego}}, [G^{\text{ego}}\ G^{\text{PS}}])$.

B. Robot Modeling Error Compensation

Using zonotopes as a representation of reachable sets allows for online compensation for the robot dynamics discrepancy between the ROM and the full-order model of our Digit robot.

We learn the modeling errors using Gaussian Process (GP) regression [68], [69], which we train offline. Our model's input is the current sagittal velocity of the robot v_q^{loc} and the MPC previous optimal solution for $(v_*^{\text{loc}}, u_*^{\Delta\theta})$. We chose these parameters as representative state variables because they effectively represent the key parameters contributing to the discrepancy between the ROM and low-level ALIP controller [1]. The GP model then outputs the expected mean deviation $\boldsymbol{\mu} = (\mu_x, \mu_y)$ and variance $\boldsymbol{\sigma} = (\sigma_x^2, \sigma_y^2)$ in robot's Euclidean position at the next walking step (See Fig. 2). The Gaussian mean from the GP model is used to design a new generator matrix $G^{\mu} = \text{diag}(\mu_x, \mu_y)$ such that the ego-agent's zonotope becomes $\hat{\mathcal{Z}}_{q+1}^{\text{ego}} = \mathscr{Z}(\boldsymbol{c}^{\text{ego}}, [G^{\text{ego}}\ G^{\text{PS}}\ G^{\mu}])$ to compensate for the anticipated mismatch between the ROM dynamics and the full-order dynamics (see Fig. 8(b)).

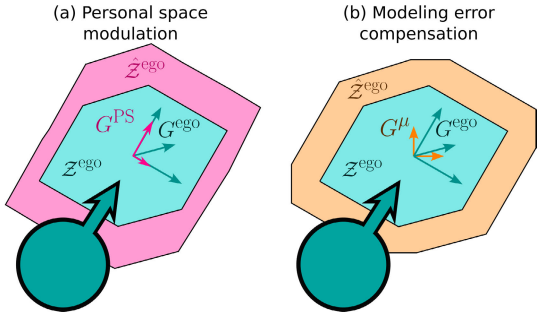


Fig. 8. Two types of zonotope refinements: (a) zonotope refinement based on personal space preference for improved social acceptability, where the generators in G^{PS} are in the local sagittal and lateral directions as shown by the pink arrows. (b) zonotope refinement based on a learned GP model of the model discrepancy between ROM and full-order models, where the generators in G^{μ} are in the global x and y directions as shown by the orange arrows.

VII. Social MPC

To safely navigate the human-crowded environment we propose to solve the following trajectory optimization problem that encodes the SZN in the previous section as constraints:

$$\min_{X,U} \sum_{q=0}^{N-1} J(\boldsymbol{x}_q, \boldsymbol{u}_q) + J_N(\boldsymbol{x}_N) \tag{13a}$$

$$\text{s.t. } \boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q), \forall q \tag{13b}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}},\ (\boldsymbol{x}_q, \boldsymbol{u}_q) \in \mathcal{XU}_q, \forall q \tag{13c}$$

$$\boldsymbol{p}_{q+1}^{\text{ego}} \in \hat{\mathcal{Z}}_{q+1}^{\text{ego}}(\Delta \boldsymbol{p}_q^{\text{ego}}, \mathcal{E}_q), \forall q \tag{13d}$$

$$\hat{\mathcal{Z}}_{q+1}^{\text{ego}}(\Delta \boldsymbol{p}_q^{\text{ego}}, \mathcal{E}_q) \bigcap \mathcal{Z}_{q+1}^{p_{k_q}} = \emptyset,\ \forall\, q, k_q \tag{13e}$$

where the decision variables include a state sequence $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and a control sequence $U = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N-1}\}$, the running and terminal costs (13a) are designed to reach the goal and promote social acceptability, subject to the ROM dynamics (13b) (Sec. III-A). Constraint (13d) requires the ego-agent at the next $(q+1)^{\text{th}}$ walking step to stay within the reachable set, while constraint (13e) requires the ego-agent to avoid collision with the pedestrians. $\mathcal{E}_q = (\mathcal{T}_{[t_p,t],q}^{p_k}, \mathcal{G})$ denotes the environment state at the q^{th} walking step. Next, we will introduce the kinematics constraints in Sec. VII-A, and navigation constraints in Sec. VII-B, social acceptability cost function (Sec. VII-C), and finally reformulate the MPC in (13) with a version for implementation (Sec. VII-D).

A. Kinematics Constraints

To prevent the LIP dynamics from taking a step that is kinematically infeasible by the Digit robot, we implement the following constraint

$$\mathcal{XU}_q = \{(\boldsymbol{x}_q, \boldsymbol{u}_q) \mid \boldsymbol{x}_{\text{lb}} \le \boldsymbol{x}_q \le \boldsymbol{x}_{\text{ub}} \text{ and } \boldsymbol{u}_{\text{lb}} \le \boldsymbol{u}_q \le \boldsymbol{u}_{\text{ub}}\} \tag{14}$$

where $\boldsymbol{x}_{\text{lb}}$ and $\boldsymbol{x}_{\text{ub}}$ are the lower and upper bounds of $\boldsymbol{x}_q$ respectively, and $\boldsymbol{u}_{\text{lb}}$ and $\boldsymbol{u}_{\text{ub}}$ are the bounds for $\boldsymbol{u}_q$ (See Fig. 4).

B. Reachability and Navigation Safety Constraints

To enforce navigation safety (i.e., collision avoidance), we require that Digit remains in the social zonotope $\mathcal{Z}^{\text{ego}}$ and outside of the surrounding pedestrians reachable set $\mathcal{Z}^{p_k}$.

1) Reachability Constraint: For the robot's CoM to remain inside the desired zonotope for the next walking step $\hat{\mathcal{Z}}_{q+1}^{\text{ego}}$, we represent the zonotope using half-space representation as shown in Prop. 7. The constraint is reformulated as such:

$$\max(\hat{A}^{\text{ego}} \boldsymbol{p}^{\text{ego}} - \hat{b}^{\text{ego}}) \le 0 \tag{15}$$

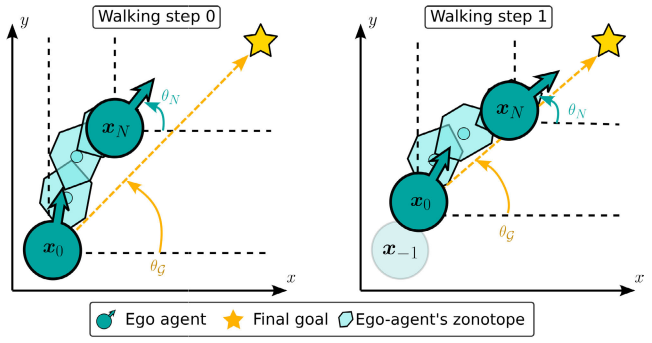


Fig. 9. Illustration of the terminal goal angle $\theta_{\mathcal{G}}$, a state dependent on the initial state $\boldsymbol{x}_0$. $\theta_{\mathcal{G}}$ is designed to guide the ego-agent towards the final goal location. SZN-MPC optimizes over N walking step horizon such that the ego-agent heading at the end of the planning horizon θ_N aligns with $\theta_{\mathcal{G}}(\boldsymbol{x}_0)$. After executing a walking step (see the figure on the right), we update $\theta_{\mathcal{G}}(\boldsymbol{x}_0)$ based on the new initial state $\boldsymbol{x}_0$ for the ego-agent.

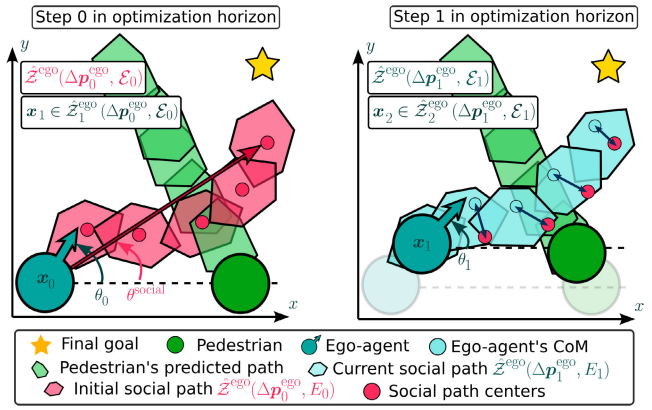


Fig. 10. Illustration of the social acceptability cost. In the initial step in the optimization horizon (figure on the left), the learned social path $\hat{\mathcal{Z}}^{\text{ego}}(\Delta\boldsymbol{p}_0^{\text{ego}}, \mathcal{E}_0)$ is shown in the red zonotopes, and the active constraint is $\boldsymbol{x}_1 \in \hat{\mathcal{Z}}_1^{\text{ego}}(\Delta\boldsymbol{p}_0^{\text{ego}}, \mathcal{E}_0)$. In the following step in the optimization horizon (figure on the right) the learned social path $\hat{\mathcal{Z}}^{\text{ego}}(\Delta\boldsymbol{p}_1^{\text{ego}}, \mathcal{E}_1)$ (shown by the cyan zonotopes) is based on the *current* environment state $\mathcal{E}_1$ and ego-agent *current* location $\boldsymbol{p}_1^{\text{ego}}$, and the active constraint is $\boldsymbol{x}_2 \in \hat{\mathcal{Z}}_2^{\text{ego}}(\Delta\boldsymbol{p}_1^{\text{ego}}, \mathcal{E}_1)$. The social acceptability cost aims to minimize the difference between the centers $\hat{\boldsymbol{c}}_q^{\text{social}}$ (red dots) of $\hat{\mathcal{Z}}_q^{\text{ego}}(\Delta\boldsymbol{p}_0^{\text{ego}}, \mathcal{E}_0)$ $\forall q$ and the ego-agent ROM CoM $\boldsymbol{p}_q^{\text{ego}}$ (cyan dots). The arrows between the social path centers $\hat{\boldsymbol{c}}^{\text{social}}$ and ego-agent ROM CoM $\boldsymbol{p}^{\text{ego}}$ indicate the distance that the social acceptability cost aims to minimize $\|\hat{\boldsymbol{c}}_q^{\text{social}} - \boldsymbol{p}_q^{\text{ego}}\|$.

2) Navigation Safety Constraint: For pedestrian collision avoidance, we require that the reachable set of the ego-agent does not intersect with that of the pedestrians for each walking step. Therefore, we design a new zonotope for the ego-agent as Minkowski sum of the ego-agent's zonotope and the pedestrian's zonotope centered around the ego-agent $\mathcal{Z}^{\text{mink}} = \mathscr{Z}(\boldsymbol{c}^{\text{ego}}, [G^{\text{ego}}\ G^{\text{PS}}\ G^{\mu}\ G^{p_k}])$ to check for collision with the pedestrians' zonotope following Prop. 6. We then represent $\mathcal{Z}^{\text{mink}}$ using a half-space representation parameterized by A_k^{mink} and b_k^{mink}, as per Prop. 7, and require that the pedestrian position is outside the Minkowski-summed zonotope. Thus, for each k^{th} pedestrian, we have the following constraint:

$$\max(A_k^{\text{mink}}\boldsymbol{p}_k - b_k^{\text{mink}}) > 0. \tag{16}$$

C. Cost Function and Social Acceptability Metric

Our MPC cost function is designed to drive the CoM state to a goal location $\mathcal{G}$ and to promote social acceptability. The terminal cost penalizes (i) the distance between the current ROM state and the global goal state $\mathcal{G}$ in the 2D world coordinate, and (ii) the ego-agent heading angle deviation from the heading angle pointing toward the final goal location (see Fig.9) to avoid abnormal walking gaits, such as the robot moving toward the goal while walking backward.

$$J_N(\boldsymbol{x}_N) = \|\boldsymbol{x}_N - \boldsymbol{x}_{\mathcal{G}}\|_{W_1}^2 + \|\theta_N - \theta_{\mathcal{G}}(\boldsymbol{x}_0)\|_{W_2}^2 \tag{17}$$

where $\boldsymbol{x}_{\mathcal{G}} = (\mathcal{G}, v_{\text{terminal}})$, $\mathcal{G} = (x^{\text{dest}}, y^{\text{dest}})$, and $\theta_{\mathcal{G}}(\boldsymbol{x}_0)$ is the angle between the ego-agent's current position and the final goal location $\mathcal{G}$ (see more details in Fig. 9).

The MPC constrains the ROM CoM $\boldsymbol{p}_q^{\text{ego}}$ to stay within the ego-agent's zonotope (13d) and avoid collisions (13e). These constraints, along with the changes in the environment ($\mathcal{E}_q \neq \mathcal{E}_{q+1}, \forall q$), might cause the generated ROM CoM $\boldsymbol{p}_q^{\text{ego}}$ trajectory to deviate from the learned social path $\hat{\mathcal{Z}}_{q+1}^{\text{ego}}(\Delta\boldsymbol{p}_0^{\text{ego}}, \mathcal{E}_0)$ with the centers $\hat{\boldsymbol{c}}_q^{\text{social}}, \forall q$. This social path is generated from the *initial* environment $\mathcal{E}_0$. Therefore, we incorporate a social acceptability metric by creating a cost that penalizes deviation of the ROM CoM $\boldsymbol{p}_q^{\text{ego}}$ from the learned social path $\hat{\mathcal{Z}}_{q+1}^{\text{ego}}(\Delta\boldsymbol{p}_0^{\text{ego}}, \mathcal{E}_0)$ as shown in Fig. 10. We set the social acceptability metric as: (1) the distance between the ROM CoM $\boldsymbol{p}_q^{\text{ego}}$ and the centers of the learned social path $\hat{\boldsymbol{c}}_q^{\text{social}}$, and (2) the difference between the ego-agent current heading θ_q and social heading angle θ^{social}, i.e., the angle between the ego-agent's initial position and the $\hat{\boldsymbol{c}}_N^{\text{social}}$, as shown in Fig. 10. Thus, we set the running cost of social acceptability as follows:

$$J_{\text{social}}(\boldsymbol{x}_q) = \|\hat{\boldsymbol{c}}_q^{\text{social}} - \boldsymbol{p}_q^{\text{ego}}\|_{W_3}^2 + \|\theta^{\text{social}} - \theta_q\|_{W_4}^2 \tag{18}$$

Including such social acceptability metric as a cost function, will guide SZN-MPC to generate the CoM trajectory that (1) tracks the learned social path $\hat{\mathcal{Z}}_{q+1}^{\text{ego}}(\Delta\boldsymbol{p}_0^{\text{ego}}, \mathcal{E}_0)$, (2) is within the next zonotope based on the current environment state $\mathcal{E}_q$, and (3) is collision-free (i.e., constraints (13d)-(13e)).

Remark 3: The initial output of the neural network $\hat{\mathcal{Z}}_{q+1}^{\text{ego}}(\Delta\boldsymbol{p}_0^{\text{ego}}, \mathcal{E}_0)$ is not guaranteed to be collision-free at every walking step in the planning horizon. Therefore, we treat social acceptability as a cost, not a constraint, to prioritize safety.

D. Social Zonotope Network MPC Formulation

To enable numerical implementation, we reformulate our Social Zonotope Network MPC (SZN-MPC) shown in (13) based on the aforementioned costs and constraints as follows:

$$\min_{X,U} \sum_{q=0}^{N-1} J_{\text{social}}(\boldsymbol{x}_q) + J_N(\boldsymbol{x}_N) \tag{19a}$$

$$\text{s.t. } \boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q), \forall q \tag{19b}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}},\ (\boldsymbol{x}_q, \boldsymbol{u}_q) \in \mathcal{XU}_q, \forall q \tag{19c}$$

$$\max(\hat{A}_{q+1}^{\text{ego}}\boldsymbol{p}_{q+1}^{\text{ego}} - \hat{b}_{q+1}^{\text{ego}}) \leq 0, \forall q \tag{19d}$$

$$\max(A_{q+1}^{\text{mink}}\boldsymbol{p}_{k_{q+1}} - b_{q+1}^{\text{mink}}) > 0,\ \forall\, q, k_q \tag{19e}$$

where the kinematics constraint in (19c) is implemented using the local dynamics $(\Delta x^{\text{loc}}, v^{\text{loc}})$. We provide detailed implementation parameters in Table I.

VIII. Results

In this section, we comprehensively analyze our framework via three experiments. The first experiment assesses social acceptability and locomotion feasibility of our proposed SZN-MPC. The second experiment benchmarks SZN-MPC agaist a baseline approach of LIP-based MPC using adiscrete-time control barrier function (DCBF-MPC) for dynamic obstacle collision avoidance and trajectory planning. The final experiment tests the feasibility of implementing SZN-MPC on our Digit robot hardware. The section starts with implementation details.

TABLE I
SZN-MPC Parameters

parameter	value	parameter	value
$v_{\text{ub}}^{\text{loc}}$	1.0 m/s	$v_{\text{lb}}^{\text{loc}}$	-0.1 m/s
$\Delta x_{\text{ub}}^{\text{loc}}$	0.2 m	$\Delta x_{\text{lb}}^{\text{loc}}$	-0.2 m
$u_{\text{ub}}^{\Delta\theta}$	15°	$u_{\text{lb}}^{\Delta\theta}$	-15°
u_{ub}^{f}	0.4 m	u_{lb}^{f}	-0.1 m
d_1	0.1	d_2	0.005
v_{terminal}	0 m/s	n_G	4
W_1	3	W_2	1
a	0.3 m	b	0.2 m
α_1, α_3	1	α_2	100

A. Implementation Details

1) Training: The social path planner module introduced in Sec. V was trained on the UCY [70] and ETH [71] crowd datasets with a standard leave-one-out approach, similar to prior studies [40], [41], [42], [72]. More specifically, we excluded the UNIV dataset from the training examples, and used it for testing.[7] To evaluate the performance of incorporating robot-locomotion-specific STL specifications into the training, we trained two neural network models with and without the added robot-specific losses introduced in Sec. V-C. We employ a historical trajectory observation $\mathcal{T}_{[-8,0]}^{p_k}$ for all neighboring pedestrians that are within a radius of 4 m and a prediction horizon $\hat{\mathcal{T}}_{[0,8]}^{\text{ego}}$. For both pedestrians and the ego-agent, the duration of 8 timesteps takes 3.2 s (8×0.4 s $=$ 3.2 s). The network architecture details are shown in Table II. The SZN is implemented and trained using PyTorch [73].

2) Pedestrian Simulation: To simulate pedestrian motion in simulation, we use SGAN (Social Generative Adversarial Network) [41], a state-of-the-art human trajectory prediction model, following the approach taken in [17]; this ensures a fair evaluation by avoiding simulating pedestrians with our own model. SGAN is designed to represent social interactions and dependencies between pedestrians by considering social context, including how people influence each other and move in groups. To achieve this, SGAN incorporates historical trajectories of pedestrians and the ego-agent. This enhances the realism of the simulation by accounting for interactions between the ego-agent and pedestrians in the environment.

3) Testing Environment Setup: The environment for all simulations is an open space of 14×14 m^2, with randomly generated pedestrians' initial trajectory. The goal location is $\mathcal{G} = (6, 12)$ m, and the ego-agent starting position is uniformly sampled along the y-axis as such $\boldsymbol{x}_0 = (0, \mathcal{U}_{[0,13]}, 0)$ with $\theta_0 = 0$. The MPC is solved with a planning horizon of $N = 4$. SZN-MPC parameters are included in Table. I. Simulations and training are conducted using a 16-core Intel Xeon W-2245 CPU and an RTX-5000 GPU with 64 GB of memory. The SZN-MPC is implemented using do-mpc Python libraray [74] and CasADi [75]. Digit is simulated using the MuJoCo simulator provided by Agility Robotics [25] and visualized using Nvidia Isaac Gym [76], which allows an animation of pedestrian characters in the environment.

[7]The datasets can be found in [40].

B. Experiment 1: Social Acceptability and Locomotion Feasibility

We now evaluate the social acceptability of our system's generated paths, and the feasibility of the generated path for bipedal locomotion.

1) Setup and Metrics:

a) Social acceptability: We quantify social acceptability using two methods. First, we measure the average displacement error (ADE) and final displacement error (FDE) of ESN's planned path compared to the ground truth:

$$\text{ADE} = \frac{\sum_{i=1}^{t_f-1} \|\mathcal{T}_{\text{mid},i}^{\text{ego}} - \boldsymbol{c}_i^{\text{ego}}\|}{t_f - 1}, \tag{20}$$

where $\boldsymbol{c}_i^{\text{ego}}$ is the ground truth data and $\mathcal{T}_{\text{mid},i}^{\text{ego}}$ is the ESN prediction output. Second, we compare our social acceptability metric when running SZN-MPC with and without the social acceptability cost function. For Digit to achieve a socially acceptable path, it must track the path produced by ESN, by assuming that ESN's ADE $= 0.218 < \epsilon$ as defined in Definition. 4.

b) Locomotion feasibility: For locomotion feasibility, we compare the tracking performance of the ROM to the social path with and without the locomotion losses introduced in Sec. V-C.

2) Results and Discussion: Fig. 11(a) shows that ESN produces an ADE$=$ 0.218 m over the prediction horizon of 7 timesteps,[8] and FDE$=$ 0.447 m. Fig. 11(b) shows that we are able to reduce the social acceptability cost when running SZN-MPC (19) with the social running cost Eq. (18), thus, promoting social acceptability by tracking the path of ESN. Fig. 12 shows the results of a planned trajectory of SZN-MPC with and without social acceptability cost (18).

We find that integrating locomotion losses into ESN training reduces the number of locomotion safety violations (i.e.,

[8]The prediction horizon timesteps is 7 and not 8, since the displacement error is calculated based on the middle points $\mathcal{T}_{\text{mid}}^{\text{ego}}$ of $\mathcal{T}_{[1,8]}^{\text{ego}}$.

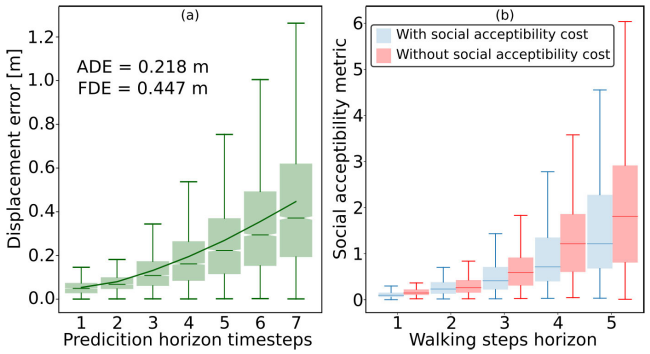


Fig. 11. Quantitative results of social acceptability: (a) Show the displacement error between the prediction of ESN $\boldsymbol{c}^{\text{ego}}$ and the ground truth data $\mathcal{T}_{\text{mid}}^{\text{ego}}$. The data is collected based on the UNIV dataset with 7831 unique frames. The solid line shows the average displacement error at each prediction horizon. (b) shows the social acceptability metric for 5 different trials with 30 pedestrians and 100 walking steps in each trial. The social acceptability metric is reduced when Problem (19) is solved with the social cost (18), thus yielding a socially acceptable path.

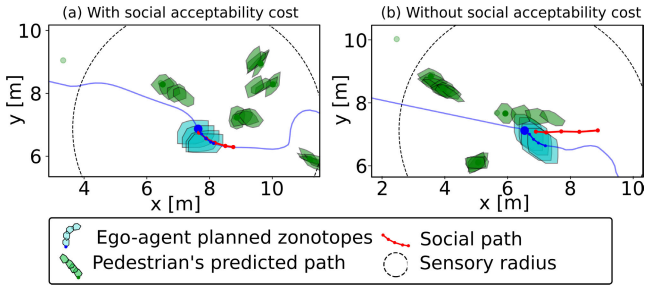


Fig. 12. Qualitative results of social acceptability: comparison of different social acceptability levels. (a) shows a socially acceptable trajectory generated by SZN-MPC with social acceptability cost (18), as the ego-agent's planned path (shown in blue) follows the predicted social path of ESN (shown in red), while (b) shows a trajectory generated by SZN-MPC without social acceptability cost (18) where a larger deviation is observed between the ego-agent's planned path and the predicted social path of ESN.

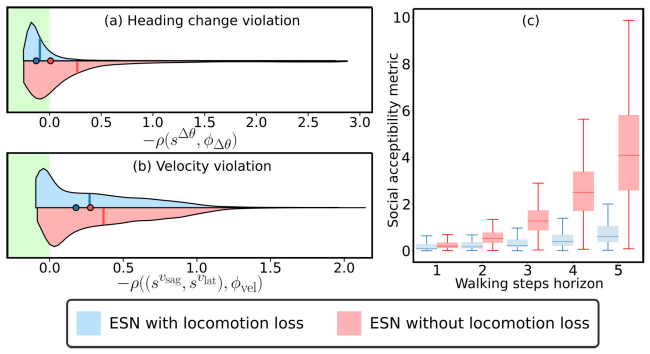


Fig. 13. Violin plots of ESN test results with and without STL locomotion losses are shown in blue and red, respectively. The shaded green region is where the heading change specification (a), and locomotion velocity specification (b) are satisfied, i.e., $\rho(s^{\Delta\theta}, \phi_{\Delta\theta}) < 0$ and $-\rho((s^{v_{\text{sag}}}, s^{v_{\text{lat}}}), \phi_{\text{vel}}) < 0$. The solid vertical line represents the mean value of $-\rho$, while the dot represents the median. The data is collected based on the UNIV dataset with 7831 unique frames. (c) Shows the social acceptability metric when ESN is integrated with SZN-MPC with and without the locomotion losses.

specifications (9) and (10), as seen in Fig. 13(a)-(b). This means that SZN-MPC can generate trajectories that achieve improved tracking of the socially acceptable path as shown in Fig. 13(c). These results indicate that locomotion losses shape our ESN output to comply with the capabilities of bipedal locomotion.

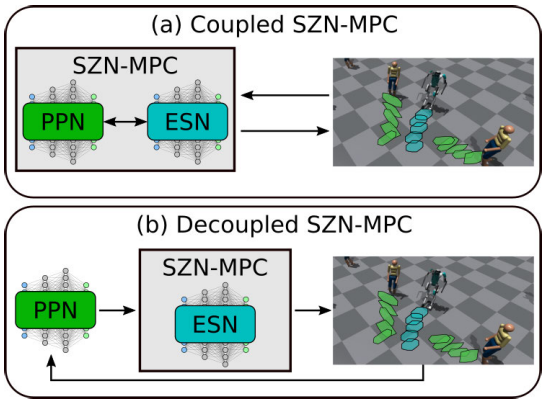


Fig. 14. Block diagram of (a) coupled SZN-MPC and (b) decoupled SZN-MPC.

C. Experiment 2: Benchmarking

We compare coupled and decoupled versions of our method with a baseline approach for LIP-based MPC that uses a discrete-time control barrier function (DCBF-MPC) for collision avoidance of dynamic obstacles [6], [57]. We compare the conservativeness of the produced trajectory, social acceptability, safety and optimality, and finally the computational cost.

1) Setup of Baselines:

a) Coupled SZN-MPC: In this setup, the prediction (PPN) and planning (ESN) networks are coupled during trajectory optimization, where the MPC reasons about the effect of the ego-agent's control on the future prediction of the pedestrians. A block diagram of this model is shown in Fig. 14(a).

b) Decoupled SZN-MPC (dec): In this setup, PPN and ESN are decoupled. Before each solve of (19), the pedestrians' future prediction is conditioned on the optimal solution from the previous MPC solve for the ego-agent and is fixed throughout the optimization for the current solve. That is, the PPN module is queried only once for each MPC solve. A block diagram of this model is shown in Fig. 14(b).

Remark 4: For clarification, our decoupled SZN-MPC is not considered as the category of the decoupled social navigation literature as explained in Sec. II-A, since the PPN module is conditioned on the ego-agent next planned walking step. Thus, our decoupled SZN-MPC can be viewed as a coupled social navigation method in the literature.

c) DCBF-MPC: We compare our path plan to that generated by a LIP-based MPC with a DCBF for navigation safety, where we substitute (19d) and (19e) with a DCBF:

$$h(\boldsymbol{p}_{q+1}^{\text{ego}}, \boldsymbol{p}_{k_{q+1}}) \geq (1-\gamma) h(\boldsymbol{p}_{q}^{\text{ego}}, \boldsymbol{p}_{k_q}), \ \forall\, k_q \tag{21}$$

where $(1-\gamma)$ is a class $\mathcal{K}_{\infty,e}$, $0 < \gamma \leq 1$ [58], [77], and $h(\boldsymbol{p}_{q}^{\text{ego}}, \boldsymbol{p}_{k_q})$ is a distance metric defined as:

$$h(\boldsymbol{p}_{q}^{\text{ego}}, \boldsymbol{p}_{k_q}) = \| \frac{1}{r} (\boldsymbol{p}_{q}^{\text{ego}} - \boldsymbol{p}_{k_q}) \| - 1 \tag{22}$$

and the optimization is initialized with $h(\boldsymbol{p}_{0}^{\text{ego}}, \boldsymbol{p}_{k_0}) \geq 0$. For this model, we also use PPN as a prediction module for the surrounding pedestrians, and SGAN as the pedestrian simulator.

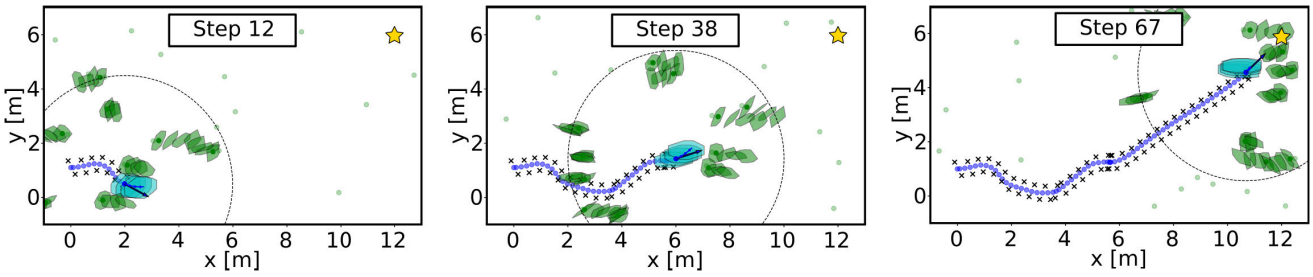


Fig. 15. Storyboard snapshots of SZN-MPC trajectory at different walking steps. The ego-agent (cyan) successfully reaches the goal (yellow ⋆) while avoiding pedestrians (green). The dashed circle is the sensory radius of the ego-agent. Blue dots represent the CoM of ROM, and black × is the desired foot placement. Green dots are unobserved pedestrians.

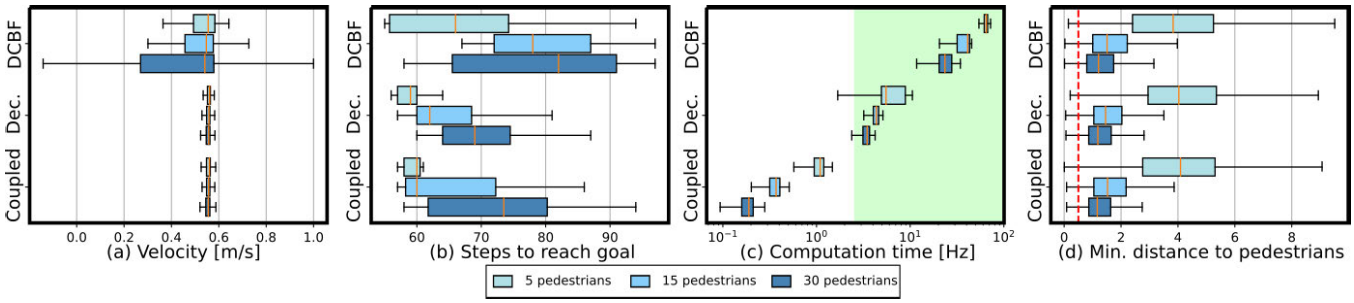


Fig. 16. Benchmarking results for the ego-agent's (a) velocity, (b) optimality: number of walking steps to reach within 1 m of the final goal, (c) frequency, and (d) safety: minimum distance to pedestrians. The data consists of 20 different trials with random initial conditions and a fixed goal location (see Sec. VIII-A3). Each trial is limited up to 100 walking steps. The velocity data is collected before reaching the goal, to avoid collecting a stopping velocity. The frequency is calculated based on a data collection of 300 walking steps, and the green shaded area in (c) is the minimum required computation time of SZN-MPC for Digit hardware implementation. (d) the dashed red line is 0.5 m which is r in (22), and what is considered as d in navigation safety in Definition. 3.

2) Results and Discussion: SZN-MPC outperforms the other baselines in creating socially acceptable robot motion while maintaining safety (snapshots of SZN-MPC are shown in Fig. 15). Our method has slower computation time, but is still fast enough for real-time implementation. We detail and discuss these results below.

a) Conservativeness: In Fig. 16(a), all three models produce relatively similar median velocities over the testing data. However, coupled SZN-MPC and decoupled SZN-MPC produce more consistent velocities. Our proposed method is more efficient with respect to the number of steps taken to reach the goal Fig. 16(b), and has less variation in velocity while maintaining similar safety performance (see Fig. 16(a) and (d)). The high variability in velocity from DCBF-MPC indicates adaptability in a dynamic environment, but this does not translate to a safer path in a densely crowded environment. Indeed, it generates the same level of safety as SZN-MPC, since it results in a similar minimum distance to the surrounding pedestrians (See Fig. 16(a) and (d)).

b) Social acceptability: SZN-MPC produces a more consistent and predictable behavior for the ego-agent compared to DCBF-MPC, as indicated by the the smaller interquartile range in Fig. 16(a). Predictability of the ego-agent behavior in a social context is desirable by pedestrians as it is perceived to be less disruptive.

Remark 5: The coupled SZN-MPC and decoupled SZN-MPC produce similar results. We believe that this is because, although the decoupled SZN-MPC only evaluates the PPN module once for each MPC solve and fixes the pedestrian prediction during the optimization, this MPC is re-solved for every walking step. Thus, this receding-horizon strategy indirectly couples the pedestrian prediction with the ego-agent planning.

c) Safety and optimality: All three methods produce comparable safety performance by maintaining a similar minimum distance to the pedestrians as shown in Fig. 16(d). However, SZN-MPC consistently generates a more optimal path, as indicated by the lower number of steps taken to reach the goal (see Fig. 16(b)). Thus, we are not sacrificing safety by moving faster. SZN-MPC generates optimal paths for the ego-agent due to its proactive approach, i.e., the SZN-MPC generates future reachable sets that take into account not only the collision avoidance but also social influence from all surrounding pedestrians. On the other hand, DCBF-MPC is reactive, as it only reacts to the PPN predictions of the pedestrians to avoid collisions. This reactivity leads to the "freezing robot problem" [39], as evidenced by the low velocities that DCBF-MPC generates, specifically in environments with high crowd densities 16(a).

Remark 6: The pedestrian simulation lacks assurance of generating non-adversarial paths. Despite considering the ego-agent's position, there are occasional cases where the simulator generates paths for pedestrians that intersect with the ego-agent and other pedestrians. In our real-world experiments, we operate under the assumption of pedestrian collaboration and non-adversarial behavior towards the ego-agent, following social norms.

d) Computational cost: The three models' computational costs are different by orders of magnitude, where DCBF-MPC is 10^2 times faster than the coupled SZN-MPC and 10 times

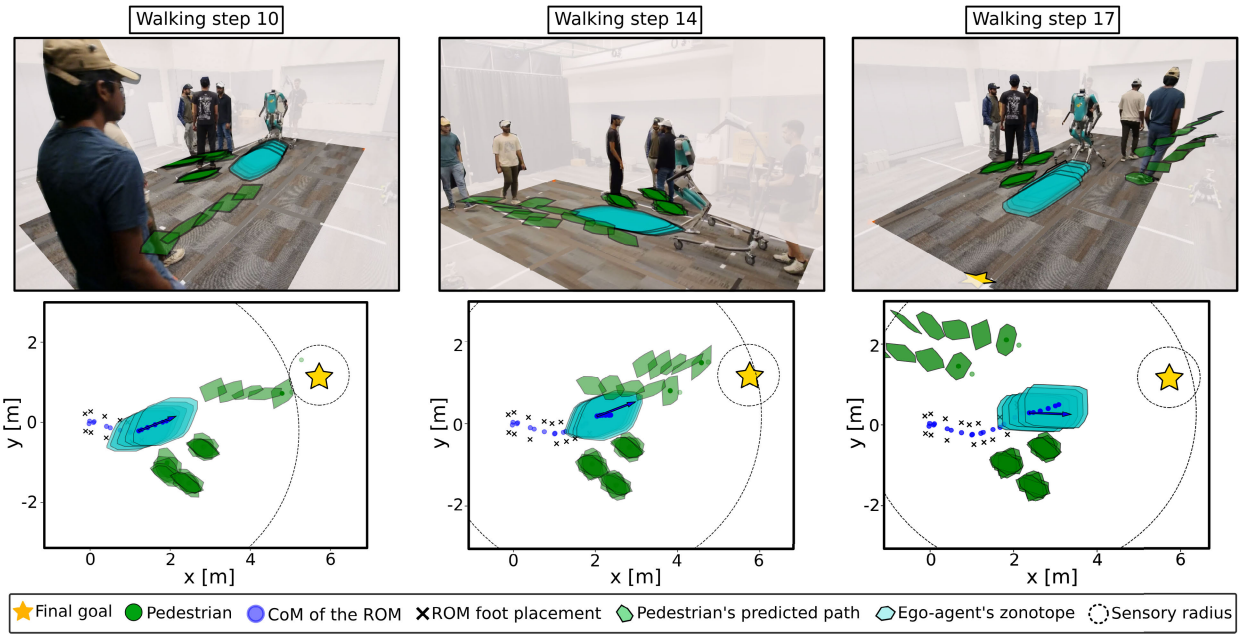


Fig. 17. A storyboard snapshots of the hardware experiments with 5 pedestrians. In this experiment, the first group consists of three stationary pedestrians standing in place, while the other group consists of two pedestrians walking towards Digit.

faster than the decoupled SZN-MPC as shown in Fig. 16(c). Note that, the MPC is used as a step planner for Digit, and it only needs to be solved once before the swing phase of the walking motion ends. Since the commanded step duration is 0.4 s, which is the minimum computational time required for hardware implementation, the SZN-MPC needs to be solved at a rate of at least 2.5 Hz (Sec.VIII-D1.a). Based on the testing parameters (Sec.VIII-A), the coupled SZN-MPC is not viable for hardware implementation. Therefore, we conclude the decoupled SZN-MPC outperforms the coupled version, and use the decoupled SZN-MPC for our hardware implementation. Computational efficiency can be improved by reducing the planning horizon N and the number of pedestrians in the environment.

D. Experiment 3: Hardware Validation in A Human Crowd

We conduct a series of hardware experiments to showcase the versatility of our proposed method in reliably navigating social settings with pedestrians across various scenarios. We also seek to validate our network architecture choices for learning the pedestrians' future path prediction and the ego-agent's future path directly as zonotopes, which facilitate real-time implementation of our framework on Digit's onboard PC.

Remark 7: The personal space modulation and modeling error compensation in Sec. VI are included in this hardware experiment. Without such refinements, we found that Digit's trajectory was often very close to the surrounding pedestrians. A key feature of the zonotope parametrization is that it allows us to compensate for desired behaviors downstream without the need to retrain the SZN.

1) Setup:

a) Low-level full-body control: At the low level we use the Angular momentum LIP planner introduced in [1], and a Digit's passivity controller with ankle actuation which we have previously shown to exhibit desirable ROM tracking results [3]. Here we set the desired walking step time and the desired lateral step width to be 0.4 s and 0.4 m, respectively.

b) Experiment design: Our experimental setup uses an indoor VICON motion capture system [78] to measure the current position of the surrounding pedestrians $\boldsymbol{p}_{k_0}$. The coordinates of pedestrians are sent over to SZN-MPC every 0.4 s to match the human crowd dataset SZN is trained on. SZN-MPC then solves for the ego-agent's social trajectory and sends the next desired CoM velocity and heading change $(v^{\text{loc}}_{q+1}, \theta_{q+1})$ to the low-level controller. The prediction, planning, and low-level control are executed on Digit's onboard PC and in real-time, with a walking step horizon $N = 4$. The experimental space in 6×3 m^2, Digit's starting position is $\boldsymbol{p}^{\text{ego}}_0 = (0, 0)$ m with $\theta_0 = 0°$, and the goal location is diagonally across the experiment space at $\mathcal{G} = (5.72, 1.18)$ m (see Fig. 17).

2) Results and Discussion: Our framework is demonstrated in various mock social scenarios.[9] In Fig. 17, we depict a social scenario involving two groups of pedestrians. The first group consists of three stationary pedestrians, while the other group consists of two pedestrians walking towards Digit. Our results show that SZN accurately predicts the two groups' behaviors and safely navigates toward the goal location marked by the yellow star. Digit adjusts its position closer to the stationary group to avoid collision with the approaching pedestrians. Three different scenarios are also shown in Fig. 18. In Fig. 18(a), pedestrians move towards the center of the space from different directions and continue their path out of the experimental space. In Fig. 18(b), one group walks alongside Digit in a row while the other group of pedestrians

[9]Videos of the experiments, code, and supplemental materials are found here https://szn-mpc.github.io/

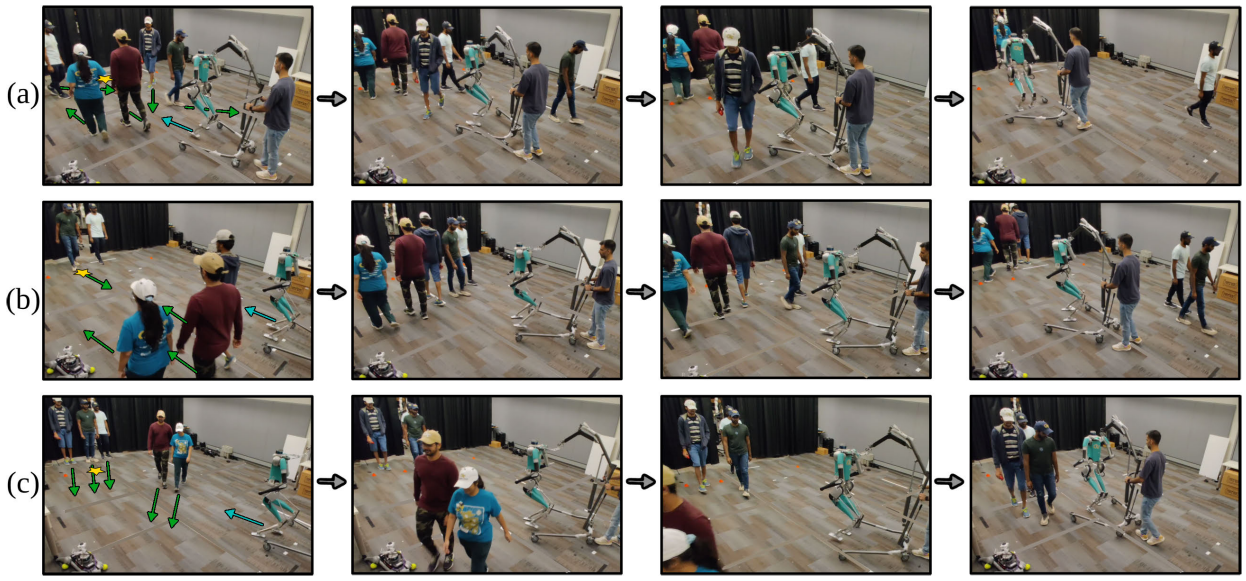


Fig. 18. Snapshots of three different testing scenarios. (a) depicts pedestrians walking in different directions. (b) illustrates a group of pedestrians walking alongside Digit in a row while the other group walks towards Digit. (c) demonstrates two groups walking in the direction opposite to Digit. In the first column, green arrows indicate the direction of the pedestrians' movement, while a cyan arrow indicates Digit's direction, and the yellow star is the final goal position for Digit.

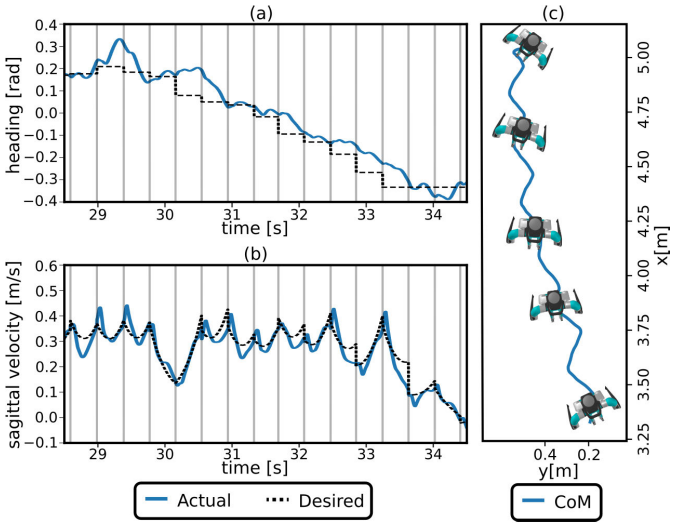


Fig. 19. Hardware experiment results: in (a) we show the torso heading tracking performance, and CoM sagittal velocity tracking performance in (b), and in (c) we show the experimental CoM position with superimposed Digit illustration. The vertical gray lines indicate the foot contact switching instants of each walking step.

walks in the opposite direction. In Fig. 18(c), two groups of pedestrians walk towards Digit but with a slightly different lateral direction. In all scenarios, Digit successfully generates socially acceptable and safe paths while maintaining proactive forward motion.

Fig. 19 illustrates the tracking performance comparisons between the desired (v^{loc}, θ) from SZN-MPC and the actual hardware responses of Digit. Fig. 19(a) shows the tracking performance of the heading for Digit θ, and Fig. 19(b) shows the CoM sagittal velocity v^{loc} tracking performance. The continuous desired velocity profile is the desired continuous ROM velocity based on v^{loc} [1], [3]. SZN-MPC updates the target parameters (v^{loc}, θ) at every walking step, with vertical gray lines marking the foot contact events. Discrepancies are bound to occur when using a ROM plan to control a full-order system. These issues become especially noticeable with elements like body orientation (see Fig. 19(a)), which the ROM does not explicitly account for. While our current framework enables compensating for this model mismatch at runtime, for future work, we aim to explore incorporating more accurate models into our SZN-MPC.

Altogether, our hardware experiments demonstrate the efficacy of our proposed approach; our design choices for the SZN architecture, the ROM-based motion planner, and the reachable-based collision avoidance collectively enabled safe and real-time social navigation on Digit.

IX. Limitations and Discussions

We now discuss the limitations of our proposed framework, including the assumptions underpinning our social acceptability evaluation, the computational demands of our approach, and the constraints related to the locomotion gait of our bipedal robot.

A. Social Acceptability Evaluation and Datasets

The core assumption underlying our framework is that the paths recorded in human crowd datasets [70], [71] represent socially acceptable trajectories. Although this assumption may hold true, we acknowledge that in practical, real-world implementations, human crowds may behave differently in the presence of a bipedal robot. This presents a significant challenge. Collecting data on human's social reaction with Digit does not ensure that pedestrians will behave naturally. Since the presence of bipedal robots in social settings is still relatively new and potentially intimidating, it may influence pedestrian's social navigation behavior. One way to potentially

address this issue would be to use a human pedestrian simulation in which the simulated pedestrians perceive the robot as another pedestrian within the scene, similar to the approach used in this work. However, accurately simulating pedestrian interactions remains a challenging task.

We base the design of socially acceptable paths for the ego-agent on the past paths of pedestrians. To achieve more comprehensive socially acceptable behavior for the ego-agent, further considerations could include (i) design of a natural walking gait for the robot such that the pedestrians are more at ease around the robot ego-agent; (ii) analysis of the facial expressions of surrounding pedestrians and adjusting the ego-agent's path accordingly; and (iii) incorporation of the social scenario context such as shopping mall, airport, and hospital.

SZN is trained to directly output zonotopes that represent the future trajectory prediction of the pedestrians and the ego-agent. While this approach inherently compensates for prediction distribution errors of the neural network, it does not explicitly incorporate confidence bounds on these errors, as seen in [26]. Unlike [26], where the zonotopes are constructed *after* predicting the Gaussian distribution of the future trajectories—an approach that is computationally expensive and limits real-time implementation—our method avoids this by directly outputting zonotopes during training, a key feature of our proposed method that significantly improves efficiency for real-time integration with our MPC. Integrating such quantitative bounds into the neural network training, particularly through loss functions that regulate the length of the generators, we could provide statistical guarantees, ensuring that the predictions remain robust to potential errors. We leave this for future work.

B. Computational Cost and Outdoor Implementation

We opted for a relatively straightforward neural network architecture to enable real-time implementation. However, processing large crowds in real-time can be challenging. This issue could be mitigated by employing a more advanced filtering scheme (rather than the simple radius threshold used in our study) to exclude pedestrians that do not impact the ego-agent's path. Ideally, this new filtering scheme could enable an optimized neural network design that does not increase the computational burden during the trajectory optimization phase. Additionally, outdoor deployments may increase computational demands due to the complexity of the environment, which includes not only pedestrians but also static and dynamic obstacles and other environmental components. Using sensors to track pedestrians and processing this data on the fly for SZN-MPC could extend the solve time beyond acceptable limits.

C. Locomotion Gait

In our implementation, we constrain the walking gait to maintain stable torso Euler angles and fixed arm motion. In human locomotion, particularly in narrow spaces such as corridors, socially acceptable behaviors often involve adjusting the torso yaw angle while continuing straight center-of-mass motion. By adjusting the torso, a person can reduce their effective width, making it easier to pass by others without direct contact. This subtle change in body orientation helps to communicate intent and awareness of social norms, facilitating smoother and more courteous interactions in confined areas. In scenarios where forward movement is obstructed by standing pedestrians, the capability to walk sideways could provide a more efficient and socially acceptable solution to navigate around other pedestrians. Without this capability, the robot is forced to either come to a complete stop or walk backward.

TABLE II
NETWORK ARCHITECTURE PARAMETERS

Pedestrian Prediction Network	
E_{ped}	$16 \to 32 \to 16$
E_{end}	$2 \to 8 \to 16$
E_{nxt}	$2 \to 32 \to 16$
P_{future}	$50 \to 32 \to 16 \to 32 \to 70$
E_{latent}	$48 \to 8 \to 16 \to 32$
D_{latent}	$48 \to 32 \to 16 \to 32 \to 2$
Ego-agent Social Network	
E_{goal}	$2 \to 8 \to 16 \to 2$
E_{future}	$16 \to 64 \to 32 \to 16$
E_{nxt}	$2 \to 64 \to 32 \to 2$
E_{traj}	$16 \to 64 \to 32 \to 16$
E_{latent}	$36 \to 8 \to 50 \to 16$
D_{latent}	$36 \to 128 \to 64 \to 128 \to 70$

D. Environment and Task

Our framework only considers an open space environment with no static obstacles and a simple reach-avoid task. To expand this capability, more complex navigation tasks in an environment with both dynamic and static obstacles can be implemented by using high-level planners hierarchically connected as a layer on top of the SZN-MPC, e.g., formal task planning methods as in [3]. This hierarchical approach allows for better handling of more sophisticated tasks and possibly adversarial environmental events. Furthermore, the modeling error GP in Sec. VI-B can be expanded to include terrain profile uncertainty, enabling the robot to navigate complex and rough terrain [69], [79].

We acknowledge that a social environment may include various dynamic agents beyond walking pedestrians, such as bicycles, cars, and scooters. These agents typically move faster and are geometrically larger than pedestrians. For faster-moving agents, their observed past trajectories implicitly encode the velocity information, since for faster-moving agents the spacing between the discrete position observations of the pedestrian ($\mathcal{T}^{p_k}_{[t_p,t],q}$) will be longer than slow-moving agents given a constant step time (0.4 s for the dataset we use). This observation will allow SZN to predict future trajectories that account for the observed velocity in an implicit way. In the case of larger agents, our framework can be further enhanced by identifying the *type* of agent [80] and adjusting the generators of the predicted zonotopes to accommodate their size and motion characteristics. Exploring these dynamic agents different from pedestrians is out of the scope of our framework, which merely focuses on pedestrian-based social navigation.

X. Conclusion

Integration of bipedal robotic systems in real-world environments is still an open problem, specifically in human-oriented environments. These environments are less predictable and require a nuanced level of social interaction between the robot and humans.

Towards solving this problem, we presented a novel approach for socially acceptable bipedal navigation. At the core of our framework is SZN, a learning architecture for reachability-based prediction of future pedestrian reachable sets and planning a socially acceptable reachable set for the robot parameterized as zonotopes. Zonotopes allowed for efficient modulation of the reachable set based on modeling uncertainty and personal space preferences. Our integration of SZN into MPC allowed for real-time pedestrian trajectory prediction with bipedal motion planning. SZN-MPC optimizes over the output of the neural network, with a novel cost function designed to encourage the generation of socially acceptable trajectories, striking a balance between efficient navigation and adherence to social norms. The extensive validation through simulations and hardware experiments solidifies the effectiveness of the proposed framework.

Appendix A
Derivation of Linear Inverted Pendulum Model

In a manner similar to the derivation of the step-to-step discrete Linear Inverted Pendulum (LIP) dynamics as in [6] and [57], the continuous motion of the LIP dynamics in the sagittal direction is governed by $\ddot{x}^{\text{loc}} = -(g/H)u^f$, where g is the gravitational acceleration, H is CoM height, and u^f is the sagittal distance of the stance foot from CoM. The closed-form solution is given by

$$\begin{bmatrix} x^{\text{loc}}(t) \\ v^{\text{loc}}(t) \end{bmatrix} = \begin{bmatrix} 1 & \sinh(\omega t)/\omega \\ 0 & \cosh(\omega t) \end{bmatrix} \begin{bmatrix} x^{\text{loc}}(0) \\ v^{\text{loc}}(0) \end{bmatrix} + \begin{bmatrix} 1-\cosh(\omega t) \\ -\omega \sinh(\omega t) \end{bmatrix} u^f$$

where $\omega = \sqrt{g/H}$. Setting each step duration to a constant T, such that the state at the $(q+1)^{\text{th}}$ step is $x_{q+1} = x_q(T)$, we obtain the step-to-step discrete LIP model as:

$$\begin{bmatrix} x^{\text{loc}}_{q+1} \\ v^{\text{loc}}_{q+1} \end{bmatrix} = \begin{bmatrix} 1 & \sinh(\omega T)/\omega \\ 0 & \cosh(\omega T) \end{bmatrix} \begin{bmatrix} x^{\text{loc}}_{q} \\ v^{\text{loc}}_{q} \end{bmatrix} + \begin{bmatrix} 1-\cosh(\omega T) \\ -\omega \sinh(\omega T) \end{bmatrix} u^f_q$$

Therefore the sagittal CoM position change in one walking step can be expressed as:

$$\Delta x^{\text{loc}} = x^{\text{loc}}_{q+1} - x^{\text{loc}}_{q} = v^{\text{loc}} \sinh(\omega T)/\omega + (1-\cosh(\omega T))u^f_q$$

To obtain the dynamics $\boldsymbol{x} = (\boldsymbol{p}, v^{\text{loc}}, \theta)$, where $\boldsymbol{p} = (x, y)$, we introduce the heading change based on $u^{\Delta\theta}_q = \theta_{q+1} - \theta_q$ to the sagittal dynamics and obtain the following set of dynamics as in (3):

$$\begin{aligned} x_{q+1} &= x_q + \Delta x^{\text{loc}} \cos(\theta_q) \\ y_{q+1} &= y_q + \Delta x^{\text{loc}} \sin(\theta_q) \\ v^{\text{loc}}_{q+1} &= \cosh(\omega T) v^{\text{loc}}_q - \omega \sinh(\omega T) u^f_q \\ \theta_{q+1} &= \theta_q + u^{\Delta\theta}_q \end{aligned}$$

Acknowledgment

The authors would like to express their gratitude to Dr. Seth Hutchinson and Dr. Sehoon Ha for allowing them to use their lab space and motion capture system. Special thanks to Nitish Sontakke for helping with the motion capture system setup and to Vignesh Subramanian, Abivishaq Balasubramanian, Meghna Narayanan, Hemanth Sai Surya Kumar Tammana, Harini Mudradi, Venkat Sai Chinta, and Manthan Joshi for their participation in the hardware experiments.

References

[1] Y. Gong and J. W. Grizzle, "Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion," *J. Dyn. Syst., Meas., Control*, vol. 144, no. 12, Dec. 2022, Art. no. 121006.

[2] H. Sadeghian, C. Ott, G. Garofalo, and G. Cheng, "Passivity-based control of underactuated biped robots within hybrid zero dynamics approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4096–4101.

[3] A. Shamsah, Z. Gu, J. Warnke, S. Hutchinson, and Y. Zhao, "Integrated task and motion planning for safe legged navigation in partially observable environments," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4913–4934, Dec. 2023.

[4] Z. Li, J. Zeng, S. Chen, and K. Sreenath, "Autonomous navigation of underactuated bipedal robots in height-constrained environments," *Int. J. Robot. Res.*, vol. 42, no. 8, pp. 565–585, Jul. 2023.

[5] J.-K. Huang and J. W. Grizzle, "Efficient anytime CLF reactive planning system for a bipedal robot on undulating terrain," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1–18, Jan. 2023.

[6] K. S. Narkhede, A. M. Kulkarni, D. A. Thanki, and I. Poulakakis, "A sequential MPC approach to reactive planning for bipedal robots using safe corridors in highly cluttered environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11831–11838, Oct. 2022.

[7] G. Gibson, O. Dosunmu-Ogunbi, Y. Gong, and J. Grizzle, "Terrain-adaptive, ALIP-based bipedal locomotion controller via model predictive control and virtual constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 6724–6731.

[8] Y. Zhao, Y. Li, L. Sentis, U. Topcu, and J. Liu, "Reactive task and motion planning for robust whole-body dynamic locomotion in constrained environments," *Int. J. Robot. Res.*, vol. 41, no. 8, pp. 812–847, Jul. 2022.

[9] S. Kulgod, W. Chen, J. Huang, Y. Zhao, and N. Atanasov, "Temporal logic guided locomotion planning and control in cluttered environments," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 5425–5432.

[10] J. Warnke, A. Shamsah, Y. Li, and Y. Zhao, "Towards safe locomotion navigation in partially observable environments with uneven terrain," in *Proc. IEEE Conf. Decis. Control*, Dec. 2020, pp. 958–965.

[11] Y. Zhao, B. R. Fernandez, and L. Sentis, "Robust optimal planning and control of non-periodic bipedal locomotion with a centroidal momentum model," *Int. J. Robot. Res.*, vol. 36, no. 11, pp. 1211–1242, Sep. 2017.

[12] C. Mavrogiannis et al., "Core challenges of social robot navigation: A survey," *ACM Trans. Human-Robot Interact.*, vol. 12, no. 3, pp. 1–39, Sep. 2023.

[13] Y. Che, A. M. Okamura, and D. Sadigh, "Efficient and trustworthy social navigation via explicit and implicit robot–human communication," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 692–707, Jun. 2020.

[14] A. Bera, T. Randhavane, and D. Manocha, "The emotionally intelligent robot: Improving socially-aware human prediction in crowded environments," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jan. 2019, p. 0.

[15] D. Paez-Granados, V. Gupta, and A. Billard, "Unfreezing social navigation: Dynamical systems based compliance for contact control in robot navigation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8368–8374.

[16] H. Nishimura, B. Ivanovic, A. Gaidon, M. Pavone, and M. Schwager, "Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 11205–11212.

[17] S. Schaefer, K. Leung, B. Ivanovic, and M. Pavone, "Leveraging neural network gradients within trajectory optimization for proactive human-robot interactions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 9673–9679.

[18] K. Majd, S. Yaghoubi, T. Yamaguchi, B. Hoxha, D. Prokhorov, and G. Fainekos, "Safe navigation in human occupied environments using sampling and control barrier functions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 5794–5800.

[19] C. Cathcart, M. Santos, S. Park, and N. E. Leonard, "Proactive opinion-driven robot navigation around human movers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 4052–4058.

[20] K. Cai, W. Chen, C. Wang, S. Song, and M. Q.-H. Meng, "Human-aware path planning with improved virtual Doppler method in highly dynamic environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 2, pp. 1304–1321, Apr. 2023.

[21] Z. Gu, R. Guo, W. Yates, Y. Chen, Y. Zhao, and Y. Zhao, "Walking-by-logic: Signal temporal logic-guided model predictive control for bipedal locomotion resilient to external perturbations," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 1121–1127.

[22] Z. Gu, N. Boyd, and Y. Zhao, "Reactive locomotion decision-making and robust motion planning for real-time perturbation recovery," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 1896–1902.

[23] L. Yang, Z. Li, J. Zeng, and K. Sreenath, "Bayesian optimization meets hybrid zero dynamics: Safe parameter learning for bipedal locomotion control," 2022, *arXiv:2203.02570*.

[24] M. Dai, X. Xiong, J. Lee, and A. D. Ames, "Data-driven adaptation for robust bipedal locomotion with step-to-step dynamics," 2022, *arXiv:2209.08458*.

[25] A. Robotics. *Digit Robot*. [Online]. Available: https://agilityrobotics.com/robots

[26] L. Paparusso, S. Kousik, E. Schmerling, F. Braghin, and M. Pavone, "ZAPP! Zonotope agreement of prediction and planning for continuous-time collision avoidance with discrete-time dynamics," 2024, *arXiv:2406.01814*.

[27] M. Selim, A. Alanwar, S. Kousik, G. Gao, M. Pavone, and K. H. Johansson, "Safe reinforcement learning using black-box reachability analysis," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10665–10672, Oct. 2022.

[28] S. Kousik, P. Holmes, and R. Vasudevan, "Safe, aggressive quadrotor flight via reachability-based trajectory design," in *Proc. Dyn. Syst. Control Conf.*, vol. 59162, 2019, p. V003T19A010.

[29] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Inst. Autom. Control Eng., Technische Universität München, Munich, Germany, 2010.

[30] A. Shamsah, K. Agarwal, S. Kousik, and Y. Zhao, "Real-time model predictive control with zonotope-based neural networks for bipedal social navigation," 2024, *arXiv:2403.16485*.

[31] M. Moder and J. Pauli, "Proactive robot movements in a crowd by predicting and considering the social influence," in *Proc. 31st IEEE Int. Conf. Robot Hum. Interact. Commun. (RO-MAN)*, Aug. 2022, pp. 644–651.

[32] X. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 4, pp. 1743–1760, Oct. 2017.

[33] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, Sep. 2016.

[34] S. Vaskov, H. Larson, S. Kousik, M. Johnson-Roberson, and R. Vasudevan, "Not-at-fault driving in traffic: A reachability-based approach," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2785–2790.

[35] S. Vaskov et al., "Towards provably not-at-fault control of autonomous robots in arbitrary dynamic environments," in *Proc. Robot., Sci. Syst.*, FreiburgimBreisgau, Germany, Jun. 2019.

[36] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4459–4466, Oct. 2019.

[37] V. B. Hoang, V. H. Nguyen, T. D. Ngo, and X.-T. Truong, "Socially aware robot navigation framework: Where and how to approach people in dynamic social environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 2, pp. 1322–1336, Apr. 2023.

[38] F. Feurtey, "Simulating the collision avoidance behavior of pedestrians," School Eng., Dept. Electron. Eng., Univ. Tokyo, Tokyo, Japan, Tech. Rep., 2000.

[39] 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," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 335–356, Mar. 2015.

[40] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 683–700.

[41] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.

[42] K. Mangalam et al., "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 759–776.

[43] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15213–15222.

[44] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1349–1358.

[45] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 507–523.

[46] Y. Hong, Z. Ding, Y. Yuan, W. Chi, and L. Sun, "Obstacle avoidance learning for robot motion planning in human-robot integration environments," *IEEE Trans. Cognit. Develop. Syst.*, vol. 15, no. 4, Feb. 2023.

[47] P. Zaytsev, W. Wolfslag, and A. Ruina, "The boundaries of walking stability: Viability and controllability of simple models," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 336–352, Apr. 2018.

[48] Y. Ding, C. Khazoom, M. Chignoli, and S. Kim, "Orientation-aware model predictive control with footstep adaptation for dynamic humanoid walking," in *Proc. IEEE-RAS 21st Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2022, pp. 299–305.

[49] G. Romualdi, S. Dafarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online non-linear centroidal MPC for humanoid robot locomotion with step adjustment," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 10412–10419.

[50] N. Smit-Anseeuw, C. D. Remy, and R. Vasudevan, "Walking with confidence: Safety regulation for full order biped models," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4177–4184, Oct. 2019.

[51] G. A. Castillo, B. Weng, S. Yang, W. Zhang, and A. Hereid, "Template model inspired task space learning for robust bipedal locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 8582–8589.

[52] Z. Li et al., "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 2811–2817.

[53] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," 2023, *arXiv:2303.03381*.

[54] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 370–387, Apr. 2018.

[55] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, "Models, feedback control, and open problems of 3D bipedal robotic walking," *Automatica*, vol. 50, no. 8, pp. 1955–1988, Aug. 2014.

[56] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Expanding Societal Role Robot. Next Millennium*, vol. 1, Oct./Nov. 2001, pp. 239–246.

[57] S. Teng, Y. Gong, J. W. Grizzle, and M. Ghaffari, "Toward safety-aware informative motion planning for legged robots," 2021, *arXiv:2103.14252*.

[58] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in *Proc. Robot. Sci. Syst.*, vol. 13. Cambridge, MA, USA, 2017, pp. 1–10.

[59] L. Guibas, A. Nguyen, and L. Zhang, "Zonotopes as bounding volumes," in *Proc. SODA*, Jan. 2003, pp. 803–812.

[60] B. Ivanovic, E. Schmerling, K. Leung, and M. Pavone, "Generative modeling of multimodal multi-human behavior," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3088–3095.

[61] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5308–5317.

[62] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Proc. Int. Symp. Formal Techn. Real-Time Fault-Tolerant Syst.* Cham, Switzerland: Springer, 2004, pp. 152–166.

[63] K. Leung, N. Aréchiga, and M. Pavone, "Back-propagation through signal temporal logic specifications: Infusing logical structure into gradient-based methods," in *Proc. 14th Workshop Algorithmic Found. Robot.* Cham, Switzerland: Springer, 2021, pp. 432–449.

[64] X. Li et al., "Vehicle trajectory prediction using generative adversarial network with temporal logic syntax tree features," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3459–3466, Apr. 2021.

[65] E. T. Hall, *The Hidden Dimension: Man's Use of Space in Public and Private the Bodley Head*, vol. 121. Toronto, ON, Canada: C. J. Hogrefe, 1969.

[66] P. Patompak, S. Jeong, I. Nilkhamhang, and N. Y. Chong, "Learning proxemics for personalized human–robot social interaction," *Int. J. Social Robot.*, vol. 12, no. 1, pp. 267–280, Jan. 2020.

[67] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *Int. J. Social Robot.*, vol. 7, no. 2, pp. 137–153, Apr. 2015.

[68] J. Jiang, S. Coogan, and Y. Zhao, "Abstraction-based planning for uncertainty-aware legged navigation," *IEEE Open J. Control Syst.*, vol. 2, pp. 221–234, 2023.

[69] K. Muenprasitivej, J. Jiang, A. Shamsah, S. Coogan, and Y. Zhao, "Bipedal safe navigation over uncertain rough terrain: Unifying terrain mapping and locomotion stability," 2024, *arXiv:2403.16356*.

[70] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, Sep. 2007.

[71] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 261–268.

[72] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6150–6156.

[73] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Jan. 2019.

[74] F. Fiedler et al., "Do-mpc: Towards FAIR nonlinear and robust model predictive control," *Control Eng. Pract.*, vol. 140, Nov. 2023, Art. no. 105676.

[75] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, Mar. 2019.

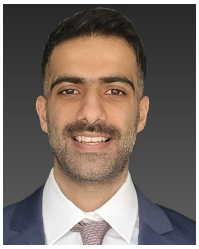
[76] V. Makoviychuk et al., "Isaac gym: High performance GPU-based physics simulation for robot learning," 2021, *arXiv:2108.10470*.

[77] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *Proc. IEEE Amer. Control Conf. (ACC)*, May 2021, pp. 3882–3889.

[78] V. M. Systems. *Vicon Vero*. [Online]. Available: https://www.vicon.com/

[79] A. Shamsah, J. Jiang, Z. Yoon, S. Coogan, and Y. Zhao, "Terrain-aware model predictive control of heterogeneous bipedal and aerial robot coordination for search and rescue tasks," 2024, *arXiv:2409.15174*.

[80] B. Ivanovic et al., "Heterogeneous-agent trajectory forecasting incorporating class uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 12196–12203.

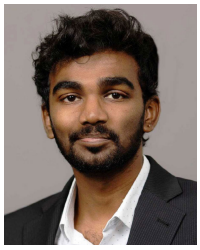


Abdulaziz Shamsah received the B.S. degree in mechanical engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2016, and the M.S.E. degree in mechanical engineering and applied mechanics from the University of Pennsylvania, Philadelphia, PA, USA, in 2018. He is currently pursuing the Ph.D. degree in mechanical engineering with Georgia Institute of Technology, Atlanta, GA, USA.

His research interests include bipedal navigation in real-world environments, formal methods, and safety and robustness in locomotion. In 2019, he was awarded the Fellowship from Kuwait University, Kuwait.

Krishanu Agarwal received the B.Tech. degree in mechatronics from Manipal University Jaipur, Jaipur, Rajasthan, India, in 2021, and the M.S. degree in robotics from Georgia Institute of Technology, Atlanta, GA, USA, in 2024.

His research interests include robot learning, path planning, generative AI, and locomotion safety.



Nigam Katta received the B.Tech. degree in electronics and communication engineering from Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India, in 2020, and the M.S. degree in robotics from Georgia Institute of Technology, Atlanta, GA, USA, in 2024.

His research interests include computer vision, photogrammetry, and bipedal robot navigation.

Abirath Raju received the B.Tech. degree in mechanical engineering with a minor in computer science from the National Institute of Technology, Tiruchirappalli, Tamil Nadu, India, in 2023. He is currently pursuing the M.S. degree in robotics with Georgia Institute of Technology, Atlanta, GA, USA.

His research interests include computer vision, bipedal robot navigation, and robot learning.

Shreyas Kousik (Member, IEEE) received the B.S. degree in mechanical engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2014, and the M.S. and Ph.D. degrees in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2020. He was a Post-Doctoral Scholar at Stanford University, Stanford, CA, USA. He is currently an Assistant Professor with George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology. His research interests include safety for full-stack robot autonomy.

Ye Zhao (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from The University of Texas at Austin, Austin, TX, USA, in 2016. He was a Post-Doctoral Fellow with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. He is currently an Assistant Professor with George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include robust task and motion planning, contact-rich trajectory optimization, and formal methods for legged locomotion and navigation.

Dr. Zhao was the Co-Chair of the IEEE RAS Student Activities Committee. He is the Co-Chair of the IEEE Robotics and Automation Society (RAS) Technical Committee on Whole-Body Control. He received the CAREER Award from the National Science Foundation in 2022 and the Young Investigator Award from the Office of Naval Research in 2023. He serves as an Associate Editor for IEEE TRANSACTIONS ON ROBOTICS (TRO), IEEE/ASME TRANSACTIONS ON MECHATRONICS (TMECH), IEEE ROBOTICS AND AUTOMATION LETTERS (RA-L), and IEEE CONTROL SYSTEMS LETTERS (L-CSS).