Following the Human Thread in Social Navigation

Luca Scofano[1]*, Alessio Sampieri[1]*, Tommaso Campari[2]*, Valentino Sacco[1],
Indro Spinelli[1], Lamberto Ballan[3], and Fabio Galasso[1]

[1] Sapienza University of Rome
[2] Fondazione Bruno Kessler (FBK)
[3] University of Padova

Abstract. The success of collaboration between humans and robots in shared environments relies on the robot's real-time adaptation to human motion. Specifically, in Social Navigation, the agent should be close enough to assist but ready to back up to let the human move freely, avoiding collisions. Human trajectories emerge as crucial cues in Social Navigation, but they are partially observable from the robot's egocentric view and computationally complex to process.

We propose the first Social Dynamics Adaptation model (SDA) based on the robot's state-action history to infer the social dynamics. We propose a two-stage Reinforcement Learning framework: the first learns to encode the human trajectories into social dynamics and learns a motion policy conditioned on this encoded information, the current status, and the previous action. Here, the trajectories are fully visible, i.e., assumed as privileged information. In the second stage, the trained policy operates without direct access to trajectories. Instead, the model infers the social dynamics solely from the history of previous actions and statuses in real-time. Tested on the novel Habitat 3.0 platform, SDA sets a novel state of the art (SoA) performance in finding and following humans.

The code is available at: https://github.com/L-Scofano/SDA.

Keywords: Embodied AI · Social Navigation · Human Trajectories

1 Introduction

Traditional navigation techniques within Embodied Artificial Intelligence (EAI) have marked a crucial advancement by introducing robots into real-life environments. However, these techniques have primarily focused on agents traversing vacant spaces. Conversely, the significance of social navigation within EAI has steadily increased. Social navigation entails agents' capacity to navigate human-centric environments while considering human movements and behaviours. These agents need to be able to locate, track, and interact with humans in a safe and
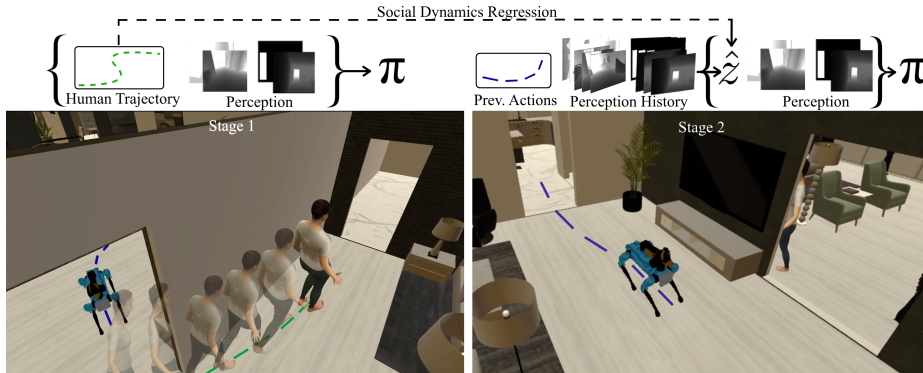
Fig. 1: We present our novel Social Dynamic Adaptation model (SDA). The framework involves two stages of training that allow the model to infer, given its past observations and actions, another agent's Social Dynamics. In the first training stage, the model embeds the followed agent's trajectory, which together with sensor perceptions compose the input to the model's Social navigation Policy (π). The knowledge obtained from the human trajectory strongly helps the navigation policy in finding and following an agent, however, this information is often not available during deployment. In the second stage, SDA learns to adapt past statuses and actions, which are always available, to the first stage's Social Dynamics embedding $\hat{z}$. As depicted in the figure, the status contains depth maps and BB detection of the person, if observable from the egocentric robot view. $\hat{z}$ is then paired with current observations as input to the frozen π.

socially acceptable manner. Previous studies predominantly characterized Social Navigation as a variation of PointGoal Navigation, wherein agents strive to reach specified destinations while considering human movements [32, 48, 50]. Habitat 3.0 [36], a significant breakthrough in EAI, introduces a lifelike environment seamlessly incorporating human avatars. This integration enables the investigation of human-agent interactions within a controlled, risk-free, dynamic environment. What sets Habitat 3.0 [36] apart is its ability to replicate complex scenarios where human intentions are constantly changing. Nevertheless, this dynamism also presents particular challenges, such as collision avoidance and achieving success in locating and following humans.

Despite notable efforts in collision avoidance and safety, most existing methods for Social Navigation either rely on privileged information that is not available in real-world scenarios or do not adequately capture the social dynamics and norms of human behaviour. For instance, [48] and [50] use a GPS and compass sensor to provide the agent with perfect localization, which might be unrealistic even if using SLAM [17, 31] methods, whenever the human is not in the line of sight. While [32] and [52] do not account for the social factors that influence human behaviour. Therefore, this limitation hinders their practical applicability and adaptability to dynamic environments. [9], instead, models some social factors in the form of Proximity Tasks but fails to account for the cooperative nature that a social agent must possess, restricting itself to merely avoid-

ing collisions with them. The current SoA model proposed in [36] necessitates privileged information, such as humanoid GPS, which offers polar coordinates, detailing the accurate distance and angle of the human from the robot, to attain high-performance outcomes. However, this requirement is highly impractical in real-world environments during inference.

This paper proposes a novel Social Dynamics Adaptation model (SDA), shown in Fig. 2, that effectively solves the robot's awareness of complex human behaviors, even temporarily losing sight of the person and fast robot motion. Specifically, the first stage trains a base policy considering human trajectories encoded into a latent vector. The latent vector is a low-dimensional, nonlinear projection of the human trajectories, and it is trained end-to-end with the base policy to extract the social factors that led the robots to choose better actions. The subsequent supervised adaptation stage regresses this latent vector using only the robot's state and action history. Unlike previous methods, such as [48] and [52], which often rely on simulated privileged information or simplified social behavior models, our approach learns to respond to real-world dynamics in real time. In summary, SDA adapts and accounts for unpredictable human behaviors by exploiting privileged information during training and recovering this fundamental signal during deployment when it is often impractical to compute. Finally, the deployed robot features the motion policy, learned in the first stage, and the social dynamics, inferred from prior statuses and actions.

Out of extensive benchmarking, SDA outperforms the approach proposed in Habitat 3.0 [36] and a second adapted best performer method [9] from Habitat 2.0 [43]. We conduct a thorough experimental evaluation on the core contribution of this work, i.e., learning to infer the social dynamics from (privileged) information about the person. Remarkably, out of ablative studies, we conclude that not only are human trajectories strong input information for the robot control policy, but they also make better supervision for inferring the social dynamics latent for the same policy. Other oracular information such as the humanoid GPS (direction and distance from the person-to-robot) are powerful sensors for the control policy but do not yield adaptable social dynamics to infer the human-robot-scene interaction.

2   Related Work

Embodied Navigation. Recently, there has been a surge in exploring indoor navigation within an embodied framework [16]. This upswing has been facilitated primarily by the availability of large-scale datasets comprising 3D indoor environments [10, 39, 42] and simulators designed for simulating navigation within these dynamic 3D spaces [23, 41, 42]. Nevertheless, these simulators are not equipped to handle human entities within the environments, restricting the investigation to navigation tasks in scenarios where the agent functions independently or, at most, alongside humans simulated via static meshes that simulate movement [49, 52]. These simulated humans are treated as dynamic obstacles and lack compliance with any social construct. This constraint has been effectively

addressed with Habitat 3.0 [36], the simulator used for this research. Habitat 3.0 introduces the capability to simulate the behaviours of humans engaging in tasks within dynamic environments, thus overcoming the limitations mentioned above. Furthermore, there is no longer a need to directly train social models in the real world, where there is a tangible risk of causing harm to actual humans. Thus, Habitat 3.0 establishes itself as a viable proxy for real-world adaptation of social models trained in simulation.

Thanks to these simulators the realm of EAI has witnessed the introduction of numerous tasks [16], including PointGoal Navigation [48], ObjectGoal Navigation [4], Embodied Question Answering [47], and Vision and Language Navigation (VLN) [2, 24]. Various modular approaches [8, 11, 12, 38, 40] have been proposed to address the challenges of navigating through static, single-agent environments. These approaches utilize maps constructed from depth images and conduct path planning directly on these maps. However, these approaches are unsuitable in social settings where dynamic objects (humans) move within the environment. This is because humans are observable only within the agent's field of view (FOV). Moreover, the agent must address the additional challenge of tracking and mapping. End-to-end RL-trained policies [7,32,46,50,51], should be adapted to learn also social clues similarly to [9], where the agent learned proxemics information about the humans moving in the environments thanks to two proximity tasks. In this paper, instead, we try to learn social behaviours directly by internally modeling the humanoid trajectories in the latent representation of the agent. Furthermore, differently from [9], the agent's aim is no longer just avoiding collisions. Still, it involves locating this dynamically acting human and following them for a specified number of steps while maintaining a safe distance. This evolution of the task demands a heightened level of social comprehension from the agent, requiring the anticipation of the person's intentions and the ability to trail them closely without compromising safety.

Socially-Aware Navigation. Research in robotics, computer vision, and the analysis of human social behavior explored socially aware representations and models [30]. Extending from the realm of collision-free multi-agent navigation [5, 6, 15, 27] and navigation in dynamic environments [3], researchers have further expanded their investigations to encompass scenarios involving human presence [13, 14, 18, 20, 29].

The approach presented in [14] incorporates collision avoidance algorithms like CADRL [15] while introducing common-sense social rules. This integration aims to reduce uncertainty while minimizing the risk of encountering the Freezing Robot Problem [44]. Other works [13,18] seek to model human-agent interaction by employing techniques such as spatiotemporal graphs [29].

Typically, these methods undergo testing in a minimal number of environments that offer complete knowledge about the human positions and velocities [13,15], featuring simple obstacles and often assuming collaboration between moving agents [13,18]. In contrast, our focus revolves around SocialNav within expansive indoor environments, characterized by partial knowledge about them, since the agent perceives the environment only through its sensors from an ego-
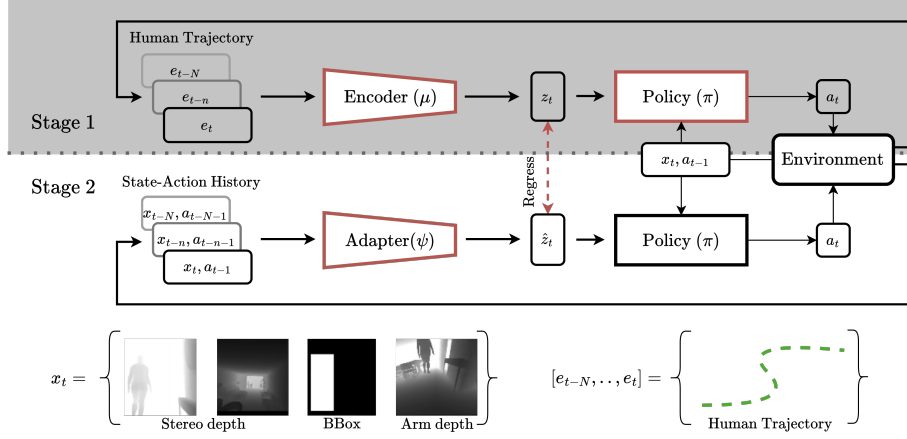
Fig. 2: Pipeline of the novel methodology proposed. First, we jointly learn to encode human trajectories and a motion policy. In the next stage, given the previous states and actions, we infer the social dynamics and pass the estimated latent vector to the frozen policy.

centric perspective, and no information about the velocity or position of the human is given to the agent. Our SDA addresses the missing information from the interacting human position, inferring it from the robot's history of actions and status.

Modelling dynamic environments. Simulated environments [26, 35, 45] offer privileged information about the scene whose exploitation can be computationally intense or unfeasible during deployment. While navigating social environments, it is vital to take into consideration human behaviour [22]. Ideally, one would want to forecast people's position for better path planning [33], but forecasting robot-person interactions is significantly slower [37] than navigation policies, hence being challenging to be considered for training or deployment. To overcome this problem, we leverage literature on system identification [1, 19] to infer the encoded privileged information during robot navigation. Once encoded in a latent space during the first training stage and used to train the primary policy, it is possible to asynchronously derive that same information from the state-action history [25, 28, 53], influenced by the signal we want to identify. Unlike previous works, we are the first to identify the social dynamics (under the form of human trajectories), with the intuition that modeling human behavior is fundamental for efficient human-robot collaboration.

3   Methodology

This section introduces a novel framework, SDA, for social navigation that incorporates human trajectories into the sequential decision-making process. The first stage of our approach focuses on encoding human trajectories into a latent social context to represent the social dynamics that are functional to the agent motion policy. In the second stage, we introduce the Adapter module, which enables the estimation of social information from the agent's past behavior. Recovering this signal allows the robot to operate in the real world without explicitly representing human behavior. We elucidate the training process, trajectory modeling, and optimization techniques utilized in our approach, highlighting its effectiveness in addressing the challenges of social navigation.

Problem formalization. The problem involves the task of locating and following a humanoid in motion within a household environment, maintaining a distance of 1 to 2 meters for at least k steps [36].

With the status x_t, we represent the agent's "perception" at time t. We exploit depth images from different cameras placed on the robot and a preprocessed version containing a humanoid detection bounding box. These perceptions are processed with a ResNet [21] before being fed to the recurrent policy network, selecting the best action a_t at time t. We use Decentralized Distributed Proximal Policy Optimization (DD-PPO) [48] to iteratively improve the agent's policy while maximizing rewards derived from interactions with several environments executed in parallel, ensuring stability through controlled policy updates and lower training times.

The pipeline described above can be considered a baseline implementation that does not contain privileged information but relies only on the robot's on-board sensors. To address the social aspect required by the task, we consider additional details on the humanoid, e.g., "social dynamics", defined as $e_{t-N:t-1}$. In the following sections, we describe how exploiting this information at training time can improve performance during deployment when it is absent.

3.1   Stage 1: Social Policy

Recurrent policies such as DD-PPO take as input the current status, in our case, a collection of depth images processed via a Resnet x_t, and the action at the previous time-step a_{t-1}. We add another input to this pipeline, namely a latent vector z_t built by encoding the humanoid privileged information $e_{t-N:t-1}$:

$$z_t = \mu(e_{t-N:t-1}) \tag{1}$$

$$a_t = \pi(x_t, a_{t-1}, z_t) \tag{2}$$

Intuitively, by training everything with the same objective z_t encodes the social dynamics that led the agent to maximize its reward, adapting to human movement patterns. Additionally, including trajectory data allows the agent to learn from past interactions and experiences. In our approach, the trajectory encoder

(μ) is implemented as Multilayer Perceptrons (MLPs). The objective retains its usual formulation without any explicit reference to the human trajectories:

$$L^{CLIP} = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1 - \epsilon, 1 + \epsilon\right) \hat{A}_t \right) \right] \tag{3}$$

where $r_t(\theta)$ is the ratio of the probability of action a_t under the current policy and the previous one that is being executed for gathering data. $\hat{A}_t$ represents the advantage function at time t, guiding the policy towards actions that yield higher expected rewards.

Defining what information is considered "privileged" and why is essential. In our context, it refers to detailed knowledge about the humanoid in the environment, such as the exact position of humanoids defining a trajectory (traj.), of the relative position with respect to the agent often denoted as humanoid GPS [36](hGPS). We can easily gather this information in simulated environments. However, collecting them in real-life scenarios is often impractical. This distinction is crucial as it highlights the challenge of transferring learned policies from simulation to the real world.

3.2   Stage 2: Social Dynamics Regression

We aim to extract and exploit social cues directly from the robot's perception and eliminate the need for auxiliary devices like GPS trackers on humanoids. Inspired by [25], we introduce the "social dynamics" module (Adapter), parametrized by an MLP ψ that takes as input the recent history of the robot's states $x_{t-N:t-1}$ and actions $a_{t-N:t-1}$ to generate a new latent vector $\hat{z}_t$:

$$\hat{z}_t = \psi(x_{t-N:t-1}, a_{t-N:t-1}) \tag{4}$$

We obtain the state-action history by deploying the agent in the environment with optimal policy π^* obtained after the first stage and the latent vector $\hat{z}_t$:

$$a_t = \pi^*(x_t, a_{t-1}, \hat{z}_t) \tag{5}$$

During this process, we optimize the Adapter, MLP, with a supervised regression objective, Mean Squared Error (MSE), to recover the original information contained in z_t that we compute relying on the preferential information trajectory:

$$\text{MSE}(\hat{z}_t, z_t) = \|\hat{z}_t - z_t\|_2^2 \tag{6}$$

Once we finalize the Adapter training, instead of relying on the privileged information, we can depend upon the robot's states $x_{t-N:t-1}$ and actions $a_{t-N:t-1}$ to generate $\hat{z}_t$, serving as an estimate of the actual latent social dynamics vector z_t. Doing so enables the agent to estimate social dynamics online, improving its performance in dynamic environments and enhancing its social navigation capabilities, freeing it from dependence upon external sensors.

3.3   Training details

In our training process, we encode social dynamics using a ResNet, which we train from scratch. The trajectory encoder μ consists of a 3-layer MLP, and the output z_t has a dimensionality of 128. The Adapter ψ comprises alternating spatial and temporal MLP layers, with the output $\hat{z}_t$ matching the dimensionality of z_t. We jointly train the policy π, the perceptions encoder (ResNet), and the social dynamics (Trajectory Encoder) encoder μ during the first stage. In the second stage, everything is frozen except the Adapter.

In Stage 1, we utilize DD-PPO [48] for 250 million steps across 24 environments, following the training protocol presented in [36]. The training process takes approximately four days. In Stage 2, we employ supervised learning for 5 million steps across the same environments. The learning process lasts around 2 hours. Both stages utilize 4 A100 GPUs for efficient computation.

4   Results

Section 4.1 presents our findings regarding Social Navigation, whereas Section 4.2 conducts an ablation analysis concentrating on adaptable information and the role of the trajectory history length. Furthermore, Section 4.3 provides a qualitative exploration of our outcomes.

Simulator. We utilize our SDA on Habitat 3.0 [36], a simulation platform designed for human-robot interaction within domestic settings. This platform offers precise humanoid simulation capabilities with a strong focus on collaborative tasks such as Social Navigation and Social Rearrangement. It offers a vast library of avatars featuring multiple gender representations, body shapes, and appearances, facilitated by SMPL-X [34]. Furthermore, it employs an oracle policy to generate movement and behavior, enabling programmable control of avatars for navigation, object interaction, and a range of other movements.

Baselines. The baselines we employ in our study are drawn from Habitat 3.0 [36] and consist of:

- *Heuristic Expert*: a heuristic baseline equipped with access to the environment map, employing a shortest path planner to devise a route to the current location of the humanoid. The heuristic expert operates on the following principles: When the agent is beyond a distance of 1.5 meters from the humanoid, it employs a "find" behavior, utilizing a path planner to approach the humanoid. Conversely, if the humanoid is within 1.5 meters, the expert executes a backup motion to prevent collision with the humanoid.
- *Baseline*: the current SoA method [36], a recurrent neural network policy trained with DD-PPO [48], operates on a "sensors-to-action" paradigm. Inputs to this policy consist of an egocentric arm and stereo depth sensors, a humanoid detector, and humanoid GPS coordinates, while the outputs

are velocity commands in the robot's local frame. Notably, this policy operates without access to an environment map. Table 1 compares our model against the most realistic configuration, wherein the humanoid GPS data is unavailable.

– *Proximity tasks*: we also adapted the Proximity Tasks defined in [9] and applied them to the baseline [36]. These tasks were proposed for a different setup of SocialNav, where the agent acts in an environment with multiple humanoids and must navigate from point A to point B while avoiding collisions. We adapted the risk and compass proximity tasks to the SocialNav setting addressed in this article. In this context, the risk has a low value (close to 0) when the agent is within 1 to 2 meters from the humanoid and a value close to 1 when the distance is less than 1 meter or greater than 2 meters. Similarly, given the presence of only one humanoid in the environment, the compass was redefined to predict the angle between the humanoid and the agent. This adjustment aims to assist during the following phase, enabling the agent to follow the humanoid while maintaining a safe distance and staying aligned with the human. The proximity tasks are jointly trained with the policy and detached during evaluation.

Metrics. We used the metrics for the SocialNav task as defined in [36], briefly summarized in the following. 1. *Finding Success (S)* is the ratio of the episodes where the agent successfully located and reached the human. 2. *Finding Success Weighted by Path Steps (SPS)* measures the optimality of the path taken by the agent w.r.t. the optimal number of steps needed to reach the human. 3. *Following Rate (F)* is the ratio of steps during which the robot maintains a distance of 1-2 meters from the humanoid while facing towards it relatively to the maximum possible following steps. 4. *Collision Rate (CR)* is the ratio of the episodes that ended with the robot colliding with the humanoid. 5. *Episode Success (ES)* measures the ratio of the episodes where the agent found the human and followed it for the required number of steps, maintaining a safe distance in the 1-2 meters range.

Privileged information. In our research, the privileged information under consideration includes humanoid GPS (hGPS) and human trajectories (traj.). Humanoid GPS is represented in polar coordinates, a method of specifying a point's position in a plane using two parameters: the distance from the point to the origin (radius) and the angle formed between a reference direction (typically the positive x-axis) and a line connecting the origin to the point (polar angle or azimuth). In our context, the origin is defined as the robot's position; thus, its position is implicitly known along with that of the human. Conversely, trajectories solely consist of information derived from the human's position within the environment.

4.1   Quantitative Results

In Table 1, we present the results obtained in Habitat 3.0 [36] for the Social Navigation task. The table is divided into three sections. The first one displays

outcomes achieved by utilizing a heuristic expert endowed with extensive information, including its position, map data, and the humanoid's position, granting it a competitive advantage over other methods. The subsequent section features models trained and tested using ground truth (GT) data (Baselines and Stage 1), thus establishing an upper limit for techniques utilizing more practical inputs feasible in real-world scenarios. Lastly, the final rows delineate results from methods conducting inference without privileged information. The models that use privileged information (GT) show that S1 has lower performance than the two Baselines, especially in S and SPS. This is to be expected, considering that while trajectories solely depict the movement of the human in the environment, hGPS furnishes crucial details on locating the humanoid, offering insights into the distance and angle between them.

In the second stage, despite the absence of human trajectory input, SDA maintains the performance level of Stage 1 by adapting to the social dynamics. Our model outperforms the best techniques in the find task, increasing S and SPS by 6%.

Our approach generally improves performance in the episode success (ES) metric, which occurs when the agent finds the humanoid and follows it for 400 steps. However, we emphasize that the test-time episode ends at 1500 steps or when the agent collides with the humanoid [36], not after the 400 follow steps. In this context, S, SPS, and F metrics demonstrate how SDA, compared to the Baseline, more frequently locates the humanoid, follows a more optimal path (on average 438 vs. 540 steps), and follows it for a longer duration (390 vs. 218 steps). Therefore, as the agent follows the humanoid longer, it has more chances to collide with it, given that the episode does not necessarily end after the required steps. In a scenario where the test-time episode concludes either after 400 follow steps or immediately upon a collision between the agent and the humanoid, SDA and the Baseline show a comparable collision rate (CR), 0.39, and 0.38, respectively.

Table 1: Main results for Social Navigation. Within the table, GT denotes ground truth privileged information and * corresponds to reproduced results.

Models	hGPS	traj.	S ↑	SPS ↑	F ↑	CR ↓	ES ↑
Heuristic Expert [36]	-	-	1.00	0.97	0.51	0.52	-
Baseline [36]	GT		$0.97^{\pm0.00}$	$0.65^{\pm0.00}$	$0.44^{\pm0.01}$	$0.51^{\pm0.03}$	$0.55^{\pm0.01}$*
Baseline+Proximity [9]*	GT		$0.97^{\pm0.01}$	$0.64^{\pm0.00}$	$0.57^{\pm0.01}$	$0.58^{\pm0.03}$	$0.63^{\pm0.02}$
SDA - S1		GT	$0.92^{\pm0.00}$	$0.46^{\pm0.01}$	$0.44^{\pm0.02}$	$0.61^{\pm0.02}$	$0.50^{\pm0.01}$
Baseline [36]			$0.76^{\pm0.02}$	$0.34^{\pm0.01}$	$0.29^{\pm0.01}$	$\mathbf{0.48}^{\pm0.03}$	$0.40^{\pm0.02}$*
Baseline+Proximity [9]			$0.85^{\pm0.02}$	$0.41^{\pm0.02}$	$0.37^{\pm0.01}$	$0.58^{\pm0.02}$	$0.41^{\pm0.01}$
SDA - S2			$\mathbf{0.91}^{\pm0.01}$	$\mathbf{0.45}^{\pm0.01}$	$\mathbf{0.39}^{\pm0.01}$	$0.57^{\pm0.02}$	$\mathbf{0.43}^{\pm0.02}$

---

* Code refactored and adapted from [9] to Habitat 3.0.

Table 2: Ablation studies for social dynamics estimation. GT denotes ground truth privileged information, and $_A$ indicates the adapted information.

Models	hGPS	traj.	hGPS$_A$	traj.$_A$	S ↑	SPS ↑	F ↑	CR ↓	ES ↑
S1	GT	GT			$\mathbf{0.94}^{\pm0.01}$	$0.58^{\pm0.00}$	$0.45^{\pm0.02}$	$0.64^{\pm0.03}$	$\mathbf{0.52}^{\pm0.01}$
S1	GT				$0.93^{\pm0.00}$	$\mathbf{0.62}^{\pm0.01}$	$\mathbf{0.46}^{\pm0.01}$	$0.64^{\pm0.02}$	$0.48^{\pm0.01}$
S1 (*Proposed*)		GT			$0.92^{\pm0.00}$	$0.46^{\pm0.01}$	$0.44^{\pm0.02}$	$\mathbf{0.61}^{\pm0.02}$	$0.50^{\pm0.01}$
S2			✓	✓	$0.57^{\pm0.06}$	$0.21^{\pm0.04}$	$0.05^{\pm0.01}$	$\mathbf{0.30}^{\pm0.02}$	$0.02^{\pm0.00}$
S2			✓		$0.70^{\pm0.02}$	$0.31^{\pm0.02}$	$0.05^{\pm0.01}$	$0.70^{\pm0.03}$	$0.03^{\pm0.01}$
S2 (*Proposed*)				✓	$\mathbf{0.91}^{\pm0.01}$	$\mathbf{0.45}^{\pm0.01}$	$\mathbf{0.39}^{\pm0.01}$	$0.57^{\pm0.02}$	$\mathbf{0.43}^{\pm0.02}$

4.2   Ablation studies

In the following section, we investigate the impact of various forms of privileged information and their adaptability. Additionally, we assess how the length of the trajectories fed in Stage 1 affects our model.

Adatptable information Table 2 presents the results from Stage 1 (S1) and Stage 2 (S2) with the utilization of various privileged information, such as Humanoid GPS (hGPS) and human trajectories (traj.). During S1, particularly in S and notably in SPS, incorporating hGPS as an additional input leads to superior results. This advantage is likely attributed to hGPS containing implicit information about human and robot positions, facilitating more efficient path selection, especially in SPS scenarios.

However, this pattern is not evident in S2, where trajectories typically provide more adaptable information. As previously mentioned, hGPS inherently encompasses the robot's position, posing challenges in regressing this data during the second stage due to the lack of initial context regarding the robot's location relative to the environment. Furthermore, hGPS may be difficult to adapt when
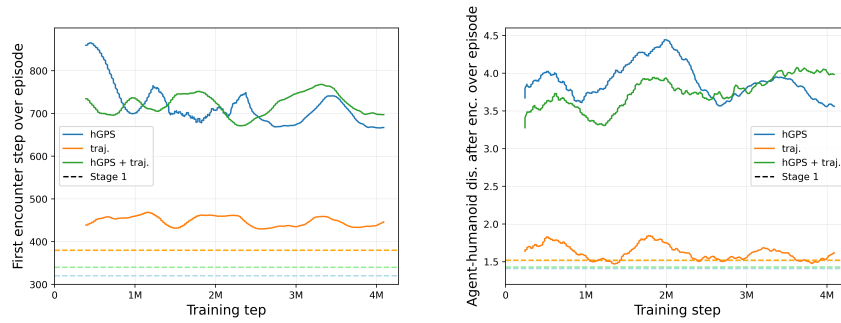
Fig. 3: *(left)* Training step (x-axis) Vs the number of steps which it takes the robot to find the humanoid (y-axis), in the finding task; *(right)* Training step (x-axis) Vs the average distance which the robot manages to keep itself at from the humanoid (y-axis), in the task of following.

the human is detected and disappears around a wall. Since hGPS comprises polar coordinates, the distance between the robot and the human spans the wall. This issue does not arise when utilizing trajectories, as they do not require any information about the robot and can be adapted solely based on depth images and detection information.

Fig. 3*(left)* illustrates the trend of the *First encounter step over episode* during Stage 2. The plot confirms that using trajectories facilitates the robot to find the person, which it achieves after only approx. 450 steps, while the hGPS and hGPS+traj take more than 700 steps. Finding the robot first results in higher S and SPS metrics, but it may incur larger chances of collision (CR metric), due to having to then follow it for longer, cf. Sec. 4.1.

Fig. 3*(right)* illustrates the average distance between the agent and the humanoid after the first encounter. While hGPS and hGPS+traj stay at approx. 4 meters, the proposed trajectory-based approach ranges between 1.5 and 1.9, thus well within the 1-2 meter range, which yields the success in the task of following (larger F metric, but encompassing a larger risk of collision– CR metric).

Table 3: Ablation on the length of trajectories to consider during Stage 1

Timeframes	S ↑	SPS ↑	F ↑	CR ↓	ES ↑
5	$0.62^{\pm0.01}$	$0.25^{\pm0.02}$	$0.05^{\pm0.01}$	$0.64^{\pm0.01}$	$0.06^{\pm0.01}$
20 (*Proposed*)	$\mathbf{0.92}^{\pm0.00}$	$\mathbf{0.46}^{\pm0.01}$	$\mathbf{0.44}^{\pm0.02}$	$\mathbf{0.61}^{\pm0.02}$	$\mathbf{0.50}^{\pm0.01}$
50	$0.70^{\pm0.02}$	$0.29^{\pm0.02}$	$0.08^{\pm0.02}$	$0.78^{\pm0.02}$	$0.08^{\pm0.01}$

Length of trajectories Table 3 illustrates the influence of trajectory length on our SDA. Employing 5 past timeframes as input yields generally inferior results, possibly due to the insufficient historical data to capture human dynamics. Similarly, using 50 timeframes leads to comparable outcomes. Although S, SPS, and F metrics improve relative to the 5-frame scenario, they still lag behind our proposed model. This could be attributed to potential redundancy and counterproductive encoding of information.

4.3   Qualitative Results

We qualitatively demonstrate the results in our proposed SDA. Firstly, we showcase the agent's ability to locate the humanoid within the environment by moving around, followed by its capability to follow the humanoid within the environment. Subsequently, we present two specific behaviors where the agent briefly spots the humanoid and one where it moves backward to create space for passage.

Fig. 4 shows an episode where the agent and the human are located in different rooms at the start. Then, the agent begins its search for the humanoid by navigating within the environment until the encounter takes place. After the encounter, the agent then transitions into the follow phase.

The episode in Fig. 5 (left) illustrates the agent's capability to track the human even when it is briefly observed. As the agent searches for the target in the bathroom, the humanoid swiftly passes in front of the door (Step 145) and disappears from the agent's view again (Step 150). Due to learned social dynamics, the agent exploits the humanoid's behavior and begins to follow it.

In the right episode in Fig. 5, the agent has already located the human and its objective is to follow it. It is observed that at Step 340, the human decides to move backward, prompting the agent to move backward as well, creating the necessary space for passage by Step 360. Then, it continues its task of following.

Limitations. Table 1 shows a dependency of stage 2 performance on the adaptability of information, highlighting the significance of this factor in determining the model's effectiveness. Another limitation arises from having benchmarked the proposed model only in simulation and the reliance on information during training that is not easily accessible outside simulation (trajectories). Mitigating these limitations, the advent of Habitat 3.0 [36], has greatly minimized the domain gap between simulated and real environments.

5   Conclusions

Our study presents the Social Dynamics Adaptation model (SDA) for Social Navigation within interactive settings. Notably, it is the first to integrate privileged human dynamics information during training while adapting it in the following stage, enabling its application in realistic environments without relying
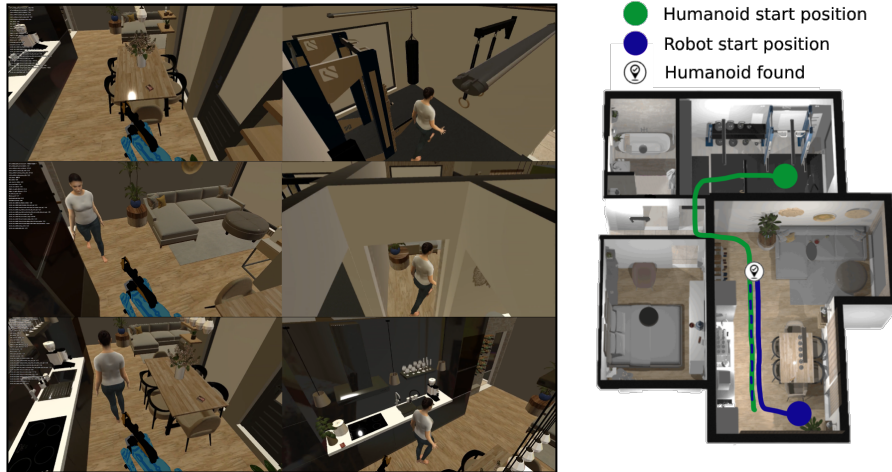
Fig. 4: The agent and the humanoid start the episode in separate rooms. The agent navigates through the environment in search of the humanoid, and once found, begins to follow it.

on such privileged information. Our findings underscore the non-trivial nature of adapting information, highlighting the necessity for selective processes. In future research, we aim to extend our model by incorporating diverse human dynamics beyond trajectories, enhancing the robot's comprehension of human movement patterns.

Fig. 5: We showcase two different episodes. On the left, the agent successfully follows the human after it swiftly moves in front of the door. On the right, an episode where the human decides to move backward and the agent steps back to make way.

References

1. Ahmed, M., Qin, N.: Surrogate-based aerodynamic design optimization: Use of surrogates in aerodynamic design optimization. International Conference on Aerospace Sciences and Aviation Technology **13**(AEROSPACE SCIENCES and AVIATION TECHNOLOGY, ASAT- 13, May 26 - 28, 2009), 1–26 (2009)
2. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: CVPR (2018)
3. Aoude, G.S., Luders, B.D., Joseph, J.M., Roy, N., How, J.P.: Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. Autonomous Robots **35**(1), 51–76 (2013)
4. Batra, D., Gokaslan, A., Kembhavi, A., Maksymets, O., Mottaghi, R., Savva, M., Toshev, A., Wijmans, E.: ObjectNav revisited: On evaluation of embodied agents navigating to objects. arXiv preprint arXiv:2006.13171 (2020)
5. Van den Berg, J., Lin, M., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: IEEE International Conference on Robotics and Automation (ICRA) (2008)
6. Berg, J.v.d., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: Robotics Research, pp. 3–19. Springer (2011)
7. Campari, T., Eccher, P., Serafini, L., Ballan, L.: Exploiting scene-specific features for object goal navigation. In: European Conference on Computer Vision. pp. 406–421. Springer (2020)
8. Campari, T., Lamanna, L., Traverso, P., Serafini, L., Ballan, L.: Online learning of reusable abstract models for object goal navigation. In: CVPR (2022)
9. Cancelli, E., Campari, T., Serafini, L., Chang, A.X., Ballan, L.: Exploiting proximity-aware tasks for embodied social navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10957–10967 (October 2023)
10. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. In: International Conference on 3D Vision (3DV) (2017)
11. Chaplot, D.S., Gandhi, D.P., Gupta, A., Salakhutdinov, R.: Object goal navigation using goal-oriented semantic exploration. In: NeurIPS (2020)
12. Chaplot, D.S., Salakhutdinov, R., Gupta, A., Gupta, S.: Neural topological slam for visual navigation. In: CVPR (2020)
13. Chen, C., Liu, Y., Kreiss, S., Alahi, A.: Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In: IEEE International Conference on Robotics and Automation (ICRA) (2019)
14. Chen, Y.F., Everett, M., Liu, M., How, J.P.: Socially aware motion planning with deep reinforcement learning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)
15. Chen, Y.F., Liu, M., Everett, M., How, J.P.: Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In: IEEE International Conference on Robotics and Automation (ICRA) (2017)
16. Deitke, M., Batra, D., Bisk, Y., Campari, T., Chang, A.X., Chaplot, D.S., Chen, C., D'Arpino, C.P., Ehsani, K., Farhadi, A., et al.: Retrospectives on the embodied ai workshop. arXiv preprint arXiv:2210.06849 (2022)
17. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: Computer Vision – ECCV 2014 (2014)

18. Ferrer, G., Garrell, A., Sanfeliu, A.: Social-aware robot navigation in urban environments. In: Proc. of thr European Conference on Mobile Robots (2013)
19. Guo, X., Li, W., Iorio, F.: Convolutional neural networks for steady flow approximation. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 481–490. KDD '16, Association for Computing Machinery, New York, NY, USA (2016)
20. Guzzi, J., Giusti, A., Gambardella, L.M., Theraulaz, G., Di Caro, G.A.: Human-friendly robot navigation in dynamic environments. In: IEEE International Conference on Robotics and Automation (ICRA) (2013)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
22. Kivrak, H., Çakmak, F., Kose, H., Yavuz, S.: Social navigation framework for assistive robots in human inhabited unknown environments. Engineering Science and Technology, an International Journal (2021)
23. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A.: Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474 (2017)
24. Krantz, J., Wijmans, E., Majumdar, A., Batra, D., Lee, S.: Beyond the nav-graph: Vision-and-language navigation in continuous environments. In: ECCV (2020)
25. Kumar, A., Fu, Z., Pathak, D., Malik, J.: RMA: Rapid motor adaptation for legged robots. In: Robotics: Science and Systems (2021)
26. Li, C., Xia, F., Mart'in-Mart'in, R., Lingelbach, M., Srivastava, S., Shen, B., Vainio, K., Gokmen, C., Dharan, G., Jain, T., Kurenkov, A., Liu, K., Gweon, H., Wu, J., Fei-Fei, L., Savarese, S.: igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. ArXiv **abs/2108.03272** (2021)
27. Long, P., Liu, W., Pan, J.: Deep-learned collision avoidance policy for distributed multiagent navigation. IEEE Robotics and Automation Letters **2**(2), 656–663 (2017)
28. Loquercio, A., Kumar, A., Malik, J.: Learning visual locomotion with cross-modal supervision. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 7295–7302. IEEE (2023)
29. Lu, Y., Ruan, X., Huang, J.: Deep reinforcement learning based on social spatial–temporal graph convolution network for crowd navigation. Machines **10**(8), 703 (2022)
30. Möller, R., Furnari, A., Battiato, S., Härmä, A., Farinella, G.M.: A survey on human-aware robot navigation. Robotics and Autonomous Systems **145**, 103837 (2021)
31. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics **31**(5), 1147–1163 (2015)
32. Partsey, R., Wijmans, E., Yokoyama, N., Dobosevych, O., Batra, D., Maksymets, O.: Is mapping necessary for realistic pointgoal navigation? In: CVPR (2022)
33. Patle, B.K., L., G.B., Pandey, A., Parhi, D.R., Jagadeesh, A.: A review: On path planning strategies for navigation of mobile robot. Defence Technology (2019)
34. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)

35. Puig, X., Ra, K.K., Boben, M., Li, J., Wang, T., Fidler, S., Torralba, A.: Virtual-home: Simulating household activities via programs. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 8494–8502 (2018)

36. Puig, X., Undersander, E., Szot, A., Cote, M.D., Yang, T.Y., Partsey, R., Desai, R., Clegg, A.W., Hlavac, M., Min, S.Y., Vondruš, V., Gervet, T., Berges, V.P., Turner, J.M., Maksymets, O., Kira, Z., Kalakrishnan, M., Malik, J., Chaplot, D.S., Jain, U., Batra, D., Rai, A., Mottaghi, R.: Habitat 3.0: A co-habitat for humans, avatars and robots (2023)

37. Rahman, M.R.U., Scofano, L., Matteis, E.D., Flaborea, A., Sampieri, A., Galasso, F.: Best practices for 2-body pose forecasting. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 3614–3624 (2023)

38. Ramakrishnan, S.K., Chaplot, D.S., Al-Halah, Z., Malik, J., Grauman, K.: Poni: Potential functions for objectgoal navigation with interaction-free learning. In: CVPR (2022)

39. Ramakrishnan, S.K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J., Undersander, E., Galuba, W., Westbury, A., Chang, A.X., et al.: Habitat-matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. arXiv preprint arXiv:2109.08238 (2021)

40. Raychaudhuri, S., Campari, T., Jain, U., Savva, M., Chang, A.X.: Mopa: Modular object navigation with pointgoal agents. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 5763–5773 (January 2024)

41. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al.: Habitat: A platform for embodied AI research. In: ICCV (2019)

42. Shen, B., Xia, F., Li, C., Martín-Martín, R., Fan, L., Wang, G., Pérez-D'Arpino, C., Buch, S., Srivastava, S., Tchapmi, L., et al.: iGibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2021)

43. Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D.S., Maksymets, O., Gokaslan, A., Vondru*vs*., V., Dharur, S., Meier, F., Galuba, W., Chang, A.X., Kira, Z., Koltun, V., Malik, J., Savva, M., Batra, D.: Habitat 2.0: Training home assistants to rearrange their habitat. ArXiv **abs/2106.14405** (2021)

44. Trautman, P., Krause, A.: Unfreezing the robot: Navigation in dense, interacting crowds. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2010)

45. Tsoi, N., Xiang, A., Yu, P., Sohn, S.S., Schwartz, G., Ramesh, S., Hussein, M., Gupta, A.W., Kapadia, M., Vázquez, M.: Sean 2.0: Formalizing and generating social situations for robot navigation. IEEE Robotics and Automation Letters **7**, 11047–11054 (2022)

46. Wani, S., Patel, S., Jain, U., Chang, A.X., Savva, M.: MultiON: Benchmarking semantic map memory using multi-object navigation. In: NeurIPS (2020)

47. Wijmans, E., Datta, S., Maksymets, O., Das, A., Gkioxari, G., Lee, S., Essa, I., Parikh, D., Batra, D.: Embodied Question Answering in Photorealistic Environments with Point Cloud Perception. In: CVPR (2019)

48. Wijmans, E., Kadian, A., Morcos, A., Lee, S., Essa, I., Parikh, D., Savva, M., Batra, D.: DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In: ICLR (2019)

49. Xia, F., Shen, W.B., Li, C., Kasimbeg, P., Tchapmi, M.E., Toshev, A., Martín-Martín, R., Savarese, S.: Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. IEEE Robotics and Automation Letters **5**(2), 713–720 (2020)
50. Ye, J., Batra, D., Das, A., Wijmans, E.: Auxiliary Tasks and Exploration Enable ObjectNav. In: ICCV (2021)
51. Ye, J., Batra, D., Wijmans, E., Das, A.: Auxiliary Tasks Speed Up Learning Point-Goal Navigation. In: Conference on Robot Learning (2020)
52. Yokoyama, N., Luo, Q., Batra, D., Ha, S.: Learning Robust Agents for Visual Navigation in Dynamic Environments: The Winning Entry of iGibson Challenge 2021. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2022)
53. Zhang, X., Wu, Y., Wang, H., Iida, F., Wang, L.: Adaptive locomotion learning for quadruped robots by combining drl with a cosine oscillator based rhythm controller. Applied Sciences **13**(19) (2023)